ISSN 2007-9737

Identificación de tópicos en un corpus utilizando Transformers

Jorge Víctor Carrera-Trejo[1], Rodrigo Cadena Martínez[2]

[1] Investigador independiente,
México

[2] Universidad Tecnológica de México,
México

jvcarrera@gmail.com, rocadmar@mail.unitec.mx

Resumen. Clasificar un corpus de textos con un base en un conjunto de clases utilizando Transformers permite construir un modelo basado en las palabras que contiene la información, sin embargo, el modelo utiliza todas las palabras con que se entrena y el orden en que se encuentran, pero ¿cuáles de éstas palabras se encuentran relacionadas directamente con la temática de los textos?, este trabajo se enfoca en proponer una metodología que permita, utilizando un corpus multi etiquetado que contiene la descripción de 1200 comics organizado en 4 clases, eliminar información no relacionada con la temática, basándose en la identificación de entidades nombradas y enunciados característicos, generando con ello un nuevo corpus con el cual entrenar y validar un Transformer, utilizando la medida de accuracy macro como medida de evaluación, como caso base se propone el valor de accuracy macro de la validación de un Transformer entrenado con datos crudos, demostrando que al utilizar datos relacionados con la temática de los textos se mejoran los resultados de clasificación pasando de 0.733 a 0.992.

Palabras clave. Bert, multi etiqueta, tópicos, clasificación, transformers, comics.

Topics Identification in a Corpus based on Transformers

Abstract. Classifying a corpus of texts based on a set of classes using Transformers allows to build a model based on the words that contain the information, however, the model uses all the words in the training process and the order in which its found, but, which of these words are directly related to the topic of the texts? This work focuses on proposing a methodology that allows, using a multi-labeled corpus that contains the description of 1200 comics organized in 4 classes, to eliminate information that are not related to the topic, based on the identification of named entities and noun phrases, thereby generating a new corpus with which to train and validate a Transformer, using the macro accuracy measure as an evaluation measure, as a base case the macro accuracy value, obtained of the validation of a Transformer trained with the original data is proposed, demonstrating that when we using data related to the subject matter of the texts, the classification results are improved from 0.733 to 0.992.

Keywords. Bert, multilabel, topics, classification, transformers, spacy, comics.

1. Introducción

La detección de información relacionada con la temática de un texto se ha convertido en un problema dentro del desarrollo de sistemas de información orientados a la clasificación. Se han desarrollado diferentes algoritmos relacionados con la identificación temática, basados en la semántica latente o implícita dentro de un texto, principalmente *Latent Semantic Analysis* (LSA) [1] y *Latent Dirichlet Allocation* (LDA) [2], que permiten generar diversos grupos temáticos a partir de un corpus de entrada, sin embargo, para el mejor funcionamiento de estos algoritmos es importante contar con información lo más relacionada posible con la temática que expresan, a partir de la cual se quieran generar los grupos temáticos, sin embargo, ¿cómo se puede identificar dicha información temática? y por otro lado, ¿cómo se puede evaluar que la información

modele adecuadamente el corpus del que se extrae?

Para responder a dichas preguntas, nos dirigimos al *procesamiento del lenguaje natural*, *nlp*, [3, 4], el cual se enfoca en resolver diferentes tareas relacionadas con el lenguaje, entre las que se incluye el procesamiento de textos, utilizándolos como entrada para su clasificación [3, 4, 5], realizando *clasificación binaria*, *multiclase*, *análisis de sentimientos*, *question-answering*, principalmente. Es así, que las preguntas planteadas anteriormente se relacionan directamente con la tarea de clasificación, en la cual se incluye un corpus, en el que los textos que lo conforman pertenecen a una o varias clases, siendo el objetivo ubicar correctamente la mayoría de los textos en las clases a las que pertenecen, a partir de una caracterización determinada [3].

Actualmente la tarea de clasificación dentro del nlp se basa principalmente en la utilización de los denominados modelos de Transformers o simplemente Transformers [6], los cuáles a su vez se basan en el uso de los modelos de lenguaje [5, 6]. En estos Transformers es importante que la información de entrada se encuentre como en el texto original, es decir, en forma de enunciados y/o párrafos, a diferencia de las herramientas de clasificación utilizadas antes de la aparición de los Transformers, la información se transformaba en una representación vectorial, la cual podía ser *tf , tf-idf* o *binaria* [3], principalmente, pero para los modelos de lenguaje son importantes las secuencias de las palabras ya que a partir de ellas, se generan espacios semánticos o de *embeddings*, es por ello que se han desarrollado herramientas [7, 8] que permiten extraer los enunciados más característicos, *noun phrases* o *chunks*, [9] utilizando un modelo de lenguaje, basadas principalmente en la tecnología de redes neuronales y posteriormente, haciendo uso de Transformers como por ejemplo las librerías *Spacy* [7] o *Stanza* [8].

El objetivo del trabajo presentado en este artículo se enfoca en tratar de responder las preguntas planteadas en ¿cómo se puede identificar dicha información temática? y ¿cómo se puede evaluar que la información modele el corpus del que se extrae?, utilizando para ello un corpus multi etiquetado, que contiene la descripción de 1200 cómics agrupados en 4 clases relacionadas con las temáticas de *Batman* y *Superman*, dónde, para responder la primer cuestión, se hace uso de un proceso semi-automático y supervisado por un experto, en el cual se elimina del corpus aquella información que no esté relacionada con la temática de los textos, siendo validado el corpus generado por el experto. Por otro lado, respecto a la segunda pregunta, esta se responderá mediante la evaluación de un clasificador basado en un modelo de Transformers utilizando como medida el *accuracy* macro de las clases a las que pertenecen los textos. Como caso base o *gold-standard* se propone el *accuracy* macro devuelto por el modelo de Transformer entrenado y validado con un corpus al cual no se le realiza un pre-procesamiento, es decir, el corpus original, dónde el valor de *accuracy* macro será comparado con el *accuracy* macro devuelto por el modelo de Transformer entrenado y probado con un corpus al cuál se le realiza un pre-procesamiento, dónde se elimina la información no relacionada con la temática del cómic al que pertenece. Finalmente, se propone, además, verificar y comparar el *accuracy* macro de un tercer Transformer entrenado con un tercer corpus que contenga sólo las frases más representativas extraídas a partir de un modelo de lenguaje utilizando para ello la librería *Spacy* [7].

En las siguientes secciones se presentarán los trabajos relacionados con este artículo, así como la metodología propuesta, los resultados obtenidos y finalmente se podrán observar las conclusiones y propuestas de trabajo futuro inferidas a partir del desarrollo del artículo.

2. Trabajos relacionados

En esta sección se muestran los trabajos relacionados con el artículo presentado, inicialmente se describen los conceptos utilizados dentro de la metodología y posteriormente los trabajos desarrollados por otros investigadores que guardan una relación con el presente trabajo.

2.1. Conceptos

El procesamiento del lenguaje natural (nlp), se puede definir [10] como "*la habilidad de la máquina para procesar la información comunicada*", para

ello desde el nlp se han definido diferentes tipos de tareas que permitan desarrollar esta habilidad utilizando diferentes perspectivas, como son la identificación de las estructuras gramaticales o la generación de árboles de dependencias, entre otras, para el caso de enunciados, o bien desde tareas que involucran una gran cantidad de información, en la forma de corpus, principalmente como son el análisis de sentimientos, agrupación temática [1, 2] o bien la clasificación de textos [3, 4].

La clasificación de textos, en la cual un texto compuesto por un conjunto de palabras, se enfoca en determinar su pertenencia a una clase con base en un conjunto de clases conocidas, para lo cual, en un ámbito supervisado, se tiene conocimiento de estas clases y de sus características, para ello existen diferentes trabajos que se enfocan en determinar las mejores características que resuelvan la tarea, un ejemplo de ello se muestra en [11], utilizando diferentes tipos de caracterizaciones, *tf* , *tf-idf* o *binaria* [3], basándose en la identificación de n-gramas [13], dónde *n* puede tomar los valores desde 1 hasta el número de palabras que contengan los textos, con lo cual se convierten los textos a representaciones vectoriales [3, 11], las cuales pueden ser procesadas por una máquina utilizando algún tipo de clasificador [3].

Sin embargo, a la aparición de trabajos como [12], en los cuales las palabras ahora son representadas mediante sus coordenadas en un espacio vectorial, también conocido como *embeddings*, construido con base en un corpus con una gran cantidad de documentos y del análisis de cada una de las palabras que contienen los diferentes textos y su relación con las otras palabras que conforman dichos textos, utilizando estos nuevos espacios de *embeddings* en la tarea de clasificación obteniéndose mejoras en esta tarea, como se muestra en [12, 14].

Por otro lado, con el advenimiento de la tecnología denominada como Transformers [6], la cual se basa en la utilización de redes neuronales, permite eliminar la necesidad de contar con grandes volúmenes de datos para la construcción de los espacios de *embeddings*, y con base en su tecnología de redes neuronales, utilizar un espacio de *embeddings* pre entrenado, el cual representa un modelo de lenguaje [6, 14], y mediante un proceso de *fine-tuning*, entrenar un modelo de Transformer con base en el espacio de *embeddings* y el corpus de trabajo de acuerdo a la información que se quiera clasificar. Actualmente existen diferentes sitios web desde los cuáles se pueden descargar diferentes modelos de Transformers pre-entrenados, como es *Hugging Face* [15], siendo uno de los más utilizados el modelo denominado BERT [16].

Finalmente, basándose en la idea del espacio de *embeddings*, así como de la tecnología alrededor de los Transformers, algunas de las tareas del procesamiento del lenguaje natural que han sido implementadas en forma de librerías de software como *Spacy* [7] para su utilización de forma automatizada, son la extracción de *entidades nombradas* y la identificación de los enunciados más característicos de un texto, *noun phrases* o *chunks*, en la primera, el objetivo es identificar textos formados por una o varias palabras que representen personas, organizaciones, lugares, expresiones de tiempo y cantidades [3], mientras que la segunda se refiere a frases que contienen una palabra dónde esa palabra es descrita por las palabras que la acompañan dentro de la frase [7].

Es importante mencionar la medida de validación utilizada en este trabajo, la cual es la de *accuracy*, la cual indica la proporción de documentos correctamente clasificados. De acuerdo a [3], esta medida se puede definir, de acuerdo a la ecuación 1:

$$\text{accuracy} = [TP + TN] / [TP + TN + FP + FN], \quad (1)$$

dónde, los valores en el numerador, *TP o true positives* y *TN o true negatives*, indican el número de documentos clasificados correctamente, y los valores en el denominador *TP, TN, FP o false positive* y *FN o false negative*, indican el número total de documentos para una clase en particular. Por otro lado, en el tenor de este trabajo al utilizar un corpus multi etiquetado, se utiliza el *accuracy* macro que es el promedio de la suma de cada *accuracy* de cada una de las clases en el corpus.

2.2. Trabajos relacionados

Se han desarrollado diferentes trabajos cuyo objetivo es observar el comportamiento del uso de

Transformers dentro de la tarea de multi-etiquetado, por ejemplo en [17] se utilizó un modelo BERT para clasificar un corpus de tweets organizado en 24 categorías, al corpus utilizado se le realizó un pre-procesamiento eliminando stop-words, url's, convirtiendo los emojis a textos y lematizando finalmente los tweets, otro trabajo es el presentado en [18], en dónde se realiza la clasificación de un corpus organizado 5 emociones, el cual no se encuentra balanceado, proponiendo utilizar un ensamble de Transformers basado en BERT, el *gold standard* propuesto se basa en realizar la clasificación del corpus utilizando clasificadores como SVM o bien modelos de redes neuronales como CNN [14].

Por otro lado en [19], se muestra un trabajo enfocado en la creación de un sistema basado en el uso de algoritmos de clasificación multi-etiqueta [11] complementados con BERT, para la clasificación de 5 categorías, cada una de ellas con 4 sentimientos, como *gold standard* el autor propone el uso algoritmos de clasificación enfocados al multi etiquetado [11] y como medida de evaluación *accuracy*, finalmente en [20] se muestra la propuesta de un modelo de transformer basado en BERT, el cual realiza la clasificación de un corpus basado en millones de etiquetas.

Por otro lado, otros trabajos se enfocan en analizar los datos de entrada a un transformer, así como proponer ciertas variaciones que se puedan construir y/o modificar la capa de *embeddings*, por ejemplo, en [21] se muestra dos estudios enfocados a comprender la estructura de la representación utilizada en BERT, el primero se enfoca en la identificación de la estructura, mientras que el segundo se enfoca en representaciones de la concordancia de verbo-sujeto y de anáfora-antecedente.

Revisando el primer estudio presentado en [21], los autores, utilizando clasificadores de diagnóstico crean representaciones del corpus basadas en propiedades lineales, secuenciales y jerárquicas, a continuación la información es representada utilizando BERT-*embeddings*, analizando esta representación se observa que conforme los datos son procesados en las capas superiores del *embedding*, la prevalencia de la información de las propiedades lineales/secuenciales se pierde, mientras que la relacionada con las propiedades jerárquicas se mantiene, para observar el comportamiento de los datos procesados utilizando dos modelos BERT pre-entrenados y como medida de evaluación utilizan *accuracy.*

Mientras en [22] se presenta un estudio experimental, en el cual utilizando técnicas de BERTopic basadas en la utilización de diferentes representaciones de *embeddings* aplicadas a un corpus de 111 728 documentos en el idioma árabe, se logra obtener mejores resultados en la identificación de tópicos que las técnicas tradicionales como *LDA*, como medida de validación se utiliza *Non-Point Mutual Information*, NPMI.

Finalmente en [23] se realiza una propuesta en la cual se analiza cada uno de los espacios de *embeddings* generado con *Word2Vec*, de dos corpus, *Twenty Newsgroups* y *Hacker News Comments*, identificando grupos de tópicos a partir del análisis de las relaciones de palabras base en un procedimiento similar a *LDA*, proponiendo medidas para relacionar palabras y documentos, como medida de validación de los grupos de tópicos se utiliza *coherence*.

Como se puede observar en los trabajos [17, 18, 19, 20] la utilización del transformer BERT en la clasificación de corpus multi etiquetados, se realiza utilizando alguna de sus características como son, su espacio de *embeddings* y/o como clasificador, basándose en un esquema de *fine-tuning*, en este trabajo, al igual que en los trabajos mostrados, se hace uso de un transformer BERT, pre entrenado, para un corpus multi etiquetado, se realiza un pre procesamiento, pero no se lematizan los textos, como si se realiza en algunos trabajos, pero a diferencia de ellos, se propone realizar una limpieza temática y posteriormente extraer los enunciados más importantes, es decir, modificar los datos de entrada

A diferencia de los trabajos presentados en [21, 22, 23], los cuales trabajan sobre los espacios de *embeddings*, este trabajo se enfoca en realizar una limpieza temática con base en la experiencia de un experto, la cual se presenta en la sección 3 relacionada con la metodología, los resultados obtenidos del proceso de clasificación utilizando el corpus limpio se observan en la sección 4, comparando dichos resultados con el *gold standard* propuesto.

3. Metodología

En esta sección se presenta la metodología utilizada en el desarrollo de este trabajo, la cual se basa en la utilización de un corpus base, con el cual se ejecutan los siguientes pasos:

- Limpieza del corpus.
- Entrenamiento del transformer.
- Prueba del transformer.

La descripción de cada uno de estos procesos se explica a detalle en las siguientes secciones.

3.1. Limpieza del corpus

La limpieza del corpus es un proceso tradicionalmente enfocado en eliminar caracteres no deseados, considerados basura, que acompañan a las palabras al momento de descargar un corpus, sin embargo, en el ámbito de este trabajo, adicionalmente a esta tarea, la limpieza del corpus se enfoca en eliminar enunciados y/o palabras que no aporten información acerca de la temática de un cómic.

Para eliminar las palabras que no estén relacionadas con el contenido de un cómic con base en la descripción con que se cuenta de éste, se realiza un proceso de limpieza, la cual aplica a todo el corpus y se basa en la utilización de diccionarios en los que se incluyen estás palabras no relacionadas, para eliminarlas de las distintas descripciones, así como aquellos signos que puedan ser considerados ruido.

Para la creación de los diccionarios y eliminación de palabras y signos (*tokens*), se sigue un proceso como el que se muestra en la figura 1. Este proceso, como se puede observar, es un proceso iterativo en el cual se cuenta con el apoyo de un experto quién revisa la información a ser eliminada y valida el corpus cuando ya no es posible eliminar más información del mismo.

El proceso de limpieza hace uso de un corpus de entrada, el cual contiene las descripciones de un número determinado de cómics, los cuáles se están organizados en cierto número de clases, a estas descripciones se les aplica los siguientes pasos:

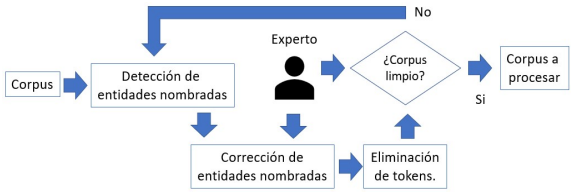


Fig. 1. Proceso de limpieza del corpus

1. Detección de *entidades nombradas*. Al corpus de entrada se le aplica una herramienta de software que permita detectar las *entidades nombradas* [3], generando con ello diferentes listas, dependiendo de la información que se busque, en el ámbito de este proyecto el objetivo es detectar nombres de autores, actividad del autor, que puede ser escritor, dibujante o entintador, publicidad insertada dentro de la revista, valor del cómic y número de páginas de la revista, principalmente.

2. Corrección de *entidades nombradas*. A partir de las diferentes listas de entidades obtenidas, un experto revisa cada una de las entidades y las compara con los textos en los que fueron detectadas y de acuerdo a su experiencia las corrige, obteniendo con ello diccionarios de entidades más fiables.

3. Eliminación de tokens. Considerando los diferentes diccionarios de entidades, se toma cada uno de estos y se eliminan cada una de las entidades en el corpus, en el caso de los nombres de los autores inicialmente se sustituyen por un token en general, y finalmente al ser asociados con una actividad, escritor, dibujante o entintador, se eliminan.

4. Corpus limpio. Finalmente, el corpus resultante es revisado por el experto, quien elige diferentes descripciones al azar, las revisa y si considera que las descripciones contienen solamente palabras relacionadas con el contenido del cómic, se considera un corpus limpio y listo para ser considerado en el entrenamiento de algún transformer, de no ser así se repite el proceso de limpieza considerando este corpus como corpus de entrada.

Fig. 2. Extracción de chunks

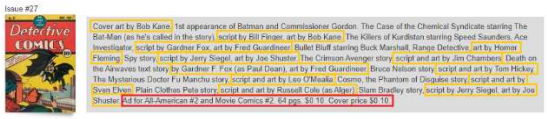


Fig. 3. Descripción del cómic Detective Comics #27 (1939)

Un proceso adicional al paso de limpieza del corpus que se propone en el presente trabajo es utilizar la potencialidad de las librerías como *Spacy* [7] y limpiar un poco más el corpus a procesar detectando los enunciados característicos o *chunks* de cada una de las descripciones limpias, para ello se propone un proceso como el que se muestran en la figura 2, el cual toma como entrada cada una de las descripciones en el corpus limpio, extrae de éste los *chunks* y utiliza éstos *chunks* como la descripción original, generando con ello un nuevo corpus basado en *chunks*.

3.2. Entrenamiento del Transformer

Para realizar el proceso de entrenamiento del Transformer, se considera un corpus, el cual se separa en dos partes, una de ellas será denominada corpus de entrenamiento, que servirá para entrenar un modelo de transformer y la otra parte será el corpus de prueba o validación que será utilizado en el proceso de prueba del transformer entrenado.

Por otro lado, en el ámbito de este trabajo se propone el uso de un transformer BERT [16] pre-entrenado, descargado del sitio *Hugging Face*[1] [15], con el objetivo de realizar el proceso de *fine-tuning* utilizando los diferentes corpus de entrenamiento con los que se cuenta.

3.3. Prueba del Transformer

En este paso, se toma transformer entrenado validándolo con el corpus de prueba. El corpus de prueba contiene descripciones que no fueron utilizadas para el entrenamiento por lo que son datos desconocidos para el transformer por lo que el resultado de su evaluación nos permitirá identificar que corpus modela adecuadamente el corpus de entrenamiento que contiene las descripciones de los diferentes cómics a partir de las cuatro clases en las que se agrupan.

Con base en la clasificación de las diferentes descripciones en el corpus de prueba se realizará el cálculo del *accuracy* macro, considerando las cuatro clases de agrupamiento, comparando posteriormente este resultado.

4. Resultados

En esta sección se presentan los resultados obtenidos de aplicar la metodología presentada en la sección 3. Inicialmente se describirá el corpus utilizado y posteriormente se mostrarán los resultados obtenidos al aplicar cada uno de los pasos de la metodología propuesta.

4.1. Corpus

El corpus utilizado, CPS-1, se basa en 1200 descripciones de cómics, las cuáles fueron descargadas del sitio web especializado en venta de comics mycomicshop.com[2]. Cada una de estas descripciones cuenta con información propia del cómic al que se refiere, la cual incluye: contenido del cómic, información de los autores, entre los que se incluye nombre del escritor, dibujante y entintador e información propia del cómic como son su precio de portada, número de páginas y en su caso anuncios que se incluyeron dentro de la revista.

En la figura 3 se muestra la descripción para el cómic Detective Comics #27 publicado en 1939, en esta figura se puede observar, marcado en color amarillo, la información relativa con los diferentes autores de las historias que se incluyen, en color rojo la información relacionada con la revista y sin marcar información relacionada directamente con el contenido del cómic.

Cada una de estas 1200 descripciones se agruparon en dos clases temáticas que son: *Batman* y *Superman*, cada una agrupando 600 descripciones, las cuáles son los temas a los que

[1] https://huggingface.co
[2] https://www.mycomicshop.com/

pertenecen los títulos que se seleccionaron, los cuáles son: *Detective Comics* y *Batman* para el caso del tema *Batman* y *Action Comics* y *Superman* para el caso del tema *Superman*, siendo 300 descripciones para cada título. Estas descripciones son las descripciones de los primeros 300 números que se publicaron para cada uno de estos títulos.

En la tabla 1, se puede observar un resumen de las clases en las que se agrupa el corpus, así como las subclases que contiene, las cuáles son indicadas por el título del cómic al que pertenece la descripción correspondiente, en paréntesis se indica el año en que se publicó el primer número del cómic.

Para efectos del trabajo presente se considera el uso del corpus dividido en las 4 clases indicadas en la columna *"Clase 2"* de la tabla 1.

4.2. Limpieza del corpus

Como se mostró en la sección 3.1, el proceso de limpieza del corpus es un proceso iterativo, de acuerdo a la figura 1, el cual se basó en una herramienta de software que hace uso de la librería *Spacy* [7], con esta herramienta se detectaron *entidades nombradas* [3], que no estuvieran relacionadas con el contenido del cómic.

La herramienta fue aplicada al corpus, CPS-1, descrito en la sección 4.1, dónde las entidades detectadas fueron revisadas por un experto, agregadas a diferentes diccionarios y seleccionadas de acuerdo al tipo de procesamiento que se le realizó, es decir, si la entidad nombrada pertenece al nombre de algún autor de cómics, es sustituida en el corpus por la etiqueta "comic_author" y si la etiqueta pertenece a una característica relacionada con la información del cómic entonces es eliminada.

Es importante el trabajo del experto ya que la herramienta no detecta correctamente todas las entidades, por lo que se hace necesario un proceso manual de verificación.

Algunas entidades detectadas se muestran en la tabla 2, se indica además como fue detectada la entidad originalmente y como fue propuesta por el experto de acuerdo al texto original, para finalmente eliminada del corpus.

Tabla 1. Clases del corpus

Clase 1	Clase 2
Batman	*Batman (1940)*
	Detective Comics (1937)
Superman	*Action Comics (1938)*
	Superman (1939)

Tabla 2. Ejemplos de entidades nombradas detectadas por *Spacy*

Característica	*Spacy*	Corrección
Nombres de autores	Kirby	Jack Kirby
	John Romita	John Romita Sr John Romita, Jr John Romita
	John B.	John B. Wentworth
Información del cómic	$0.10	$0.10.cover price $0.10
	100 pages	100-page super spectacular
	first 15-cent	first 15-cent cover price
		first 15-cent issue
	plot by	plot by comic_author
		plot by comic_author and comic_author

El proceso se repitió iterativamente hasta que ya no se detectaron entidades del corpus procesado, es importante mencionar que el proceso de eliminación de entidades genera ruido en la forma de signos de puntuación, por lo que se eliminaron los signos de puntuación que no estuvieran asociados a alguna palabra en el corpus procesado, el corpus resultante fue denominado CPS-2.

Al corpus CPS-2, se le aplicó un proceso de extracción de *chunks* [3, 7], basado en la utilización de la librería *Spacy* [7], con el objetivo de obtener las frases más características, es así que cada descripción del corpus CPS-2 contiene después de aplicar este proceso la lista de *chunks* únicos y característicos de la descripción,

Tabla 3. Descripciones del comic *Detective Comics #27*

Corpus	Descripción
CPS-1	Cover art by Bob Kane. 1st appearance of Batman and Commissioner Gordon. The Case of the Chemical Syndicate starring The Bat-Man (as he's called in the story), script by Bill Finger, art by Bob Kane. The Killers of Kurdistan starring Speed Saunders, Ace Investigator, script by Gardner Fox, art by Fred Guardineer. Bullet Bluff starring Buck Marshall, Range Detective, art by Homer Fleming. Spy story, script by Jerry Siegel, art by Joe Shuster. The Crimson Avenger story, script and art by Jim Chambers. Death on the Airwaves text story by Gardner F. Fox (as Paul Dean), art by Fred Guardineer. Bruce Nelson story, script and art by Tom Hickey. The Mysterious Doctor Fu Manchu story, script and art by Leo O'Mealia. Cosmo, the Phantom of Disguise story, script and art by Sven Elven. Plain Clothes Pete story, script and art by Russell Cole (as Alger). Slam Bradley story, script by Jerry Siegel, art by Joe Shuster. Ad for All-American #2 and Movie Comics #2. 64 pgs. $0.10.Cover price $0.10.
CPS-2	1st appearance of Batman and Commissioner Gordon. The Case of the Chemical Syndicate starring The Bat-Man (as he's called in the story). The Killers of Kurdistan starring Speed Saunders, Ace Investigator. Bullet Bluff starring Buck Marshall, Range Detective. Spy story.
CPS-3	1st appearance, Batman, Commissioner Gordon, The Case, the Chemical Syndicate, the story, The Killers, Kurdistan, Speed Saunders, Ace Investigator, Bullet Bluff, Buck Marshall, Range Detective, Spy story, The Crimson Avenger, Death, the Airwaves, Bruce Nelson, The

generando así un nuevo corpus denominado CPS-3.

Es importante resaltar que mientras el corpus CPS-1 contiene toda la información descargada, los corpus CPS-2 y CPS-3 contienen solamente información relacionada directamente con la temática de cada uno de los cómics. En la tabla 3, tomando como ejemplo la descripción del cómic mostrada en la figura 3, se muestran las descripciones correspondientes al mismo cómic, *Detective Comics #27*, obtenidas después del proceso de limpieza, en los corpus CPS-2 y CPS- .

Finalmente, cada uno de los corpus resultantes fue divido en corpus de entrenamiento y pruebas, siguiendo un esquema de 80-20, es decir, el corpus de entrenamiento contiene el 80 % de las descripciones totales del corpus y el de pruebas el 20 % restante, cada uno de los corpus fue etiquetado para identificar si es un corpus de prueba o entrenamiento, por ejemplo para el corpus CPS-1 se generó un corpus CPS-E1 de entrenamiento y CPS-P1 de pruebas, se hace notar que los corpus de entrenamiento y de pruebas de los tres corpus, CPS-1, CPS-2 y CPS-3 contienen las descripciones de los mismos cómics para así poder realizar una comparación adecuada de los Transformers que se generan y prueban en pasos posteriores.

4.3. Entrenamiento de Transformers

En este paso se utilizaron los parámetros mostrados en la tabla 4 para el entrenamiento de un transformer para cada uno de los corpus de entrenamiento, es decir, se utilizaron los mismos parámetros, cambiando solamente el corpus de entrenamiento, el cual puede ser CPS-E1, CPS-E2 o bien CPS-E3.

El tiempo de entrenamiento de cada uno de los transformers, fue de alrededor de 1 hora 40 minutos, utilizando una implementación basada en Python [24] y Tensorflow [25], cada transformer se identifica de acuerdo al corpus de entrenamiento utilizado, es decir, el transformer entrenado con el corpus CPS-E1 se identifica como TRF-1, el relacionado con el CPS-2 es TRF-2 y finalmente TRF-3 es el transformer entrenado con el corpus CPS-3. En la tabla 5 se muestran los valores de *accuracy* obtenidos en el proceso de entrenamiento, dónde se resalta la fila con el mejor valor de *accuracy* obtenido.

Tabla 4. Parámetros de entrenamiento

Parámetro	Valor
Modelo Transformer	bert-base-uncased
Tokenizador	bert-base-uncased
Longitud de entrada	510
Batch size	32
Número de épocas	3
Clases	4
Optimizador	Adam learning_rate=5e-05, epsilon=1e-08, decay=0.01, clipnorm=1.0
Validación	Accuracy

Tabla 5. Accuracy macro de entrenamiento utilizando Transformers

Transformer	*Accuracy* macro
TRF-1 (*gold standard*)	0.865
TRF-2	0.960
TRF-3	0.813

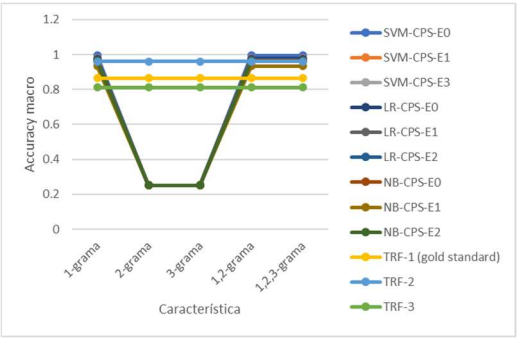


Fig. 4. *Accuracy* macro de entrenamiento

Por otro lado, con el objetivo de observar el compartimiento de los corpus utilizados en el proceso de *fine-tuning* de los diferentes transformers, se realizó el entrenamiento de diversos clasificadores, basados en máquinas de soporte vectorial o SVM, en técnicas probabilísticas o Naive Bayes y en técnicas de regresión, definiendo un esquema k-Fold cross validation con *k* igual a 10 y una caracterización basada en n-gramas [3], dónde *n* es igual 1, 2 y 3, realizando combinaciones entre los *n*-gramas agregando caracterizaciones basadas en la utilización de uni-bi-gramas y uni-bi-tri-gramas, para la construcción del espacio vectorial se utilizó *tf-idf* [3, 11], todo ello, utilizando diferentes procesos basados en la librería *Scikit-Learn* [26] de Python [24].

Comparando los valores de *accuracy* macro se seleccionaron diferentes tres clasificadores, uno por cada tipo de los antes mencionados, que obtuvieron los valores más altos en el proceso de entrenamiento, como máquina de soporte vectorial se seleccionó LinearSVC [3], basado en Naive Bayes se seleccionó la implementación de MultiNomial NaiveBayes [3] y finalmente el clasificador basado en técnicas de regresión seleccionado fue LogisticRegressión [3].

En la tabla 6 se muestran los valores de *accuracy* macro obtenidos en el proceso de entrenamiento para las diferentes caracterizaciones basadas en *n*-gramas, se resaltan los valores que superiores al mejor valor obtenido de *accuracy* macro en el proceso de entrenamiento del transformer TRF-2, que de acuerdo a la tabla 5 es de 0.960.

Finalmente, en la figura 4 se muestra un comparativo de los *accuracy* macro obtenidos en el proceso de entrenamiento de los Transformers y clasificadores, es importante recordar que el *gold standard* propuesto es el transformer TRF-1 y que los corpus son los generados de acuerdo a la sección 4.2 por lo que no se realizaron otros procesos de limpieza y/o lematizado.

4.4. Prueba de Transformers

Para realizar la prueba de los Transformers y de los clasificadores entrenados, se utilizaron cada uno de los diferentes corpus de prueba, en la tabla 7 se muestran los diferentes *accuracy's* obtenidos, dónde se resaltan las filas con los mejores valores de *accuracy* obtenidos. Para el caso de los clasificadores se muestran los resultados de validación utilizando modelos de clasificación basados en una caracterización utilizando

Tabla 6. *Accuracy* macro de entrenamiento utilizando clasificadores tradicionales

Clasificador/Transformer	*Corpus*	*Caracte-Rización*	*Accuracy* macro
LinearSVC (SVM)	CPS-E0	1-grama	0.995
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.995
		1,2,3-grama	0.995
	CPS-E1	1-grama	0.980
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.980
		1,2,3-grama	0.980
	CPS-E2	1-grama	0.976
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.976
		1,2,3-grama	0.976
LogisticRegression (LR)	CPS-E0	1-grama	0.978
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.978
		1,2,3-grama	0.978
	CPS-E1	1-grama	0.964
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.964
		1,2,3-grama	0.964
	CPS-E2	1-grama	0.969
		2-grama	0.25
		3-grama	0.25
		1,2-grama	0.969
		1,2,3-grama	0.969
Multinomial NaiveBayes	CPS-E0	1-grama	0.964
(NB)		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.964
		1,2,3-grama	0.964
	CPS-E1	1-grama	0.934
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.934
		1,2,3-grama	0.934
	CPS-E2	1-grama	0.963
		2-grama	0.250
		3-grama	0.250
		1,2-grama	0.963
		1,2,3-grama	0.963

unigramas, ya que de acuerdo a los resultados obtenidos en el proceso de entrenamiento mostrados en la tabla 6 se observa que los modelos de clasificación mejor entrenados son aquellos que se basan en la utilización de unigramas, es por ello que en la etapa de prueba se utilizan únicamente dichos clasificadores.

En la figura 5 se muestra un diagrama que concentra los diferentes valores de *accuracy* macro obtenidos en el proceso de pruebas, en esta figura se puede observar que los modelos de Transformers obtienen los mejores resultados de validación ya que de acuerdo a la tabla 7 el valor mínimo de *accuracy* macro es de 0.596 mientras que el valor máximo obtenido por un clasificador es de 0.275, en promedio los modelos de Transformers obtienen un *accuracy* macro de 0.793 y los clasificadores de 0.251. En la tabla 6 se muestran los valores de *accuracy* obtenidos en el proceso de pruebas.

5. Conclusiones y trabajo a futuro

Como se puede observar en los diferentes experimentos realizados de acuerdo a la Tabla 7, el transformer que obtiene el mejor promedio de *accuracy*, el cual es 0.927, al ser validado con los 3 conjuntos de pruebas es el transformer TRF-2, siendo el TRF-3 el siguiente con un *accuracy* promedio de 0.769 y finalmente es transformer TRF-1, en promedio tiene un *accuracy* de 0.683, es importante recordar que los Transformers TRF-2 y TRF-3 fueron entrenados, el primero con un corpus, el cual fue procesado eliminando aquellas palabras que no guardaran una relación con su contenido, mientras que en el caso de TRF-3 fue entrenado con sólo los enunciados más características extraídos a partir del corpus de entrenamiento del transformer TRF-2, mientras que el TRF-1 se entrenó utilizando las descripciones de los cómics como fueron descargadas originalmente, con ello se puede identificar que el corpus de entrenamiento utilizado en TRF-2 permite que el transformer modele adecuadamente el fenómeno y que puede considerarse de acuerdo al preprocesamiento aplicado un corpus temático validado por un experto, con el cual también se obtiene el mejor valor de *accuracy* que es de 0.992 utilizando el

conjunto de validación, CPS-P2, que en este caso recibió un pre-procesamiento similar al conjunto de entrenamiento, sin embargo, es importante mencionar que el segundo mejor valor de *accuracy* obtenido fue de 0.950 el cual fue obtenido con el conjunto de validación, CPS-P1, el cual no recibió ningún pre-procesamiento y que mejora el valor de *accuracy* propuesto como *gold standard*, el cual fue de 0.733 obtenido utilizando el transformer TRF-1 con su correspondiente conjunto de validación CPS-P1, es decir, se obtiene una mejora de 0.217, en el caso de los valores de *accuracy* obtenidos utilizando los clasificadores, éstos presentan un valor promedio igual a 0.251, siendo el *accuracy* máximo obtenido de 0.275, por lo que se observa claramente que los modelos de Transformers modelan adecuadamente los corpus con los que se entrenan.

Por otro lado, el mejor valor de *accuracy* obtenido en el proceso de entrenamiento, de acuerdo a la tabla 5, es de 0.960, el cual corresponde al transformer TRF-2, el cual es mejor que el valor obtenido con el *gold standard* propuesto que es de 0.865, por otro lado, de acuerdo a la tabla 7, en las fases de entrenamiento utilizando clasificadores que utilicen una caracterización de 1-grama obtienen valores similares de *accuracy*, sin embargo, estos resultados son engañosos, ya que como se observa en la tabla 7 en el proceso de prueba los clasificados no ofrecen buen resultados, lo cual gráficamente se observa en la figura 5.

Dados los resultados anteriores se puede observar la importancia de los datos de entrada, ya que en los que se resaltó la parte temática permitieron que el transformer fuese entrenado mejor, por lo que una vertiente a este trabajo es la de mejorar los proceso de limpieza modelando el conocimiento del experto, desarrollando modelos de Transformers que permitan realizar este proceso con un mayor de número de datos y entonces realizar el entrenamiento de diferentes modelos de transformer y entonces revisar los resultados.

Otra línea interesante es trabajar con textos más grandes, desarrollando para ello modelos de Transformers que permitan entradas de longitud mayor ya que al día de hoy los modelos BERT permiten reciben entradas de 512 o 1024 tokens y poder así identificar textos que indiquen no sólo

Tabla 7. *Accuracy* macro de pruebas

Transformer/ Clasificador	Corpus	*Accuracy* macro
SVM (CPS-E0)	CPS-P0	0.250
	CPS-P1	0.250
	CPS-P2	0.238
SVM (CPS-E1)	CPS-P0	0.250
	CPS-P1	0.250
	CPS-P2	0.242
SVM (CPS-E2)	CPS-P0	0.233
	CPS-P1	0.250
	CPS-P2	0.254
LR (CPS-E0)	CPS-P0	0.250
	CPS-P1	0.267
	CPS-P2	0.233
LR (CPS-E1)	CPS-P0	0.246
	CPS-P1	0.246
	CPS-P2	0.250
LR (CPS-E2)	CPS-P0	0.233
	CPS-P1	0.242
	CPS-P2	0.250
NB (CPS-E0)	CPS-P0	0.258
	CPS-P1	0.258
	CPS-P2	0.258
NB (CPS-E1)	CPS-P0	0.254
	CPS-P1	0.263
	CPS-P2	0.258
NB (CPS-E2)	CPS-P0	0.275
	CPS-P1	0.267
	CPS-P2	0.271
TRF-1	CPS-1	0.733
(*gold standard*)	CPS-2	0.721
	CPS-3	0.596
TRF-2	CPS-1	0.950
	CPS-2	0.992
	CPS-3	0.838
TRF-3	CPS-1	0.625
	CPS-2	0.783
	CPS-3	0.900

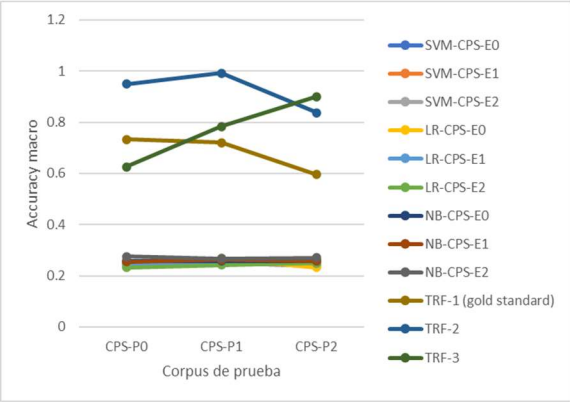


Fig. 5. Accuracy macro de pruebas

temáticas en forma general sino poder identificar sub temáticas implícitas en los textos.

De acuerdo a los trabajos relacionados, principalmente en [21, 22, 23], a los resultados obtenidos y conclusiones presentadas se observan oportunidades de investigación en desarrollar trabajos relacionados con el manejo de los datos de entrada, por lo que como trabajo futuro se propone relacionar los datos de entrada, con diferentes propiedades como son lineales, secuenciales y jerárquicas, con la creación de espacios de *embeddings* temáticos que permitan describir temáticas de acuerdo al corpus de entrada, en el cual, los procesos de limpieza (pre procesamiento) que se apliquen no se realicen de forma semi automática, sino en un entorno automático basándose en la utilización de diferentes modelos de Transformers.

Referencias

1. **Landauer, T. K., Foltz, P. W., Laham, D. (1998).** An introduction to latent semantic analysis. Discourse Processes, Vol. 25, No. 2-3, pp. 259–284, DOI: 10.1080/01638539809545028.
2. **Blei D. M., Ng A. Y. Jordan, M. I. (2003).** latent Dirichlet allocation. Journal of Machine Learning Research, Vol. 3, pp. 993–1022.
3. **Manning, C. D., Schütze, H. (1999).** Foundations of statistical natural language processing. Cambridge, Massachusetts, MIT Press.
4. **Gelbukh, A. (2018).** Introduction to the thematic issue on natural language processing. Computación y Sistemas, Vol. 22, No. 3, pp. 721–727. DOI: 10.13053/cys-22-3-3032.
5. **Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao J. (2022).** Deep learning based text classification: A comprehensive review. ACM Computing Surveys Vol. 54, No. 3, pp 1–40, DOI: 10.1145/3439726.
6. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017).** Attention is all you need. Conference Advances in Neural Information Processing Systems, pp. 5998–6008.
7. **Honnibal, M., Montani, I. (2017).** spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. Sentometrics Research, Vol. 7, No. 1, pp. 411–420.
8. **Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C. D. (2020).** Stanza: A Python natural language processing toolkit for many human languages. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 101–108.
9. **Krishnamurthy, J., Mitchell, T. (2011).** Which noun phrases denote which concepts? Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 570–580, Portland, Oregon, USA. Association for Computational Linguistics, pp. 570–580.
10. **Gelbukh, A., (2010).** Procesamiento de lenguaje natural y sus aplicaciones. Komputer Sapiens, Sociedad Mexicana de Inteligencia Artificial, Vol. I, pp. 6–11.
11. **Carrera-Trejo, V., Sidorov, G., Miranda-Jiménez, S., Moreno-Ibarra, M., Cadena-Martínez, R. (2015).** Latent dirichlet allocation complement in the vector space model for Multi-Label text classification. International Journal of Combinatorial Optimization

Problems and Informatics, Vol. 6, No. 1, pp. 7–19.

12. **Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013).** Efficient estimation of word representations in vector space. CoRR, abs/1301.3781. DOI: 10.48550/arXiv.1301.3781.
13. **Sidorov, G. (2019).** Syntactic n-grams in computational linguistics. Springer, pp. 92. DOI: 10.1007/978-3-030-14771-6.
14. **Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, K. N., Asgari-Chenaghlu, M., Gao, J. (2021).** Deep learning-based text classification: A comprehensive review. ACM Computing Surveys, Vol. 54, No. 3, pp. 1–40, DOI: 10.1145/3439726.
15. **Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et. al. (2020).** Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.
16. **Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. ArXiv, abs/1810.04805. DOI: 10.48550/arXiv.1810.04805.
17. **Zahera, H. M. (2019).** Fine-tuned BERT model for multi-label tweets classification. TREC.
18. **Tang, T., Tang, X., Yuan, T. (2020).** Fine-tuning BERT for multi-label sentiment analysis in unbalanced code-switching text. IEEE Access, Vol. 8, pp. 248–256, DOI: 10.1109/ACCESS.2020.3030468.
19. **Bhamare, B. R., Prabhu, J. (2021).** A multilabel classifier for text classification and enhanced BERT system. Revue d'Intelligence Artificielle, Vol. 35, No. 2, pp. 167–176. DOI:10.18280/ria.350209.
20. **Chang, W., Yu, H., Zhong, K., Yang, Y., Dhillon, I. S. (2019).** X-BERT: eXtreme multi-label text classification with using bidirectional encoder representations from transformers. arXiv:Learning.
21. **Lin, Y., Tan, Y. C., Frank, R. (2019).** Open sesame: Getting inside BERT's linguistic knowledge. ArXiv, abs/1906.01698. DOI: 10.48550/arXiv.1906.01698.
22. **Abuzayed, A., Al-Khalifa, H. S. (2021).** BERT for Arabic topic modeling: An experimental study on BERTopic technique. Procedia Computer Science. Vol. 189, pp. 191–194. DOI: 10.1016/j.procs.2021.05.096.
23. **Moody, C. E. (2016).** Mixing Dirichlet topic models and word embeddings to make lda2vec. ArXiv, abs/1605.02019. DOI: 10.48550/arXiv.1605.02019.
24. **Van-Rossum, G., Drake, F. L. (2009).** Python 3 reference manual. Scotts Valley, CreateSpace100 Enterprise Way, Suite A200Scotts ValleyCA, pp. 242.
25. **Martín, A., Agarwal, A., Barham,P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et. al. (2015).** Tensor Flow: Large-scale machine learning on heterogeneous systems. DOI: 10.48550/arXiv.1603.04467
26. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., et. al. (2011).** Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, Vol. 12, pp. 2825–2830.

Article received on 13/03/2022; accepted on 18/04/2022. Corresponding author is Jorge Victor Carrera-Trejo.

ISSN 2007-9737

A Product Review Writing Recommender System based on LDA and TF-IDF

Pradnya Bhagat, Jyoti D. Pawar

Goa Business School,
Goa University,
India

{dcst.pradanya, jdp}@unigoa.ac.in

Abstract. Twitter is a micro-blogging platform where people broadcast their views and opinions to fellow users in crisp messages called Tweets. However, the platform's format of restricted character limit makes it challenging for many users to express their views exhaustively. The paper proposes a recommender system to help in writing effective product review Tweets within the restricted character limit of Twitter. The approach is divided into two phases where, the first phase uses the Latent Dirichlet Allocation (LDA) algorithm to find pivotal features from the training corpus and suggests them to the users while writing new Tweets. In the second phase, the approach suggests the most appropriate opinion words to describe the respective features by using an method based on the occurrence frequency of opinion words and TF-IDF. The evaluation results show significant improvement in the quality of product review Tweets. The percentage of good reviews corresponding to a parameter such as correct usage of feature words is found to be 17.85% higher, whereas an improvement of 23.22% is reported with regard to the correct use of opinion words using the generated recommendations.

Keywords. Recommender systems, product review tweets, feature words, topics, opinion polarity, opinion intensity, latent dirichlet allocation, term frequency, inverse document frequency.

1 Introduction

Twitter is a social media platform that allows people to broadcast their views and opinions in the form of brief messages known as Tweets. Tweets are basically short messages with a limit of 280 characters.

The platform differs from most other social media platforms in a way that the relationships formed can be asymmetric [11, 9]. For instance, if user A follows user B, user A receives all Tweets from user B, whereas user B does not receive user A's Tweets unless he/she is interested in user A's Tweets (unless user B follows user A), thereby preserving his/her interests.

Hence, Twitter acts as an interest-specific service that allows users to preserve their interests and at the same time connect with people all over the world. Furthermore, since it is a free service with a worldwide presence, people all over the globe have started using the platform to share their views and benefit from other similar-minded people.

This has driven companies as well to make their presence felt on the platform and connect with customers all over the world. Many companies have dedicated customer-relation profiles on the platform to advertise new products and to communicate directly with the public [27].

A trend has also been springing up on Twitter where people tweet about various products/services used by them and exchange their ideas and opinions with other followers. Fig. 1 shows some sample Tweets maintaining the anonymity of the authors.

As seen, people effectively use the platform to lodge complaints or express satisfaction about goods and services used by them. Twitter being a broadcasting medium, the tweeted message has the power to instantly reach the destined company

Fig. 1. Example Tweets

and thousands of other customers, who in turn can base their purchase decisions on the read Tweet. Also, the short message format of the platform assures the companies that the messages tweeted on Twitter are concise yet informative, unlike on other social media platforms/product reviews on ecommerce websites.

This drives the companies to give immediate attention to the suggestions and complaints raised by the customers. As a result, Twitter is emerging as an effective medium for the public to make their problems heard and addressed. A study has shown that as many as 83% of customers who used Twitter for customer service had their issues addressed and fixed [17].

However, unlike other social media platforms like Facebook or Instagram, Twitter has still not succeeded in reaching the masses. One of the major reasons is attributed to the 280 character short message format, which acts as a great challenge to express one's opinion in brief. Specifically, for people with native languages other than English, expressing one's opinion in a constrained manner using English is not always feasible.

Also, addressing of technical issues specifically, in brief, requires the use of domain specific terms; a terminology which most of the users may not be well versed with. In character intensive languages like English, where the language uses several characters just to convey one word, the users easily run out of characters while expressing themselves [20].

Moreover, although many applications exist in the market to increase the users' engagement on Twitter, we observe that none specifically focus on simplifying the process of writing the Tweets.

As a result, even after being an extremely powerful medium, its use has not been realized to its complete potential. This paper is an extension of the research work presented in [4]; an approach designed to aid people in composing better quality product review Tweets for a specific domain of products by recommending them with various product-related features and feature specific opinion words.

The recommendations to help generate review Tweets for our study are mined from the reviews on Amazon.com with the domain being restricted to cell phones and its related accessories [10, 14]. Since there is no restriction in the number of words used to write a review on Amazon.com, an average Amazon Review is more elaborate and more informative than a Twitter Tweet; hence we select it as the training corpus.

The approach works in two phases; in the first phase, we extract product features and topics from the corpus using the Latent Dirichlet Allocation (LDA) [5] algorithm. Features are the attributes that describe a particular product and a group of features together form a topic.

In the second phase, the system calculates the polarity and the intensity of opinion words and suggests the appropriate opinion words corresponding to a particular feature using Term Frequency- Inverse Document Frequency (TF-IDF) statistical measures.

Polarity of a opinion word refers to the positive/negative/neutral orientation of the word and intensity refers to how intensely the word expresses its corresponding polarity. The orientation of the sentiment of a word can largely depend on the domain in which it is used. A word that expresses a positive opinion in one domain can express negative opinion in another domain.

For example, the word "unpredictable" has shown to denote positive sentiments in the movie domain; but the same word has been shown to carry a negative sentiment in the domain of automobiles [21]. Hence, in this paper we propose a method to calculate domain dependent polarity of opinion words overcoming the limitations of a generic sentiment lexicon. The experimental study is conducted on a set of 28 users.

The users were asked to write two Tweets about the cell phones they use in 280 characters as per the specifications of Twitter on a dummy website. The first Tweet is written by the users without the aid of any recommendations. In the second Tweet, the users are provided with the recommendations generated by our system.

Both the Tweets, i.e., the one written without recommendations and the one written with the help of recommendations are evaluated by two in-house subject experts, chosen by us as judges. The generated Tweets are analyzed based on the following aspects: use of correct feature words, appropriate opinion words used to describe the features, quality of the sentences constructed, and the overall helpfulness/usefulness of the composed review Tweets.

The analysis results have shown that the quality of the reviews improved in all the aspects considered for evaluation with the introduction of the generated recommendations. The remainder of this paper is organized as follows: Section 2 presents a review on the role of social media platforms in the success of brands. The section also reviews various methods employed in the literature for feature extraction and sentiment analysis from user reviews.

Section 3 describes the proposed two-step approach for feature and opinion word recommendation. Section 4 elaborates on the implementation details explaining the tools and techniques used in the implementation of the proposed method. Section 5 presents the data sets used in the study and the experimental evaluation, section 6 asserts and discusses on the results obtained and finally section 7 states the conclusion and future work.

2 Literature Review

Social media has been a continuously evolving field with every platform rolling out new features every few days to retain their position in the market. As a result, it has been extensively studied in literature trying to cover its varied aspects. [16] explores the power of social media platforms in the incessant expansion of brands by engaging the consumers on networking platforms for regular feedback. The study shows that a brand that collaborates with its consumers online can create, or modify, it's Corporate Social Relationship (CSR) strategies to fit consumer needs in a better way. [1] does a comparative study between various social media platforms and also states that the time spent by the users on Twitter is the least compared to all social networking platforms.

[6, 7] demonstrate one of the first works in guiding users while writing content on the internet. It presents the work as a browser plugin that can be used with e-commerce websites to help users write better quality reviews. The system uses association rule mining to recommend various product features to users while writing reviews.

[8] presents an incremental work of Reviewer's Assistant and reports the use of LDA [5], for detection of keywords. [3] presents a survey and comparison of some of the major methods used by researchers for feature extraction from textual product reviews. [12] proposes a probabilistic rating framework that mines user preferences from reviews and maps them to a rating scale.

The algorithm is a step towards improving Collaborative Filtering (CF) algorithms that allow text reviews to predict user preferences. [26] works on the problem of identifying feature nouns that also imply opinions. The method works by determining the polarity of feature words by identifying the opinion words that modify the feature and analyzing the surrounding context.

[22] goes beyond the zero-one polarity and tries to compare adjectives that share a similar sentiment orientation using a semi-supervised approach. The approach tries to work on the FrameNet data [2] and derives the polarity-intensity ordering among adjectives for specific categories.

The presented approach is not entirely corpus dependent, hence the approach even attempts to find the intensity of sentiment words absent in the corpus. [21] proposes a scheme to detect domain dedicated sentiment words through an application of Chi-Square test based on the difference in the counts of the word in positive and negative documents.

3 Proposed Methodology

The work addresses the challenge of assisting users in writing better quality product review Tweets using an approach based on a recommendation of two phases:

1. Recommending Product Feature Words: Recommends specific feature words to describe a product.
2. Recommending Appropriate Opinion Words: Recommends correct opinion words to describe the corresponding product features.

3.1 Recommending Product Feature Words

Reviews are broken into sentences, assuming that a single sentence describes a single feature or a topic. The sentences are Parts-of-Speech (POS) tagged [23] [24] to identify the various parts of speech in the review.

Next, we identify Nouns as the parts of speech that convey the product features most of the time. The challenge lies in distinguishing feature nouns from non-feature nouns and we are interested in only the former.

We proceed our experiment on an assumption that feature nouns occur in close proximity to adjectives since the users are interested in expressing their opinions about them. On the other side, this is not the case with non-feature nouns. For example, given a sentence:

My friend advised me to buy this awesome mobile because it has this stunning look and attractive features. [4] POS tagging of the above sentence would give us;

My_PRP friend_NN advised_VBD me_PRP to_TO buy_VB this_DT awesome_JJ mobile_NN because_IN it_PRP has_VBZ this_DT stunning_JJ look_NN and_CC attractive_JJ features_NNS

The nouns occurring in the above sentence are *friend*, *mobile*, *look*, and *features*. Out of these, the nouns we would be interested in are *look*, *mobile* and *features* since they belong to the domain of cell phones.

As seen, the nouns *mobile*, *look* and *features* have some adjectives associated with them since the users want to express their opinions about the features but the noun *friend* does not have any adjectives associated with it. As it is not a feature related to mobile phones, the user is not interested in expressing an opinion on it in a review post.

We utilize this observation to differentiate candidates to feature nouns from non-feature nouns. Next, the position of the candidate feature nouns in the sentence is retained and the pre-processed file is given to the LDA algorithm. LDA [5] algorithm helps to identify latent topics from a dataset.

Topics are basically formed by a group of words related to each other. In the case of a review dataset, it is a group of feature words related to each other forming a topic. For example, feature words like *flash, pixel, front,back, digital, wide* etc., can be grouped together under a single latent topic *camera*.

These related feature words can be used in making the Tweet on the topic more informative. Thus, whenever a person starts composing a Tweet on any topic, the extracted relevant features of the same topic get displayed to the user as recommendations which helps in making the Tweet more informative. Algorithm 1 summarizes the process in the form of a pseudocode.

Algorithm 1 Identify Product features and feature topics

Input: Product reviews from the corpus
Output: Product features and feature topics

```
for each review R in the corpus do
    for each sentence S in R do
        for each word W in S do
            POS_Tag(W)
            if POS_Tag(W) == NN then
                if POS_Tag(W) preceded by JJ then
                    Retain W in S as Candidate Feature Noun
                else
                    Delete W
Product features and topics= LDA (Review Sentences with
    Candidate Feature Nouns)
```

3.2 Recommend Appropriate Opinion Words

We take the POS Tagged reviews and extract all the adjectives that occur in the dataset. The review dataset has a numeric rating associated with every review given by the reviewers in addition to the review text.

The rating is in the form of stars and it ranges from 1 star to 5 stars. We group the reviews according to the number of stars associated with the review.

To find the polarity of the sentiment words, we take the adjectives found using POS tagging and find their occurrence across all five groups of reviews. Algorithm 2 elaborates on the process of classification of sentiment words according to the star rating.

Algorithm 2 Identify the domain based polarity of sentiment words

Input: POS Tagged Review text, star rating associated with the reviews. **Output:** Sentiment words and associated polarity.

for each review R in the POS tagged Reviews **do**
  for each sentence S in R **do**
    for each word W in S **do**
      if $POS\_Tag(W) == JJ$ **then**
        Add W to the Adjectives_List
for each review R in the corpus **do**
  if R has 1 star rating **then**
    Classify R as 1 star review.
  else if R has 2 star rating **then**
    R as 2 star review
  else if R has 3 star rating **then**
    Classify R as 3 star review.
  else if R has 4 star rating **then**
    Classify R as 4 star review
  else if R has 5 star rating **then**
    Classify R as 5 star review
  else
    Discard R
for `Adjective A from Adjectives_List` **do**
  if if A occurs majority times in 4 or 5 star reviews **then**
    Label A as positive
  else if if A occurs majority times in 1 or 2 star reviews **then**
    Label A as negative
  else
    Label A is neutral

Next, we identify the occurrence frequency of all positive, negative and neutral sentiment words with respect to every feature.

To find feature specific adjectives, we calculate the Term Frequency-Inverse Document Frequency (TF-IDF) [18] of sentiment adjectives in each sentiment category (positive, negative and neutral) with respect to each feature.

The sentiment adjective having the highest TF-IDF is considered to be the most intense sentiment word in that category for that feature.

Algorithm 3 Generating feature based sentiment words recommendations

Input: 1. Product Features Extracted using Algorithm 1 denoted as F,
2. Set of Sentiment categories: Positive, Negative, Neutral Extracted using Algorithm 2 referred as S
Output: Appropriate sentiment words with intensities for features identifies.

for `Feature` F_i `in` F **do**
  for `Sentiment Category` S_i `in` S **do**
    for Sentiment Word SW_i in S_i **do**

$$TF(SW_{iFi}) = \frac{\text{Frequency}(SW_i)_{Fi}}{\sum_{i=1}^{n}(SW_i)_{Fi}}$$

$$IDF(SW_{iFi}) = \log_e \frac{\text{Number of features}}{\text{Features with} SW_i}$$

$$TF - IDF(SW_{iFi}) = F(SW_{iFi}) * IDF(SW_{Fi})$$

for `Feature` Fi `in` F **do**
  for Sentiment Category Si in S **do**
    Sort SW based on $TF - IDF(SW_{iFi})$

The top 25 sentiment adjectives with highest TF-IDF scores in positive, negative and neutral sentiment categories for every features words are retained.

These resultant opinion words are the candidates to be shown as recommendations to the users while writing their opinions about respective features in the Tweets. Algorithm 3 elaborates the process in the form of a pseudocode.

4 Implementation Details

The proposed work is implemented using Python programming language [25]. The pre-processing operations and POS tagging is carried out using the Natural Language Processing ToolKit (NLTK) [13].

Gensim [19] topic modelling library is used for LDA implementation. It is an open source library for topic modelling and Natural Language Processing tasks using machine learning techniques.

	Topic 1				
	battery	**device**	**phone**	**charger**	**charge**
Positive	good	great	great	great	good
	great	good	good	portable	long
	low	portable	smart	good	great
	new	easy	new	nice	fast
	portable	compatible	easy	compact	free
Negative	low	little	little	full	little
	long	full	hard	long	cheap
	dead	high	last	last	high
	little	long	full	little	full
	bad	second	difficult	less	long
Neutral	extended	other	other	much	other
	external	mobile	same	quick	usb
	spare	electronic	much	initial	dual
	original	multiple	more	single	same
	extra	same	able	same	original

	Topic 2			
	case	**protect**	**color**	**plastic**
Positive	great	good	bright	hard
	good	great	great	clear
	nice	adequate	nice	cheap
	protective	decent	vibrant	soft
	hard	excellent	favourite	flexible
Negative	hard	little	little	hard
	little	full	dark	cheap
	cheap	hard	hard	little
	bad	top	wrong	flimsy
	bulky	less	cheap	rigid
Neutral	other	much	fifferent	thin
	slim	more	black	thick
	much	minimal	other	glossy
	thin	added	red	small
	more	extra	more	inner

	Topic 3		
	protector	**screen**	**part**
Positive	good	good	best
	great	clear	top
	easy	great	good
	clear	easy	hard
	first	responsive	soft
Negative	little	little	top
	hard	hard	hard
	last	difficult	little
	cheap	fast	back
	back	full	bad
Neutral	screen	touch	most
	other	other	bottom
	bubble	anti-glare	plastic
	same	small	rubber
	anti-glare	big	other

	Topic 4				
	cable	**port**	**car**	**cord**	**tip**
Positive	great	accessible	great	retractable	soft
	good	easy	good	long	retractable
	retractable	great	long	great	replaceable
	nice	top	new	good	great
	long	nice	easy	nice	good
Negative	short	little	long	short	little
	little	top	little	long	hard
	long	second	second	little	last
	cheap	full	cheap	flat	
	flat	high	last	heavy	
Neutral	usb	usb	rental	usb	stylus
	other	dual	other	other	mesh
	included	other	dual	longer	pen
	own	micro-usb	usb	coiled	ear
	micro-usb	standard	different	extra	micro-knit

	Topic 5				
	quality	**product**	**time**	**price**	**sound**
Positive	good	great	long	great	good
	high	good	first	good	great
	great	excellent	good	low	clear
	excellent	nice	hard	reasonable	better
	better	pleased	great	cheap	excellent
Negative	high	cheap	long	low	lou
	poor	little	hard	cheap	little
	low	bad	little	little	full
	cheap	high	last	bad	high
	bad	last	short	high	low
Neutral	sound	other	same	same	tiny
	build	sample	few	more	other
	audio	similar	several	retail	ear
	higher	more	many	other	overall
	call	same	more	current	small

Fig. 2. Features and corresponding sentiment words

5 Dataset and Experimental Evaluation

The dataset for the study presented is sourced from Amazon.com and is made available by Stanford Network Analysis Project (SNAP) [10, 14]. It is a collection of Amazon.com reviews and the associated metadata on Cell Phones and its related Accessories spanning from May 1996 to July 2014. We consider only the review text from the product and the product ratings from the dataset for our experiment. A group of 28 students with computer science background was selected for the experimental study.

The age range of the selected students for the experimental study is from 22 to 24 years, as this age group falls in the category of users found to be most enthusiastic about mobiles and electronic accessories.

Out of the 28 students, 16 were males and remaining 12 were females. The students were asked to write reviews about the mobile phone used in the form of Tweets on a dummy website created by us expressing their satisfaction or dissatisfaction about their currently used phone.

The experiment was carried out in two phases, wherein; in the first phase, the students were told to write Tweets independently without any aid.

We imposed a restriction of 280 characters as presented by Twitter.

Without Recommendations	***With Recommendations***
I bought my ASUS Zenfone 2 laser way back in the year 2015. It is still working flawlessly.There is no hang,no virus till date.I recommend those who want to buy budgeted mobile phone under 10000 Rs.	*Asus zenfone 2 laser battery is enough.Device is perfect for middle range mobile.Mobile touch is still working flawlessly even after 3 years.It has corning gorilla glass 3 display which helps your device screen from damage.*
I have Redmi Note 3, Until now I'm very satisfied with my phone. It has all the features necessary which is required to carry out all the basic functions. The Only thing I want to be better is the OS. It's full of bloatware, but the hardware suffices all the need.	*I have Redmi Note 3, which in a lot of ways is still a good phone. The Battery is lasting and comes through the whole day usage without any problems. It charges in decent pace, and the screen is very responsive. The device using experience is also good and new apps runs smoothly.*
xaomi redmi note 4 is quite good on battery and overall performance. I will recommend you to buy this phone if ur looking for good phone in budget category.	*Xaomi Redmi note 4 a good device with excellent battery back up, fast charger, descent sound quality, responsive screen. I will recommend u to buy this phone.*
Samsung grand prime has a major problem of storage plz increase it.	*battery is removable and good ,phone is good only problem is the storage*
Superb build quality. We easily get Android update. Stock camera is not so good. Battery life is poor.	*Battery life is fine. Phone supports fast charging but the charger is slow charger. Case is durable and protective. Color is awesome. Plastic is durable. Screen is responsive. Cable is of high quality. Price is reasonable.*

Fig. 3. Some example Tweets written by subjects without and with recommendations

Table 1. Features and topics discovered using LDA algorithm

Topic 1	battery	device	phone	charger	charge
Topic 2	case	protect	color	plastic	phone
Topic 3	protector	screen	thing	one	part
Topic 4	cable	port	car	cord	tip
Topic 5	quality	product	time	price	sound

In the second phase, again the students were told to write another Tweet in 280 characters, but this time they were provided with the recommendations in the form of features and opinion words mined using the system.

Two human judges were assigned to judge the quality of the Tweets written by each student independently. The judges were given 4 parameters to judge every Tweet on a two-point scale (1 - poor or average, 2 - good).

The parameters selected for evaluation are are:

1. Use of correct feature words in the Tweet.
2. Appropriate use of adjectives/sentiment words to describe the features.
3. Overall quality of the sentences constructed.
4. Helpfulness/usefulness quotient of the review.

The agreement between both the judges was validated using Cohen's Kappa [15] inter rater reliability measure.

Table 2. Sentiment words and corresponding Normalized occurrence frequencies across review categories

1 Star	2 Star	3 Star	4 Star	5 Star
less	right	bad	last	excellent
1	1	0.57269	0.622323	1
poor	second	cheap	clear	happy
1	0.452949	0.355925	0.592842	1
second	bad	clear	good	perfect
0.547051	0.297234	0.277108	0.422779	0.800259
bad	cheap	full	little	good
0.42731	0.267953	0.235836	0.301183	0.703959
cheap	last	good	nice	great
0.376122	0.170843	0.172343	0.281524	0.695143

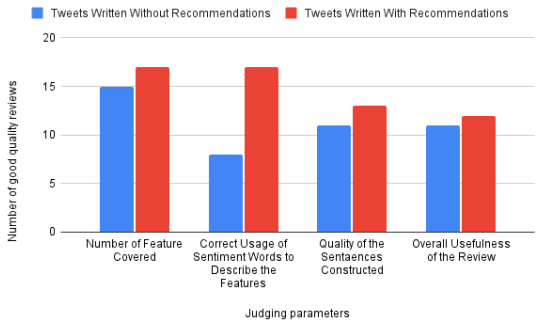


Fig. 4. Graph of ratings given by Judge 1

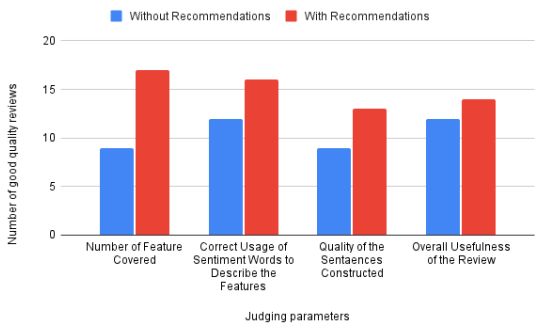


Fig. 5. Graph of ratings given by Judge 2

6 Results and Discussions

Table 1 displays the features and topics obtained using LDA algorithm. As it can be seen in the first topic, features like *battery, phone, device, charger and charge* have got displayed which signifies that the features in topic 1 are related to each other i.e. they are cohesive and together form one topic. Similarly features in Topic 2 correspond to the *phone case* used for protection.

The same explanation applies to remaining topics discovered. These are the features that were displayed to the users while writing on the corresponding topics by dropping the stop words like *thing, one*, etc. using manual filtering. The second phase of the experiment deals with discovering the domain specific polarity of the sentiment words without the use of a sentiment lexicon.

The normalized occurrence frequency of first five sentiment words across review categories are shown in Table 2. As can be seen, words like *less, poor, bad* occur most of the times in 1 or 2 star reviews hence we classify them as negative sentiment words in our dataset; whereas 4 or 5 star reviews have words like *excellent, happy, perfect* and hence these words are classified as positive words.

The results of the next step that deals with finding the most appropriate positive, negative and neutral opinion words for each feature using TF-IDF measure are shown in Fig. 2. For example; in Topic 2, for a feature by name *case*, the positive words discovered are *great, good, nice, protective, hard*; whereas, the negative sentiment words discovered are *hard, little, cheap, bad, bulky*.

These extracted words are recommended as sentiment words to the users to help them in explaining the respective features more effectively. The next Fig. 3 shows some example Tweets written by the subjects with and without recommendations. As can be seen, Tweets written with the help of recommendation clearly show more effective usage of feature words and sentiment words to describe them thereby making the written Tweet more effective.

The graphs in Fig. 4 and Fig. 5 show the performance of the students in writing Tweets; with and without the aid of the system as evaluated by the judges.Only good quality review Tweets were considered for plotting the graphs disregarding the poor and average quality review Tweets.

Table 3. Agreement between the judges as calculated by Cohen's Kappa

Categories	With Recommendations	Without Recommendations
Cohen's Kappa Measure	0.598361	0.515152

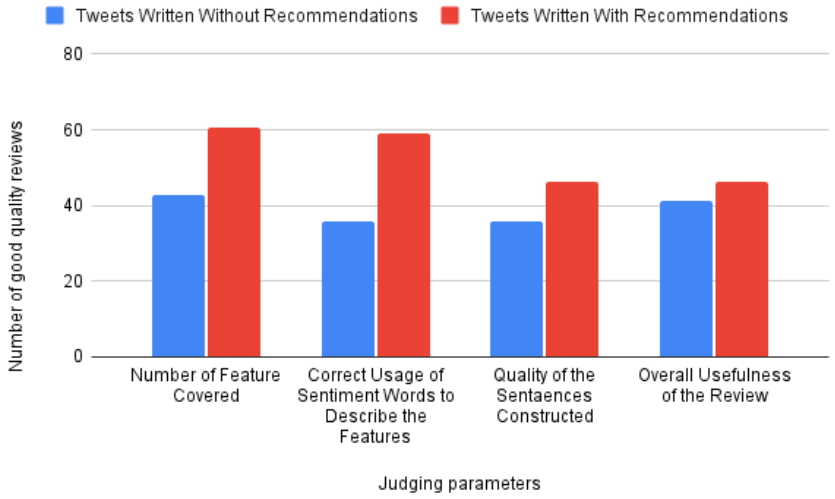


Fig. 6. Graph of average of the ratings by Judge 1 and Judge 2

This can be justified from the fact that the poor and average quality reviews have insignificant contribution in guiding other users. The parameters used for comparison of reviews were number of feature words used in a particular review, correct usage of opinion words to describe the respective features, quality of the sentences constructed and overall usefulness of the review.

It can be seen from the graph that the percentage of good quality reviews written using recommendations provided by proposed methodology are higher for all the parameters considered.

For instance, according to judge 1 in Fig. 4, only 8 people were able to make appropriate use of sentiment words while writing reviews without use of any recommendations.

Whereas, after the use of the recommender system 17 people wrote better quality reviews in terms of correct usage of sentiment words. I.e., the percentage of good reviews for a parameter such as correct usage of sentiment words was found to be 32% higher with recommendations than without.

Similar improvement is noted in all other parameters considered for evaluation as can be seen from the graph. Also, the ratings given by Judge 2 are plotted in graph shown in Fig. 5. The product review Tweets written using the recommendations generated by the system have scored higher across all four categories.

These graphs serve as a strong evidence for the validation and usefulness of the proposed methodology. Furthermore, the validity of the ratings given by the judges is confirmed by the agreement between the two judges using Cohen's Kappa statistical measure as shown in Table 3.

Since the Cohen's Kappa score is more than 0.5 in both categories, we can clearly say that the agreement between both the judges is validated and we proceed to find the average of the ratings given by both the judges. The graph of the average ratings given by both the judges is given as Fig. 6.

As can be seen in Fig. 6 there is an overall improvement of 17.85% is observed with regard to correct usage of feature words using the recommendations. Consequently, usage of appropriate sentiment words improved a 23.22%.

Also, Both the judges found a boost of 10.72% and 5.36% with respect to the quality of the sentences constructed and the helpfulness quotient of the reviews respectively with the use of the proposed system.

7 Conclusion and Future Work

In this paper, we presented a method to help Tweeter users compose better quality product review Tweets in the restricted character limit. The method aims to generate effective and well-composed product review Tweets that are expected to help users get their Tweets desired attention and his/her problems being heard and addressed.

The approach uses LDA algorithm that helps combine related features from the training corpus and displays them as suggestions to a user while composing new Tweets. The paper also shows that the feature based polarity and intensity of the sentiment words can be calculated based on frequency of occurrence and the TF-IDF score in the dataset and we need not rely on an universal sentiment lexicon.

The Experimental results confirm that the presented method is promising to help users write better quality product review Tweets. The ratings given by both the judges and the validated inter rater agreement indicate that the Tweets written with the use of the recommendations generated by the system were of better quality than the ones written without using the system. As future work we intend to incorporate cross domain knowledge transfer in the proposed work as getting manually labelled data for every domain is not always feasible.

Acknowledgments

This publication is an outcome of the research work supported by Visvesvaraya PhD Scheme, MeitY, Govt. of India (VISPHD-MEITY-2002).

References

1. **Alhabash, S., Ma, M. (2017).** A tale of four platforms: Motivations and uses of Facebook, Twitter, Instagram, and Snapchat among college students? Social media + society, Vol. 3, No. 1.
2. **Baker, C. F., Fillmore, C. J., Lowe, J. B. (1998).** The Berkeley Framenet project. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, pp. 86–90.
3. **Bhagat, P., Pawar, J. D. (2018).** A comparative study of feature extraction methods from user reviews for recommender systems. Proceedings of the ACM India joint international conference on data science and management of data, pp. 325–328.
4. **Bhagat, P., Pawar, J. D. (2021).** A two-phase approach using LDA for effective domain-specific tweets conveying sentiments. In Computational Intelligence and Machine Learning. Springer, pp. 79–86.
5. **Blei, D. M., Ng, A. Y., Jordan, M. I. (2003).** Latent Dirichlet allocation. Journal of Machine Learning Research, Vol. 3, pp. 993–1022.
6. **Dong, R., McCarthy, K., O'Mahony, M., Schaal, M., Smyth, B. (2012).** First demonstration of the intelligent reviewer's assistant. Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, pp. 337–338.
7. **Dong, R., McCarthy, K., O'Mahony, M., Schaal, M., Smyth, B. (2012).** Towards an intelligent reviewer's assistant: recommending topics to help users to write better product reviews. Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, pp. 159–168.
8. **Dong, R., Schaal, M., O'Mahony, M. P., McCarthy, K., Smyth, B. (2012).** Unsupervised topic extraction for the reviewer's assistant. International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer, pp. 317–330.
9. **Gupta, P., Goel, A., Lin, J., Sharma, A., Wang, D., Zadeh, R. (2013).** Wtf: The who to follow service at twitter. Proceedings of the 22nd international conference on World Wide Web, pp. 505–514.
10. **He, R., McAuley, J. (2016).** Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. Proceedings of the

25th international conference on world wide web, pp. 507–517.

11. **Kwak, H., Lee, C., Park, H., Moon, S. (2010).** What is twitter, a social network or a news media? Proceedings of the 19th international conference on World wide web, pp. 591–600.
12. **Leung, C. W.-K., Chan, S. C.-F., Chung, F.-L., Ngai, G. (2011).** A probabilistic rating inference framework for mining user preferences from reviews. World Wide Web, Vol. 14, No. 2, pp. 187–215.
13. **Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., McClosky, D. (2014).** The stanford coreNLP natural language processing toolkit. Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55–60.
14. **McAuley, J., Targett, C., Shi, Q., van den Hengel, A. (2015).** Image-based recommendations on styles and substitutes. Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp. 43–52.
15. **McHugh, M. L. (2012).** Interrater reliability: the kappa statistic. Biochemia medica, Vol. 22, No. 3, pp. 276–282.
16. **Okazaki, S., Plangger, K., West, D., Menéndez, H. D. (2020).** Exploring digital corporate social responsibility communications on twitter. Journal of Business Research, Vol. 117, pp. 675–682.
17. **Picazo, S. (2016).** Customer service on twitter and the impact on brands.
18. **Ramos, J. (2003).** Using TF-IDF to determine word relevance in document queries. Proceedings of the first instructional conference on machine learning, volume 242, Citeseer, pp. 29–48.
19. **Řehřek, R., Sojka, P., others (2011).** Gensim—statistical semantics in python. Retrieved from genism. org.
20. **Rosen, A., Ihara., I. (2017).** Giving you more characters to express yourself.
21. **Sharma, R., Bhattacharyya, P. (2013).** Detecting domain dedicated polar words. Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 661–666.
22. **Sharma, R., Gupta, M., Agarwal, A., Bhattacharyya, P. (2015).** Adjective intensity and sentiment analysis. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2520–2526.
23. **Taylor, A., Marcus, M., Santorini, B. (2003).** The penn treebank: an overview. Treebanks, pp. 5–22.
24. **Toutanova, K., Klein, D., Manning, C. D., Singer, Y. (2003).** Feature-rich part-of-speech tagging with a cyclic dependency network. Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 252–259.
25. **Van Rossum, G., others (2007).** Python programming language. USENIX annual technical conference, volume 41, pp. 1–36.
26. **Zhang, L., Liu, B. (2011).** Identifying noun product features that imply opinions. Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies, pp. 575–580.
27. **Zhang, M., Jansen, B. J., Chowdhury, A. (2011).** Business engagement on twitter: a path analysis. Electronic Markets, Vol. 21, No. 3, pp. 161.

Article received on 23/09/2021; accepted on 06/05/2022.
Corresponding author is Pradnya Bhagat.

ISSN 2007-9737

Comprehensive Performance Analysis on Classical Machine Learning and Deep Learning Methods for Predicting the COVID-19 Infections

Prabhat Kumar, Selvam Suresh

Banaras Hindu University, Institute of Science,
Department of Computer Science,
India

{prabhat.kumar13, suresh.selvam}@bhu.ac.in

Abstract. The COVID-19 (coronavirus disease) has been declared a pandemic throughout the world by the WHO (World Health Organization). The number of active COVID-19 cases is increasing day by day and clinical laboratory findings consume more time while interpreting the COVID-19 infected result. There are limited treatment facilities and proper guidelines for reducing infection rates. To overcome these limitations, the requirement of clinical decision support systems embedded with prediction algorithms is raised. In our study, we have architected the clinical prediction system using classical machine learning, deep learning algorithms, and experimental laboratory data. Our model estimated which patients were likely infected with COVID-19 disease. The prediction performances of our models are evaluated based on the accuracy score. The experimental dataset has been provided by Hospital Israelita Albert Einstein at Sao Paulo, Brazil, which included the records of 600 patients from 18 laboratory findings with 10% COVID-19 disease infected patients. Our model has been validated with a train-test split approach, 10-fold cross-validation, and AUC-ROC curve score. The experimental results show that the infected patients with COVID-19 disease are identified at an accuracy of 91.88% through the deep learning method (Convolutional Neural Network (CNN)) and 89.79 % through classical machine learning (Logistic Regression) respectively. This high accuracy is evidence that our prediction model could be readily used for predicting the COVID-19 infections and assisting the health experts in better diagnosis and clinical studies.

Keywords. COVID-19, coronavirus disease, WHO, machine learning, deep learning, decision support system.

1 Introduction

The origin of the novel coronavirus (2019-nCoV) was spotted in Wuhan province, China on December 31, 2019, named COVID-19 by WHO [1]. The World Health Organization (WHO) has declared the novel CoV outbreak as Public Health Emergency worldwide on January 30, 2020, conveyed according to the act of International Health Regulations [2].

The clinical characteristics of CoV are classified as most common symptoms (fever, dry cough, and tiredness), less common symptoms (aches and pains, sore throat, diarrhea, conjunctivitis, headache, loss of taste or smell, and a rash on the skin, or discoloration of fingers or toes) and serious symptoms (difficulty breathing or shortness of breath, chest pain or pressure, and loss of speech or movement) [3, 4].

The prevention strategies regarding interruption of the CoV spreading were noted as early detection, isolating and treating cases, contact tracing, and social distancing. The transmission of CoV can be occurred by directly connected to the infected person via coughing or sneezing within closed connecting (<1m) and indirectly also infected by immediately touching or using the infected surfaces or objects [5].

Recently, the article published in the New England Journal of Medicine has produced evidence of the COVID-19 virus spreading through airborne transmission. The home quarantine is enough for a healthy person affected with mild CoV

symptoms, on average 5–6 days are essential to show the symptoms otherwise in worst cases takes up to 14 days [6].

Over 80% of infected persons are recovered from COVID-19 who had low levels of antibodies of SARS-CoV-2 in their blood. The careful observation of the development of antibodies in infected persons helps to develop the vaccines and treatment for COVID-19 [7].

Due to lack of vaccines or the proper treatment, the person can slow down the transmission of COVID-19 by regularly washing the hands with soap and water, maintaining the social distance of at least 1 meter (3 feet) between yourself and others, avoiding going to crowded places, don't touch eyes, nose, and mouth unnecessary, stay home and self-isolate while minor symptoms, and up to date with the latest information from trusted sources, such as WHO or your local and national health authorities [8].

The WHO has brought the world's scientists and global health professionals to collaborate to accelerate the research and development process, and develop the treatment and vaccines for controlling the coronavirus pandemic [9]. Various studies on epidemiologic history, laboratory conditions, analyze the clinical characteristics, treatment regimens, and prognosis of patients are commenced regarding the instantiation of the outbreak of COVID-19 [10,11].

The clinical characteristics have been studied on mild symptoms affected patients, the outcomes are varied greatly [12, 13]. This is very difficult to find out a highly risky group by concerning the only age and gender factors. Furthermore, it is necessary to predict the infected groups, provide the real treatments with constraints hospitality resources, and health practitioners faced difficulties while treating the patients without any previous experience. Out of these limitations, artificial intelligence (AI) can analyze the data, learn effective patterns, and suggest while decision-making processes. Over the last two decades, AI has achieved countless milestones in the field of health care and advisory systems such as biomedical information processing, disease diagnostics and prediction, and application to radiology, pathology, ophthalmology, and dermatology [14–16].

The machine learning algorithms also effort to early detect and predict the health care issues in the application area of latent diseases [17], Health Monitoring System [18], Brain Stroke [19], early-stage disease risk prediction [20], and Acute Kidney Injurious prediction [21]. Similarly, the deep learning methods are extremely dedicated to the application area of health such as Alzheimer's disease [22], emotion analysis towards mental health care [23], Cancer Care [24], and prediction of pain progression in knee osteoarthritis [25].

We have observed that the contribution of AI, machine learning, and deep learning are considerable in the health care system and the application area of such techniques can also be extended to predict the COVID-19 infection.

In this paper, we provide the classical supervised machine learning algorithms and deep learning methods for the prediction of COVID-19 infection. Twelve classifiers (nine classical supervised machine learning algorithms and three deep learning methods) are designed and applied to laboratory datasets for finding the infected patients.

The performance of our implemented models is compared based on the classification accuracy rate. The main objectives are covered in this paper are summarized as follows:

- Processing towards introducing the prediction system for the identification of COVID-19 infected persons using machine learning and deep learning algorithms rather than Chest X-ray or CT-Scan Images.
- Our research work compared with various machine learning and deep learning algorithms mention in this paper. Further, we also analyzed the experimental results with other recently published research works.
- Our work will motivate the researchers to further architect and build more effective models and include additional parameters such as genders, travel details, previous medical treatment details, etc for boosting the prediction of COVID-19 infection outcomes.

The remainder of the paper is organized as follows. Section 2 elaborates the related works regarding the prediction of COVID-19 infections. The essential information for the implementation of the experimental dataset and basic introduction to

methodology is described in section 3. Section 4 provides the initial configuration setup for the method used in this paper.

The evaluation metrics such as accuracy, precision, recall, F1-score, and AUC-ROC score used for evaluating the classification performance are presented in section 5. The evaluation parameters and experimental results of the proposed classification model's performance are comprehensively analyzed with recently published works are detailed in section 6. Finally, section 7 contains the conclusion of our research work and future scope.

2 Related Works

This is very important to continue monitoring and predicting health care tasks. The computer-aided clinical systems are widely used as assisting tools for caring the various health-related issues such as diagnosis of breast cancer [26], diagnosing early gastric cancer [27], brain pathology identification [28], computer-aided drug discovery [29], health care facilities management [30].

The medical experts can use these techniques as assistance for better prediction of diagnosing related issues. This study is extremely dedicated to building the recent methodological model for predictive the COVID-19 infection. Recently, various literature was published related to deep learning methods for COVID-19 infection prediction using chest X-ray or CT-Scan Images [31–33].

The authors [34] have obtained the clinical data set from the institutional ethics board of Wenzhou Central Hospital and Cangnan People's Hospital in Wenzhou, China. The effective features were extracted using eleven feature selection algorithms (ALT, Myalgias, Hemoglobin, Gender, Temperature, Na+, K+, Lymphocyte Count, Creatinine, Age, and White Blood Count). Six machine learning algorithms (Logistic Regression, KNN (k=5), Decision Tree based on Gain Ratio & Gini Index, Random Forests, and Support Vector Machine (SVM)) were applied and accuracy measured on 10-fold cross-validation.

The SVM was obtained the maximum accuracy of 80% among the listed classifiers. The paper [35] have also applied the machine learning (neural networks, random forests, gradient boosting trees, logistic regression, and support vector machines) techniques on the clinical dataset and measured the performance based on AUC, sensitivity, specificity, F1-score, Brier score, Positive Predictive Value (PPV), and Negative Predictive Value (NPV).

This paper was used as the clinical dataset, obtained from Hospital Israelita Albert Einstein in São Paulo, Brazil, and split into 70% for training and 30% for testing. The SVM and random forests classifiers were achieved the best score valued regarding measured parameters (AUC = 0.847, Sensitivity = 0.677, Specificity = 0.850, F1-score = 0.724, Brier score = 0.160, PPV = 0.778, and NPV = 0.773).

The paper [36] was used the same experimental clinical dataset as in [35] and applied the various machine learning algorithms including Logistic Regression (LR), Neural Network (NN), Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting (XGB). The predictive performance was compared in terms of AUC, AUPR, sensitivity, specificity, and specificity at greater than 95% sensitivity (Spec. @95%Sens.). The XGB was obtained the best experimental result, noted as AUC = 0.66, AUPR = 0.21, Sensitivity = 0.75, Specificity = 0.49, Spec.@95% Sens. = 0.23.

In the paper [37], the deep learning methods (Artificial Neural Networks (ANN), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), CNN + LSTM, CNN + RNN were applied on the clinical dataset used in [35], experimental results were evaluated with train-test split and 10 fold cross-validation approach, and scores are measured based on precision, F1-score, recall, AUC, and accuracy scores. The hybrid model CNNLSTM of deep learning methods was achieved the best predictive score accuracy = 86.66%, F1-score = 91.89%, precision = 86.75%, recall = 99.42%, and AUC = 62.50%.

3 Experimental Dataset and Methodology

The purpose of this section is to outline the necessary background information regarding the

experimental dataset and methodology used in this paper.

3.1 Dataset Description

Here, we provide a detailed description of the experimental dataset, available at Hospital Israelita Albert Einstein at Sao Paulo Brazil, and accessed through [36].

The samples were collected to test the infection of SARS-CoV-2 in the early month of 2020 and available on [38]. This dataset contained a sample record of 5644 patients with a contribution of 111 different laboratories. The infection rate of patients was around 10% of which around 6.5% and 2.5% required hospitalization and critical care. The rest of 90% of patients reported negative SARS-CoV-2. The information related to the gender of patients is not mentioned in this dataset.

The dataset consists of a total of ten columns (Patient ID, Patient age quantile, SARS-Cov-2 exam result (negative/positive), Patient admitted to the regular ward (yes/no), Patient admitted to the semi-intensive unit (yes/no), Patient admitted to intensive care unit (yes/no), Hematocrit, Hemoglobin, Platelets, and Mean platelet volume). We apply the split-test approach and randomly divide the dataset into training (80%) and testing (20%) respectively for validating our models. Furthermore, the 10-fold cross-validation is also used to approximately balance the accuracy rate of models.

3.2 Methodology

Artificial Intelligence (AI) is a loose interpretation of human intelligence into the machine, accomplished through learning, reasoning, and self-correction. The AI-based machine can make decisions based on predefined rules and algorithms without interfering with human beings. Machine learning (ML) and deep learning (DL) are considered a subset of AI and adopt additional features to beat the human being in terms of intelligence and accuracy. The working performance of ML is differing from DL due to the way data is presented in the system. The ML is always required structured data whereas deep learning relies on the reassembling of artificial neural networks (ANN). It is essential to hand over the control to the human beings for handling the applied areas of the ML concept. The DL system aims to adopt the same features without supplementary interference with human beings. The large amount of data processed and used the complex mathematical calculations in the algorithms; DL systems require much more powerful computing power rather than simple ML systems. So, a deep learning system consumes much time (a few hours to a few weeks) to train the model as compared to a simple ML model (a few seconds to a few hours).

In this study, we serve a reasonable framework for validating the developed clinical predictive models to predict the COVID-19 infection. We developed the twelve different models (nine ML and three DL) for evaluation of the study: logistic regression, K-Neighbors classifier, support vector classifier, decision tree classifier, random forest classifier, AdaBoost classifier, GaussianNB, linear discriminant analysis, quadratic discriminant analysis, Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). The logistic regression classification algorithm is used to predict the probability of a categorical dependent variable, which should be a binary variable that contains the binary coded (true/false).

This model of ML is used for predicting the risk of developing chronic diseases [39], Trauma and Injury Severity Score (TRISS) [40], diabetes [41], heart disease identification [42], breast cancer [43], Alzheimer [44], etc. The K-Nearest Neighbors has supervised classification algorithms in ML, stores all available cases, and classifies new cases based on a similarity measure, e.g., hamming or standardized distance function.

This is a popular method with a wide variety of applications in many different areas of voice disorder identification [45], brain tumor classification [46], and more.

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification, regression, or outlier detection. The SVM assembles the hyperplane or set of hyperplanes in a high or infinite-dimensional space. The hyperplane achieved a good classification result that has the largest distance to the nearest training data point of any class. This is a popular method with a wide variety of

applications in many different areas of skin disease detection [47], heart disease diagnosis [48], etc.

Decision tree is supervised machine learning and classifies the instances according to their feature values [49]. This classifier follows the concept of a divide-and-conquer algorithm that splits the data into subsets based on homogeneous properties. This method has innumerable applications in many different areas such as detection of Hepatocellular carcinoma (HCC) in clinical data [50], Opioid Use Disorder (OUD) understanding [51], etc.

Random forest is a supervised learning algorithm and can be used both for classification and regression. It usually draws the set of decision trees from a randomly selected subset of the training set and then merges them to occupy a more accurate and stable prediction result. Its applications can be found in many areas such as identification of human vital functions related to the respiratory system [52], prognosis prediction [53], etc.

AdaBoost or Adaptive Boosting classifier is an iterative approach that learns from the incorrectly classified instances by weak classifiers and fits additional copies of the classifier for turning them into strong classifiers. The application areas of the AdaBoost classifier are relevant to the early prediction of lung cancer [54], pinus diseased recognition [55], etc.

GaussianNB classifier provides the way of implementing the concept of the Gaussian Naïve Bayes algorithm for classification. This method could be extended to solve the problems in various areas such as diabetes prediction [56], a prediction model for the detection of cardiac arrest [57], etc.

Linear Discriminant Analysis (LDA) classifier is based on the value of the prediction, estimated by the probability new inputs set belongs to each class. The output class is designated by the highest probability value and built the prediction. This convention is mostly preferred for measuring the models for human health effects [58], detection of epileptic seizures using EEG signals [59], etc.

Quadratic discriminant analysis is used as both classifier and dimensionality reduction technique but cannot be used as a dimensionality reduction technique. This approach is a variation of the LDA classification technique that also allows for non-linear separation of data. This method is applied in various application areas such as epileptic seizure detection [60], pre-diagnosis of Huntington's disease [61], etc.

Design of CNN architecture is inspired by the biological vision system and is composed of four subsequent stages of layers: convolutional layer, pooling layer, activation layer, and the fully-connected layer. Each distinct layer is responsible for transforming the input volume to the output volume through different hidden layers to achieve the predefined goal. We can apply the CNN method in different application areas such as automatic skin disease diagnosis [62], pneumonia detection [63], breathing disorder detection [64], arrhythmia classification [65], small lesion detection [66], etc. A plausible and useful theory behind the RNN method is to make use of sequential information that means output from the previous stage provided as input to the current stage. The RNN has a concept of storage that stores all calculated information and also exhibits temporal dynamic behavior. For this reason, this method represents an attractive option for arrhythmia detection [67], hemoglobin concentration prediction [68], Heart sound classification [69], etc.

The LSTM is a special kind of RNN and explicitly designed to avoid the problem of long-term dependency. The LSTM carried out the chain structure that contains four neural networks and various memory blocks, called cells. These cells are responsible for retaining the information and gates to manipulate the memory, named as forget gate, input gate, and output gate. We can use the LSTM approach in various fields such as EEG-based emotion classification [70], analyze psychological effects [71], abnormal heart sound detection [72], chronic laryngitis classification [73]. Figure 1 shows the logical diagram of our experimental prediction model used in this paper.

4 Configuration of Experimental Methods

In this section, we are addressing the detailed description regarding the configuration of ML and DL methods, used in this paper for the prediction of COVID-19 infection. For the exposure of ML algorithms as compared to DL methods, we have

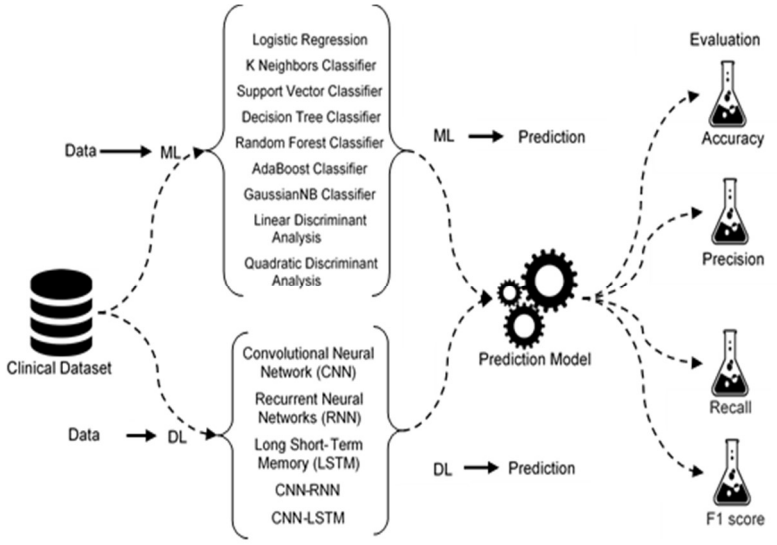


Fig. 1. Conceptual view of experimental models used in this paper: from the flow of dataset to ML (Machine Learning) and DL (Deep Learning) model, prediction model, and evaluation results

used the Scikit-learn machine learning classifiers (logistic regression, K-neighbors classifier, support vector classifier, decision tree classifier, random forest classifier, AdaBoost classifier, GaussianNB, linear discriminant analysis, and quadratic discriminant analysis).

These methods are publically accessible with full documentation and can be imported from the sklearn library [74]. The initial values of parameters for each classifier and reference section contain the user guideline URL, mentioned in table 1. The layer architecting, the details and parameters of each DL classifier used in this study are mentioned in table 2.

The parameters are named as number of layers, activation function, learning rate, loss function, number of the epoch, optimizer, dropout, batch size, and total parameters are responsible for framework the DL methods. Each layer is configured with different values that can be optimized and manipulate the input data.

The gentle introduction of the activation function is used for deciding whether a neuron should be activated or not with the help of calculating the weighted sum. How quickly and slowly, the neural network model learns a problem depends on learning rate values. The loss function calculates the prediction error due to estimation loss by the neural networks. The epoch values count the pass of the full training data set through the model. The optimization function is responsible for reducing the losses and accelerating the accuracy rates as much as possible. The dropout is commonly used in deep neural networks to prevent overfitting problems. The batch size specifies the number of training samples processed before the model execution. The total number of parameters aggregates all weights and biases.

5 Evaluation Metrics

The evaluation metrics exist with the Sklearn method to compare the performance of classifiers. In this section, we discuss the evaluation metrics, experimental result analysis based on train-test split and 10-folds validation approach, and result comparison with published works.

To evaluate the classification performance of models, we can use accuracy (A), precision (P), recall (R), and F1-score (F1). For a binary classification problem, the confusion matrix holds the entries of True Positive (TP), False Positive (FP), True Negatives (TN), and False Negatives (FN). The diagonal entries hold the correct

Table 1. ML classifiers parameter adjustment

classifier	scikit-learn method	parameters	Ref.
Logistic Regression	sklearn.linear_model.LogisticRegression	C=1.0, max_iter=100, penalty='l2', solver='lbfgs', tol=0.0001	[75]
K-Neighbors	sklearn.neighbors.KNeighborsClassifier	leaf_size=30, metric='minkowski', n_neighbors=3, p=2	[76]
Support Vector	sklearn.svm.SVC	C=0.025, cache_size=200, degree=3, kernel='rbf', max_iter=-1	[77]
Decision Tree	sklearn.tree.DecisionTreeClassifier	criterion='gini', min_samples_leaf=1, presort='deprecated', splitter='best'	[78]
Random Forest	sklearn.ensemble.RandomForestClassifier	n_estimators=100, min_samples_split=2, min_samples_leaf=1	[79]
AdaBoost	sklearn.ensemble.AdaBoostClassifier	algorithm='SAMME.R', learning_rate=1.0, n_estimators=50	[80]
GaussianNB	sklearn.naive_bayes.GaussianNB	var_smoothing=1e-09	[81]
Linear Discriminant Analysis	sklearn.discriminant_analysis.LinearDiscriminantAnalysis	solver='svd', tol=0.0001	[82]
Quadratic Discriminant Analysis	sklearn.discriminant_analysis.QuadraticDiscriminantAnalysis	reg_param=0.0, tol=0.0001	[83]

Table 2. Architecting configuration of DL methods

Parameters	CNN	RNN	LSTM
Number of layers	1	1	1
Activation function	ReLU, Softmax	ReLU, Softmax	ReLU, Softmax
Learning rate	0.0005	0.0005	0.0005
Loss function	Sparse categorical crossentropy	Sparse categorical crossentropy	Sparse categorical crossentropy
Number of epoch	30	30	30
Optimizer	Adam	Adam	Adam
Dropout	0.4, 0.6	0.4, 0.6	0.4, 0.6
Batch size	512	1024	1024
Total parameters	9,442,306	2,102,274	4,728,322

prediction TP and incorrect prediction denoted by TN. The classifier made the wrong prediction, referred to as FP and FN.

Accuracy measures the ratio between the number of correct predictions and the total number of input samples. The classification model is observed as perfect when the number of predicted samples is equal to the total number of samples. For the multiclass classification problem, the numbers of classes are denoted by the value of k.

Precision equation measures the number of correct positive results divided by the number of

Table 3. Equations for evaluating the classification performance

Evaluation Metric	Equation
Accuracy (A)	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}$
Precision (P_k)	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Recall (R_k)	$\frac{TP}{TP+FN}$
F1-score ($F1_k$)	$2 \times (\frac{P_k \times R_k}{P_k + R_k})$

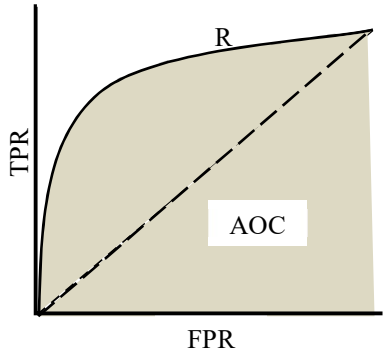


Fig. 2. AUC-ROC Curve

Table 4. Summary of experimental results of all ML and DL with the train-test split approach.

Classification Models			
Machine Learning Methods		**Deep Learning Methods**	
Methods	Accuracy (%)	Methods	Accuracy (%)
Logistic Regression	**85.00**		
K-Neighbors	**85.00**	**CNN**	**91.88**
Support Vector	84.16		
Decision Tree	82.50		
Random Forest	84.16	RNN	90.27
AdaBoost	**85.00**		
GaussianNB	84.16		
Linear Discriminant Analysis	84.16	LSTM	90.00
Quadratic Discriminant Analysis	80.83		

positive results predicted by the classifier. The recall evaluates the number of correct positive results divided by the number of all relevant samples.

F1-score is primarily used to measure the model's test accuracy and the score varies between 0 and 1 values. The high precision value and low recall value achieve a great accuracy rate but avoid the large number of samples that are difficult to classify. Table 3 illustrates the equation to measure the classification accuracy, precision, recall, and F1-score, extracted from the confusion

matrix. The ROC-AUC (Receiver Operating Characteristics - Area under the Curve) is frequently used to evaluate the classification and prediction model's performance. This examines the model's ability while distinguishing between positive and negative classes. The higher AUC score indicates the better model for the prediction of patients with infected or not infected. The ROC curve is plotted with False Positive Rate (FPR) on X-axis and True Positive Rate (TPR) on Y-axis (figure 2).

The FPR and TPR score are calculated using the expression (1) and (2), respectively:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad (1), \quad \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

The idea behind the calculation of the AUC score (exists between 0 and 1) is the measurement of separability. The AUC score exists near 1, which means has good separation capability. For the multi-class problem, we can plot the N number of AUC ROC curves for multiple classes.

6 Experimental Results and Discussion

This section consists of the experimental results of ML and DL methods for the prediction of COVID-19 infection, considering a total of 600 patients using 18 different laboratory findings. These results are evaluated based on the train-test split approach, 10-folds cross-validation, ROC-AUC score, and comparison of results with published works.

6.1 Train-Test Split Approach

We can observe in Table 4, the accuracy results of all ML models have reached at least 80% and above. The Logistic Regression, K-Neighbors, and AdaBoost classifier have achieved the best evaluation performance with an 85.00% accuracy score. The Support Vector, Random Forest, GaussianNB, and Linear Discriminant Analysis were observed as the second-best models. The experimental results of all DL application methods through the aggregation of the mean values of accuracy score. In terms of accuracy predictive performance, we observed that the overall best score was achieved by CNN with a 91.88% score followed by RNN (accuracy = 90.27%), then LSTM (accuracy = 89.99%).

Table 5. Summary of experimental results of all ML and DL with 10 folds cross-validation approach

Classification Models			
Machine Learning Methods		**Deep Learning Methods**	
Methods	Accuracy (%)	Methods	Accuracy (%)
Logistic Regression	89.79	CNN	87.66
K-Neighbors	87.29		
Support Vector	87.29		
Decision Tree	83.75	RNN	86.49
Random Forest	89.16		
AdaBoost	87.91		
GaussianNB	83.54	LSTM	86.66
Linear Discriminant Analysis	87.70		
Quadratic Discriminant Analysis	82.29		

Table 6. Performance measurement based on AUC score

AUC-Score	Model's characteristics
0	Inaccurate Test
0.5	No Discrimination
0.6 to 0.8	Acceptable
0.8 to 0.9	Excellent
> 0.90	Outstanding
1	Accurate Test

Table 7. AUC values of all deep learning methods

Deep Learning Methods	AUC- Score
CNN	0.6211
RNN	0.6876
LSTM	0.5000

6.2 10-Fold Cross-Validation Approach

In the 10-fold cross-validation, the experimental dataset is randomly partitioned into 10 equal sub-datasets. Out of these sub-datasets, a single sub-dataset is assigned as a validating dataset, and the rest of the nine sub-datasets are retained as

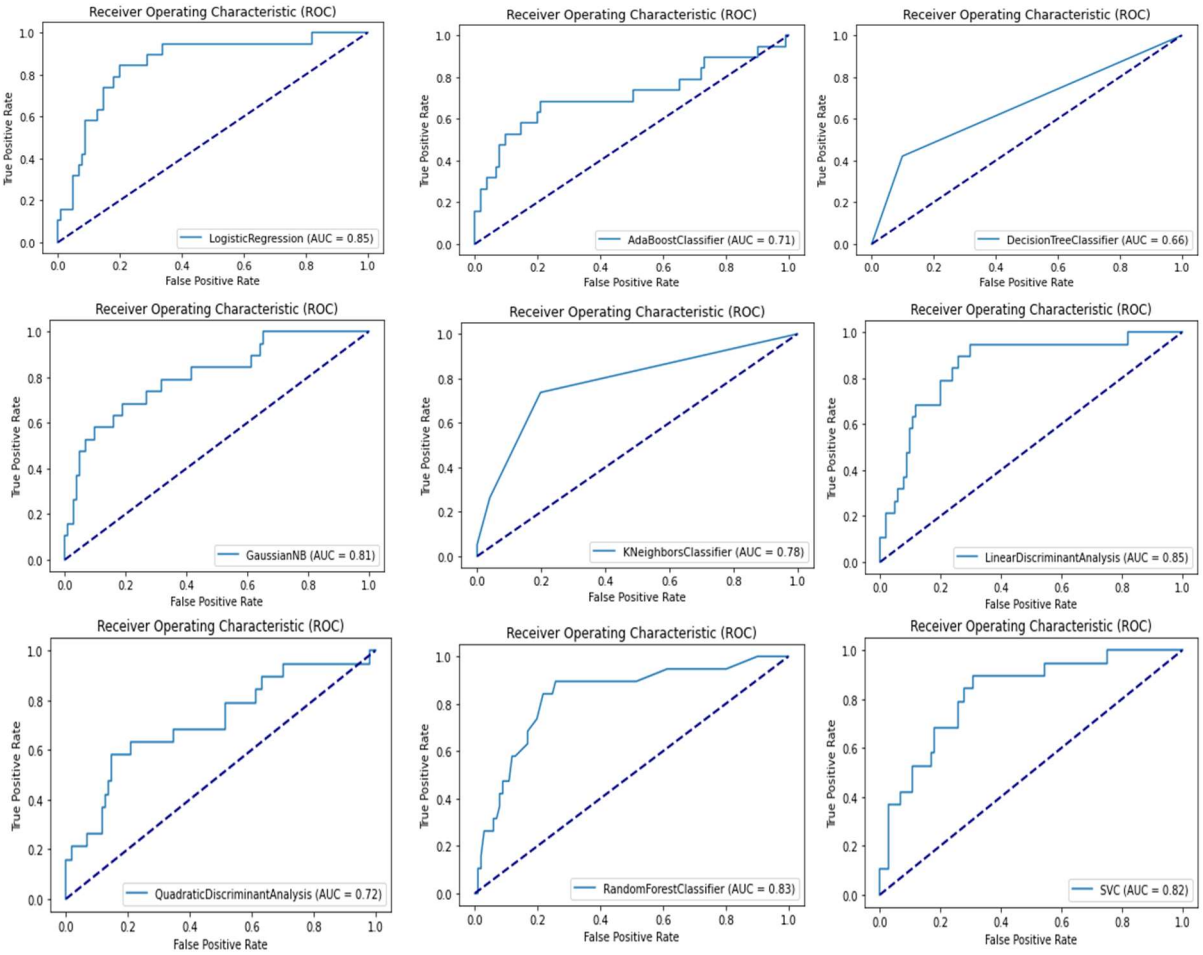


Fig. 3. AUC values of all classical machine learning algorithms

training data. The cross-validation technique repeats the process ten times and each of the 10 - subsamples is used exactly once as the validation data. The final result of 10-folds can be produced by aggregating the average results of each folding. Table 5 shows the experimental results of all machine learning application models with 10 folds cross-validation approach.

In cases of relatively small samples, the k fold cross-validation approach is frequently used to measure the accurate classification performance of classifiers especially in health studies [35]. In table 5, we have observed the accuracy score of all ML and DL classifications methods based on 10 folds cross-validation techniques. The performance of all ML algorithms was better in 10 folds cross-validation approach comparison with the train-test split strategy but opposite performance results with DL methods. The accuracy results of all ML models have reached at least 82.29% and above.

Logistic Regression has achieved the best accuracy performance with an 89.79% score, followed by Random Forest as the second-best model with 89.16% accuracy. Moreover, the experimental results of all DL application methods were observed using the mean values of accuracy score. CNN has achieved the overall best accuracy 91.88% score, followed by RNN (accuracy = 86.49%) then LSTM (accuracy = 86.66%).

Table 8. Comparison of experimental results with recently published works

Ref.	Dataset Source	Techniques	Classification methods	Accuracy (%)	AUC	F1 – Score
[34]	Wenzhou Central Hospital and Cangnan People's Hospital in Wenzhu, China	ML	Support Vector Machine	80.00	-	-
[35]	Hospital Israelita Albert Einstein at Sao Paulo, Brazil	ML	Support Vector Machine, Random Forest	-	0.87	0.72
[36]	Hospital Israelita Albert Einstein at Sao Paulo, Brazil	ML	Logistic Regression	89.00	0.85	-
Our work	Hospital Israelita Albert Einstein at Sao Paulo, Brazil	DL	CNN, RNN, LSTM	91.88, 90.27, 90.00	-	-

6.3 Results Interpretation of Area Covered Under the ROC Curve

The AUC-Score determines the best model prediction on classes and ranges in value from 0 to 1. The various points on the ROC curve determine the different characteristics of the model's performance. The following table 6 determines the model's characteristics based on range value [84]. The classification models achieved more than 0.60 AUC score value; we can say that those models were accepted for clinical prediction of COVID-19.

The AUC score of logistic regression is considered acceptable since the results range between 0.8 and 0.9 (figure 3). The AUC scores of the remaining ML methods were excellent, all of the results were higher than 0.66. In the DL methods, the RNN achieved the highest score (AUC = 0.68), followed by CNN (AUC = 0.62), and then the LSTM approach (AUC = 0.50) (table 7).

6.4 Comparison of Experimental Results with Recently Published Articles

The paper [34] and [35] have used the classical machine learning algorithms i.e. Support Vector Machine, Random Forest, respectively. Similarly, the paper [37] compared the prediction performance of six different classical and hybrid deep learning algorithms i.e. ANN, CNN, RNN, LSTM, CNNRNN, and CNNLSTM. However, we have used both classical ML and DL algorithms. We can observe in Table 8, the best classification is obtained by deep learning methods (CNN, RNN, and LSTM). Yet, in our study, we have exposed the performance of both ML and DL methods. It has shown that the AUC-Score of all methods is acceptable for the prediction of COVID-19 infection.

7 Conclusion and Future Works

In this study, we have designed and developed deep learning-based machine learning models for predicting the COVID-19 infection. We have carried out the nine classical machines (logistic regression, K-Neighbors classifier, support vector classifier, decision tree classifier, random forest classifier, AdaBoost classifier, GaussianNB, linear discriminant analysis, quadratic discriminant analysis) and three deep learning methods (CNN, RNN, and LSTM) to accomplish the clinical prediction work.

The experimental data were preprocessed using standardization and then fed to our models. Further, the classification results were measured based on the accuracy score. To validate our model, we have used the train-test split approach, 10-fold cross-validation, and AUC-ROC curve score. In the train-test split approach, the best result was achieved using CNN with an accuracy of 91.88% and an AUC score of 62.11% in the deep learning application.

However, Logistic Regression, K-Neighbors, and AdaBoost classifiers have obtained a similar accuracy of 85.00% and AUC score of 85.00%, 78.00%, and 71.00%, respectively. The best accuracy value was achieved by CNN (Deep Learning) with an accuracy of 87.66% and Logistic Regression (Machine Learning) with an accuracy

of 89.79%. All the ML and DL algorithms used in this study, have achieved an accuracy of over 80%.

This study carried out a major limitation with a small and imbalanced experimental dataset. The performance of our prediction model can be enhanced by increasing the size of the dataset either combining the data from different laboratories or using data augmentation techniques. Further studies were carried out from our study with findings the additional parameters such as genders, travel details, previous medical treatment details, etc. for enhancing the prediction rate. Based on our experimental results, we conclude that the clinical system should explore the use of artificial intelligence for prioritizing the models as decision support systems while reducing the personalizing infection risks.

Acknowledgments

Prabhat Kumar sincerely acknowledges the University Grants Commission (UGC), New Delhi, India, for awarding the Non-Net Fellowship Scheme [FILE NO.: R/Dev/IX-Sch .(UGC Res. Sch.) 2020-21/13674, Dated: 01-10-2019]. The work was supported under Institute of Eminence (IoE) Seed Grant by Banaras Hindu University.

References

1. **Aylward, B., Liang, W. (2020).** Report of the WHO-China joint mission on Coronavirus disease 2019 (COVID-19). WHO-China Jt. Mission Coronavirus Dis. 2019. pp. 6–24.
2. **WHO (2020).** WHO Director-General's statement on IHR Emergency Committee on Novel Coronavirus (2019-nCoV), August 11, 2020.
3. **Wang, D., Hu, B., Hu, C., Zhu, F., Liu, X., Zhang, J., Wang, B., Xiang, H., Cheng, Z., Xiong, Y., et al. (2020).** Clinical characteristics of 138 hospitalized patients with 2019 novel Coronavirus-Infected pneumonia in Wuhan, China. JAMA, Vol. 323, No. 11, pp. 1061–1069. DOI: 10.1001/jama.2020.1585.
4. **Holshue, M. L., DeBolt, C., Lindquist, S., Lofy, K. H., Wiesman, J., Bruce, H., Spitters, C., Ericson, K., Wilkerson, S., Tural, A., et al. (2020).** First case of 2019 novel coronavirus in the United States. New England Journal of Medicine, Vol. 382, pp. 929–936. DOI: DOI: 10.1056/NEJMoa2001191.
5. **WHO (2020).** Modes of transmission of virus causing COVID-19: implications for IPC precaution recommendations, Scientific brief, August 13, 2020.
6. **WHO (2020).** Coronavirus, August 13, 2020. Available from: https://www.who.int/health-topics/coronavirus#tab=tab_3.
7. **NIH (2020).** Potent antibodies found in people recovered from COVID-19, August 13, 2020**.** Available from: https://www.nih.gov/news-events/nih-research-matters/potent-antibodies-found-people-recovered-covid-19.
8. **WHO (2020).** Advice for the public, August 13, 2020.
9. **WHO. (2020).** Global research on coronavirus disease (COVID-19), August 13, 2020.
10. **Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., Zhang, L., Fan, G., Xu, J., Gu, X., et al. (2020).** Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. Lancet, Vol. 395, pp. 497–506. DOI: 10.1016/S0140-6736(20)30183-5.
11. **Liu, K., Fang, Y. Y., Deng, Y., Liu, W., Wang, M. F., Ma, J. P., Xiao, W., Wang, Y. N., Zhong, M. H., Li, C. H., et al. (2020).** Clinical characteristics of novel coronavirus cases in tertiary hospitals in Hubei Province. Chinese Medical Journal, Vol. 133, pp. 1025–1031. DOI: 10.1097/CM9.0000000000000744.
12. **Kam, K. Q., Yung, C. F., Cui, L., Lin, R. T. P., Mak, T. M., Maiwald, M., Li, J., Chong, C. Y., Nadua, K., Tan, N. W. H., et al. (2020).** A well infant with Coronavirus disease 2019 (COVID-19) with High Viral Load. Clinical Infectious Disiseases, Vol. 71, No. 15, pp. 847–849. DOI: 10.1093/cid/ciaa201.
13. **Jiehao, C., Jin, X, Daojiong, L, Zhi, Y., Lei, X., Zhenghai, Q., Yuehua, Z., Hua, Z., Ran, J., Pengcheng, L., et al. (2020).** A case series of children with 2019 novel coronavirus infection: clinical and epidemiological features. Clinical Infectious Diseases, Vol. 71, No. 6, pp. 1547-1551. DOI: 10.1093/cid/ciaa198.

14. **Kulkarni, S., Seneviratne, N., Baig, M. S., Khan, A. H. A. (2020).** Artificial intelligence in medicine: where are we now?. Academic Radiology, Vol. 27, No. 1, pp. 62–70. DOI: 10.1016/j.acra.2019.10.001.
15. **Rowe, J. P., Lester, J. C. (2020).** artificial intelligence for personalized preventive adolescent healthcare. Journal of Adolescent Health, Vol. 67, No. 2, pp. S52–S58. DOI: 10.1016/j.jadohealth.2020.02.021.
16. **Rong, G., Mendez, A., Assi, E. B., Zhao, B., Sawan, M. (2020).** artificial intelligence in healthcare: review and prediction case studies. Engineering, Vol. 6, No. 3, pp. 291–301. DOI: 10.1016/j.eng.2019.08.015.
17. **Wang, Y., Zhao, Y., Therneau, T. M., Atkinson, E. J., Tafti, A. P., Zhang, N., Amin, S., Limper, A. H., Khosla, S., Liu, H. (2020).** Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records. Journal of Biomedical Informatics, Vol. 102, pp. 1–10. DOI: 10.1016/j.jbi.2019.103364.
18. **Sheela, K. G., Varghese, A. R. (2019).** Machine learning based health monitoring system. Materials Today: Proceedings, Vol. 24, pp. 1788–1794. DOI: 10.1016/j.matpr. 2020.03.603.
19. **Sirsat, M. S., Fermé, E., Câmara, J. (2020).** Machine learning for brain stroke: A review. Journal of Stroke and Cerebrovascular Diseases, Vol. 29, No. 10, pp. 1–17. DOI: 10.1016/j.jstrokecerebrovasdis.2020.10516
20. **Hossain, M. A., Ferdousi, R., Alhamid, M. F. (2020).** Knowledge-driven machine learning based framework for early-stage disease risk prediction in edge environment. Journal of Parallel Distributed Computing, Vol. 146, pp. 25–34. DOI: 10.1016/j.jpdc.2020.07.003.
21. **Yang, X., Yu, Y., Xu, J., Shu, H., Xia, J., Liu, H., Wu, Y., Zhang, L., Yu, Z., Fang, M., et al. (2020).** Clinical course and outcomes of critically ill patients with SARS-CoV-2 pneumonia in Wuhan, China: a single-centered, retrospective, observational study. The Lancet Respiratory Medicine, Vol. 8, No. 5, pp. 475–481. DOI: 10.1016/S2213-2600(20)30079-5.
22. **An, N., Ding, H., Yang, J., Au, R., Ang, T. F. A. (2020).** Deep ensemble learning for Alzheimer's disease classification. Journal of Biomedical Informatics, Vol. 105, pp. 1–11. DOI: 10.1016/j.jbi.2020.103411.
23. **Fei, Z., Yang, E., Li, D. D. U., Butler, S., Ijomah, W., Li, X., Zhou, H. (2020).** Deep convolution network based emotion analysis towards mental health care. Neurocomputing, Vol. 388, pp. 212–227. DOI: 10.1016/j.neucom.2020.01.034.
24. **Coccia, M. (2020).** Deep learning technology for improving cancer care in society: New directions in cancer imaging driven by artificial intelligence. Technology in Society, Vol. 60, pp. 1–11. DOI: 10.1016/j.techsoc. 2019.101198.
25. **Guan, B., Liu, F., Matthew, P., Mirzaian, A. H., Demehri, S., Neogi, T., Guermazi, A., Kijowski, R. (2020).** Deep learning approach to predict pain progression in knee osteoarthritis. Osteoarthritis and Cartilage, Vol. 28, pp. S316. DOI: 10.1016/j.joca. 2020.02.489.
26. **Rahman, M. M., Ghasemi, Y., Suley, E., Zhou, Y., Wang, S., Rogers, J. (2020).** Machine learning based computer aided diagnosis of breast cancer utilizing anthropometric and clinical features. IRBM, Vol. 42, No. 4, pp. 215-226. DOI: 10.1016/j.irbm.2020.05.005.
27. **Horiuchi, Y., Hirasawa, T., Ishizuka, N., Tokai, Y., Namikawa, K., Yoshimizu, S., Ishiyama, A., Yoshio, T., Tsuchida, T., Fujisaki, J., et al. (2020).** Performance of a computer-aided diagnosis system in diagnosing early gastric cancer using magnifying endoscopy videos with narrow-band imaging (with videos). Gastrointestinal Endoscopy, Vol. 92, No. 4, pp. 856–865. DOI: 10.1016/j.gie.2020.04.079.
28. **Gudigar, A., Raghavendra, U., Hegde, A., Kalyani, M., Ciaccio, E. J., Acharya, U. R. (2020)**. Brain pathology identification using computer aided diagnostic tool: A systematic review. Computer Methods and Programs in Biomedicine, Vol. 187, pp. 1–18. DOI: 10.1016/j.cmpb.2019.105205.

29. **Ebhohimen, I. E., Edemhanria, L., Awojide, S., Onyijen, O. H., Anywar, G. (2020).** Advances in computer-aided drug discovery. Phytochemicals as Lead Compounds for New Drug Discovery, Elsevier, pp. 25-37. DOI: 10.1016/B978-0-12-817890-4.00003-2.
30. **Iadanza, E., Luschi, A. (2019).** Computer-aided facilities management in health care. Clinical Engineering Handbook, 2nd ed. Academic Press, pp. 42–51. DOI: 10.1016/B978-0-12-813467-2.00005-5.
31. **Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Singh, V. (2020).** Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet. Chaos, Solitons and Fractals. Vol. 138, pp. 1–8. DOI: 10.1016/j.chaos.2020.109944.
32. **Panwar, H., Gupta, P. K., Siddiqui, M. K, Morales-Menendez, R., Bhardwaj, P, Singh, V. (2020).** A deep learning and Grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. Chaos, Solitons and Fractals, Vol. 140, pp. 1–12. DOI: 10.1016/j.chaos.2020.110190.
33. **Das, N. N., Kumar, N., Kaur, M., Kumar, V., Singh, D. (2020).** Automated deep transfer learning-based approach for detection of COVID-19 infection in chest X-rays, IRBM, Vol. 43, No. 2, pp. 114–119. DOI: 10.1016/j.irbm.2020.07.001.
34. **Jiang, X., Coffee, M., Bari, A., Wang, J., Jiang, X., Huang, J., Shi, J., Dai, J., Cai, J., Zhang, T., et al. (2020).** Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. Computers, Materials & Continua, Vol. 63, No. 1, pp. 537–551. DOI: 10.32604/cmc.2020.010691.
35. **Batista, A. F. M., Miraglia, J. L., Donato, T. H. R., Chiavegatto Filho, A. D. P. (2020).** COVID-19 diagnosis prediction in emergency care patients: a machine learning approach. MedRxiv. DOI:10.1101/2020.04.04.20052092.
36. **Schwab, P., Schütte, A. D., Dietz, B., Bauer, S. (2020).** predCOVID-19: A Systematic Study of Clinical Predictive Models for Coronavirus Disease 2019. ArXiv:2005.08302, Vol. 76, pp. 1–8.
37. **Alakus, T. B., Turkoglu, I. (2020).** Comparison of deep learning approaches to predict COVID-19 infection. Chaos, Solitons and Fractals, Vol. 140, pp. 1–7. DOI: 10.1016/j.chaos.2020.110120.
38. **Kaggle (2020).** Diagnosis of COVID-19 and its clinical spectrum. August 18, 2020. Available from: https://www.kaggle.com/einsteindata4u/covid19.
39. **Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., Wong, T. Y., Cheng, C. Y. (2020).** Logistic regression was as good as machine learning for predicting major chronic diseases, Journal of Clinical Epidemiology, Vol. 122, pp. 56–69. DOI: 10.1016/j.jclinepi.2020.03.002.
40. **Schluter, P. J. (2011).** The trauma and injury severity score (TRISS) revised. Injury, Vol. 42, No. 1, pp. 90–96. DOI: 10.1016/j.injury.2010.08.040.
41. **Aboagye-Mensah, E. B., Azap, R. A., Odei, J. B., Gray, D. M., Nolan, T. S., Elgazzar, R., White, D., Gregory, J., Joseph, J. J. (2020).** The association of ideal cardiovascular health with self-reported health, diabetes, and adiposity in African American males. Preventive Medicine Reports, Vol. 19. DOI:10.1016/j.pmedr.2020.101151.
42. **Ahmed, H., Younis, E. M. G., Hendawi, A., Ali, A. A. (2020).** Heart disease identification from patients' social posts, machine learning solution on spark. Future Generation Computer Systems, Vol. 111, pp. 714–722. DOI: 10.1016/j.future.2019.09.056.
43. **Morais-Rodrigues, F., Silvério-Machado, R., Kato, R. B., Rodrigues, D. L. N., Valdez-Baez, J., Fonseca, V., San, E. J., Gomes, L. G. R., dos Santos, R. G., Viana, M. V. C., et al. (2020).** Analysis of the microarray gene expression for breast cancer progression after the application modified logistic regression. Gene, Vol. 726, pp. 1–8. DOI: 10.1016/j.gene.2019.144168.
44. **Fukunishi, H., Nishiyama, M., Luo, Y., Kubo, M., Kobayashi, Y. (2020).** Alzheimer-type dementia prediction by sparse logistic regression using claim data. Computer Methods and Programs in Biomedicine, Vol.

196, pp. 1–8. DOI: 10.1016/j.cmpb.2020.105582.

45. **Chen, L., Wang, C., Chen, J., Xiang, Z., Hu, X. (2021).** voice disorder identification by using Hilbert-Huang Transform (HHT) and K Nearest Neighbor (KNN). Journal of Voice, Vol. 35, No. 6, pp. 932.e1–932.e11. DOI: 10.1016/j.jvoice.2020.03.009.
46. **Kaplan, K., Kaya, Y., Kuncan, M., Ertunç, H. M. (2020).** Brain tumor classification using modified local binary patterns (LBP) feature extraction methods. Medical Hypotheses, Vol. 139. DOI: 10.1016/j.mehy.2020.109696.
47. **Balaji, V. R., Suganthi, S. T., Rajadevi, R., Kumar, V. K., Balaji, B. S., Pandiyan, S. (2020).** Skin disease detection and segmentation using dynamic graph cut algorithm and classification through Naive Bayes classifier. Measurement, Vol. 163. DOI: 10.1016/j.measurement.2020.107922.
48. **Shah, S. M. S., Shah, F. A., Hussain, S. A., Batool, S. (2020).** Support vector machines-based heart disease diagnosis using feature subset, wrapping selection and extraction methods. Computers & Electrical Engineering, Vol. 84. DOI: 10.1016/j.compeleceng.2020.106628.
49. **Panigrahi, R., Borah, S. (2019).** Classification and analysis of facebook metrics dataset using supervised classifiers. Social Network Analytics, Computational Research Methods and Techniques, Elsevier, pp. 1–19. DOI: 10.1016/B978-0-12-815458-8.00001-3.
50. **Radha, P., Divya, R. (2020).** An efficient detection of HCC-recurrence in clinical data processing using boosted decision tree classifier. Procedia Computer Science, Vol. 167, pp. 193–204. DOI: 10.1016/j.procs.2020.03.196.
51. **Wadekar, A. S. (2020).** Understanding opioid use disorder (OUD) using tree-based classifiers. Drug and Alcohol Dependence, Vol. 208. DOI: 10.1016/j.drugalcdep.2020.107839.
52. **Proniewska, K., Pregowska, A., Malinowski, K. P. (2020).** Identification of human vital functions directly relevant to the respiratory system based on the cardiac and acoustic parameters and random forest. IRBM, Vol. 42, No. 3, pp. 174–179. DOI: 10.1016/j.irbm.2020.02.006.
53. **Li, J., Tian, Y., Zhu, Y., Zhou, T., Li, J., Ding, K., Li, J. (2020).** A multicenter random forest model for effective prognosis prediction in collaborative clinical research network. Artificial Intelligence in Medicine, Vol. 103. DOI: 10.1016/j.artmed.2020.101814.
54. **Tan, C., Chen, H., Xia, C. (2009).** Early prediction of lung cancer based on the combination of trace element analysis in urine and an Adaboost algorithm. Journal of Pharmaceutical and Biomedical Analysis, Vol. 49, No. 3, pp. 746–752. DOI: 10.1016/j.jpba.2008.12.010.
55. **Hu, G., Yin, C., Wan, M., Zhang, Y., Fang, Y. (2020).** Recognition of diseased pinus trees in UAV images using deep learning and AdaBoost classifier. Biosystems Engineering, Vol. 194, pp. 138–151. DOI: 10.1016/j.biosystemseng.2020.03.021.
56. **Mujumdar, A., Vaidehi, V. (2019).** Diabetes prediction using machine learning algorithms. Procedia Computer Science, Vol. 165, pp. 292–299. DOI: 10.1016/j.procs.2020.01.047.
57. **Javan, S. L., Sepehri, M. M., Javan, M. L., Khatibi, T. (2019).** An intelligent warning model for early prediction of cardiac arrest in sepsis patients. Computer Methods and Programs in Biomedicine, Vol. 178, pp. 47–58. DOI: 10.1016/j.cmpb.2019.06.010.
58. **Worth, A. P., Cronin, M. T. D. (2003).** The use of discriminant analysis, logistic regression and classification tree analysis in the development of classification models for human health effects. Journal of Molecular Structure: THEOCHEM, Vol. 622, Nos. 1–2, pp. 97–111. DOI: 10.1016/S0166-1280(02)00622-X.
59. **Nkengfack, L. C. D., Tchiotsop, D., Atangana, R., Louis-Door, V., Wolf, D. (2020).** EEG signals analysis for epileptic seizures detection using polynomial transforms, linear discriminant analysis and support vector machines. Biomedical Signal Processing and Control, Vol. 62. DOI: 10.1016/j.bspc.2020.102141.

60. **Bari, M. F., Fattah, S. A. (2020).** Epileptic seizure detection in EEG signals using normalized IMFs in CEEMDAN domain and quadratic discriminant classifier. Biomedical Signal Processing and Control, Vol. 58. DOI: 10.1016/j.bspc.2019.101833.
61. **Georgiou-Karistianis, N., Gray, M. A., Domínguez D, J. F., Dymowski, A. R., Bohanna, I., Johnston, L. A., Churchyard, A., Chua, P., Stout, J. C., Egan, G. F. (2013).** Automated differentiation of pre-diagnosis Huntington's disease from healthy control individuals based on quadratic discriminant analysis of the basal ganglia: The IMAGE-HD study. Neurobiology of Disease, Vol. 51, pp. 82–92. DOI: 10.1016/j.nbd.2012.10.001.
62. **Shanthi, T., Sabeenian, R. S., Anand, R. (2020).** Automatic diagnosis of skin diseases using convolution neural network. Microprocessors and Microsystems, Vol. 76. DOI: 10.1016/j.micpro.2020.103074.
63. **Jain, R., Nagrath, P., Kataria, G., Kaushik, V. S., Hemanth, D. J. (2020).** Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. Measurement, Vol. 165. DOI: 10.1016/j.measurement.2020.108046.
64. **Cimr, D., Studnicka, F., Fujita, H., Tomaskova, H., Cimler, R., Kuhnova, J., Slegr, J. (2020).** Computer aided detection of breathing disorder from ballistocardiography signal using convolutional neural network. Information Sciences, Vol. 541, pp. 207–217. DOI: 10.1016/j.ins.2020.05.051.
65. **Atal, D. K., Singh, M. (2020).** Arrhythmia classification with ECG signals based on the optimization-enabled deep convolutional neural network. Computer Methods and Programs in Biomedicine, Vol. 196. DOI: 10.1016/j.cmpb.2020.105607.
66. **Savelli, B., Bria, A., Molinara, M., Marrocco, C., Tortorella, F. (2020).** A multi-context CNN ensemble for small lesion detection. Artificial Intelligence in Medicine, Vol. 103. DOI: 10.1016/j.artmed.2019.101749.
67. **Zhang, J., Liu, A., Gao, M., Chen, X., Zhang, X., Chen, X. (2020).** ECG-based multi-class arrhythmia detection using spatio-temporal attention-based convolutional recurrent neural network. Artificial Intelligence in Medicine, Vol. 106. DOI: 10.1016/j.artmed.2020.101856.
68. **Pellicer-Valero, O. J., Cattinelli, I., Neri, L., Mari, F., Martín-Guerrero, J. D., Barbieri, C. (2020).** Enhanced prediction of hemoglobin concentration in a very large cohort of hemodialysis patients by means of deep recurrent neural networks. Artificial Intelligence in Medicine, Vol. 107. DOI: 10.1016/j.artmed.2020.101898.
69. **Deng, M., Meng, T., Cao, J., Wang, S., Zhang, J., Fan, H. (2020).** Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. Neural Networks, Vol. 130, pp. 22–32. DOI: 10.1016/j.neunet.2020.06.015.
70. **Yang, J., Huang, X, Wu, H., Yang, X. (2020).** EEG-based emotion classification based on bidirectional long short-term memory network. Procedia Computer Science, Vol. 174, pp. 491–504. DOI: 10.1016/j.procs.2020.06.117.
71. **Ghosh, L., Saha, S., Konar, A. (2020).** Bi-directional long short-term memory model to analyze psychological effects on gamers. Applied Soft Computing, Vol. 95. DOI: 10.1016/j.asoc.2020.106573.
72. **Zhang, W., Han, J., Deng, S. (2019).** Abnormal heart sound detection using temporal quasi-periodic features and long short-term memory without segmentation. Biomedical Signal Processing and Control, Vol. 53. DOI: 10.1016/j.bspc.2019.101560.
73. **Guedes, V., Junior, A., Fernandes, Teixeira, F., Teixeira, J. P. (2018).** Long short term memory on chronic laryngitis classification. Procedia Computer Science, Vol. 138, pp. 250–257. DOI: 10.1016/j.procs.2018.10.036.
74. **Scikit Learn (2020).** Supervised learning, scikit-learn-0.23.2 documentation, August 31, 2020.
75. **Scikit Learn (2020).** Logistic Regression, sklearn.linear_model. LogisticRegression, scikit-learn-0.23.2 documentation, August 31, 2020.
76. **Scikit Learn (2020).** KNN, sklearn.neighbors. KNeighborsClassifier, scikit-learn-0.23.2 documentation, August 31, 2020.

77. **Scikit Learn (2020).** sklearn.svm.SVC, scikit-learn-0.23.2 documentation, August 31, 2020.
78. **Scikit Learn (2020).** Decision Tree, sklearn.tree.DecisionTreeClassifier, scikit-learn-0.23.2 documentation, August 31, 2020.
79. **Scikit Learn (2020).** Random Forest, sklearn.ensemble.RandomForestClassifier, scikit-learn-0.23.2 documentation, August 31, 2020.
80. **Scikit Learn (2020).** AdaBoost, sklearn.ensemble.AdaBoostClassifier, scikit-learn-0.23.2 documentation, August, 31, 2020.
81. **Scikit Learn (2020).** Naïve Bayes, sklearn.naive_bayes.GaussianNB, scikit-learn-0.23.2 documentation, August, 31, 2020.
82. **Scikit Learn (2020).** LDA, sklearn.discriminant_analysis. LinearDiscriminantAnalysis, scikit-learn-0.23.2 documentation, August, 31, 2020.
83. **Scikit Learn (2020).** Quadratic Discriminant Analysis, sklearn.discriminant_analysis. QuadraticDiscriminantAnalysis, scikit-learn-0.23.2 documentation, August, 31, 2020.
84. **Mandrekar, J. N. (2010).** Receiver operating characteristic curve in diagnostic test assessment. Journal of Thoracic Oncology, Vol. 5, No. 9, pp. 1315–1316. DOI: 10.1097/JTO.0b013e3181ec173d.

Article received on 16/10/2020; accepted on 20/12/2021.
Corresponding author is Prabhat Kumar.

ISSN 2007-9737

Luminescence Properties of Nanomaterials

María Elena Aguilar Jáuregui[1], Cuauhtémoc Peredo Macías[1],
Sandra Dinora Orantes Jiménez[1], Paulina Alejandra Flores de los Rios[2],
Eduardo San Martín Martínez[2*]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada-Legaria,
Mexico

{maguilar, cperedo, dinora}@cic.ipn.mx,
pfloresd1900@alumno.ipn.mx, esanmartin@ipn.mx

Abstract. Undoped and $CeCl_2$-doped ZnS nanomaterials were prepared at room temperature by a chemical synthesis method. X-ray diffraction analysis confirms a cubic crystal structure of zinc and the estimated size of the nanocrystals is in the range of 2-3 nm on average. Scanning electron microscopy observation shows a non-homogeneous structure due to the agglomeration of the nanomaterial. The characterization of PL indicates the high intensity of luminescence in the doped nanoparticles, so the application of this new nanomaterial in the medical and optoelectronic areas is possible.

Keywords. Nanomaterials, nanoparticles, lanthanides, photoluminescense, zinc sulfur.

1 Introduction

Nanostructured materials, such as nanoparticles, are considered to be less than 100 nm in size, so they have very different physical and chemical properties at their natural or raw scale. Its properties are determined by the crystallographic structure of the nanoparticle surface, its size, and its shape. Semiconductor nanoparticles have electrical, magnetic, and optical properties due to quantum confinement, so they can be used in various applications such as luminescent materials, optoelectronic devices (lasers) [1], sensors, plasma television screens, light-emitting diodes white (LED) [2], fluorescent lamps [3] biosensors among others. As luminescent nanoparticles, they can be highly fluorescent when excited by ultraviolet light, so they can be used as an alternative for organic and inorganic staining.

In applications associated with luminescence, a high luminescence efficiency is usually required, which can be modified by the presence of impurifiers, by an active ion dispersed in the crystal structure of the matrix or by the type of crystal structure itself [4].

There are several compounds that are used as a matrix to obtain luminescent materials and that can be inpurified with oxides, phosphates, fluorides or lanthanides. Zinc sulfide (ZnS) is a semiconductor used commercially as a photoluminescent and especially if it is doped [5-6].

It has been investigated that the lanthanides or rare earths used as dopants for semiconductor nanoparticles, in addition to being luminescent compounds, reduce their environmental impact, are biocompatible, and have less toxicity and greater chemical and thermal stability [7-11].

Lanthanides have properties that allow tuning the optical properties of semiconductor materials.

An important feature of these materials is their narrow emission bandgap, which represents a great advantage for their use in flat panel displays, medical labeling, disease detection, and imaging [12], as well as in fluorescent probes and can be used in vivo, and in vitro [13].

Quantum confinement in semiconductor nanomaterials is important because it creates new optical, electrical, and mechanical properties of the materials. Therefore, the size and shape of the particles [14] have a dramatic effect on the density of electronic states and thus on the optical response.

In this work, we present a nanoprecipitation method for the synthesis of undoped and $CeCl_2$ doped ZnS nanoparticles, to observe their optical properties, where nanoparticles or quantum dots with sizes around 10 nm can emit red and smaller sizes emit red. in orange, yellow, and green [15].

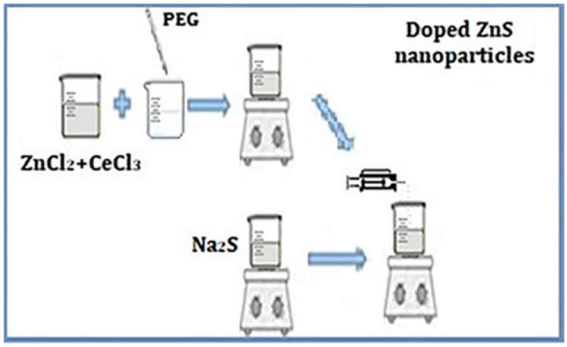


Fig. 1. Synthesis of doped semiconductor nanoparticles

2 Materials and Methods

2.1 Synthesis

The synthesis method used to obtain metallic nanoparticles is based on nanoprecipitation reactions, which are carried out through controlled release processes of precipitating cations or anions at moderate temperatures. It is a simple and low-cost colloidal method of processing [16].

2.1.1 ZnS Nanoparticles

The nanoparticles were developed by a colloidal method and chemical precipitation [16] using zinc chloride ($ZnCl_2$) and sodium sulfide (Na_2S), purchased from Sigma Aldrich, as initiators. 0.039 $ZnCl_2$ was dissolved in 50 mL of deionized water and the solution was magnetically stirred for 20 min at room temperature (RT). Subsequently, 1.22 g of Na_2S were dissolved in 72 ml of distilled water and stirred magnetically for 20 minutes at room temperature.

The resulting solution was added dropwise to the previously prepared solution. A solution of 0.5 g of PEG in 10 mL of deionized water was prepared, as a passivating agent and added to the ZnS solution until complete dissolution was achieved.

2.1.2 Doped ZnS Nanoparticles

Using zinc chloride ($ZnCl_2$) and sodium sulfide (Na_2S) as initiators and $CeCl_3$ as a doping agent. A solution of 0.7039 g of $ZnCl_2$ and 1% w/v $CeCl_3$ was dissolved in 50 ml of deionized water and the solution was stirred magnetically for 20 min at room temperature. Separately, 1.22 g of Na_2S was dissolved in 72 ml of distilled water, and the solution was magnetically stirred for 20 minutes at room temperature. The resulting solution was added dropwise, to the previously prepared solution. 0.5 g of PEG solution was added as a passivating agent to obtain the precipitate (Figure 1)

2.1.3 Dispersion and Filtration

The solution was subjected to ultrasonic dispersion at 45 kHz for 30 min (TI-H-5, Elma, Germany) and then filtered using 0.22 µm Millipore membrane filters.

In this type of synthesis, the nanoparticles are dispersed within a dispersing medium (water). They are prepared from colloidal solutions of nanoparticles with a surfactant coating that controls the surface energy by intermolecular forces to prevent their aggregation and increase their size and shape and modify their optical luminescence properties [17].

2.1.4 Centrifugation and Washes

The resulting precipitate from each sample was centrifuged and washed [18].

2.1.5 Drying of the Nanomaterial

The precipitate was dried at 60 °C for 6 h to remove any organic residue, water, and other by-products formed that might evaporate at this temperature.

After drying, the precipitate was crushed with an agate mortar to obtain a fine semiconducting nanoparticle powder.

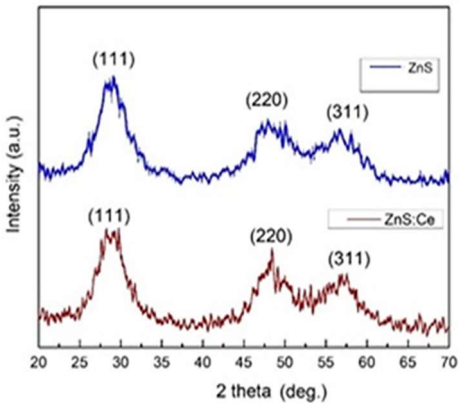


Fig. 2. XRD patterns of ZnS and Ce^{+3} doped ZnS nanoparticles

2.2 Characterization of Nanoparticles

2.2.1 X- Ray Diffraction Pattern (XRD)

The synthesized nanoparticle samples were examined by X-ray diffraction (XRD), Bruker-AXS, model D8-advance coupled to a 1.5406 A° copper anode X-ray tube, using a diffracted beam monochromator to select the radiation. Kα, to analyze the crystalline structure of the nanomaterial and estimate the size of the crystal. The diffractogram obtained shows a series of maxima called diffraction patterns, which correspond to the constructive interference of waves elastically scattered by the atomic planes of the crystal [19].

The position and intensity of these peaks provide information about the crystal structure. To determine the average size of the crystals in the analyzed sample, the Debye-Scherrer equation was used considering the main diffraction peak in the (111) plane, the broadening of the maximum diffraction peak defines the size of the crystal.

2.2.2. Scanning Electron Microscopy (SEM)

Through the technique of scanning electron microscopy or SEM (JEM-6390 LV Scanning Electron Microscope, Jeol, Japan) and energy-dispersive X-ray spectroscopy (EDS), the microstructure of the nanoparticles and the redistribution of elements in the sample were determined.

2.2.3. Optical Properties

The excitation and emission wavelengths depending on the size and shape of the nanoparticles, as well as the synthesis conditions, were considered [20]. The optical properties were characterized by employing a spectrofluorimeter (Edinburgh Instruments FS5, United Kingdom). and country). To determine the intensity of the photoluminescence of the granules of the nanomaterial, and obtain the excitation and emission PL spectra of the nanoparticles.

3 Results

3.1 X- ray Diffraction Analysis

The diffractograms of the samples were obtained in the three main peaks, corresponding to the peaks that are assigned to the planes (111), (220), and (311) of the cubic phase of ZnS, which is observed in Figure 2. Small shifts were observed in the peaks of the diffraction pattern of the doped sample, so it can be thought that there is a good integration of the doping material in the parameters of the lattice due to the concentration of $CeCl_3$.

Using the Debye-Scherrer equation (1), information on the average size of the synthesized nanoparticles and their internal structure was obtained [20]:

$$\tau = \frac{k\lambda}{\beta \cos\theta}. \quad (1)$$

where τ gives the average diameter of the nanocrystals in the direction perpendicular to the related planes. The values of the total width at half of the main peak maximum (FWHM) of the undoped and doped samples and the estimated values of the nanoparticle size are presented in Table 1.

3.2 SEM Analysis

In the SEM micrographs, the morphology of the pure and doped ZnS nanoparticles was verified, being found to be very similar. A homogeneous surface could be expected, however, the surface may have defects due to the crystal growth process or due to agglomeration presenting an

inhomogeneous surface. SEM micrographs show, that quasi-spherical nanoparticles with sizes less than 100 nm can be observed (Figure 3).

3.3 Photoluminescence Studies

The photoluminescence spectra of pure and doped nanoparticles are presented. The excitation wavelength is 330 nm and the emission wavelength in the blue is 368 nm for the pure nanoparticles as shown in figure 4 (a) y (b).

It can be observed that for Ce^{+3} doped nanoparticles there is a shift in the wavelength towards blue at 447 nm and with 2 times more luminescence intensity than pure nanoparticles.

4 Conclusions

Undoped and Cerium-doped ZnS nanoparticles were successfully synthesized using a precipitation reaction that requires a relatively short time for its reaction and was carried out at RT, obtaining nanoparticles with sizes between 2-100 nm. It was observed that the size of the nanoparticles depends on the passivating or stabilizing agent, so these sizes can be changed by modifying the concentration of the passivating agent and the concentrations in the precursors.

The ZnS nanoparticles present a cubic structure as observed in the X-ray diffraction patterns. The resulting sizes after applying the Debye-Scherrer equation are 2.25 nm and 2.98 nm for the doped material.

The PL analysis shows an emission with a wavelength of 368 nm for the undoped nanoparticles applying UV light and under an excitation spectrum of 330 nm, while for the doped nanoparticles the light emission was presented with a shift at 433 nm under an excitation spectrum of the same wavelength.

This favors their potential application of the doped nanoparticles in optoelectronic devices and for their possible use in imaging for disease detection. Lanthanide ion doped nanoparticles have higher luminescence intensity almost twice as much as undoped nanoparticles. From the analysis of the quantum confinement effect in the literature it is known that it is the nanoparticles

Table 1. FWHM values and estimated size of the synthesized nanoparticles

Nanoparticles	FWHM	Size (nm)
ZnS	0.06981	2.25
ZnS:Ce	0.05270	2.98

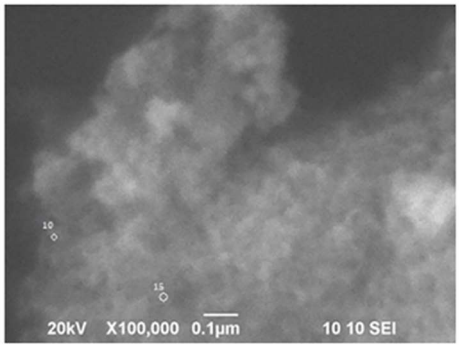


Fig. 3. SEM micrograph of the nanomaterial

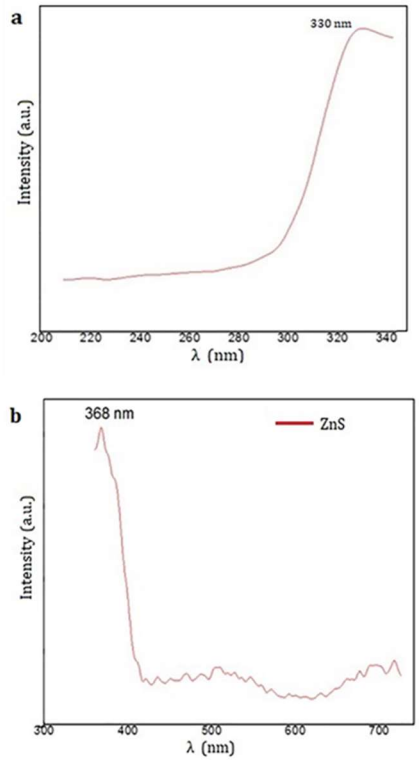


Fig. 4. (a) Excitation spectra at 330 nm and (b) Emission spectra at 368 nm of ZnS nanoparticles

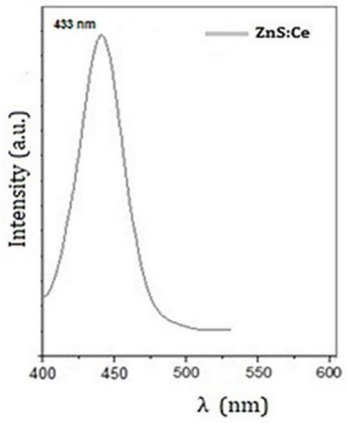


Fig. 5. Emission spectrum at 433 nm of ZnS nanoparticles doped with Ce^{+3}

smaller than 10 nm that meet the characteristics of intense light emission and long decay time.

Acknowledgments

The authors are grateful for the support provided by the Instituto Politécnico Nacional under the project “Study of the Luminescent Properties of Semiconductor Nanoparticles Based on Rare Earths” to carry out this research.

References

1. **Henderson, B., Imbusch, G. T. (1989).** Optical spectroscopy of inorganic solids. Clarendon Press, Oxford.
2. **Ye, S., Xiao, F., Pan, Y. X., Ma, Y. Y., Zhang, Q. Y. (2010).** Phosphors in phosphor-converted white light-emitting diodes: Recent advances in materials, techniques and properties. Materials Science and Engineering: R: Reports. Vol. 71, pp. 1–34.
3. **Terna, A. D., Elemike, E. E., Mbonu, J. I., Osafile, O. E., Ezeani, R. O. (2021).** The future of semiconductors nanoparticles: Synthesis, properties and applications. Materials Science and Engineering: B, 272, 115363.
4. **Vo-Dinh, T. (2003).** Biomedical photonics handbook. Second Edition, III Volume Set, CRC Press, Taylor & Francis, USA.
5. **Iranmanesh, P., Saeednia, S., Nourzpoor, M. (2015).** Characterization of ZnS nanoparticles synthesized by co-precipitation method. Chin. Phys. B, Vol. 24, No. 4, 046104.
6. **Sharma, R. K., Mudring, A. V., Ghosh, P. (2017).** Recent trends in binary and ternary rare-earth fluoride nanophosphors: How structural and physical properties influence optical behavior. Journal of Luminescence, Vol. 189, pp. 44–63.
7. **Wang, M., Abbinenib, G., Clevenger, A., Mao, C., Xu, S. (2011).** Upconversion nanoparticles: synthesis, surface modification, and biological applications. Nanomedicine, Vol. 7, No. 6, pp. 710–729.
8. **Voncken, J. H. L. (2016).** The rare earth elements: An Introduction. Springer, pp. 106.
9. **Hardman, R. A. (2006).** Toxicologic review of quantum dots: Toxicity depends on physicochemical and environmental factors. Environmental Health Perspectives, Vol. 114, No. 2, pp. 165–172.
10. **Auffan, M., Rose, J., Bottero, J. Y., Lowry, G. V., Jolivet, J. P., Wiesner, M. R. (2009).** Towards a definition of inorganic nanoparticles from an environmental, health and safety perspective. Nature Nanotechnology, Vol. 4, No. 10, pp. 634–641.
11. **Shen, J., Sun, L. D, Yan, C. H. (2008).** Luminescent rare earth nanomaterials for bioprobe applications. Dalton Transactions, Vol. 42, No. 9226, pp. 5687–5697.
12. **Harish, G. S., Sreedhara, R. P., Yan, C. H. (2015).** Synthesis and characterization of Ce, Cu co-doped ZnS nanoparticles. Physica B, Vol. 473, pp. 48–53.
13. **Parak, W.J; Gerion, D., Pellegrino, T., Zanchet, D., Micheel, C., Williams, S. C., Boudreau, R., LeGros, M. A., Larabell, C. A. Alivisatos, A. P. (2003)**. Biological applications of colloidal nanocrystals. Nanotechnology 14, R15-R27.
14. **Jeevanandam, J., Barhoum, A., Chan, Y. S., Dufresne, A., Danquah, M. K. (2018).** Review on nanoparticles and nanostructured

materials: History, sources, toxicity and regulations. Beilstein J. Nanotechnology, Vol. 9, pp. 1050–1074.

15. **Onyia, A. I., Ikeri, H. I., Nwobodo, A. N. (2018).** Theoretical study of the quantum confinement effects on quantum dots using particle in a box model. Journal of Ovonic Research, Vol. 14, No. 1, pp. 49–54.
16. **Hedayati, K., Zendehnam, A., Hassanpour, F. (2016).** Fabrication and characterization of zinc sulfide nanoparticles and nanocomposites prepared via a simple chemical precipitation method. J. Nanostruct, Vol. 6, No. 3, pp. 207–212.
17. **Guerrini, L., Alvarez, R. A., Pazos, N. (2018).** Surface modifications of nanoparticles for stability in biological fluids. Materials, Vol. 11, pp. 1154.
18. **Uekawa, N., Matsumoto, T., Kojima, T., Shiba, F., Kakegawa, K. (2010).** Synthesis of stable sol of ZnS nanoparticles by heating the mixture of ZnS precipitate and ethylene glycol. Colloids and Surfaces A: Physicochem. Eng. Aspects, Vol. 361, pp. 132–137.
19. **Giannini, C., Ladisa, M., Altamura, D., Siliqi, D., Silbano, T., De Caro, L. (2016).** X-ray diffraction: A powerful technique for the multiple-length-scale structural analysis of nanomaterials. Crystals, 6, 87.
20. **Hu, Y., Wei, Z., Wu, B. (2018).** Photoluminescence of ZnS:Mn quantum dot by hydrothermal method, AIP Advances Vol. 8, pp. 015014.

Article received on 14/10/2021; accepted on 01/12/2022.
Corresponding author is Eduardo San Martin-Martinez.

ISSN 2007-9737

Convolutional Neural Network for Improvement of Heart Valve Disease Detection

Blanca Tovar-Corona[2], Santiago Isaac Flores-Alonso[1], René Luna-García[1]

[1]Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2]Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas,
Mexico

{bltovar, rlunag}ipn.mx
sfloresa2010@alumno.ipn.mx

Abstract. Heart Valve Disease (HVD) encompasses a number of common cardiovascular conditions that account for a significant percentage of heart diseases. At present, the acoustic phenomena generated by the abnormal functioning of the heart valves can be recorded and digitized using electronic stethoscopes known as phonocardiographs. The analysis of the phonocardiographic signals has made it possible to indicate that the normal and pathological records differ in terms of both temporal and spectral characteristics. The present work describes the construction and implementation of a Deep Learning (DL) algorithm for the binary classification of normal and abnormal heart sounds. The performance of this approach reached an accuracy higher than 98 % and specificities in the "Normal" class of up to 99 %.

Keywords. Artificial intelligence, deep neural network, phonocardiography, heart valve disease.

1 Introduction

Heart noises are the expression of the opening and closing of the four cardiac valves, where the muscular contraction that drives the blood from one cavity to another generates a high acceleration and delay of the blood flow causing a pressure differential [12, 15]. Its normal physiological functioning is unidirectional, which allows the correct circulation of blood through the cardiovascular circuit. However, abnormal noises can be produced when the heart valves do not close or open completely, causing leaking backwards and the interruption of laminar blood flow by turning into a turbulent flow. These sounds are called murmurs, and their correct identification during auscultation, as part of the diagnosis procedure, is crucial to detect potentially life-threatening heart conditions.

Apart from traditional auscultation, these sounds can be recorded and digitized using electronic stethoscopes, which generate phonocardiographic (PCG) signals. The identification of abnormalities of the mechanical functioning of the heart is based on a series of features extracted from the PCG recordings, where computer-aided analysis allows to identify between normal and abnormal records, since these vary among themselves with respect to their temporal and spectral characteristics.

Therefore, the precise feature extraction is key for a correct classification of heart sounds and can play an important role in assisting the medical community in speeding up and improving the diagnosis.

This article addresses the problem of identifying abnormal heart conditions using features from the PCG recording in both time and spectral domain, extracted through a technique known

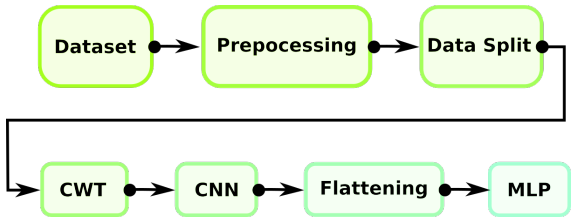


Fig. 1. Block diagram of the complete process: Dataset pre-processing and splitting, feature extraction, classification using a deep neural network

as Continuous Wavelet Transform, and a based Deep Learning (DL) methodology known as Convolutional Neural Networks (CNN). The output of the network grants the probability that a particular PCG recording belongs to a normal or abnormal class. The summary of the proposed model as a block diagram is shown in Fig. 1.

2 Related Work

Several laboratories, using particular datasets, have approached the heart sounds classification problem using their own distinctive AI methodology [14]. However, to make a correct comparison, it is necessary to select those works that use the same database as in the present work. Table 1 summarizes the feature extraction techniques and classifiers used, along with the results respectively obtained, using the same open access dataset [17].

However, a point noted is the trifle with which they approach the training of their models. It is possible to observe in [17, 18, 13] that the reported results are those obtained during training since the number of the samples shown in the confusion matrices, sums as the total of samples in the dataset.

This has important implications for the interpretation of the reported results since through training it is only possible to know the memorization capacity of the classification algorithm and the degree of compaction of the data. It is not possible to evaluate an actual performance if it is not through a test data set that the classifier has never seen.

Furthermore, the use of Convolutional Neural Networks (CNN), along with the spectral decomposition known as Continuous Wavelet Transform (CWT), has never been used to classify heart valve disease, placing the present work as a new methodological proposal.

3 Materials and Methods

This section summarizes the feature extraction techniques and Deep Learning algorithm used to address the problem apropos the HVD detection, along with the dataset description. The algorithmic proposal was developed in Python 3.9 on the Ubuntu 20.04 distribution. In particular, the deep learning algorithm was built on Keras 2.4.3.

3.1 Dataset

The PCG signals used in this article were obtained from an open database [17][1], containing 200 records for each of the following five classes:

- Aortic stenosis (AS).
- Mitral regurgitation (MR).
- Mitral stenosis (MS).
- Mitral valve prolapse (MVP).
- Normal (N).

Each signal was sampled at 8000 Hz, with durations of at least one second. To maintain uniformity in the data analysis, two windows of 6144 data points (0.768 s) were taken from each signal, each one containing at least one complete cardiac cycle, therefore, duplicating the number of samples from 200 to 400 for each class.

It is possible to notice that the Normal (N) and Pathological (AS, MR, MS, MV) classes, with a ratio of 4:1, are strongly unbalanced. This has implications for the model training, as mentioned in the previous section. Since the Normal class was separated into training and test subsets containing 320 (80%) and 80 (20%) time series, respectively, it was necessary to select the same subsets of the Pathological class to avoid the related bias. Therefore, 80 random samples of each subclass

[1] https://github.com/yaseen21khan/

Table 1. Comparative table between works that used the same dataset

Author	Feature Extraction	Classifier	Precision	Recall	Specificity	Global Accuracy
Son et al. 2018 [17]	DWT and MFCCs	SVM, KNN, DNN	–	98.2%	99.4%	97.9%
Alqudah, A. M. 2019 [4]	Eight statistical moments from the Instantaneous Frequency Estimation + PCA	KNN* and Random Forest	100%	98.28%	100%	94.8%
Ghosh, S.K. et al. 2019 [7]	Wavelet Synchrosqueezing Transform	Random Forest	–	–	–	95.13%
Upretee, P., and Yuksel, M. E. 2019 [18]	Centroid Frequency Estimation	SVM and KNN*	99.6%	99.76%	98.83%	96.5%
Ghosh, S.K. et al. 2020 [6]	Local energy and entropy from Chirplet Transform	WaveNet	98.0%	98.1%	99.3%	98.33%

(AS, MR, MS, MVP) were selected to structure the other half of the training subset.

Afterward, each time series was transformed using CWT. The implications of using this extraction technique and the procedure are discussed forward.

3.2 Continuous Wavelet Transform

CWT is a spectral decomposition method which is based on representing the signal in the form of wavelets with different displacement and scaling factors, where the use of the correct mother wavelet (MW) drives the enhancement of the waveforms of interest.

The MW is an effectively limited waveform in duration, with an average equal to zero. The MW used in the CWT was a Morlet, described by:

$$\psi(t) = e^{-\pi t^2} e^{i\pi t}. \tag{1}$$

And starting with an MW ψ, the family $\psi_{\tau,s}$ of "daughters wavelets" can be obtained by simply scaling and moving ψ:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{|s|}}\psi(\frac{t-\tau}{s}), \quad s, \tau \in \mathbb{R}, s \neq 0, \tag{2}$$

where s is a scaling or dilation factor that controls the width of the wavelet and τ is a translation parameter controlling its location. Scaling a wavelet simply means stretching it (if $|s| > 1$) or compressing it (if $|s| < 1$), while translating it simply means shifting its position in time [2].

Thus, the CWT of a signal $f(t)$ is given by [16]:

$$CWT(\tau, s) = \langle f, \psi_{\tau,s} \rangle \sum_{0}^{+\infty} f(t)\psi(\frac{t-s}{\tau})dt, \tag{3}$$

where the integral is solved for τ, s (shifting and scaling parameters), which performs a transformation of the signal $f(t)$ from the time domain to a function in the time domain and scale.

However, as a previous step to the CWT calculation, the Hilbert transform was implemented since this transform is an efficient tool to extract the time-localized amplitude and phase of a mono-component signal, with scale and translation invariance, and its energy-conserving (unitary) nature [5, 11]. The Hilbert transform $\hat{s}(t)$ of a function $s(t)$, is defined as the convolution of $(s(t) * 1/(\pi t))$ such that [9]:

$$\hat{s}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau} d\tau. \tag{4}$$

It is possible to observe that this gives us a complex representation. To retrieve all the information of the signal, it is necessary to select a complex MW such as Morlet. By applying the

ISSN 2007-9737

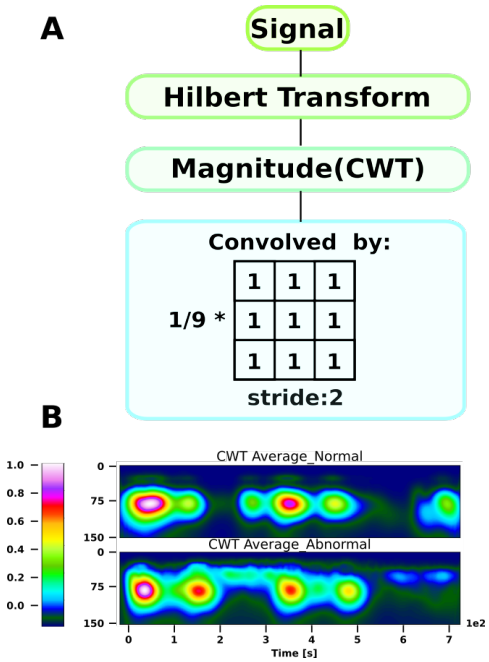


Fig. 2. CWT Results: **A)** Block diagram of the full CWT algorithm and post-processing to reduce dimensionality. **B)** Magnitude of the coefficients obtained for each scale at each time point, averaged for each of the class sets

CWT, we obtain a matrix representation of the coefficients of size NxM, as shown in Fig. 2B. To reduce computational demand, it was necessary to apply an averaging 3×3 filter as shown in Fig. 2A, which highly reduces the matrix size.

3.3 Convolutional Neural Network (CNN)

Deep learning refers to AI models capable of extracting features, with multiple levels of abstraction and learning representations of data, without the need for a human expert agent that transformed the raw data into suitable internal features from which the learning subsystem, could detect or classify patterns in the input [8].

In particular, CNN discovers intricate patterns in datasets by using the backpropagation to optimize how a set of filters need to change their internal parameters to compute the attributes that best represent the data in a highly compact depiction [10].

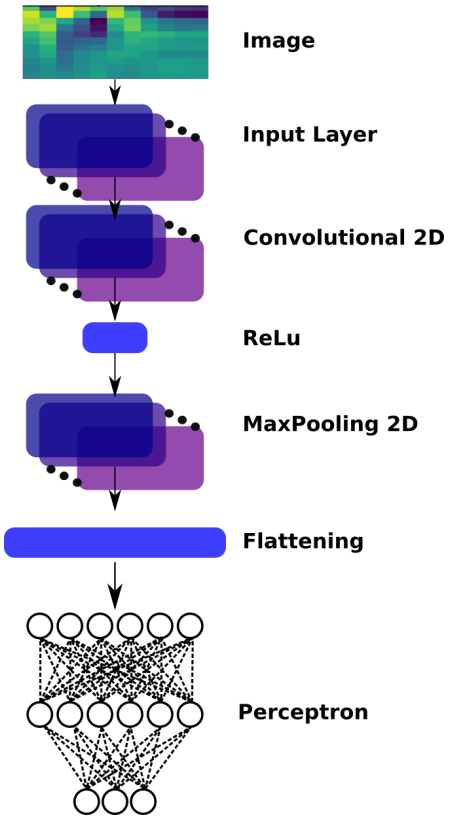


Fig. 3. Block diagram of the complete process: Dataset pre-processing and splitting, feature extraction, classification using a deep neural network

The proposition of the decomposition into a spectral space through CWT may be counterintuitive. However, since CNN uses filters that look for local spatial patterns (the locality depends on the size of the filter), the frequency dynamics of the PCG records over time contain richer information than the simple temporal dynamics of the time series.

As it is possible to see in the flow chart shown in Fig. 3, the CNN introduces a special network structure, which consists of the so-called convolution and grouping layers alternately that allow extracting the main characteristics of the coefficient matrices [1].

When using CNN for pattern recognition in phonocardiographic sounds, the input data must be organized as a series of feature maps. Since

CWT was used to find spectral coefficients along time, the expected input structure for a 2D CNN occurred naturally, where each of the coefficients represents the pixel values.

Once the input feature maps are formed, the convolution and grouping layers apply their respective operations to generate the activation of the units in those layers, in sequence. The discrete convolution between the filter and the coefficient matrix is mathematically defined as:

$$conv(I,K)_{x,y} = \sum_{i=0}^{n_{f1}-1} \sum_{j=0}^{n_{f2}-1} K_{(i,j)} I_{(x+i,y+j)}. \quad (5)$$

It is possible to deduce that, if the image dimension is given by (n_H, n_W) and, the filter dimensions is given by (f_1, f_2), the dimension of the convolution will be:

$$dim(I * K) = \Big[\frac{n_H - f1}{s} + 1, \frac{n_W - f2}{s} + 1\Big]. \quad (6)$$

Max-pooling is a particular case of a convolutional layer, where the filter is a matrix of ones and, after the convolution, a maximum function is applied. By convention, we consider a square filter with dimensions $f1 = f2 = 2$ and $s = 2$. This operation is defined as:

$$max\big(K_{(i,j)} I_{(x+i,y+j)}\big). \quad (7)$$

In CNN terminology, the pair of convolution and max-pooling layers in succession is often referred to as a convolutional layer [3]. Each of these layers is in charge of finding, building attributes and reducing the dimensionality of the input matrix to a characteristic pattern.

Finally, this pattern is vectorized (flattened) and fed to a multilayer perceptron network (MLP), which will act as a classifier. In reality, nothing prevents the use of any other architecture or classification model, however by convention MLP is the most commonly used.

The proposed architecture of the CNN is described as pseudocode in the Algorithm 1.

3.4 Performance

The evaluation and validation of the machine learning algorithm is an essential part of any AI project. The model can give satisfactory results when it is evaluated using a metric, such as accuracy, but most of the time using a single metric is not enough to judge the performance of our model. That is why, in this section, the four evaluation metrics used are defined, where the primary building blocks are the true positive(tp), true negative(tn), false positive(fp) and false negative(fn) instantiations. In our particular case, the tp cases are the PCG recordings labelled as Normals. Therefore, the golden goal is to build a classifier with 0% fp, thus ensuring that no patient with any HDV is classified as Normal, which could pose a risk to their health and even death.

Accuracy: It is the ratio between the number of correct predictions and the total number of input samples:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}. \quad (8)$$

Due to its construction, this metric is not ideal when the classes are unbalanced (as is the case with the dataset used). The problem arises when the cost of misclassifying samples from minor classes is very high. If we are faced with a rare but fatal disease, the cost of not diagnosing a sick person's illness is far greater than the cost of sending a healthy person for further tests. Therefore, it is necessary to use metrics based on relevance, that is, that do take into account the imbalance of the classes, such as precision and recall.

Precision and recall: Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances:

$$Precision = \frac{tp}{tp + fp}. \quad (9)$$

While recall (also known as sensitivity) is the fraction of relevant instances that were retrieve:

$$Recall = \frac{tp}{tp + fn}. \quad (10)$$

Algorithm 1 CNN Architecture

1: **input:** CWT
2: $CNN \leftarrow Convolution\ layer\ (16\ Filters\ (21 \times 21))$
3: $Batch\ Normalization\ +\ Nonlinear\ layer\ +\ MaxPooling$
4: $CNN \leftarrow Convolution\ layer\ (8\ Filters\ (11 \times 11))$
5: $Batch\ Normalization\ +\ Nonlinear\ layer\ +\ MaxPooling$
6: $CNN \leftarrow Convolution\ layer\ (4\ Filters\ (7 \times 7))$
7: $CNN \leftarrow Flatten()(CNN)$
8: $MLP \leftarrow 2\ output\ neurons$
9: **output:** $Membership\ probability$

Finally, since in a clinical test the goal is to accurately identify people who have a particular condition (where its misclassification into a non-pathological class could be fatal), the ratio between true negatives and false positives should be accounted for, giving rise to a metric known as specificity. In other words, specificity measures how the test is effective when used on negative individuals:

$$Specificity = \frac{tn}{tn + fp}. \tag{11}$$

4 Results

During the construction of the model, the experimentation focused on two variables: the number of scales to be used in the CWT and the generation of the training subset, which as mentioned in section 3.1, is partially built from 320 pseudorandomly selected items from the Pathological (AS, MR, MS, MVP) subclasses.

For the case of CWT, the value of the power coefficients obtained for each scale at each time point, averaged for each of the class sets, with 150 scales is shown in Fig 2. The number of scales depends on the MW used to perform the decomposition, since each MW has a specific morphology and central frequency that will change as a function of scale. There is an approximate relationship between scale and frequency defined as:

$$s(fr) = \frac{ln(\frac{cf * fs}{fr})}{ln(2)}, \tag{12}$$

where s is the approximate scale, cf is the central frequency of the MW, fs is the sampling frequency and fr is the target frequency to approximate. However, this approximation is not exact and that is why the selection of the MW, number of scales and subscales can be defined as a hyperparameter. For the PCG records used in the present work, the 150 scales of the complex Morlet proposed as MW showed the level of detail sought.

On the other hand, to ensure that the CNN's performance was since the optimum (local) minimum was found, which ensures the generalizability of the model, and not from the pseudo-random selection of the data, a 6-fold cross-validation method was applied, where the overall accuracy obtained was 97.70 $\pm$ .432.

By having an overall accuracy with a standard deviation of less than 0.5%, the proposed model execution can be attributed to its generalizability, which allows us to select the best of the runs of our classifier to evaluate its performance. Fig. 4B shows the detailed accuracy, precision, recall and specificity obtained using 20 % of the dataset as the test set. It is possible to observe that 98.2% of the classes were correctly classified according to the binary accuracy.

Furthermore, the confusion matrix, from where all the metric calculations were based, is shown in Figure 4A, where each column of the matrix represents the number of predictions of each class, while each row represents the instances in the real class.

5 Conclusion

This article focused on the classification of HVD from 1000 PCG records, combining a deep learning algorithm with time-frequency analysis wherein the time-series recognition problem is transformed into an image recognition problem. To do so, the spectral characteristics through time were extracted using CWT, and given the dimensional nature of these features, it was decided to use a CNN to classify each recording as Normal or Pathological, since this is the first step in the diagnostic procedure. If an abnormality is present, further clinical tests must be carried out to determine the type of abnormality. This approach

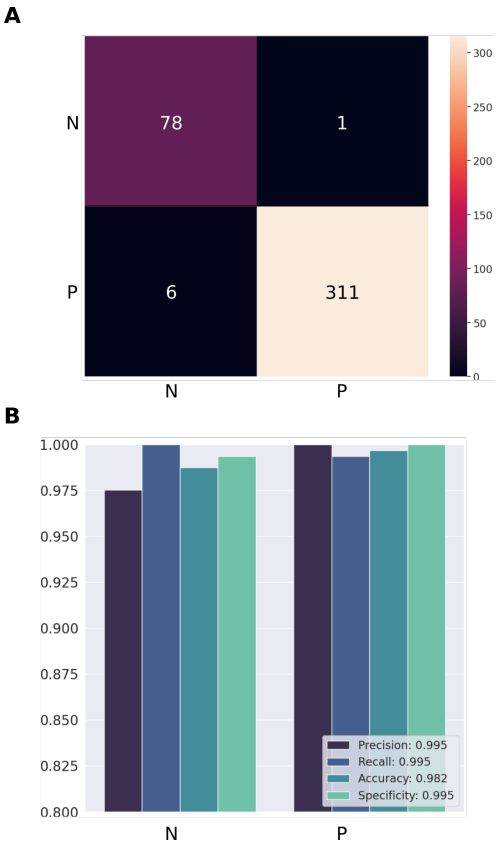


Fig. 4. DL performance results (20% of the dataset as the test set): **A** Confusion matrix used to calculate the metrics. **B** Precision, Recall, Specificity and Accuracy for the test dataset classification. The vertical axis shows only the percentage from 0.8 to 1.0 to facilitate the visualization of the results. ”N” and ”P” stand for Normal and Pathological class respectively

has never been used, placing it as an innovative methodological proposal.

Furthermore, the model had a performance, measured through its accuracy, above 98.2%, surpassing four of the five models described in the literature (Table 1), placing it as a competitive and efficient model for the classification of valvular diseases.

In addition, one of the necessary metrics to measure competitiveness in clinical diagnostic systems, and where the present work takes into account and stands out, is specificity (section 3.4), obtaining 99.5%, which means that less than 1% of the Pathological PCG records will be classified as Normal.

This provides robustness to the model and invites to implement it in a system for the assisted diagnosis of heart valve diseases to improve the prognosis of patients, reducing the error associated with the experience of the medical crew.

Acknowledgments

This research was funded by the Instituto Politécnico Nacional through the project SIP20210473. We thank CONACyT for partial support of the present work.

References

1. **Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., Yu, D. (2014).** Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, Vol. 22, No. 10, pp. 1533–1545.
2. **Aguiar-Conraria, L., Soares, M. J. (2014).** The continuous wavelet transform: Moving beyond uni-and bivariate analysis. Journal of Economic Surveys, Vol. 28, No. 2, pp. 344–375.
3. **Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017).** Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET), Ieee, pp. 1–6.
4. **Alqudah, A. M. (2019).** Towards classifying non-segmented heart sound records using instantaneous frequency based features. Journal of medical engineering & technology, Vol. 43, No. 7, pp. 418–430.
5. **Chaudhury, K. N., Unser, M. (2009).** Construction of Hilbert transform pairs of wavelet bases and Gabor-like transforms. IEEE Transactions on Signal Processing, Vol. 57, No. 9, pp. 3411–3425.
6. **Ghosh, S. K., Ponnalagu, R., Tripathy, R., Acharya, U. R. (2020).** Automated detection of heart valve diseases using chirplet transform and multiclass composite classifier with pcg signals. Computers in biology and medicine, Vol. 118, pp. 103632.

7. **Ghosh, S. K., Tripathy, R. K., Ponnalagu, R., Pachori, R. B. (2019).** Automated detection of heart valve disorders from the pcg signal using time-frequency magnitude and phase features. IEEE Sensors Letters, Vol. 3, No. 12, pp. 1–4.
8. **Goodfellow, I., Bengio, Y., Courville, A. (2016).** Deep learning. MIT press.
9. **Johansson, M. (1999).** The Hilbert transform. Mathematics Master's Thesis. Växjö University, Suecia. Disponible en internet: http://w3. msi. vxu. se/exarb/mj_ex. pdf, consultado el, Vol. 19.
10. **Liu, T., Fang, S., Zhao, Y., Wang, P., Zhang, J. (2015).** Implementation of training convolutional neural networks. arXiv preprint arXiv:1506.01195.
11. **Mahato, S., Teja, M. V., Chakraborty, A. (2017).** Combined wavelet–Hilbert transform-based modal identification of road bridge using vehicular excitation. Journal of Civil Structural Health Monitoring, Vol. 7, No. 1, pp. 29–44.
12. **Mondal, A., Kumar, A. K., Bhattacharya, P., Saha, G. (2013).** Boundary estimation of cardiac events s1 and s2 based on Hilbert transform and adaptive thresholding approach. 2013 Indian Conference on Medical Informatics and Telemedicine (ICMIT), IEEE, pp. 43–47.
13. **Oh, S. L., Jahmunah, V., Ooi, C. P., Tan, R.-S., Ciaccio, E. J., Yamakawa, T., Tanabe, M., Kobayashi, M., Acharya, U. R. (2020).** Classification of heart sound signals using a novel deep wavenet model. Computer Methods and Programs in Biomedicine, Vol. 196, pp. 105604.
14. **Rajagopalan, V., Cao, H. (2022).** Cardiovascular applications of artificial intelligence in research, diagnosis, and disease management. In Biomedical and Business Applications Using Artificial Neural Networks and Machine Learning. IGI Global, pp. 80–127.
15. **Randhawa, S. K., Singh, M. (2015).** Classification of heart sound signals using multi-modal features. Procedia Computer Science, Vol. 58, pp. 165–171.
16. **Sinha, S., Routh, P. S., Anno, P. D., Castagna, J. P. (2005).** Spectral decomposition of seismic data with continuous-wavelet transform. Geophysics, Vol. 70, No. 6, pp. P19–P25.
17. **Son, G. Y., Kwon, S. (2018).** Classification of heart sound signal using multiple features. Applied Sciences, Vol. 8, No. 12, pp. 2344.
18. **Upretee, P., Yüksel, M. E. (2019).** Accurate classification of heart sounds for disease diagnosis by a single time-varying spectral feature: Preliminary results. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT´19), IEEE, pp. 1–4.

Article received on 08/04/2022; accepted on 25/05/2022.
Corresponding author is René Luna-García.

ISSN 2007-9737

A New Fuzzy Vault based Biometric System Robust to Brute-Force Attack

Gina Gallegos-Garcia[1], Mariko Nakano-Miyatake[2], Alfonso Francisco De Abiega-L'Eglisse[2], Mario Rosas Otero[3], Vladimir Azpeitia Hernández[4]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Culhuacán,
Mexico

[3] Universidad Nacional Autónoma de México,
Facultad de Estudios Superiores Cuautitlán,
Mexico

[4] Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Mexico

{ggallegosg, mnakano}@ipn.mx, adeabiegal1900@alumno.ipn.mx,
malttz15@comunidad.unam.mx, vazpeitiah@gmail.com

Abstract. Fuzzy vault based biometric systems use fuzzy vaults during the coding that occurs within the enrollment stage inside a biometric system. In that stage the biometric system creates a vault from the biometric data, user's key and chaff points. In the verification stage, a new biometric data is introduced and user's key can be recovered through the use of the Lagrange polynomial interpolation method. As a consequence, in this kind of systems brute force attack would be successful because fuzzy vaults are finite. Most of the related works focus on designing secure fuzzy vault systems through the use of a password or using hybrid systems. This leads to security falling on the same user or increasing the number of security elements such as chaff points, the degree of the polynomial, or multiple biometric samples. This paper proposes a new system that considers cryptography to achieve a fuzzy vault biometric system robust against brute-force attacks. It is important to say that our system does not need a higher number of chaff points or even a higher polynomial degree. Obtained results show that this new fuzzy vault biometric system not only would be secure for current times but also would be for the future time.

Keywords. Authentication, fuzzy vault, biometric system, confidentiality, cryptography, encryption.

1 Introduction

Fuzzy vault based biometric systems use an algorithm for hiding a secret string S in such way that a user who has the biometric template T can easily recover S. The biometric template T can be fuzzy in the sense that the secret S is locked by some related, but not identical data T' [12]. However, this kind of systems are susceptible to various attacks such as attack against the vault with minutiae descriptors, false-accept attack, intermediate discussion, cross-matching or brute-force attack [6, 20]. Roughly speaking, a brute-force attack consists of an attacker submit

many passwords or passphrases with the hope of eventually guessing the one. The attacker systematically checks all possible passwords and passphrases until the correct one is found.

Related work shows that brute-force attack has attempted to be mitigated through the use of symmetric-key cryptographic schemes that base their security on the difficulty of solving mathematical problems with higher complexity such as the integer factorization problem, the discrete logarithm problem, or the elliptic-curve discrete logarithm problem [8, 17, 20].

Considering the aforementioned, this paper proposes a new fuzzy vault based biometric system that considers three cryptographic primitives in order to mitigate the brute-force in the following manner. The first primitive is a hash function. It is used to obtain a hash value from the original vault. The second primitive is a key encapsulation mechanism. This is used to agree a cryptographic secret key.

Then, in the enrollment stage, the binding data composed by the secret key and user's biometric data are encrypted with a symmetric key encryption. This last one as the third primitive we consider. Subsequently, in the verification stage the biometric template is compared with the new user sample, all this without the need to decrypt them.

Finally, to recover the cryptographic key, Lagrange polynomial interpolation method is executed. As a consequence, our biometric system is able to keep the minutiae values in a confidential way, even thought an attacker steals the templates values, without needing a higher number of chaff points or even a higher polynomial degree. The rest of the paper is organized as follows.

Section 2 shows the related work that describes the way to harden the fuzzy vault biometric system. Section 3 describes the brute-force attack, the fuzzy vault biometric system, and what is a brute-force attack over a fuzzy vault biometric system. Section 4 explicates the cryptographic considerations and describes the work inside this paper as well as the notation and the algorithm.

Section 5 reports all the data needed to made the experiment and shows all the steps in the experiment of brute-force attack over a fuzzy vault biometric system. Section 6 shows the results of the experiment of brute-force attack between no encrypted vault and encrypted vault and Section 7 shows the conclusion of this work.

2 Related Work

This section analyzes the papers related to biometric systems and their improvements in terms of security. These improvements within biometric systems are to prevent attacks such as brute force attack specifically in biometric systems based on fuzzy vaults.

In [10], the idea of the fuzzy vault for the retinal biometric template is presented through a multi-modal biometric fuzzy vault. It includes points from the retina and fingerprint in order to obtain a combined vault, which is hardened with user password for achieving high-level security. The security of the combined vault is measured using min-entropy.

The proposed password hardened multi biometric fuzzy vault is robust towards stored biometric template attacks. In [11], a brute force attack which improves upon the one described in [2] in an implementation of the vault for fingerprints is presented. On base of this attack, they show that the implementations of the fingerprint vault are vulnerable and cannot be avoided by mere parameter selection in the actual frame of the procedure.

They introduce the idea of the fuzzy vault based on information resources not used by the current version of the vault. In [13], a scheme for hardening a fingerprint minutiae-based fuzzy vault using password is proposed. Benefits of the proposed password-based hardening technique include template revocability, enhanced vault security and a reduction in the False Accept Rate of the system without significantly affecting the False Reject Rate.

Since the hardening scheme utilizes password only as an additional authentication factor (independent of the key used in the vault), the security provided by the fuzzy vault framework is not affected even when the password is compromised.

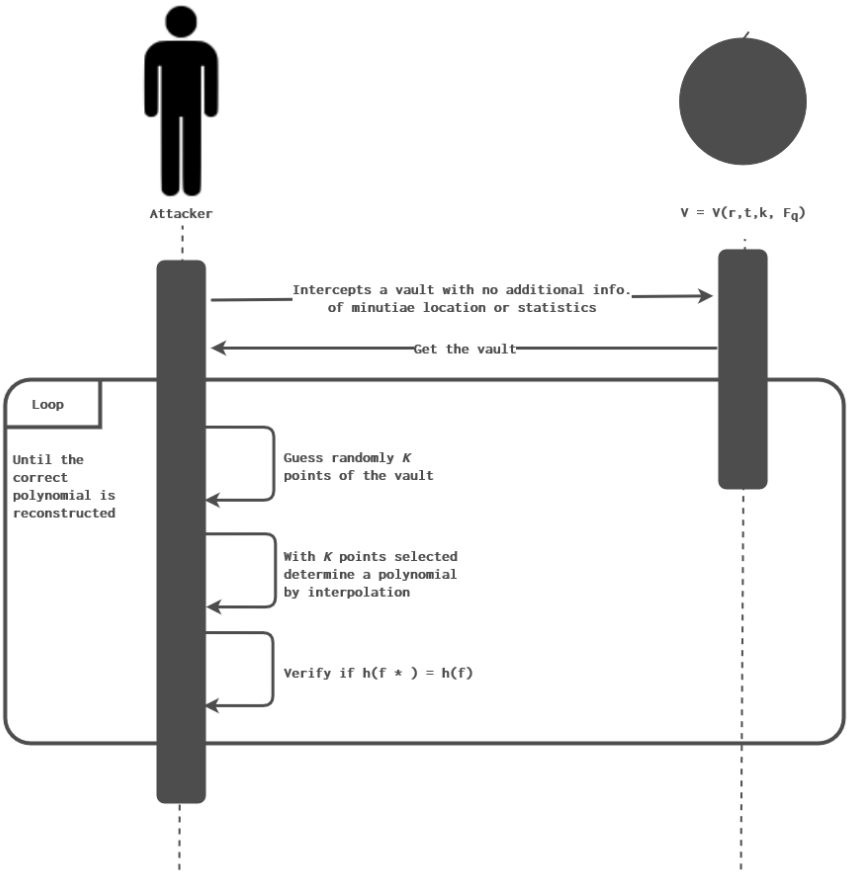


Fig. 1. Brute force attack process

In [16], a review of techniques like hybrid model and multimodal biometrics for security are proposed. They show how such techniques can be effective in enhancing the security of the system. In [18], some of the other known attacks against biometric fuzzy vault and biometric encryption techniques is reviewed.

They introduce three disturbing classes of attacks against Privacy Enhanced Technologies (PET) techniques including attack via record multiplicity, surreptitious key-inversion attack, and novel blended substitution attacks. In [22], a minutiae-based fuzzy vault implementation preventing an adversary from running attacks via record multiplicity is redesigned. Furthermore, they propose a mechanism for robust absolute fingerprint prealignment. Together, they obtain a fingerprint-based fuzzy vault that resists known record multiplicity attacks and that does not leak information about the protected fingerprints from auxiliary alignment data.

In [5], the vulnerabilities of the scheme in [13] is analyzed. After studying various schemes using special data like password a new scheme which is secure against various attacks to fuzzy vaults is proposed to enforce the security.

In [9], a new attack based on the alteration of original user data on fuzzy vault biometric cryptosystem is investigated. They assume that the attacker uses a modified version of the real user image to gain unauthorized access to the system (mobile phone).

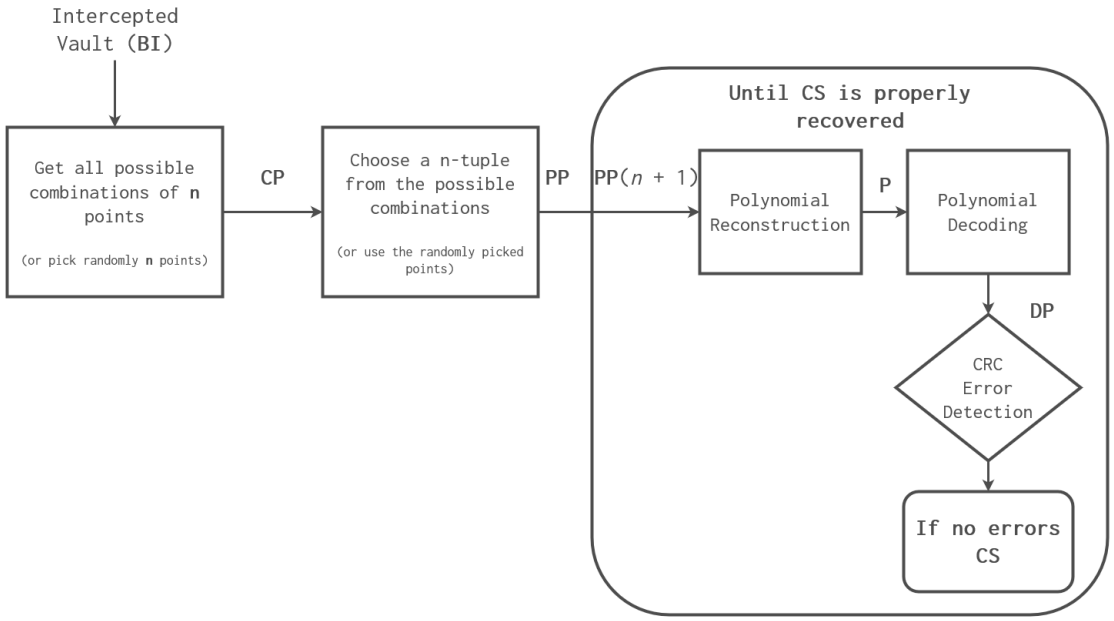


Fig. 2. Modular diagram of the brute force attack process

Experimental results carried out using fingerprint and face modalities show that this assumption has serious impact on the security of this type of biometric cryptosystem.

As we can see most of the related work focuses on designing secure fuzzy vault systems through the use of a password, using hybrid systems, or increasing the number of security elements such as chaff points, the degree of the polynomial, or multiple biometric samples.

In this work, we propose a new system that considers cryptography to achieve a fuzzy vault biometric system robust against brute-force attacks without needing a higher number of chaff points or even a higher polynomial degree.

3 Brute-Force Attack on Fuzzy Vault based Biometric Systems

The attack that we address in this work is brute-force attack. In this sense, on the one hand, we give the definition of the attack. On the other hand, we describe how a fuzzy vault biometric system works.

Finally, we describe the attack over such kind of system. The brute force attack is shown in the literature as viable to violate the fuzzy vault scheme based on fingerprints. Having the advantage that explicit knowledge of the operation of the scheme or of the implementation in the system to be attacked is not necessarily required.

A disadvantage compared to other attacks is that it has a high computational cost. One of the terms that must be taken into account is that to facilitate the verification of success at the time of implementation, it is assumed that the result of the coefficients of the polynomial or the secret is known. Fig. 1 shows the high level brute force attack process diagram, establishing the lines of each entities involved in the attack and their participation in its development.

Fig. 2 shows the modular diagram of the brute force attack where it identifies the transformation of the data and presents the comparison of the data identifying whether it is correct or not. A fuzzy vault based biometric system works over a field $\mathbb{F}$ of cardinality q and a universe $\mathbb{U}$. It assumes in the exposition that $\mathbb{U} = \mathbb{F}$.

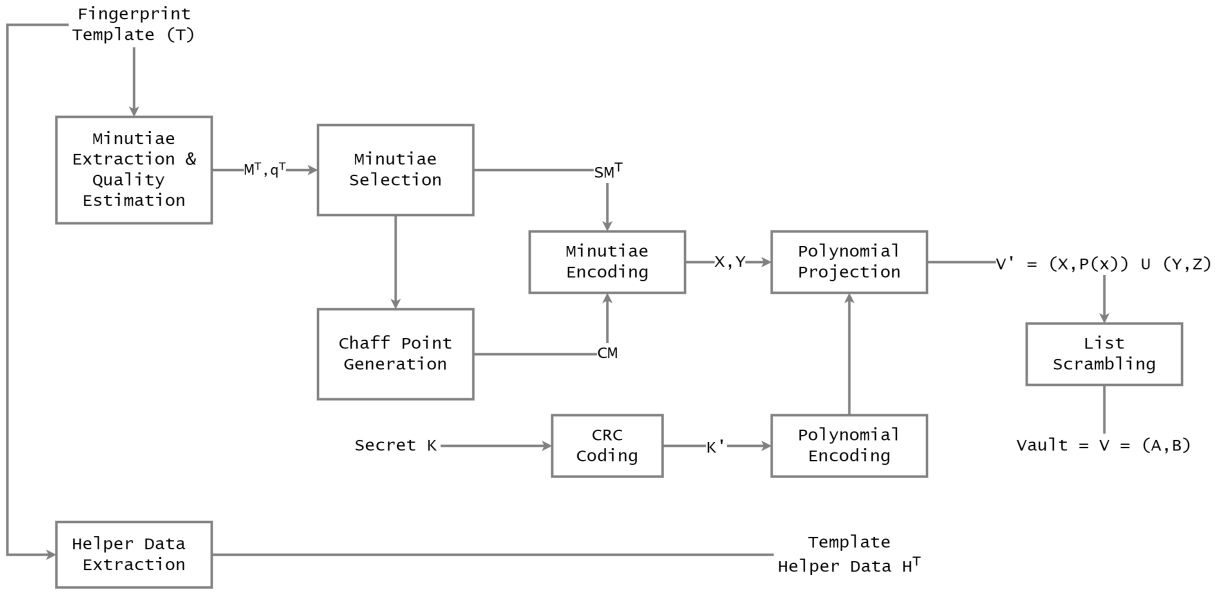


Fig. 3. Fuzzy vault scheme

The target in this kind of systems is to lock a secret value $\mathcal{K} \in \mathbb{F}^k$ under a secret set $\mathbb{A} \in \mathbb{U}^t = \mathbb{F}^t$, for protocol parameter $\mathcal{K}$ and t. Fig 3 shows the original fuzzy vault scheme. It considers a fuzzy vault encryption algorithm that takes, as input, a secret $\mathcal{K}$, a set $\mathcal{A}$ and outputs a vault $\mathcal{V}_\mathcal{A} \in \mathcal{F}^r$ for some security parameter r.

A corresponding decryption algorithm takes as input a vault $\mathcal{V}_\mathcal{A} \in \mathcal{F}^r$ and a decryption set $\mathcal{B} \in \mathcal{U}^t$. The output of the algorithm is a plain text value $\mathcal{K}' \in \mathcal{F}^k$ or null if the algorithm is unable to extract a plain text [12]. The brute-force attack process essentially consists of obtaining a fuzzy vault of size T that is of interest to be breached.

Then all the possible combinations of points existing in the vault are obtained in groups of n elements, where $0 \leq n \leq T$ and T is the total of points that the vault contains. It is also possible to choose n points at random.

If a range of values $[V_1, V_2]$ is known that could be the degree of the polynomial then the combinations are made in groups of $V_1 \leq n \leq V_2$ points of the vault. If the degree V of the polynomial is known then the combinations of points will be made in groups of $n = V$ points.

Subsequently, each of the combinations of points are passed through the Lagrange polynomial interpolation method, the result is obtained and it is verified if the secret obtained is equal to the original secret. If they are the same, the vault has been breached.

4 A New Fuzzy Vault based Biometric System

4.1 Cryptographic Considerations

Cryptographic Hash Functions. take a message as input and produce an output referred to as a hash code, hash-result, hash-value or simply hash. A hash function h maps bit-strings of arbitrary finite length to strings of fixed length, e.g. n bits. For a domain D and range R with $h : D \rightarrow R$ and $\mid D \mid > \mid R \mid$. Hash functions are one-way function, in other words, they are practically infeasible to invert [4].

Key Encapsulation Mechanism. are a class of encryption method designed to protect symmetric cryptographic key material from transmission using a public key scheme. In other words, KEM is a set of functions that can be used to obtain a symmetric encryption key from asymmetric keys.

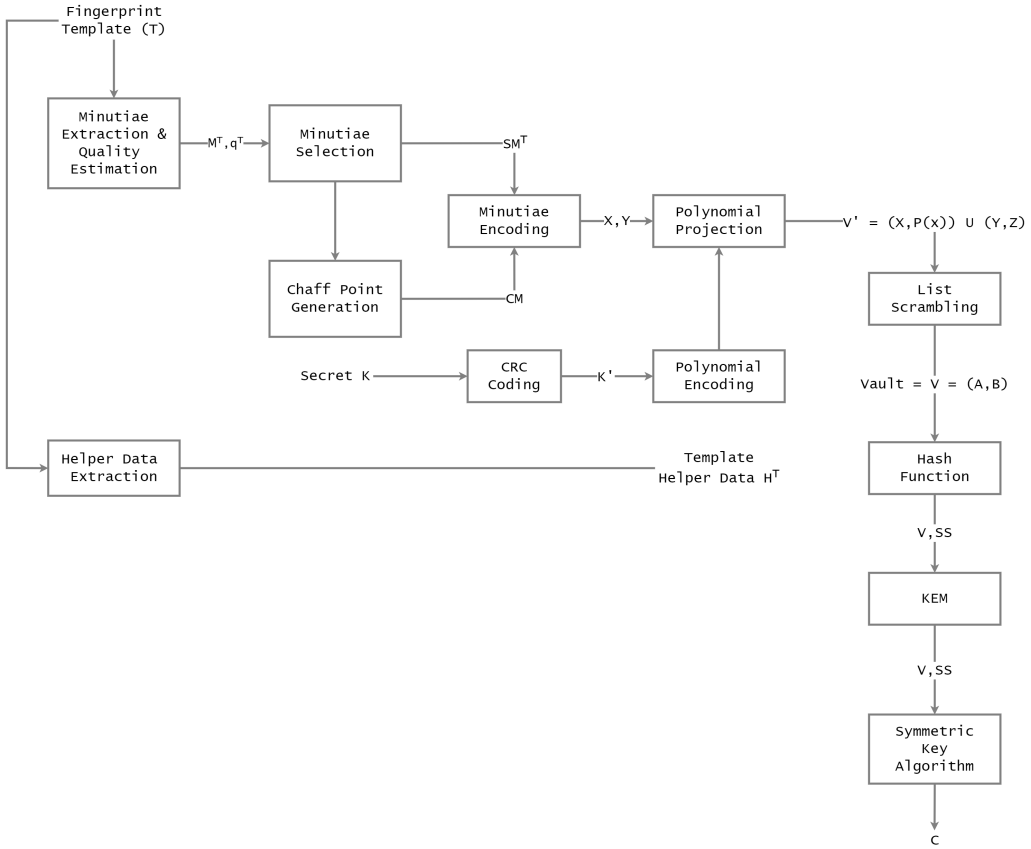


Fig. 4. Cryptographic considerations in our fuzzy vault based biometric system

Then, the symmetric key is used to encrypt the longer message. KEM simplifies the process by generating a random element in the finite group underlying the asymmetric scheme. Then, the symmetric key is derived by hashing it as a consequence the need for padding is eliminated.

KEM is composed of three algorithms named KeyGen, Encaps and Decaps. KeyGen generates a public key pk and a private key sk. Encaps returns a symmetric key K and a ciphertext ct. Decaps returns K [14].

Symmetric Key Algorithm. These are used in the most modern block ciphers. They often incorporate a sequence of permutation and substitution operations.

An iterated cipher is a commonly used design. It requires the specification of a round function, a key schedule and the encryption of a plain text will proceed trough Nr similar rounds [3].

4.2 Description

Our system considers encrypting template values with a symmetric key algorithm. The symmetric key is generated with a hash function of variable length. It is transmitted to the database with a key encapsulation mechanism. The template is protected by encrypting the data, so if it is stolen, a brute force attack can be avoided, since the template is not in plain text.

The entities that interact in our proposed solution during the enrollment are the biometric scanner that takes the fingerprint template, a personal computer used to capture user data and a server that storage all the user data captured. The entities

that interact during the verification are the biometric scanner that takes a new fingerprint and a server to verify the data.

Considering the approach presented in [19] and the entities aforementioned, as we can see in Fig. 4, we take as the input to our first algorithm, a data matrix denoted by $Vault = V = (A, B)$. After that its hash value is obtained with $SS = H(V)$, whose size is the exact length of the input of the key encapsulation mechanism. Once this hash value SS is obtained, it is necessary to transmit it to server. It is made by using the key encapsulation mechanism between the server and the personal computer as follows.

Firstly, the server generates a key pair $\mathsf{KeyGen}(\mathsf{Sk}, \mathsf{Pk})$ to start the encapsulation process. Then, the server sends the public key to the personal computer. Now, it can encapsulate the value SS called secret shared. Secondly, the personal computer encapsulates the secret shared SS by using the public key and sends it to the server. Finally, the server receives the information and recovers the secret shared SS with his private key and the decapsulation algorithm.

When the secret shared SS is obtained, the server stores it for its later usage. After that, the symmetric key algorithm is used with the secret shared, recovered by both entities, to encrypt the Vault. When it is done, the Vault is transmitted, from the personal computer to the server, to be storage in the database.

All of this is depicted in Fig. 5 and can be seen in Table 1. Considering that the decoding stage maintains in the same way with [19], with the difference that the $Vault$ is processed in the filter process.

It must be decrypted with the symmetric key algorithm $V = D_{SS'}(C)$ used in the coding stage and using the secret shared, previously saved, as the key to decrypt such $Vault$. All of this is depicted in Fig. 6 and Fig. 7.

4.3 Cryptographic Protocol in our New Fuzzy Vault Based Biometric System

In this section, we describe the cryptographic protocol defined for the encrypted fuzzy vault biometric system.

The notation of variables used in the protocol is described in Table 2.

Table 1. Cryptographic protocol definition

Cryptographic protocol definition for our new fuzzy vault based biometric system
1 : $M^T, q^T, H^T \leftarrow Ext(T)$
2 : $SM^T \leftarrow (M^T, q^T)$
3 : $CM \leftarrow Ch.P.Gen(u, v, \theta)$
4 : $(X, Y) \leftarrow M.Encod(CM, SM^T)$
4a: $P \leftarrow Polyencod(K')$
4b: $K' \leftarrow CodingCRC(K)$
5 : $V' \leftarrow Polyproyec(X, Y, P)$
6 : $V \leftarrow L.S(V')$
7 : $h \leftarrow H(V)$
8 : $Sk \leftarrow KEM(h)$
$C^V \leftarrow Enc(Sk, V)$

Table 2. Notation

Symbol	Definition
T	Fingerprint template
M^T	Minutiae extracted
q^T	Quality estimation
SM^T	Minutiae selection
CM	Chaff points
(X, Y)	Minutiae encoding
K'	CRC Coding
P	Polynomial encoding
$V'(X, P(x) \cup (Y, Z))$	Polynomial projection
$V = (A, B)$	Vault
E/D	Encryption/Decryption symmetric key algorithm

ISSN 2007-9737

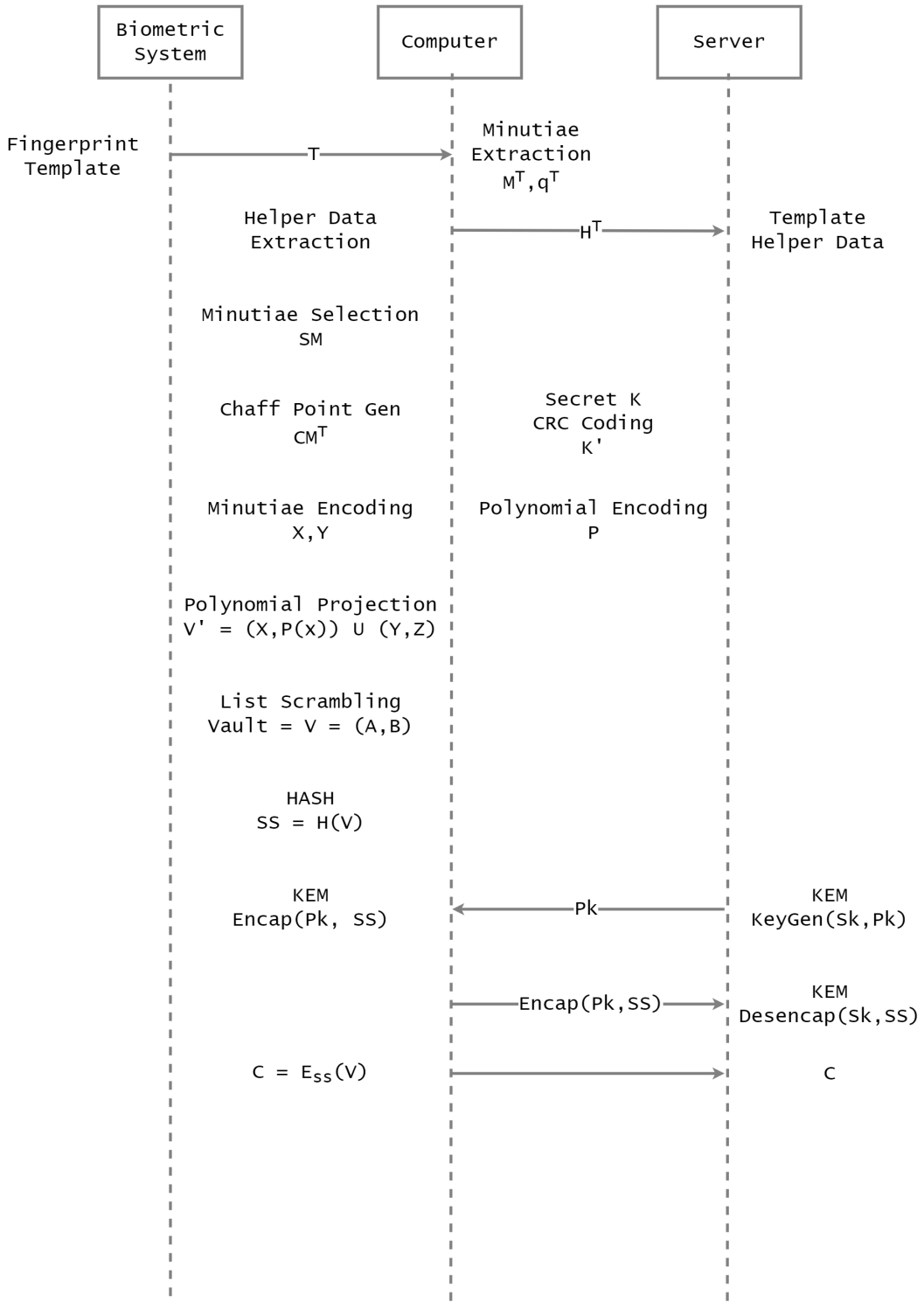


Fig. 5. Interaction between the three entities of our solution proposed

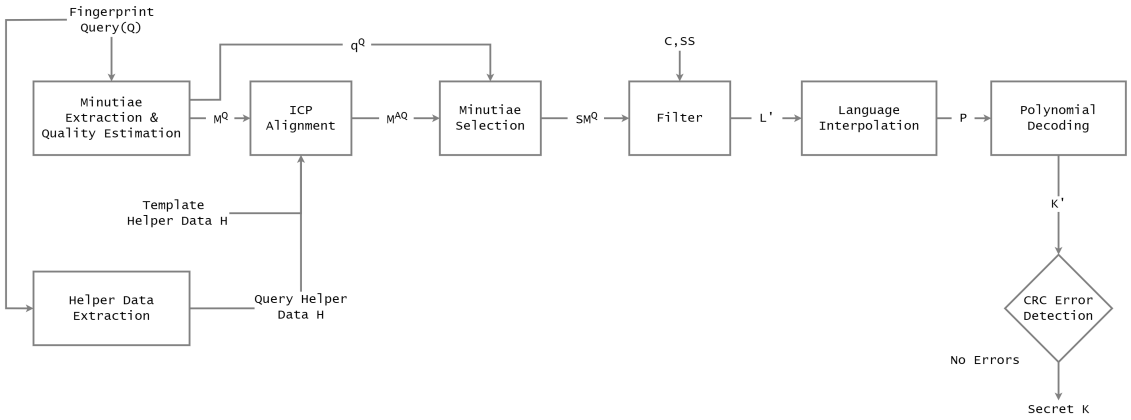


Fig. 6. Verification of fuzzy vault based biometric system considers outputs obtained from cryptographic considerations

5 Experimentation

5.1 Fingerprint Database Overview

For the experimentation, a database of 80 images of fingerprints in total was used. All the images contain different fingerprint impressions, such impressions belong to 10 different users. That is, there are eight different impressions for each user.

With each of the impressions, a different fuzzy vault is obtained. Three types of vaults were created for each fingerprint impression resulting in a total of three sets of 80 fuzzy vaults. Each of the vault types created are described below.

Free Size Vaults. These vaults are based on the quality of the minutiae, a quality filter is used at the time of extraction to select the best samples, resulting in the total set of minutiae being those with sufficient quality for use at the time of an authentication request later. This in turn results in each template having a fuzzy vault with a number of different genuine points and the overall size of the vaults would be randomly sized differently. All these vaults possess 50 chaff points and S genuine points, with S being the amount of high-quality minutiae found.

Standard Size Vaults. These vaults are created by sacrificing a little the quality of the minutiae in order to obtain a certain number of R in all the impressions thus achieving that all the vaults produced have the same size regardless of whether they come from different fingerprint impressions. All these vaults contain 50 chaff points and 23 genuine points.

Encrypted Vault. These vaults are created from standard sized vaults that have the same R elements. All These vaults contain 50 chaff points and 23 genuine points.

5.2 Brute-Force Attack Experimentation

Having the fuzzy vaults created from the available database it is possible to test them for vulnerability relatively easily. For the implementation of the attack, a uniform random distribution was used to apply the Lagrange polynomial interpolation method. It is assumed that the degree of the hidden polynomial is previously known in the experimentation.

The results provided in the Table 3 and Table 4 show the performance against iterations and the time required to successfully break the vaults. The criterion considered that a vault cannot be violated is when the correct polynomial is not found after a million iterations of polynomial reconstruction.

The vaults are named in the following way. Firstly the name of the vault, then the user number and finally the sample number of the user. As in example Vault101_1, Vault102_1,$\cdots$, Vault110_8.

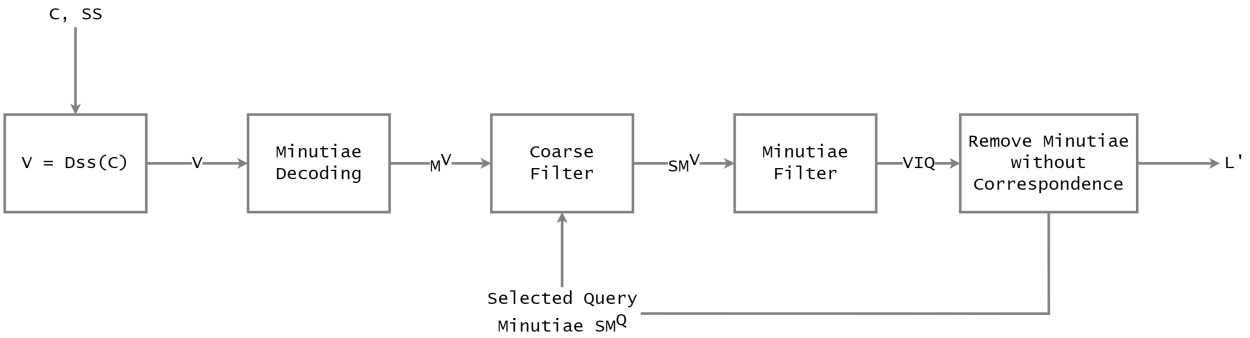


Fig. 7. The process of filtering during the verification stage should obtain the original vault

5.3 Cryptographic Schemes Used in the Experimentation

The cryptographic hash function used into the experimentation is named SHAKE-128 with an output size of 32 bits. SHAKE or SHA-3 extendable-output function (XOFs) is defined from the KECCAK[c] function by appending a four-bit suffix to M, for any output length d : SHAKE$128(M, d)$ = KECCAK$[256](M\|1111, d)$.

KECCAK is the family of sponge functions with the KECCAK $-$ p$[b, 12 + 2l]$ permutation as the underlying function and with $pad10^*1$ as the padding rule. The family is parameterized by any choices of the rate r and the capacity c such that $r + c$ is in $25, 50, 100, 200, 400, 800, 1600$ [4].

The key encapsulation mechanism used into the experiment to calculate the symmetric key is Kyber-1024. CRYSTALS - Kyber is an IND CCA2 secure key encapsulation mechanism (KEM), whose security is based on the hardness of solving the learning-with-errors (LWE) problem over module lattices. Kyber is one of the finalists in the NIST post-quantum cryptography project.

The submission lists three different parameter sets aiming at different security levels. Specifically, Kyber-512 aims at security roughly equivalent to AES-128, Kyber-768 aims at security roughly equivalent to AES-192, and Kyber-1024 aims at security roughly equivalent to AES-256 [1].

The symmetric key algorithm used into the experiments to encrypt the vault is AES-256-CBC. The Advanced Encryption Standard commonly called AES has a specification for the encryption of digital data established by the United States National Institute of Standards and Technology (NIST) [15]. It has a fixed block size of 128 bits and a key length of 128, 192 or 256 bits.

The relation between the number of rounds and the key length is as follows. There are 10 rounds for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys.

According to the specification, AES works on a 4x4 array of bytes, named the state and most of the operations are done in the finite field [3, 7]:

$$\mathbb{F}_{2^8} = \mathbb{Z}_2[x]/(x^8 + x^4 + x^3 + x + 1). \qquad (1)$$

6 Results

As we mentioned before, our new system considers Cryptographic Hash Functions, a Key Encapsulation Mechanism, and Symmetric Key Algorithm in order to be robust against brute-force attacks. Before including cryptographic schemes in our simulation, the attacker executes a brute-force attack on the vault in order to find the polynomial coefficients that contain the secret that was defined during the enrollment process of a user.

The probability of finding the correct combination that forms this polynomial is $1/n$ with n equal to the number of combinations. By encrypting the vault with the symmetric encryption scheme resistant to attacks from computers, robustness is added to the biometric system, since the security of this type of scheme lies on two parameters. The first one is the length of the symmetric key used to transform or encrypt the vaults of the biometric system.

Table 3. Iterations and time results per user to breach the free-size vaults

User	# Vaults	Min. Iterations	Max. Iterations	Min. Time	Max. Time
1	3	7,613	637,964	7.8s	573s
2	3	54,486	462,515	65.4s	545.4s
3	4	517	655,593	0.6s	718.8s
4	7	1,120	79,873	1.2s	85.2s
5	5	15,759	881,182	16.2s	909.6s
6	8	1,309	214,215	1.2s	238.2s
7	6	17,671	979,271	19.8s	1555.8s
8	8	901	134,253	5.4s	149.4s
9	3	89,734	427,978	100.2s	463.2s
10	5	4,029	484,881	7.2s	635.4s

The second one is the ciphertext obtained from the vaults. In other words, the patterns that the attacker can use to guess the polynomial coefficients are absent, as stated in [12]. The existence of symmetric encryption leads to the generation and establishment of symmetric key between two entities, the computer that captures the user's data and the server that stores it.

During this interaction there exist a possibility that the attacker wants to intercept the symmetric key. Our proposed solution incorporates a key encapsulation mechanism (KEM), in charge of agreeing the symmetric key between the computer and server.

KEM scheme uses a hash function, which takes as input parameters one of the vault coefficients of the biometric system, selected randomly. The function output feeds the KEM scheme to agree the symmetric key they use to encrypt the vault coefficients.

6.1 Results in Proposed Solution Against Brute-Force Attack with no Encrypted Vaults

6.1.1 Free Size Vault Experimentation

Table 3 shows the results of the number of iterations that were necessary to violate the fuzzy vaults. In the experimentation, not all the vaults were violated and the table shows the number that could have been violated. For all users, there are cases in which the number of iterations to violate the vaults was low, however, there are also cases in which the number of iterations was very high.

In the end, there were only two users to whom all the vaults were successfully breached. The same Table 3 shows the results measured in seconds needed to break the vaults are shown. The results of the vaults that were successfully breached averaged at least 25 seconds and averaged a maximum of 734 seconds.

In some vaults the polynomials could not be calculated by means of Lagrange polynomial interpolation method because they did not contain enough genuine points to calculate the polynomial reconstruction.

The reason is because the minutiae quality filter did not allow for enough genuine points. Vaults with fewer genuine points than necessary were Vault101_4, Vault101_5, Vault102_5, Vault102_8, Vault107_8, Vault109_6.

Some vaults that needed more than a million iterations in the polynomial reconstruction to be able to find the coefficients of the correct polynomial are Vault101_2, Vault101_3, Vault101_8, Vault102_3, Vault102_4, Vault102_7, Vault103_2, Vault103_3, Vault103_4, Vault103_8, Vault104_5, Vault105_4, Vault105_6, Vault105_8, Vault107_3, Vault109_1, Vault109_3, Vault109_4, Vault109_8, Vault110_1, Vault110_2, Vault110_4.

Table 4. Iterations and time results per user to breach the standard-size vaults

User	# Vaults	Min. Iterations	Max. Iterations	Min. Time	Max. Time
1	8	488	208,412	1.2s	346.2s
2	8	8,128	83,129	16.8s	159s
3	8	9,403	617,940	9s	885.6s
4	8	8,123	211,829	7.8s	288s
5	8	80,589	218,181	123s	294.6s
6	8	710	60,441	1.2s	71.4s
7	8	26,040	482,439	30s	596.4s
8	8	25,476	253,338	24.6s	277.8s
9	8	214	406,164	0.18s	632.7s
10	8	1,558	242,562	1.8s	211.8s

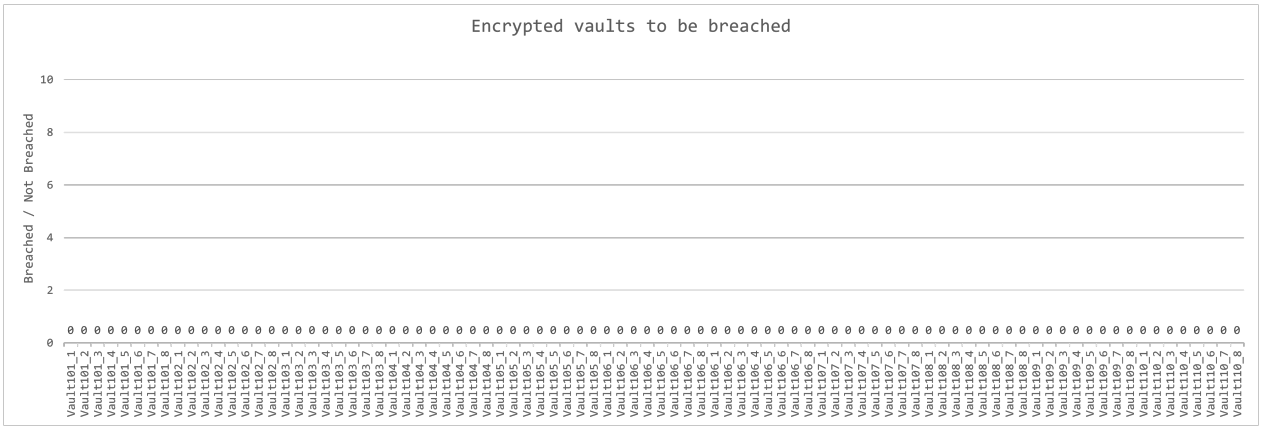


Fig. 8. Encrypted vaults breached

6.1.2 Standard Size Experimentation

Table 4 shows the results of the number of iterations that were necessary to breach the vaults. During the development of the experiment, 100% of the vaults used could be successfully breached and 65% of the vaults could be breached with less than 100,000 iterations.

The same Table 4 shows the results measured in seconds of the time required to breach the vaults.

During the experiment, most of the vaults were breached in less than 300 seconds, that is, the attack shows low times to be a brute force attack considering that all the vaults were breached.

6.2 Results in Proposed Solution Against Brute-Force Attack with Encrypted Vaults

Since the viability and performance of the brute force attack on the fuzzy vault scheme has been verified in this work, a test was carried out with the same vaults but that were passed through an encryption function, specifically using the symmetric key algorithm.

Encrypted vaults, instead of being a set of points, are a string of coded characters. Because of that, it is not feasible to apply a polynomial reconstruction directly on that character string. The alternative to show a comparison is to transform the encoded character string to a set of points that are contained in the finite field bounded by the vault values.

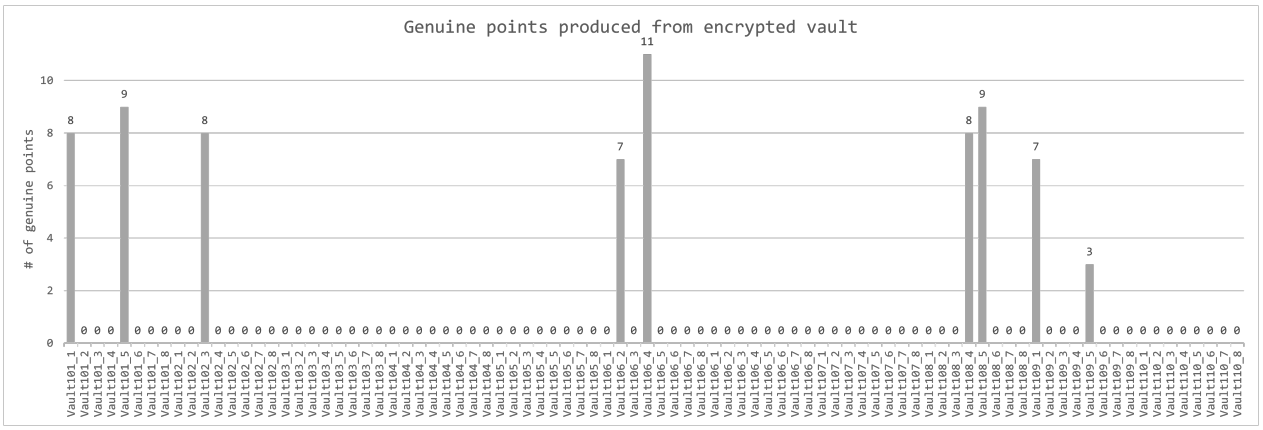


Fig. 9. Genuine point results recovered on all vaults from their encrypted versions

To achieve this, the following process was done. The encrypted vault is read as a binary file. The values obtained are transformed to their equivalent in the decimal system at values between 0 and 255. Later, these values are normalized and re-scaled to values between 0 and 2^{16}. Finally, it is verified that with the transformation made to the set of points, sufficient points have been obtained, this is done by comparing the points obtained in the encrypted vaults with the genuine points of the unencrypted vaults.

The number of points that must be obtained are at least $(k + 1) \times 2$, where k is the degree of the hidden polynomial inside the vault, since both the dependent and independent values that intervene in the polynomial are needed to be able to rebuild it.

In case of finding the number of points necessary to carry out the polynomial reconstruction, the correct order of the data pairs must still be known to be able to interpolate successfully, which increases complexity and computational cost.

In addition, it was necessary to take into account that when transforming the encrypted vault to numerical values, the amount of values obtained depends on the length of the chain that represents the encrypted vault, which for our experiments more than 2000 values are obtained.

This increases the complexity of finding the correct set of coefficients out of such a large number of possible values. So the attacker must try all the combinations of points and prove the polynomial reconstruction since he did not know the genuine points. Fig. 9 shows all the vaults where genuine points could be recovered.

These points were recovered from the encrypted vaults. However, none of them obtained a sufficient number of genuine points for the polynomial reconstruction, since having a polynomial of degree 8 requires at least 18 genuine points recovered to attempt the reconstruction. Given the results shown in the previous graph and due to no vault's enough genuine points were found to even attempt a polynomial reconstruction, that is $(k+1)\times 2$ points, there is no encrypted vault at risk of being compromised as shown in Fig. 8.

Even considering that in a real attack, the attacker would not have a quick way to verify that he is recovering genuine points.

7 Conclusion and Future Work

Fuzzy vault based biometric systems use fuzzy vaults within the enrollment stage inside a biometric system. These systems are susceptible to multiple security attacks. In this paper, we propose a new system that considers cryptography to achieve a fuzzy vault based biometric system robust against brute-force attacks. Our solution was designed based on the effects generated by this kind of attack due to fuzzy vaults are finite.

Most of the related works focus on designing secure fuzzy vault systems through the use of

a password or using hybrid systems. The main difference between our new fuzzy vault biometric system and related work is that we do not need a higher number of chaff points or even a higher polynomial degree. This leads to security falling on the same user or increasing the number of security elements such as chaff points, the degree of the polynomial, or multiple biometric samples.

Obtained results show that an important piece of information is that to recover many more polynomials, it is necessary to have a standard vault size since when there was a free-sized vault, the recovery of the polynomial was lower.

Moreover, we were able to verify that when the vault was encrypted the brute force attack was not successful in recovering the polynomial and therefore the security of this system could not be violated. The test in this paper was made with 256 security bits, as a consequence, this new fuzzy vault biometric system not only would be secure for current times but also would be for the future.

In future work, it is necessary to test other types of attacks such as a correlation attack or an attack through the multiplicity of records to continue testing the security of the proposed solution. In this way, it could be demonstrated that this solution proposal can be effective for protection against multiple attacks.

Acknowledgments

The authors thank the Instituto Politecnico Nacional and the Consejo Nacional de Ciencia y Tecnologia. The research for this paper was financially supported by SIP-IPN 20221427 and CONACYT 321068.

References

1. **Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., Schwabe, P., Seiler, G., Stehlé, D. (2021).** CRYSTALS-Kyber algorithm specifications and supporting documentation. NIST PQCRYPT round 3 submission, pp. 43.
2. **Clancy, T. C., Kiyavash, N., Lin, D. J. (2003).** Secure smartcard-based fingerprint authentication. WBMA '03: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, pp. 42–52.
3. **FIPS-197 (2001).** Advanced encription standard (AES). Last updated October 05, 2021.
4. **FIPS-202 (2015).** SHA-3 standard: Permutation-based hash and extendable-output functions. Last updated November 11, 2020.
5. **Hong, S., Jeon, W., Kim, S., Won, D., Park, C. (2008).** The vulnerabilities analysis of fuzzy vault using password. FGCN '08: Proc of the 2008 Second Int Conf on Future Generation Communication and Networking, pp. 76–83.
6. **Jain, R., Kant, C. (2015).** Attacks on biometric systems: An overview. International Journal of Advances in Scientific Research, Vol. 1, pp. 283.
7. **Katz, J., Lindell, Y. (2020).** Introduction to Modern Cryptography. CRC Press.
8. **Kholmatov, A., Yanikoglu, B. (2008).** Realization of correlation attack against the fuzzy vault scheme. Proceedings of SPIE, Vol. 6819, pp. 7.
9. **Lafkih, M., Lacharme, P., Rosenberger, C., Mikram, M., Ghouzali, S., Haziti, M., Aboutajdine, D. (2015).** Vulnerabilities of fuzzy vault schemes using biometric data with traces. 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 822–827.
10. **Meenakshi, V., Ganapathi, P. (2009).** Security analysis of password hardened multimodal biometric fuzzy vault. World Academy of Science, Engineering and Technology, Vol. 32, pp. 312–320.
11. **Mihailescu, P., Munk, A., Tams, B. (2009).** The fuzzy vault for fingerprints is vulnerable to brute force attack. BIOSIG 2009 - Proceedings of the Special Interest Group on Biometrics and Electronic Signatures, pp. 43–54.
12. **Nandakumar, K., Jain, A., Pankanti, S. (2008).** Fingerprint-based fuzzy vault: Implementation and performance. Information Forensics and Security, IEEE Transactions on, Vol. 2, pp. 744–757.
13. **Nandakumar, K., Nagar, A., Jain, A. (2007).** Hardening fingerprint fuzzy vault using password. ICB 2007: Advances in Biometrics, pp. 927–937.

14. NIST (2009). SP 800-56b - Recommendation for pair-wise key establishment schemes using integer factorization cryptography. Last updated March, 2019.

15. NIST (2017). Post-quantum cryptography standardization. Last updated December 02, 2021.

16. Panwar, A., Singla, P., Kaur, M. (2018). Techniques for enhancing the security of fuzzy vault: A review. Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, pp. 205–213.

17. Rathgeb, C., Wagner, J., Tams, B., Busch, C. (2015). Preventing the cross-matching attack in bloom filter-based. 3rd International Workshop on Biometrics and Forensics, IWBF 2015.

18. Scheirer, W., Boult, T. (2007). Cracking fuzzy vaults and biometric encryption. Biomet Symp, pp. 1–6.

19. Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. Proceedings of 35th Annual Symposium on Foundations of Computer Science, pp. 124–134.

20. Tams, B. (2013). Attacks and countermeasures in fingerprint based biometric cryptosystems. PhD. Thesis, pp. 32.

21. Tams, B. (2013). Cryptanalysis of the Fuzzy Vault for Fingerprints: Vulnerabilities and Countermeasures. Ph.D. thesis, Georg-August-Universität Göttingen, 37073 Göttingen.

22. Tams, B., Mihailescu, P., Munk, A. (2015). Security considerations in minutiae-based fuzzy vaults. IEEE Transactions on Information Forensics and Security, Vol. 10, No. 5, pp. 985–998.

23. Uludag, U., Pankanti, S., Jain, A. (2005). Fuzzy vault for fingerprints. volume 3546, pp. 310–319.

Article received on 03/02/2022; accepted on 25/05/2022.
Corresponding author is Gina Gallegos-Garcia.

ISSN 2007-9737

Natural Language Semantic Answering Applied to Medicinal Plant and Coronavirus

Alma Delia Cuevas-Rasgado[1], Maricela Claudia Bravo-Contreras[2], Franz Ludwig Lake-Moctezuma[3], Adolfo Guzmán-Arenas[3]

[1] Universidad Autónoma del Estado de México, Computer Engineering, Estado de México, Mexico

[2] Universidad Autónoma Metropolitana, Systems Department, Ciudad de México, Mexico

[3] Instituto Politécnico Nacional, Centro de Investigación en Computación, Ciudad de México, Mexico

adcuevasr@uaemex.mx, mcbc@azc.uam.mx, lakemoctezuma@gmail.com, aguzman@ieee.org

Abstract. A question answering system that receives as input a question in Spanish and returns the answer is presented. Preguntas y Respuestas {questions and answers} (PryRe) has two main components: 1) An information retrieval component that identifies the meaning of the question using its semantic properties. This component transforms the question into a triplet: R (C, V), where R is the relation or link, C is the concept or main idea, and V is the value of the concept. Example: ¿Cuál es la hierba que mejora la digestión? {What is the herb that improves digestion?} becomes R(C, V) = mejora (hierba, digestión) {improves(herb, digestion)}. This component uses natural language processing modules; 2) a component that uses the triplet to carry out a query analysis on PryRe's ontology, to identify the answer, which in the example is Manzanilla {Chamomile}. This component performs the semantic identification of the question while traveling on parts of the ontology. Details of the PryRe system are given, as well as tests on herbalism and Coronavirus. It shows an acceptable accuracy (82%). Resources used in this work are (A) a notation used to describe ontologies, and (B) the deductive capability of PryRe.

Keywords. Semantic analysis, ontology, question-answering, knowledge retrieval, natural language processing.

1 Introduction

PryRe is a question-answering system that integrates Natural Language Processing (NLP), ontologies, and Information Retrieval (IR) methods. It interprets a given question in Spanish and finds an answer using as a source an ontology of the interest domain.

People who at some time need to ask specific questions about a knowledge domain for support inspired it.

The coronavirus pandemic was its target. Its deductive functionality PryRe (Sections 5.1 to 5.5) usually provides correct, nontrivial answers (82% accuracy). PryRe can be used with other ontologies.

Information is important in the Technology Age, and one way to obtain it is by questions in natural language.

Nevertheless, people usually do not express themselves rigorously when communicating, because the receiver is another person who provides the lacking information by using context and common sense. However, this does not apply with programs and computers. If the receiver is a

computer, then all the details should be explicitly described compare "John is a blonde engineer" with "The color of the hair of John is blond and his profession is engineer."

For this reason, ontologies, which are interpretations for computers, must have explicit and well-defined conceptualizations.

NLP research, an area of Artificial Intelligence, handles the automation of communication between human beings and "artificial agents." Parts of the NLP repertoire used in this paper are:

- Tagging (nouns, verbs, articles, adverbs, and named entities) to identify the elements of a sentence.
- Tools, such as WordNet [1], set of tags for probabilistic tagging based on the Expert Advisory Group on Language Engineering Standards (EAGLES)[1] group. Since WordNet does not provide disambiguation resources for prepositions in Spanish, a previously developed solution [2] for disambiguation of prepositions in Spanish is used.
- Ontologies [3], a type of knowledge base mainly developed in Knowledge Representation in Artificial -Intelligence, which represents a concept (usually described by a word or a set of words) according to its meaning. PryRe uses the Ontology Merging notation [4], a formal notation based on a local logic foundation that facilitates the representation of ontology concepts. Since it uses an XML-based syntax, ontology engineers find it easy to understand and use.
- Freeling [5], an open-source language analysis tool suite. It provides effective use of the dependency tree, easing the translation from Spanish phrases into triplets.

To evaluate its functionality and efficiency, PryRe uses the following:

- Coronavirus;
- Health information concerning the nervous system was used for experimentation. Specifically, the use of the Galphimia Glauca plant (growing in Mexico) was selected;
- Nutritional and medicinal aspects of fruits, vegetables, herbs, and seeds are incorporated, as well as their compositions: minerals, essential amino acids, vitamins, and other useful elements.

All examples are in Spanish, followed by their English translation in brackets {}, manually introduced.

The main contributions of this work are:

- The way in which PryRe splits the natural language question into portions and transforms these into a triplet R(C, V), and its deductive capabilities, embodied in the search that PryRe performs on the ontology to provide the correct answer.

The organization of the paper is as follows. Section 2 presents related work and a comparison analysis of question-answering approaches. The methodology is explained in Section 3, while Section 4 contains a description about OM notation; the PryRe system is detailed in Section 5. Section 6 deals with an evaluation that employs documents about medicinal plants and Coronavirus. Finally, Section 7 contains the results, conclusions, and future works.

2 Related Work

Question-Answering systems have become very popular since the evolution of semantic models based on ontologies and Knowledge Graphs (KG). This section presents a review of related work, and a comparative analysis of solution approaches (see Table 1).

Qanda [6] is a question-answering system that integrates IR, Knowledge Representation (KR) and NLP methods into a hybrid algorithm whose input is a question in natural language. Qanda converts its input into a logical representation, a proposition enhanced by inferencing hierarchical relations of the variables involved. In these propositions, each variable is replaced by a keyword from the question. A small set of the most relevant documents containing the keywords are retrieved, ranked by its relevance. The closest answer (the

[1] https://www.cs.upc.edu/~nlp/tools/parole-sp.html

most common answer, according to the search) is returned. The main characteristic of Qanda is its domain independence, since it does not use knowledge bases. The disadvantage is the time consumed, because the approach requires the preprocessing, indexing, and ranking of the collection of documents to execute queries.

AQUA [7] is a question-answering system that integrates NLP, Logic, Ontologies, and IR methods. Its process model has four phases:

– User interaction. The user introduces a question in natural language and obtains a list of ranked answers.
– Question processing, which executes a NLP parser (an interpreter that uses the unification and resolution algorithms to find a logical proof of the query). It uses the AQUA lexical resource and the Ontology. The Question processing phase has a failure analysis system, and a module to classify and reformulate questions.
– Document processing, that extracts a set of paragraphs from a collection of documents by identifying the focus of the question.
– Answer processing, which produces answers by extracting passages from the documents, groups, and scores answers.

AQUA uses the AKT reference ontology to provide more information about each question. AQUA's main features are the translation of English questions into a logical form of Query Logic Language (QLL), and enhanced answers using the AKT ontology. The disadvantage is the translation of English questions into the QLL form.

A multi-agent question-answering system [8] uses students to support learners in collaborative environments. A set of agents forward the question to students. Their responses are analyzed using extracted documents from course materials and other resources. This analysis verifies and ranks the responses. All questions and answers are stored, verified, and reused. The main features of this work are:

– It is an agent-based system; it supports collaborative learning, and it uses answers verification. Its disadvantage is the lack of a logic-based formal language for query processing and knowledge representation, which prevents the use of reasoners to obtain more information.

A Machine Learning (ML) question-answering system [9] is trained with a set of question-answers pairs from the WebQuestions dataset. It generates answers using Freebase, a general fact knowledge base represented by N-triples in Resource Description File (RDF). The system learns low-dimensional vector embeddings of words that appear in questions and produces a joint embedding space where questions and answers are close. In doing so, it takes advantage of the graph-based representation of knowledge to obtain answer paths and subgraph representations. Its main features are: a vector space model to represent pairs of questions and answers and the use of graph knowledge representation to obtain more information. Its disadvantages are: the Freebase API has been shut down, and its training depends on the use of a benchmark of paired questions and answers.

Aqqu [10] is a question-answering system that uses the Freebase API. It identifies all entities from the knowledge base that matches part of the question. Then, based on tree templates, it generates a set of SPARQL Protocol and RDF Query Language (SPARQL) query candidates, from which the answers are obtained. Its main feature is its translation of a natural language question to a SPARQL query. Its disadvantages are: the Freebase API has been shut down, and Aqqu depends on the use of a benchmark of paired questions and answers for training.

Convolutional Neural Networks (CNN) are used in [11], a question-answerer that does inference on the knowledge base (entity linking and relation extraction), and then refines the answer. For relation extraction, it relies on a Multi-Channel Convolutional Neural Network (MCCNN), which uses one channel for syntactic, and the other for sentential information.

For answer refinement, it uses Freebase to retrieve candidate answers, and validates answers using Wikipedia. Its main feature is the use of a MCCNN. Its disadvantages are: the Freebase API has been shut down, and its training depends on the use of a benchmark of paired questions and answers.

Table 1. Comparison of related work

Question-Answering Approach	Dataset and additional resources	Knowledge Base Representation Language
Hybrid algorithm based on IR, KR and NLP [6].	Text REtrieval Conference (TREC) Question Answering collection.	Formal logic representation of the query.
NLP, Logic, Ontologies, and Information Retrieval methods [7].	Documents retrieved from the Web by using the search engine Google. AKT reference ontology.	Query Logic Language (QLL).
Multi-agent system that coordinates the reception and forwarding of questions. It uses IR and NLP [8].	Course materials, a collection of Frequently Asked Questions (FAQ) and the learners' answers.	There is not a formal language representation of questions and answers.
Learning vector embeddings of words appearing in questions and answers [9].	WebQuestions based on Freebase, using 5,810 question answer pairs.	N-triples RDF
Template matching and relation matching [10].	Freebase 917 and WebQuestions.	RDF and SPARQL
Learning approach based on MCCNN. Use of dependency tree patterns to decompose questions [11].	Freebase and Wikipedia to validate the results.	RDF and SPARQL
Template-based over a knowledge base using a learning approach [12].	ClueWeb09-FACC1, a corpus of 500M Web pages annotated with Freebase	RDF and SPARQL
Template-based over the SNOMED KG [13].	Uses SNOMED KG. WordNet for word similarity.	SNOMED query templates

QUINT [12] is a question-answering approach that implements a learning approach to automatically generate question-query templates and manage complex questions.

For experimentation, the authors used Freebase and WebQuestions benchmarks. The general process is divided into two: template generation and question-answering.

The main difference with previous efforts (and its main feature) is the fully automatic generation of question templates that link a question to a triplet pattern query over a KG.

A question-answering system for SNOMED medical ontology [13] uses a question-template library based on ontology definitions. Its main feature is the use of a template matching inference method combined with semantic similarity.

Considering the different solution approaches analyzed, it can be observed that some of them strongly depend on the use of pre-configured pairs of question-answering benchmarks to train a learning model. Additionally, the question-answer benchmarks use a knowledge base that may no longer be publicly available. Two questions arise: is it feasible to offer an open knowledge base about any topic? Is it possible to build question-answer approaches that do not require a training stage using previously validated benchmarks?

Template-based approaches have shown good results, some based on previously defined template libraries and others, more advanced, generate question templates using ML methods.

PryRe represents knowledge using KG, identifies and generates templates using graphs, generates answers by identifying and mapping question types, provides a friendly NL interface, translates the NL query into a logic query (a triplet), answers that query using a simple inference engine that exploits the laws of equivalence of inference logic.

3 Methodology

In this section, the research and development methodology consisting of five stages described in Figure 1.

Descriptions about medicinal plants and Coronavirus: First, data from plants related to the central nervous system was obtained [14].

Manual analysis: From them, the concepts, implicit and explicit relationship, classifications, and partitions are identified, to obtain an idea about how to organize the ontology.

Building the ontology: To have an ontology that contains information about a specific domain, relevant aspects must be decided. First, it is necessary to determine the formal language in which the ontology will be implemented.

In this area, there are several methods to represent ontologies such as Protegè [15]. A complete study of this area is presented in [16], where an analysis of different recent tools to visualize ontologies is given, as well as its main features. For the representation of the ontology, the OM notation is used. For the manual design, a graphic ontology builder, OM Edit [17] is presented, a graphical editing tool through which an ontology can be implemented without having to deal with the strict syntax of the language, it automatically generates code in OM notation [17]. This language is supported with tools to edit and verify the conceptualizations that will be included. It was decided to use the OM notation**,** which has a formal basis based on local logic.

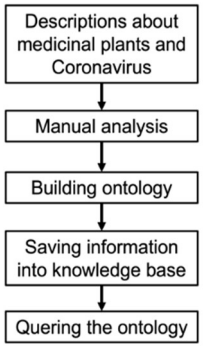


Fig. 1. PryRe method

Saving the information in the knowledge base: There are two ways to save the ontology: 1) graphically, creating a JPG file, or 2) saving the OM notation in an ONT file (a text file).

Querying the ontology: Using the ontology in OM notation, allows different types of searches to be performed, considering several scenarios and domains. A tag-based language (XML) was used to implement OM notation. Thus, the query algorithms using tree representations and making use of the semantic relationships described in the language formalization were built.

In addition, a method of analysis and translation of the input questions made in natural language into a representation in triplets R (C, V) was developed.

This process is simple because the question is typed in natural language. For example, *¿Cuál es la relación entre Organismo y Planta Vascular?* {*What is the relationship between Organism and Vascular plant?*}. **PryRe** converts the main objective of the search into a triplet R(C, V). R represents the name of the relationship or link, C is the concept where the relationship starts, and V is

the concept where the relationship finishes. In this example, PryRe returns the triplet: subset (Planta vascular, organism), R=subset, C=Planta vascular and V=organism. The **PryRe** answer is: *Vascular plant es un subconjunto de Organismo* {*Vascular plant is a subset of Organism*}.

At the end, a whole set of test cases and scenarios was designed to evaluate both the efficiency of the algorithm and the responses obtained.

4 Ontology Construction

This section describes OM notation, the language selected for the implementation of the ontologies. First, the formal foundations of OM notation are described, and then its implementation is given.

4.1 Description of the OM Notation

The Ontology Merge (OM) notation was introduced in 2010 by [4] because of [18]. The purpose of the OM notation is to design ontologies with concepts and relationships described in a high degree of detail. These descriptions help the question-answering process since the OM notation can represent nested concepts, synonyms, implicit and explicit relationships (a relationship is a link that connects the concepts among themselves).

Some of types of relationships provided by OM notation are explained below (these appear between <> symbols, for example: <subset>, <word>) and shown in Figures 11 to 14:

1 **Implicit relations**. They represent the hierarchical structure of the ontology. The concepts in an ontology are classified into sets and subsets, parts of, members of and type of. For instance, the concept; *Girasol* {*Sunflower*} is a subset *(<subset>)* of *Flor* {*Flower*} but it is possible that *Girasol* {*Sunflower*} is a type *(<type>)* of *Flor* {*Flower*}. Referring to linguistics, the *subset* is known as Hyponym. In the same order *Pistilo* {*Pistil*} is a part of the *Girasol* {*Sunflower*}. The *Pistilo* {*Pistil*} is known as Meronym and *Girasol* {*Sunflower*} is known as its Holonym.

2 **Synonyms Concepts**. Many concepts have synonyms. In reference to [19], the OM notation represents synonyms with the tag <word>.

3 **Explicit Relations**. These types of relations increase the meaning of the concepts. For example, it is known that Pistil is a part of Girasol {Sunflower} but to know what tamaño {size}, grosor {thickness}, and textura {texture} the Pistil has, the <relation> tag is used.

4 **Definition**. It is the set of words that describes the concept, taken from dictionaries and expressed in natural language, its tag is *<gloss>*.

4.2 OM Notation and the Ontology Languages

OM notation, like Ontology Web Language (OWL)[2] are structured languages, using tags as in XML. These structures allow their interpretation by computers, as well as by people.

OM notation supports:

1 Conjunctions of classes and cardinality (arity) relations.

2 Simplicity in categorizing (object taxonomies) and relationships.

3 Greater expressiveness using <relation>, <word> and <gloss> that define the concept in detail, where the relationship can be both a concept and a relation. They express its meaning, including logical properties such as transitivity, symmetry, and inheritance, as OWL and Description Logic (DL).

4 Linking several ontologies at the same time, merging them (in a binary way). This is the degree of freedom that OM notation provides but guaranteeing the consistency of ontologies.

[2] http://w3.org/OWL/

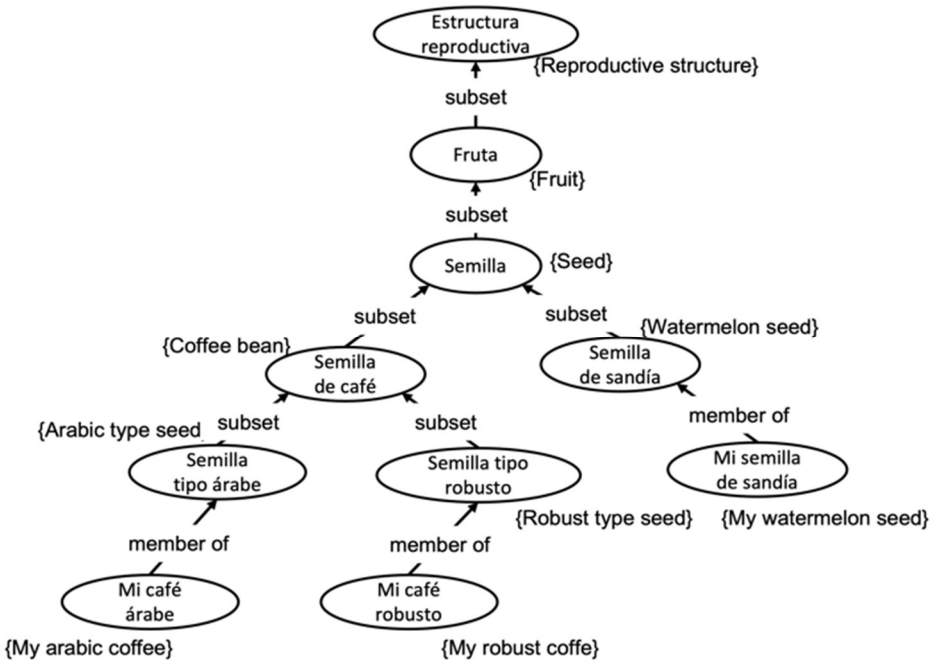


Fig. 2. In the ontology some instances are Mi Café Arabe {My Arabic coffee}, Mi café Robusto {My Robust coffee} and Mi semilla de Sandía {My Watermelon seed} and some types are Semilla Tipo Árabe {Arabic type seed}, Semilla tipo Robusto {Robust type seed]. The link member of Mi café Robusto {My Robust coffee} member of Semilla tipo Robusto {Robust type seed} means that the first is an instance, object or example of the second

In OWL 2 the OWL ontology languages have been improved with more expressiveness, extensions on data types and annotation capacity, simple metamodeling, among others. The OM notation has those capabilities, too, except metamodeling, and is simple to use.

The names of the concepts in this article are their usual names. However, they can be changed by URL, or memory addresses. All concepts have one address, as in OWL. For example, [20] shows the large number of nodes that the Facebook Ontology can have. Google, Instagram, and Facebook find relationships between users and events. Their challenge is to take advantage of the data found in its pages and combine them with other more structured databases to strengthen its knowledge. For this they use URL addresses. Obviously, the idea is to have control of links and a large amount of structured data. Facebook has the resources to store 50 million entities (primary nodes), Microsoft 2 billion entities, Google 1 billion, eBay around 100 million, IBM more than 100 million entities.

Our idea is to build small ontology fragments of descriptions of an event and a user. It is small because it is formed by a set of concepts that are born from a theme, broken down into triplets (relationship, object, value) and merge these small ontologies until obtaining a large common ontology as in [18].

The work of OM notation is to represent useful information, describing the concepts found. For instance, the information that a biography, or descriptive or technical documents contain. It permits merging such information locally and uses it. It is not intended to represent figurative language: poetry, stories, or political speeches. OM notation uses a set of labels that have a specific function. There is no graphical user interface like Protégé [15], that allows us to

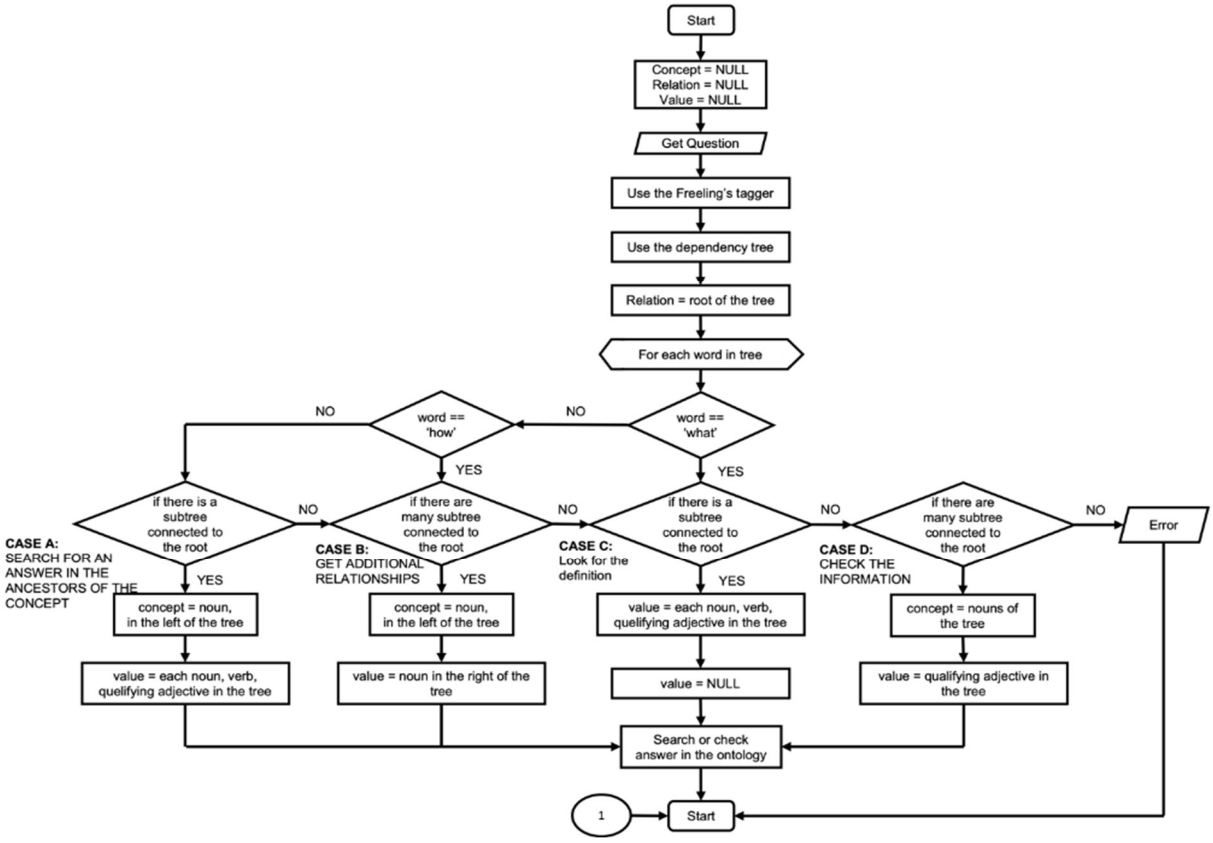


Fig. 3. The *PryRe* system (first part), showing how it creates the triplet in cases A to D

visualize the structure of the ontology, restrictions, and classifications of each class or instance.

All answers of *PryRe* are concrete responses, except when the query includes a concept definition taken from dictionaries.

5 Design of PryRe

The general purpose of *PryRe* is to interpret a question in natural language and generate a response. Figure 3 shows in detail the control flow of the *PryRe* question interpreter. *PryRe* first receives a question in natural language, then performs a syntactic and semantic analysis of the question, labels the grammatical elements of the question using a PoS tagger, and integrates a dependency tree with the elements of the question.

Those elements are identified as Concept, Relationship, and Value to obtain a triplet representation of the question. Once the triplet is obtained, PryRe identifies the type of question and obtains a response according to the following cases:

a. Search for an answer in the ancestors of the concept.

b. Obtain additional relationships.

c. Look for the definition.

d. Check the information.

e. Search relation and value in all the ontology.

The second part of *PryRe* is shown in Figure 4: finding the possible answer (case E) and looking for the triplet in the ontology. The answer is produced with two precisions: low or high, as explained below.

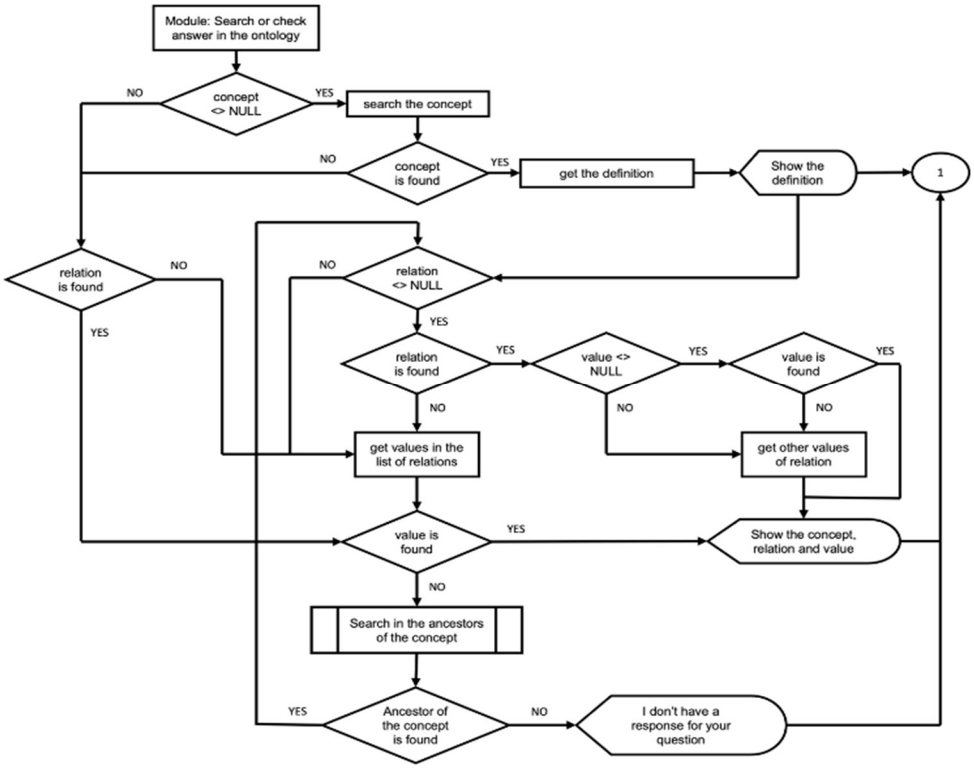


Fig. 4. The PryRe system, second part. Case E: Search for relation and value in all the ontology. This part of PryRe the case E is applied if the answer is not found inside of ancestors of the Concept. The recursion of the algorithm happens when PryRe needs to look for the same triplet in the whole ontology and applies the whole process to whatever it found related to this triplet. The feedback going from the "YES" exit of the bottom diamond to diamond "relation <> null" in this figure, and the box "Search in the ancestors of the concept", show the recursion of PryRe

5.1 Case A. Search for an Answer in the Ancestors of the Concept

This case occurs when searching for the answer using the inheritance relationships of the concept. For this, three subcases are considered:

a. Subcase A.1: when the concept is found in the ontology, but the relationship or the value is not found. For instance, the triplet *forma* (*Galphimia Glauca*, *arbusto verde*) {*form* (*Galphimia Glauca*, *green bush*)}, has low possibilities of precision when searching among its ancestors. In this example, the relationship *form* and *the green bush* is searched. If *Plant* is an ancestor of *Galphimia Glauca*, it may have a relationship: form, since *Galphimia* is a *bush* and a *bush* is a plant, therefore the triplet *form* (*Plant*, *green bush*) represents a closer response, semantically. The accuracy of this example is: C = 1, R = 0, V = 0, it is equal to: (1+ 0 + 0) / 3 = 0.3, it would change to C = 0.5 because it was found in an ancestor. The ancestors are semantically more general than the concept, R = 1, V = 1 it is equal to: (0.5 + 1 + 1)/3 = 0.8. The result ranges from 0 to 1, the worst is 0 and the best is 1.

b. Subcase A.2: when the concept is found in the ontology, but the relationships are not found in the concept, and it is not among the synonyms of the relationship (subcase E.1). In this case, *PryRe* searches for the most common relationship of the concept. For example, consider the triplet: forma(*Galphimia Glauca*, *arbusto verde*) {*form* (*Galphimia Glauca*, *green bush*)} whose most common relationship is:

orden(*Galphimia Glauca*, *Polygalales*) {*order* (*Galphimia Glauca*, *Polygalales*)}. The result of this example is low: C = 1, R = 0, V = 0 therefore: (1.0 + 0 + 0) /3 = 0.3.

c. Subcase A.3: when the relationship was not found in any of the ancestors of C. Therefore, the answer is: *No tengo respuesta para su pregunta* {*I have no answer for your question*}. Result: C = 0, R = 0, V = 0 has: (0 + 0 + 0)/3 = 0.0.

5.2 Case B. Get Additional Relationship

a. Case B.1: when *PryRe* searches for the answer using the implicit relationships *subset* and *type* of the concept. Recalling that the *subset* is known as Hyponym, while *type* is the Hyperonym. For example, consider the question *¿Qué animales salvajes portan coronavirus?* {*What wild animals carry coronaviruses?}.* The resulting dependency tree of this question is shown in Figure 5, which is represented with the triplet: *portan (animales salvajes, coronavirus)* {*carry (wild animals, coronaviruses*)}. For this question *PryRe* will search for the concept: *animales salvajes* {*wild animals*}, for the relation: *portan* {carry} and for the value: *coronavirus*. Then *PryRe* will search using the implicit relationships *subset* and *type* of the concept: *wild animals*. The resulting relationships are *subset (animales salvajes, murciélago)* {*subset (wild animals, bat*)}, and *type (animales salvajes, rata)* {*type (wild animals, rat*)}. Therefore, the answer for this question will be: *murciélago, rata* {bat, rat}. That is, all *subset* and *type* relationships that are contained in the concept: *animales salvajes* {*wild animals*} will be displayed. The result of this example is: C = 1, R = 1, V = 1 therefore: (1 + 1 + 1) /3 = 1.

b. Case B.2: This case happens when *PryRe* searches for the answer using the explicit *relation* and *partition* of the concept. For example, consider the question: *¿Cuál es la forma y la estructura del Coronavirus?* {*What is the shape and structure of Coronavirus?}* The question word is: *Cuál* {*What*}, the root is: *ser* {*is*}*,* the nouns connected (relation label) with the conjunction are: *forma* {*form*} and *estructura* {*structure*}. *PryRe* will search in the ontology for the *Coronavirus* concept and the relation between *forma* {*form*} and *estructura* {*structure*} (they are explicit relations in the ontology). Then, it will obtain the values of these two relationships. Figure 6 shows the code (A) and dependency tree (B) of the question. The triplet is: *forma*(*Coronavirus*, null) {form(Coronavirus, null)}. The precision is: C=1, R=1, V=1 therefore: (1 + 1 + 1)/3 = 1. For form, its precision is: C=1, R=1, V=1 although *part* was found instead of *structure* (as synonym), rendering a precision of 1, and the overall result will be (1 + 1)/2= 1.

5.3 Case C. Look for the Definition

This case occurs when *PryRe* searches for the definition of a given concept. Given a triplet, if the concept is found in the ontology, *PryRe* searches in all the relationships of the concept using the following keywords: *significa* {*it means*} (and its different lemmas) *es* {*it is*}, *tener {to have}, entender {to understand*), using the question word: *Cuál* {*What*} and the value: null.

For instance, consider the following questions:

a. *¿Qué significa Galphimia Glauca?* {*What does Galphimia Glauca mean?}* The triplet representation that corresponds to this question is: *significa (Galphimia Glauca, null) {means (Galphimia Glauca, null*)}.

b. *¿Cuál es el significado de Galphimia Glauca ?* {*What is the meaning of Galphimia Glauca?}* The triplet representation that corresponds to this question is: *significa (Galphimia Glauca, null)* {*meaning has (Galphimia Glauca, null)}.*

c. *¿Qué se entiende por Gauphimia Glauca*? {*What is meant by Gauphimia Glauca?}.* The triplet representation that corresponds to this question is: *entendido por (Gauphimia Glauca, null)* {*understand by (Gauphimia Glauca, null*)}.

In the three cases above, the relationship indicates the *type* for search. Another way to answer the same questions is by obtaining the definition in the <*gloss*> tag. The answer for the questions is: *Arbusto o planta medicinal mexicana para el sistema nervioso central. También llamado en el Bajío mexicano como: "ojo de gallina"* {*Shrub*

or Mexican medicinal plant for the central nervous system. Also known in the Mexican Central Area as: "eye of the hen"}. The accuracy of this example is: C = 1, R = 1, V = 0, (1 + 1 + 0) / 3 = 0.6. However, the question word is a key to obtain the precise answer. In this case, it is *Qué* {*What*} and complements the word in the relationship *significa* {*means*}. The correct answer is found in the definition. Therefore, the value is V = 1. The result of this example is: C = 1, R = 1, V = 1, (1 + 1 + 1) / 3 = 1.

5.4 Case D. Check the Information

This case occurs when *PryRe* searches for the information verifying that the triplet is found. There are two possibilities:

a. Subcase D.1. If the concept, relationship, and value are found (which represents the best case), only the question needs to be confirmed. For instance, consider the question *¿Es Galphimia Glauca un arbusto verde?* {*Is Galphimia Glauca a green shrub?*} The triplet corresponding to this question is *forma* (*Gauphimia Glauca*, *arbusto verde*) {*shape (Gauphimia Glauca, green bush*)}. The result of this example is C = 1, R = 1, V = 1. Then accuracy = (1 + 1 + 1) / 3 = 1.

b. Subcase D.2. If the concept and its relationship are found, but the value is not. For instance, consider the question *¿Cuál es la forma de la Galphimia Glauca?* {*What is the shape of Galphimia Glauca?},* Triplet corresponding to this question: *forma (Gauphimia Glauca, null)* {*shape (Galphimia Glauca, null)}*. In this example, unlike subcase A.1, the *forma* {*shape*} relation is not obtained from the definition, or it would be difficult to find it since not all the definitions cite the shapes of the defined objects. Therefore, the value of V is null, and the value of V will be found in the set of values of the shape relation. The answer is *arbusto verde* {green bush}. The triplet representation of this question is: *forma(Gauphimia Glauca, arbusto verde)* {*shape*(*Gauphimia Glauca*, *green bush*)}. The result of this example is: C = 1, R = 1, V = 0, (1 + 1 + 0) / 3 = 0.6.

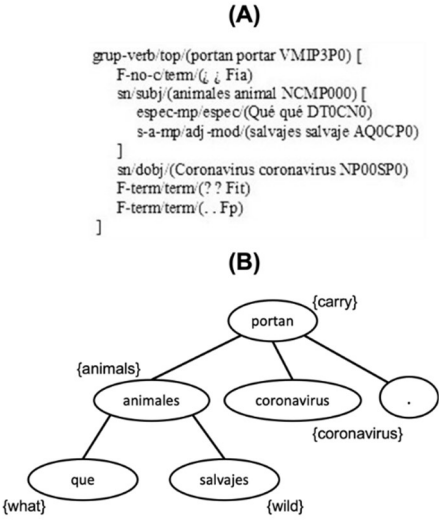


Fig. 5. Dependency tree of the question *¿Qué animales salvajes portan coronavirus?* {*What wild animals carry coronavirus?*}. The dependency tree is represented in a code (A). This code is explained in [5], but for simplicity, the graphical mode is used (B)

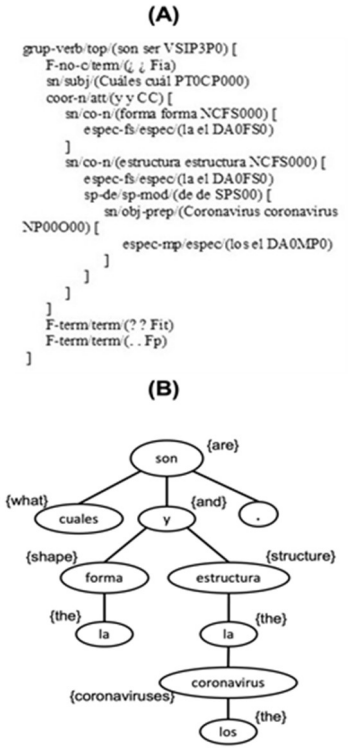


Fig. 6. Dependency tree of the question: *What is the shape and structure of Coronavirus*?

5.5 Case E. Search Relation and Value in All the Ontology

This case occurs when *PryRe* searches for the relation and value of a concept. The following subcases can occur:

a. Subcase E.1. If the concept is found in the ontology, the relation is not found, and the value is found, PryRe searches for a relationship that is synonymous with the missing relationship. For example, consider the triplet: *forma (hoja, redonda)* {*shape (leaf, round)}* a synonym is: *forma (hoja, circular)* {*form (leaf, circular)}.* Synonyms can be found in the <*word*> tag only if the relationship is also a concept (it is one of the advantages of OM notation). For this case, the result is: C = 1, R = 1, V = 1, (1 + 1 + 1) / 3 = 1.

b. Subcase E.2. Based on the previous case, if a synonym is not found, the value is searched in all the relations of the concept, but it can be erroneous. For example, *forma (hoja, redonda)* {*form (leaf, round)}* and *forma (tallo, redondo)* {*steam (leaf, round*)} This solution is ambiguous since the shape of the bush is round, but the stem of the bush is also round. Therefore, the result is: C = 1, R = 0, V = 1, (1 + 0 + 1) / 3 = 0.6.

c. Subcase E.3. If the concept is not found in the ontology, but the relation and value are found in the ontology, PryRe will find a concept in the ontology that satisfies the relation and the value of the same triplet. However, there is a risk to find a more general or semantically different concept. For example, consider the triplet: forma (Galphimia Glauca, arbusto verde) {shape (Galphimia Glauca, green bush)}. If Galphimia Glauca is not found in the ontology, the triplet may be misinterpreted as: la planta es verde {plant is green} (too general) forma (planta, arbusto verde) {shape (plant, green bush)}. The precision is: C = 0, R = 1, V = 1, (0 + 1 + 1)/3 = 0.6. In this case, although the answer has been found, there is not a guarantee that the result will be semantically close to the correct answer. The result would be: 0.6.

6 Evaluation

To evaluate PryRe, this section describes experiments from diverse domains: medicinal plants, and Coronavirus. For each of the tests, the question, the tagging of the question, the dependency tree, the integration of the triplet, the search for the triplet in the ontology, and the correct answer are shown.

6.1 Test 1 Meaning

¿Que significa Citrus Aurantium? {*What does Citrus Aurantium mean?*}, after Freeling's labeling, Table 2 shows the tags assigned.
Figure 7 shows the dependency tree generated.

The next step is to choose nouns, verbs, and adjectives. Only one sub-tree remains (Figure 8).

Triplet Integration

For each sub-tree of the dependency tree, the root is taken as the relation, that is, the link to be searched in the ontology. The first noun is taken as the main concept: Citrus Aurantium, since there are no more nouns, the value is null. Therefore, the triplet is as follows:

Significa (Citrus Aurantium, null)
{*means (Citrus Aurantium, null*)}

Now *PryRe* will look for the answer in the ontology.
There are two ways to find it:

a) In the ontology, the relationship *significa* {*means*} of the concept *Citrus Aurantium* points to the value *Naranjo amargo* {*Bitter orange*} because the relationship *significa* {*means or meaning*}, would be found (the lemma is taken from the word), then this value is connected to the relation, see Figure 9 (A). PryRe would apply case B.
b) Another option is: that the value of V is found in its gloss or description. In this case, only the value of the <gloss> tag is obtained in the Citrus Aurantium concept, see Figure 9 (B). In this case, PryRe would apply case D.

6.2 Test 2 Explanation

¿Cómo se conocen los Espasmolíticos?

(A)

```
grup-verb/top/(significa significar VMIP3S0) [
    F-no-c/term/(¿ ¿ Fia)
    sn/subj/(Qué qué PT0CN000)
    sn/dobj/(Citrus_Aurantium citrus_aurantium NP00SP0)
    F-term/term/(? ? Fit)
    F-term/term/(. . Fp)
]
```

(B)

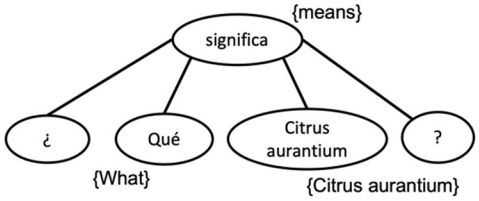


Fig. 7. Dependency tree of the question: *¿Qué significa Citrus Aurantium?* {What does Citrus Aurantium mean?}

Table 2. Words and its tags accord to EAGLES

Word	Tag
¿	Fia
Qué {*What*}	PT0CN000
Significa {*mean*}	VMIP3S0
Citrus_ Aurantium {*Citrus Aurantium*}	NP00SP0
?	Fit

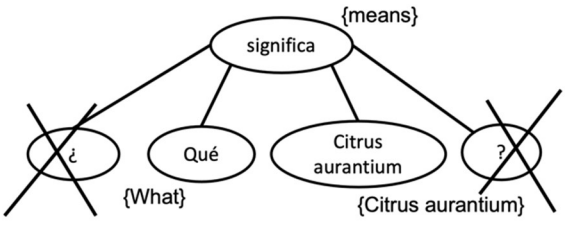


Fig. 8. Question marks are removed from the tree

{*How are Spasmolytics known?*}
The labels for each word are shown in table 3. The dependency tree of the question is as follows:

```
grup-verb/top/(conocen conocer VMIP3P0) [
    F-no-c/term/(¿ ¿ Fia)
    sadv/cc/(Cómo cómo PT000000)
    morfema-verbal/es/(se se P00CN000)
    sn/dobj/(Espasmolíticos espasmolíticos NP00O00) [
        espec-mp/espec/(los el DA0MP0)
    ]
    F-term/term/(? ? Fit)
    F-term/term/(. . Fp)
]
```

For now, no more details of the graphical representation of the dependency tree will be given. The triplet that corresponds to the question is *conocido como (Espasmoliticos,* null*)* {*known as* (*Spasmolytic, null)},* as seen in Figure 10.

In ontology there are two ways to find the answer: (A) with the link: *conocido como* {*known as*} and (B) with the link: *mejor conocido como* {*best known as*}. Both answers are valid. PryRe has applied the Case B (section 5.2).

6.3 Other Ways to Find Answer in the Ontology

Several examples of inferences and queries using Medicinal Plants are explained below. Details about the inference process can be found in [21].

Test 3: implicit relationship: *hoja suave* {*soft leaf}* is a part of *Galphimia Glauca*, the set of *Galphimia Glauca* has a part that is *hoja suave* {soft leaf} that is also type of *hoja* {*leaf*}. See Figure 11.

Test 4: Redundancy correction in triplet: If *Galphimia Glauca* is a subset of *planta medicinal* {*medicinal plant*} and this is a subset of *planta vascular* {*vascular plant*} then *Galphimia Glauca* is a subset of *planta vascular* {*vascular plant*}. It is not necessary to set a relation between *Galphimia Glauca* and *Planta vascular* {*vascular plant*} because shifting through the trajectories: by the route of *planta medicinal* {*medicinal plant*}, *Planta vascular* {*vascular plant*} ancestor is arrived at (see Figure 12).

Test 5: Explicit relationship in triplet: *Galphimia Glauca* is shaped like *Arbusto verde* {*Green Bush*} and blooms in *tiempos lluviosos* {*rainy times*}. The explicit relationship gives more semantics to the concepts. (See Figure 13).

Test 6: Synonyms: *Galphimia Glauca* has the following synonyms enclosed in parentheses

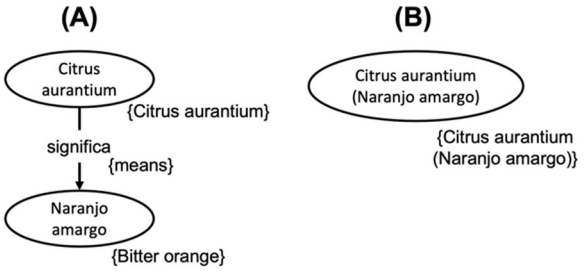


Fig. 9. The Concept Citrus Aurantium and its definition in (A) are found (means points to bitter orange), and in (B), this is in the gloss (Naranjo amargo {bitter orange}) inside of Citrus aurantium

Table 3. Words and its tags, accord to EAGLES [5]

Word	Tag
Cómo {*How*}	PT000000
Se {*are*}	P00CN000
Conocen {*known*}	VMIP3P0
Los {the}	DA0MP0
Espasmolíticos {*Spasmolytics*}	NP00000
?	Fit
.	Fp

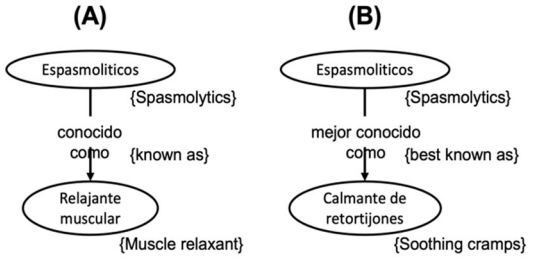


Fig. 10. Question marks are removed from the tree

(*Thyallis glauca*, *Galphimia gracilis*, *Galphimia humboldtiana*, *Galphimia multicaulis*) (see Figure 14).

Test 7: Requested Information is not part of the response in the ontology. In the question: *¿Las ratas de bambú portan coronavirus?* {*Do bamboo rats carry coronavirus?}* PryRe says No, Ibecause Bamboo rat is not part of *animal salvaje* {*wild animal*} whose *carries* Coronavirus. *Rata de bambú* {*bamboo rat*} is subset of *rata* {rat} that is a subset of Animal (above of *animal salvaje* {*wild animal*}) in Figure 15. This *rata de bambú* {*bamboo rat*} does not appear in the *animal salvaje* {*wild animal*} concept, for this reason, the answer is: *No*, although by common sense it is true.

6.4 Examples Related to Coronavirus

In the early days of the outbreak, China quickly shared its understanding of the Coronavirus with the world through the World Health Organization. (WHO)[3]. Tested and tempered by the viral epidemic such as the SARS epidemic, professionals, and experts in the first line focused in the "epicenter" Wuhan, China. They decided to share their invaluable experiences and lessons from the current outbreak as well as during their internships and experiences in China and various countries making it possible to edit The Coronavirus Prevention Handbook [22].

Pryre has been tested with 100 questions about coronavirus obtained from this book [22]. Only 16 questions are shown in this section, all of them appear here[4].

1 **Question:** *¿Qué son los virus asociados con las infecciones respiratorias? {What are viruses associated with respiratory infections?}* Case applied: C.

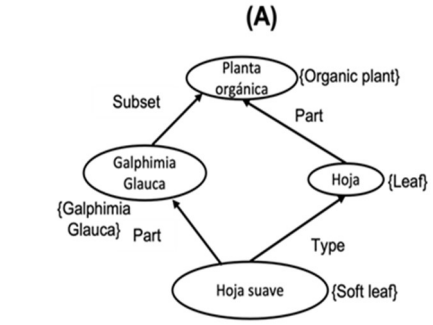
Book answer: "Viruses associated with respiratory infections" refer to viruses that invade and proliferate in the epithelial cells of the respiratory tract that can cause respiratory and systemic symptoms.

PryRe triplet: *significa (Virus asociados con infecciones respiratorias, null)* {*means (viruses associated with respiratory infections, null)}*.

PryRe answer: L*os virus que invaden y proliferan en las células epiteliales de las vías respiratorias que pueden causar síntomas*

[3] World Health Organization, www.who.int/

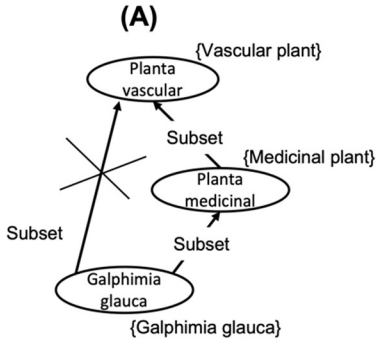
[4] https://tinyurl.com/4pk7r2sj



(B)

```
<concept> Planta orgánica
   <Language> Spanish <word> Planta Orgánica
</word></Language>
   <concept> Galphimia glauca
      <Language> Spanish <word> Galphimia glauca
</word></Language>
      <subset> Planta orgánica </subset>
  </concept>
   <concept> Hoja
      <Language> Spanish <word> Hoja </word></Language>
      <part> Planta orgánica </part>
         <concept> Hoja suave
            <Language> Spanish <word> Hojuela
blanda</word></Language>
            <type> Hoja </type>
            <part> Galphimia glauca </part>
         </concept>
   </concept>
</concept>
```

Fig. 11. (A) shows a part of the ontology (predecessor of Galphimia Glauca) and (B) shows the code of this part in OM notation



(B)

```
<concept> Planta vascular
   <Language> Spanish <word> Planta vascular </word></Language>
   <subset> Planta </subset>
    <concept> Planta medicinal
      <Language> Spanish <word> Planta medicinal
</word></Language>
       <subset> Planta vascular </subset>
       <concept> Galphimia Glauca
          <Language> Spanish <word> Galphimia Glauca
</word></Language>
          <subset> Planta medicinal </subset>
          <subset> Planta vascular </subset>
       </concept>
   </concept>
</concept>
```

Fig. 12. The part of ontology in (A) has redundancy in the relation *subset (Galphimia Glauca, Planta vascular)* {*subset (Galphimia Glauca, Vascular plant*)}. Although the relation appears in the code (B), it is eliminated in the ontology in memory. PryRe eliminates redundant relations.

respiratorios y sistémicos {Viruses that invade and proliferate in the epithelial cells of the respiratory tract that can cause respiratory and systemic symptoms}.

2 **Question:** *¿Cuáles son los virus comunes asociados con las infecciones respiratorias?* {*What are the common viruses associated with respiratory infections?}*

Case applied: B.

Book answer: Influenza virus, syncytial virus respiratory, measles virus, mumps virus, virus Hendra, Nipah virus, rubella virus, rhinovirus, SARS coronavirus.

PryRe triplet: *type (virus comunes asociados con las infecciones respiratorias, null)* {*type (common viruses associated with respiratory infections, null*)}.

PryRe answer: *Virus de la influenza, virus sincicial respiratorio, virus del sarampión, virus de la parotiditis, virus Hendra, virus Nipah y metapneumovirus humano,virus de la rubéola, la familia Picornaviridae (rhinovirus), y la familia Coronaviridae (coronavirus del SARS).*

3 **Question:** *¿Qué es el síndrome respiratorio de Oriente Medio (MERS)?* {*What is Middle East Respiratory Syndrome (MERS)?}*

Case applied: C.

Book answer: it is an illness caused by MERS-CoV.

PryRe triplet: *significa (Sindrome Respiratorio de Medio Oriente (MERS), null)* {*means*

(Middle East Respiratory Syndrome (MERS)}, null).

PryRe answer: E*s una enfermedad causada por MERS-CoV* {*Is an illness caused by MERSCoV}*.

4 **Question:** *¿Qué es el nuevo coronavirus?* {*What is the new coronavirus?*}

Case applied: C.

Book answer: It is a mutated novel coronavirus (genus B), which is named 2019-nCoV by WHO and SARS-CoV-2 by ICTV.

PryRe triplet: *significa (nuevo cororavirus, null) {means (new coronavirus, null*)}.

PryRe answer: E*s un nuevo coronavirus mutado (género B), que la OMS denomina 2019-nCoV y la ICTV el SARS-CoV-2* {*It is a novel mutated coronavirus (genus B), which is named 2019-nCoV by WHO and SARS-CoV-2 by ICTV*}.

5 **Question:** *¿Qué es la neumonía adquirida en la comunidad?* {*What is community-acquired pneumonia?}*

Case applied: C.

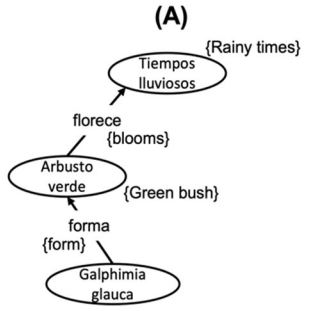
Book answer: Refers to infectious pneumonia of the lung parenchyma (included in the alveolar wall, which belongs to the pulmonary interstitium in a broad sense) contracted outside the hospital, including pneumonia from known pathogens that occur after admission within their period of average incubation.

PryRe triplet: significa (neumonía adquirida en la comunidad, null) {*means (community-acquired pneumonia, null)*}.

PryRe answer: *Se refiere a la neumonía infecciosa del parénquima pulmonar, contraída fuera del hospital* {*Refers to infectious pneumonia of the lung parenchyma, contracted outside the hospital*}.

6 **Question:** *¿Qué patógenos causan neumonía adquirida en la comunidad? {What pathogens cause community-acquired pneumonia*?} Case applied: B.

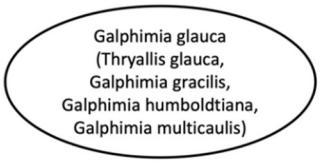
Book answer: The most common pathogens causing acute respiratory diseases include bacteria, viruses, or a combination of bacteria and viruses. New pathogens, such as the new



(B)

```
<concept> Arbusto
   <Language> Spanish <word> bush </word></Language>
   <subset> Planta leñosa </subset>
   <concept> Arbusto verde
      <Language> Spanish <word> Green bush </word></Language>
      <subset> Bush </subset>
      <relation> florece = tiempos lluviosos </relation>
   </concept>
</concept>
<concept> Planta medicinal
   <Language> Spanish <word> Medicinal plant </word></Language>
   <subset> Planta vascular </subset>
   <concept> Galphimia glauca
      <Language> Spanish <word> Galphimia glauca
</word></Language>
      <subset> Planta medicinal </subset>
      <subset> Planta leñosa </subset>
      <relation> forma = Arbusto verde </relation>
   </concept>
</concept>
```

Fig. 13. Explicit relations (bold letter) give more information about concepts



(B)

```
<concept> Galphimia glauca
   <Language> Spanish <word> Galphimia glauca,
                             Thryallis glauca,
                             Galphimia gracilis,
                             Galphimia humboldtiana,
                             Galphimia multicaulis
   </word></Language>
   <subset> Planta medicinal </subset>
   <subset> Planta leñosa </subset>
</concept>
```

Fig. 14. In (A) synonyms inside parentheses are shown. In (B) these synonyms are represented with <word> label (bold letter)

coronavirus, can cause an epidemic or pandemic of acute respiratory disease. Bacteria is the leading cause of community-acquired pneumonia. Strep pneumonia is one of the most common types of bacterial pneumonia. Other bacterial pathogens include Mycoplasma, Chlamydia, Klebsiella pneumonia, Escherichia coli, Staphylococcus aureus, and Acinetobacter baumannii; Pneudonia caused by Pseudomonas aeruginosa has also been reported.

PryRe triplet: causa (neumonía adquirida en la comunidad, null) {*cause (community-acquired pneumonia, null*)},

PryRe answer: Bacteria, virus, nuevo coronavirus, Mycoplasma, Chlamydia, Klebsiella pneumonia, Escherichia coli, Staphylococcus aureus, Pseudomonas aeruginosa, Acinetobacter baumannii {*Bacteria, virus, new coronavirus, Mycoplasma, Chlamydia, Klebsiella pneumonia, Escherichia coli, Staphylococcus aureus, Pseudomonas aeruginosa, Acinetobacter baumannii*}.

7 **Question:** *¿Cómo es transmitida la neumonía adquirida en la comunidad? {How is community-acquired pneumonia transmitted?}*
Case applied: B.

Book answer: Theoretically, all pathogens that cause community-acquired pneumonia have the potential for transmission from person to person. Transmission routes from the source of infection to susceptible populations are transmission by droplets, contact transmission, and air transmission.

PryRe triplet: transmisión (neumonía adquirida en la comunidad, null) {*transmission (community-acquired pneumonia, null*)}.

PryRe answer: Persona a persona, contacto, gotitas, por aire {*Person to person, contact, droplets, by air}.*

8 **Question:** *¿Cuáles son los factores de riesgo de transmisión de la neumonía adquirida en la comunidad?* {*What are the risk factors for the transmission of community-acquired pneumonia?}*
Case applied: B.

Book answer: The Fall and Winter are seasons prone to seeing the prevalence of respiratory viruses such as influenza, and when other respiratory infections occur. These factors are: Environmental conditions, accessibility, and effectiveness of medical care services and Infection prevention measures, host factors, pathogen characteristics.

PryRe triplet: factores de riesgo de transmisión (neumonía adquirida en la comunidad, null) {*transmission risk factors (community acquired pneumonia, null*)}.

PryRe answer: Condiciones ambientales, Accesibilidad y efectividad de los servicios de atención médica, medidas de prevención e infección, factores de huésped, características del patógeno {*Environmental conditions, Accessibility, and effectiveness of medical care services and infection prevention measures, host factors, pathogen characteristics*} (See Figure 16).

9 **Question:** *¿Cómo prevenir la neumonía adquirida en la comunidad? {How to prevent community-acquired pneumonia*?}

Case applied: B.

Book answer: (1) Maintaining a balanced diet, ensuring proper nutrition, and maintaining oral health can help prevent infection. (2) Exercise regularly to increase immunity. (3) Quit smoking, limit alcohol consumption, and try to be positive. (4) Ensure interior ventilation, natural ventilation and/or use of exhaust fans for better airflow. (5) Get vaccinated if available.

PryRe triplet: prevención (neumonía adquirida en la comunidad, null) {*prevention (community-acquired pneumonia, null*)}.

PryRe answer: Mantener una dieta equilibrada, hacer ejercicio con regularidad, DEJAR de fumar, dejar de consumir alcohol, mantenerse de buen humor, asegurar una ventilación interior natural, utilizar extractores de aire, vacunarse { *Maintain a balanced diet, Exercise regularly, Stop smoking, stop consuming alcohol, stay in a good mood, ensure natural indoor ventilation, use air extractors, get vaccinated*}.

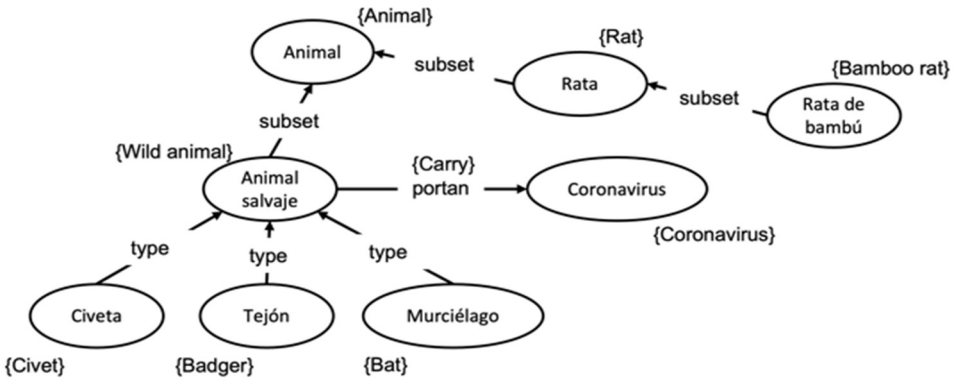


Fig. 15. The problem here is the lack of the *rata de bambú* {*bamboo rat}* concept in *animal salvaje {wild anmal*} that are related to the *coronavirus*. If *rata* {*rat*} was connected to *animal salvaje* {*wild animal*}*,* then Case A could look up the answer and the answer would be: *Sí* {*Yes*}

10 Question: *¿Quién es susceptible al 2019-nCoV? {Who is susceptible to 2019-nCoV?}*
Case applied: B.
Book answer: The coronavirus is newly emerging in humans. Therefore, the general population is susceptible because they lack immunity against it. 2019-nCoV can infect people with normal or compromised immunity. The amount of exposure to the virus also determines whether it becomes infectious. If one is exposed to many viruses, one can get sick even if their immune function is normal. For people with poor immune function, such as the elderly, pregnant women, or people with liver or kidney dysfunction, the disease progresses relatively quickly, and symptoms are more serious.
PryRe triplet: susceptible, vulnerable (2019-nCoV, null).
PryRe answer: Población general, población expuesta, personas con función inmunológica deficiente, personas con disfunción hepática o renal. {*General population, exposed population, people with poor immune function, people with liver or kidney dysfunction*}.

11 Question: *¿Cuáles son las características epidemiológicas de COVID-19?* {*What are the epidemiological characteristics of COVID-19?}*
Case applied: B.
Book answer: The emerging epidemic of COVID-19 has undergone three stages: local outbreak, community communication, and general stage (epidemic). Communication stages: the COVID-19 epidemic went through three stages: 1) the local outbreak stage (the cases of this stage are mainly related to exposure to seafood); 2) the stage of community communication (interpersonal communication and transmission of grouping in communities and families); 3) generalized stage (rapid diffusion, with large population flow, to the whole country of China and even to the world).
PryRe triplet: características epidemiológicas (COVID-19, null) {*epidemiological characteristics (COVID-19, null*)}.
PryRe answer: Etapa de brote local, Etapa de comunicación comunitaria, Etapa generalizada *{Stage of local outbreak, Stage of community communication, Generalized stage*}.

12 Question: *¿Cuáles son las rutas de transmission de* COVID-19*?* {*What are the 2019-nCoV transmission routes?}*
Case applied: B.
Book answer: Currently, transmission via droplets and respiratory contacts is believed to be the primary route, but there is a risk of fecal-oral transmission. Aerosol transmission, mother-to-child transmission, and other routes are not yet confirmed.
PryRe triplet: Rutas de transmission (COVID-19, null) {*transmission routes (COVID-19, null*)}.

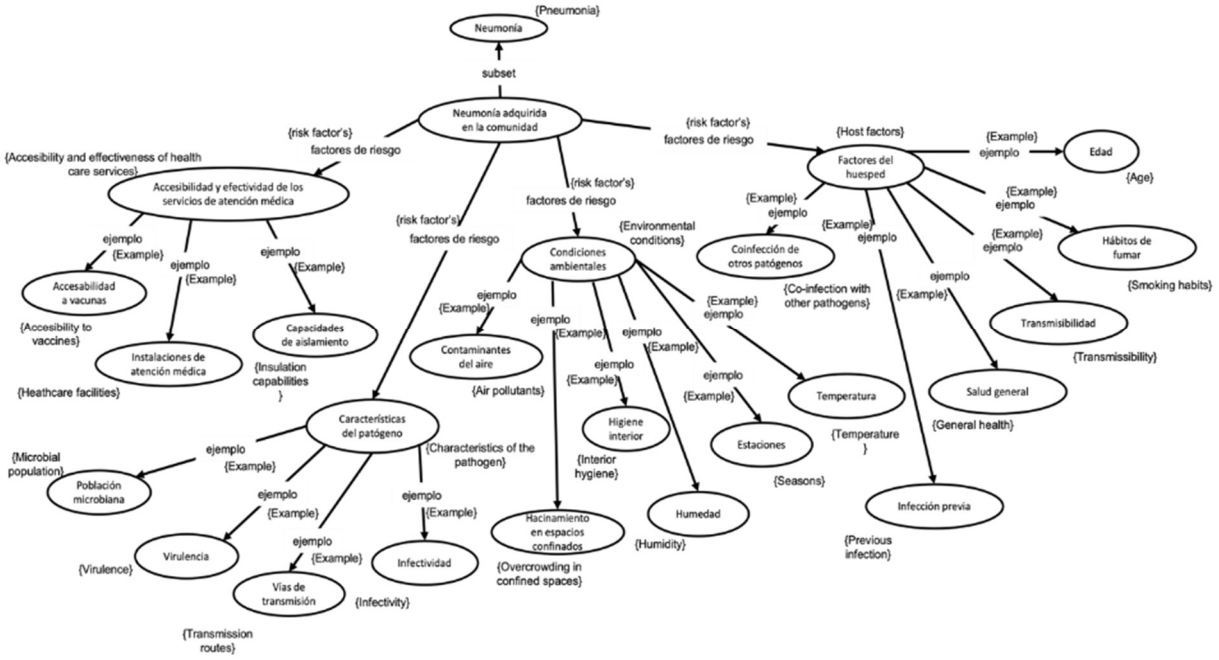


Fig. 16. Part of the ontology that represents the answer to question number 8, in which only the examples of each factor are presented. However, more information can be shown in each example. For example, air pollutants, overcrowding in confined spaces

PryRe answer: Contactos respiratorios, gotitas *{Respiratory contacts, droplets*}.

13 Question: *¿Qué es la transmisión de gotas?* {*What is the transmission of drops?}*

Case applied: C.

Book answer: Drops can enter mucous surfaces within a certain distance (usually 1m). Due to the relatively large size and weight of the droplets, they cannot remain airborne for long.

PryRe triplet: significa (transmisión de gotas, null) *{means (droplet transmission, null)}.*

PryRe answer: Gota es una partícula que contiene agua con un diámetro mayor a 5 mm, las gotas pueden ingresar a las superficies mucosas dentro de una cierta distancia (generalmente 1 m). Debido al tamaño y peso relativamente grandes de las gotas, no pueden permanecer en el aire por mucho tiempo. {*Droplet is a particle that contains water with a diameter greater than 5 mm, the droplets can enter the mucous surfaces within a certain distance (generally 1 m). Due to the relatively large size and weight of the droplets, they cannot remain airborne for long}.*

14 Question: *¿Qué es la transmisión aérea?* {*What is air transmission?}*

Case applied: C.

Book answer: Also known as Aerosol Transmission. Aerosols are suspensions of small particles or droplets that can be transmitted through the air.

PryRe triplet: significa (*transmisión aérea*, null) *{means (airborne transmission, null)*}.

PryRe answer: También conocido como transmisión por aerosol. Los aerosoles son suspensiones de pequeñas partículas o gotitas que se pueden transmitir a través del aire. {*Also known as aerosol transmission. Aerosols are suspensions of small particles or droplets that can be transmitted through the air}.*

Table 4. The fifty questions answered by PryRe

Number	Case Applied	Precision
1	C	1.0
2	B	1.0
3	C	1.0
4	C	1.0
5	C	1.0
6	B	1.0
7	B	1.0
8	B	1.0
9	B	1.0
10	B	1.0
11	B	1.0
12	B	1.0
13	C	1.0
14	C	1.0
15	C	1.0
16	C	1.0
17	B	1.0
18	B	1.0
19	B	1.0
20	B	1.0
21	B	1.0
22	Unsolved	0.0
23	Unsolved	0.0
24	C	1.0
25	B	1.0
26	B	1.0

27	B	1.0
28	Unsolved	0.0
29	B	1.0
30	Unsolved	0.0
31	B	1.0
32	B	1.0
33	Unsolved	0.0
34	B	1.0
35	Unsolved	0.0
36	B	1.0
37	B	1.0
38	B	1.0
39	B	1.0
40	C	1.0
41	B	1.0
42	B	1.0
43	B	1.0
44	Unsolved	0.0
45	B	1.0
46	D	1.0
47	B	1.0
48	Unsolved	0.0
49	Unsolved	0.0
50	B	1.0

15 Question: *¿Qué es la transmisión por contacto?* {*What is contact transmission*?} Case applied: D.

Book answer: Contact transmission refers to the transmission of pathogens through direct or indirect contact through fomites (objects carrying pathogens): (1) Direct contact. Pathogens are transmitted by direct contact of the mucous or skin with an infected host. Blood or bloody fluids enter the body through mucous membranes or non-intact skin (mainly viruses).

Transmission is caused by contact with secretions that contain certain pathogens, commonly due to infections by bacteria, viruses, parasites, etc. (2) Indirect contact. Pathogens are transmitted through contaminated objects or people. The pathogens of intestinal infectious diseases are transmitted mainly through indirect contact. (3) Other important pathogens transmitted by indirect contact: MRSA (Staphylococcus aureus resistant to benzoxazole/methicillin), VRE (vancomycin-resistant enterococcus), Clostridium difficile.

PryRe triplet: significa (transmisión por contacto, null) {*means (contact transmission, null)}.*

PryRe answer: Se refiere a la transmisión de patógenos por contacto directo o indirecto a través de fomites (objetos portadores de patógenos): 1) Contacto directo. (2) Contacto indirecto. (3) Otros patógenos importantes transmitidos por contacto indirecto {*Refers to the transmission of pathogens through direct or indirect contact through fomites (objects carrying pathogens): (1) Direct contact. (2) Indirect contact. (3) Other important pathogens transmitted by indirect contact}.*

16 Question: *¿Qué es un contacto cercano?* {*What is close contact?*}

Case applied: D.

Book answer: Close contacts refer to people who have contact with a confirmed or suspected 2019-nCoV infection patient), including the following situations. (1) Those who live, study, work, or have other forms of close contact with a patient, (2) Medical personnel, family members, or others who have had close contact with a patient without taking effective protective measures during diagnosis, treatment, nursing and visits, (3) Other patients and their companions who share the same room with an infected patient, (4) Those who shared the same transport or lift with the patient, (5) Those who are considered as such through field investigations.

PryRe triplet: significa (contacto cercano, null) *{means (close contact, null)}.*

PryRe answer: Referirse a personas que tienen contacto con un paciente con infección confirmada o sospechada de 2019-nCoV, incluidas las siguientes situaciones: (1) Quienes viven, estudian, trabajan o tienen otras formas de contacto cercano con un paciente, (2) Personal médico, familiares u otras personas que han tenido contacto cercano con un paciente sin tomar medidas de protección efectivas durante el diagnóstico, tratamiento , enfermería y visitas, (3) Otros pacientes y sus acompañantes que comparten la misma habitación con un paciente infectado, (4) Los que compartieron el mismo transporte o ascensor con el paciente, (5) Los que son considerados como tales a través de investigaciones de campo {*It refers to people who have contact with a confirmed or suspected 2019-nCoV infection patient, including the following situations: (1) Those who live, study, work, or have other forms of close contact with a patient, (2) Medical personnel, family members, or others who have had close contact with a patient without taking effective protective measures during diagnosis, treatment, nursing and visits, (3) Other patients and their companions who share the same room with an infected patient, (4) Those who shared the same transport or lift with the patient, (5) Those who are considered as such through field investigations.}*

In Question 16, PryRe only presents the types of transmission by contact but does not define each of them: only *Contacto directo* {Direct contact} appears, and not its description.

In Question 16 PryRe presents the types of people who can transmit the virus since this answer is an important part of the question. That is, (2) *Personal médico* {*medical personnel}, familiares {relatives*}, etc.

The two books that were used had different structures. The book about herbs and medical plants [14] is based on tables and lists. The coronavirus book is based on questions/answers [22].

The output of PryRe is text. Thus, the results presented in this section are copied and pasted by hand. 50 questions answered by PryRe, are presented here.

The ontology about both topics (medical plants and coronavirus book) was based in the WordNet

hierarchy; this ontology is downloaded in Spanish from [4].

7 Conclusions and Future Work

Based on an ontology, PryRe interprets questions in Spanish, and provides the answers in Spanish. It does this with acceptable precision and can be used, as it is, with other ontologies to answer related questions in natural language.

The knowledge base was built manually obtaining the information from reliable sources [14, 22]: one hundred questions about coronavirus and 85 descriptions of medicinal plants with its features: scientific name, place of collection, location, coordinates, name of the collector, year of collection, altitude, and observations are represented. This yields an average of eight relations or links per description. Building the knowledge base in this manner was a tedious process. Then, PryRe tested (using 185 questions) the complete knowledge. The accuracy (percentage of correct answers) obtained by PryRe has been 82%. Table 4 shows only fifty of these questions.

In Table 4, 41 answers have precision 1 (perfect answer), and 9 have 0 (no match), because the book's answer is ambiguous, not direct or concrete. These 9 responses were considered unsolved as PryRe. The total precision of PryRe for this test, is therefore (41)/50 = 0.82 (82%).

Availability of the complete ontology is here[5], and the code of PryRe is publicly available here[6].

The next work is to use crawlers and scrappers to retrieve from the web other useful information, and to review manually its suitability. Then, transform it into OM notation with the help of the Ontology Merger [18]. This path increases the body of knowledge for PryRe, in a semi-automated, supervised way.

References

1. **Landes, S., Leacock, C., Tengi, R. I. (1998).** Building semantic concordances. In: **Fellbaum, C.**, **ed.,** WordNet: an electronic lexical database, chapter 8, MIT Press, pp. 199–216.
2. **Villanueva, D., Cuevas-Rasgado, A. D., Juárez, O., Guzmán-Arenas, A. (2013)**. Using frames to disambiguate propositions. Expert Systems with Applications, Vol. 40, No. 2, pp. 598–610. DOI: 10.1016/j.eswa.2012.07.061.
3. **Gruber, T. R. (1995).** Toward Principles for the Design of Ontologies Used for Knowledge Sharing? International Journal of Human-computer Studies, Vol. 43, No. 5–6, pp. 907–928. DOI: 10.1006/ijhc.1995.1081.
4. **Guzmán-Arenas, A., Cuevas, A. D. (2010).** Knowledge accumulation through automatic merging of ontologies. Expert Systems with Applications, Vol. 37, No. 3, pp. 1991–2005. DOI: 10.1016/j.eswa.2009.06.078.
5. **Padró, L., Stanilovsky, E. (2012).** FreeLing 3.0: Towards Wider multilinguality. 8th International Conference on Language Resources and Evaluation (LREC), pp. 2473–2479.
6. **Breck, E., Burger, J., House, D., Light, M., Mani, I. (1999).** Question Answering from Large Document Collections. AAAI Fall Symposium on Question Answering Systems.
7. **Vargas-Vera, M., Motta, E. (2004).** AQUA-ontology-based question answering system. Lecture Notes in Computer Science, Vol. 2972, pp. 468–477. DOI: 10.1007/978-3-540-24694-7_48.
8. **Alinaghi, T., Bahreininejad, A. (2011).** A multi-agent question-answering system for e-learning and collaborative learning environment. International Journal of Distance Education Technologies (IJDET), Vol. 9, No. 2, pp. 23–39. DOI: 10.4018/jdet.2011040103.
9. **Bordes, A., Weston, J., Chopra, S. (2014).** Question answering with subgraph embeddings. EMNLP: Conference on Empirical Methods in Natural Language Processing, pp. 615–620. arXiv:1406.3676. DOI: 10.48550/arXiv.1406.3676.
10. **Bast, H., Haussmann, E. (2015).** More accurate question answering on freebase.

[5] https://tinyurl.com/524m8fj4

[6] https://tinyurl.com/y37p9tph

24th ACM International Conference on Information and Knowledge Management, pp. 1431–1440. DOI: 10.1145/2806416.2806472.

11. **Xu, K., Reddy, S., Feng, Y., Huang, S., Zhao, D. (2016).** Question answering on freebase via relation extraction and textual evidence. 54th Annual Meeting of the Association for Computational Linguistics, pp. 2326–2336. DOI: 10.18653/v1/P16-1220.
12. **Abujabal, A., Yahya, M., Riedewald, M., Weikum, G. (2017).** Automated template generation for question answering over knowledge graphs. 26th international conference on world wide web, pp. 1191–1200. DOI: 10.1145/3038912.3052583.
13. **Zhu, X., Yang, X., Chen, H. (2018).** A biomedical question answering system based on SNOMED-CT. Lecture Notes in Computer Science, Vol. 11061, pp. 16–28. DOI: 10.1007/978-3-319-99365-2_2.
14. **Estrada, E., Lara, A. (2008).** Sistema nervioso y herbolaria. Universidad Autónoma Chapingo.
15. **Noy, N. F., McGuinness, D. L. (2001).** Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory.
16. **Swaminathan, V., Sivakumar, R. (2012).** A Comparative Study of Recent Ontology Visualization Tools with a Case of Diabetes Data. International Journal of Research in Computer Science (IJORCS), Vol. 2, No. 3, pp. 31–36. DOI: 10.7815/ijorcs.23.2012.026.
17. **Castañeda, E., Cortés, O. (2011).** Construcción de una base de conocimiento sobre el uso de herramientas de carpintería. Instituto Politécnico Nacional.
18. **Cuevas, A. D. (2008).** Unión de Ontologías usando propiedades semánticas. Doctoral Dissertation Thesis, Centro de Investigación en Computación, IPN.
19. **Pavon, L. (1999).** Clases de partículas: preposición, conjunción y adverbio. In: **Bosque, I., DeMonte V., eds.**, Gramática descriptiva de la lengua española, Capítulo 9. Espasa-Calpe. Vol. 1, pp. 565–656.
20. **Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J. (2019).** Industry-Scale Knowledge Graphs: Lessons and Challenges: Five diverse technology companies show how it´s done. Communications of the ACM, Vol. 62, No. 8, 36-43. DOI: 10.1145/3331166.
21. **Cuevas, R. A. D., Niño, M. Y. E., Lamont, F. G. (2017).** Semantic analyzer for Spanish, using ontologies. Komputer Sapiens, Vol. 3, pp. 13–36.
22. **Zhou, W. (2020).** The Coronavirus Prevention Handbook 101 Science-based Tips that Could Save your Life. Skyhorse Publishing.

Article received on 09/06/2021; accepted on 24/01/2022.
Corresponding author is Alma Delia Cuevas-Rasgado.

ISSN 2007-9737

Exploratory Data Analysis and Sentiment Analysis of Drug Reviews

Bijayalaxmi Panda, Chhabi Rani Panigrahi, Bibudhendu Pati

Rama Devi Women's University,
Department of Computer Science,Bhubaneswar,
India

bijayalaxmi.panda81@rediffmail.com,
{panigrahichhabi, patibibudhendu}@gmail.com

Abstract. The exponential increase in the volume of data in our daily life needs to be managed and analyzed properly to get knowledge and benefit out of that. A drug review dataset obtained from the UCI machine learning repository has six parameters namely drug-id, name, condition, review, rating, and usefulness count out of which we have filtered a subset of the dataset based on the eight conditions. Exploratory Data Analysis (EDA) and Sentiment Analysis (SA) are then applied to the filtered data set. EDA shows the total number of medicines used for all light conditions, number of reviews per condition, five most popular drugs based on usefulness count, number of drugs per condition, etc. SA is performed on the filtered dataset, in which twenty-eight drugs are compared based on rating and polarity where three drugs *Lisdexamfetamine*, *Vyvanse*, *Lamotrig ine* are found to be the best in the view of customers as per their rating and positive polarity and *Suvorexant* is the drug found to have negative polarity and least rating.

Keywords. Drug review, exploratory data analysis, adverse drug reaction, subjectivity, polarity.

1 Introduction

In this digital world, people do not have time to spend for their day-to-day activities physically. Therefore, they depend on several electronic activities such as purchasing goods, an appointment with doctors, bank transactions, and so on. In the current time, every need of human beings is satisfied by electronic means. While purchasing goods, a user's first choice is a trusted website then the user checks the customer feedback and rating. By analyzing such things, we get ready for purchasing such products. Many datasets are available related to Amazon product review [19].

Similarly, there are several aspects of health sectors on which we focus such as patient review on choosing a hospital for treatment, health check-up, drug review, a side effect of drugs dosage and effectiveness, etc. [11].

Clinical-social-personality is the standard of measurement in health-related sentiment analysis [12]. This helps the drug makers by giving them opinion of drug users.

The rest part of the paper is organized as follows. Section 2 presents the related work. Section 3 presents the methodology for the proposed approach, which contains data description, exploratory data analysis, and sentiment analysis. Section 4 describes the experimental results and discussion. Section 5 concludes the paper and identifies certain future research directions.

2 Related Work

In the present scenario, the drug is an essential aspect of human life. Many researchers are working on drug reviews so that common people can get an idea about the best drug for a particular disease. Cavalcanti *et al.* [1] suggested a new unsupervised and knowledge-based method for the extraction of aspects in drug reviews.

Hiremath *et* al. [2] focused on a case study to develop a clinical decision support system for

personalized therapy process using aspect- based sentiment analysis.

The process is carried out on drug review data to determine whether the patient's behavior towards a medicine, product, treatment, etc is positive, negative, or neutral using Natural Language Processing techniques. The polarities obtained are compared for further analysis of the patient reviews for a better clinical decision system.

Das *et al*. [3] developed a learning model that can be trained to predict the disease type when provided with a drug name and its corresponding review. To mitigate the above- mentioned issue, the authors presented and compared various machine learning-based prediction models and their performance compared based on metrics such as precision, recall, F1-Score, and accuracy.

Vijayaraghavan et al. [4] worked on analyzing reviews of various drugs which have been reviewed in the form of texts and have also been given a rating on a scale from 1-10. We had obtained this data set from the UCI machine learning repository which had 2 data sets: train and test (split as 75-25%). We had split the number rating for the drug into three classes in general: positive (7-10), negative (1-4), or neutral (4-7). There are multiple reviews for the drugs that belong to a similar condition and we decided to investigate how the reviews for different conditions use different words impact the ratings of the drugs.

Our intention was mainly to implement supervised machine learning classification algorithms that predict the class of the rating using the textual review. We had primarily implemented different embedding such as Term Frequency Inverse Document Frequency (TFIDF) and the Count Vectors (CV). Authors had trained models on the most popular conditions such as "Birth Control", "Depression" and "Pain" within the data set and obtained good results while predicting on the test data sets.

Shiju *et al.* [5] built different classification models to classify user ratings of drugs with their textual review. Multiple supervised machine learning models including Random Forest and Naive Bayesian classifiers were built with drug reviews using TF-IDF features as input. Also, transformer-based neural network models including BERT, BioBERT, RoBERTa, XLNet,ELECTRA, and ALBERT were built for classification using the raw text as input.

Overall, BioBERT model outperformed the other models with an overall accuracy of 87%. Compagner *et al.* [6] focused on characterizing the sentiment of online medication reviews of Selective Serotonin Reuptake Inhibitors (SSRIs) and Serotonin–Norepinephrine Reuptake Inhibitor (SNRIs) used to treat depression. The publicly available data source used was the Drug Review Dataset from the University of California Irvine Machine Learning Repository. This study utilized a sentiment analysis of free- text, online reviews via the sentimentr package.

The result shows that average sentiment was higher in SSRIs compared to SNRIs (0.065 vs. 0.005, $p < 0.001$). The average sentiment was also found to be higher in high-rated reviews than in low-rated reviews (0.169 vs. −0.367, $p < 0.001$). Ratings were similar in the high-rated SSRI group and high-rated SNRI group (9.19 vs. 9.19).

Gräßer *et al.* [7] proposed a new approach that includes different steps. First, extra parameters are added to the review data by applying VADER sentimental analysis to clean the review data. Then, different machine learning algorithms are applied, namely linear SVC, logistic regression, SVM, random forest, and Naive Bayes on the drug review dataset. To improve this, a stratified K-fold algorithm was applied in combination with Logistic regression. With this approach, the accuracy obtained was increased to 96%.

Mishra [19] performed sentiment analysis of the reviews of drugs given by the patients after the usage using the boosting algorithms in machine learning. The dataset used, provides patient reviews on some specific drugs along with the conditions the patient is suffering from and a 10-star patient rating reflecting the patient satisfaction.

EDA is carried out by the customers to get more insight and engineer features. To classify the reviews as positive or negative three classification models such as LightGBM, XGBoost, and CatBoost were trained and the feature importance is plotted.

The results show that LGBM is the best performing Boosting algorithm with an accuracy of 88.89%. In most of the works related to drug review dataset, it was found that researchers used

supervised machine learning algorithms for classification and comparison.

EDA was performed on the whole dataset in [19] to obtain inside features. In this work, we have performed EDA on drugs of specific disease so that relevant drugs can be compared.

For this, we have taken drugs having more number of reviews for comparison.

3 Methodologies

In this section, we have presented the description of the dataset used along with EDA which includes the result analysis of the drug review dataset, and SA which is used to analyze the subjectivity and polarity of the dataset.

3.1 Data Set Description

The drug review data set was collected from the UCI machine learning repository [18]. The dataset contains patient reviews subject to specific drugs, along with conditions and a 10- point rating depending on the fulfillment of the needs of patients. The dataset contains six attributes such as the name of the drug, disease name marked as condition, patient review as review, rating, review entry date, number of users who found review as useful marked as usefulCount.

The data was collected from drug review sites like druglib.com, drugs.com, etc. The data set is represented in two tsv files as train and test. In our work, we have joined the two files and extracted one subset of the data set by filtering out the data based on eight conditions like depression, Insomnia, anxiety, anxiety and stress, Bipolar disorder, major depressive disorder, ADHD, and panic disorder. Then we performed exploratory data analysis on the filtered data set.

3.2 Exploratory Data Analysis (EDA)

EDA is the evaluative process of execution on data to uncover the designs, solve problems, testing hypotheses in virtue of analytical and pictorial representations. EDA is performed based on sample data sets [9]. We can acquire more and more perceptions by performing EDA on sample data set. In this work, we applied certain analytical processes to the filtered data set of drug review data set.

3.3 Sentiment Analysis (SA)

The process of classifying text into positive, negative, and neutral sentiment using different methods of NLP is known as sentiment analysis. This is used in several areas like a movie review, product review, and drug review, and so on. In this research we have focused on drug review, detecting and monitoring adverse drug reactions, or identifying negative or positive sentiment of patients is the part of research where sentiment analysis is applied on UCI drug review data set. [15, 17] We applied text blob analysis on reviews obtained from drug users.

Textblob analysis shows the classification of positive, negative, and neutral reviews. Polarity obtained lies between the range of -1 and +1. Then a comparison of several drugs was made to find out the best one.

4 Experimental Setup and Results

This section describes the experimental setup required for the proposed approach along with results and discussion.

4.1 Experimental Setup

In this work, we have used the drug review dataset for our experimentation. The dataset was taken from the UCI machine learning repository [8]. The dataset contains parameters such as drug name, disease, review of users for a specific drug, rating, etc. EDA and SA are performed on the drug review dataset. We created a subset of the main dataset taking into account eight conditions.

EDA shows an insight of the subset of dataset such as details of medicines used for the said conditions, number of drugs per condition, most popular drugs according to their usefulness. In sentiment analysis part by using textblob in python we performed subjectivity analysis to find out positive and negative opinions of patients who are using these drugs. Polarity was obtained and compared with drugs.

Table 1. Reviews with usefulness count

Id	Condition	Rating	Drug Name	Review	Useful Count
96616	Depression	10	Sertraline	"I remem ber reading people 's opinions, on...	1291
119151	Depression	9	Zoloft	"I&#039 ;ve been on Zoloft 50mg for over two ye...	949
62688	Anxiety and Stress	8	Citalopram	"I work for a large Fire Depart ment. I was ha...	693

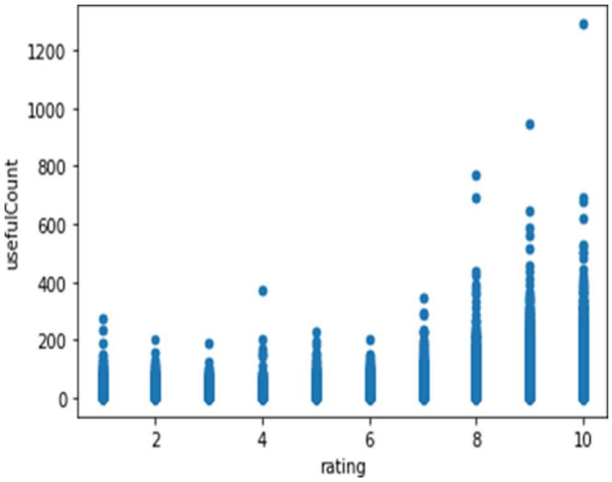


Fig.1. Rating vs. useful count

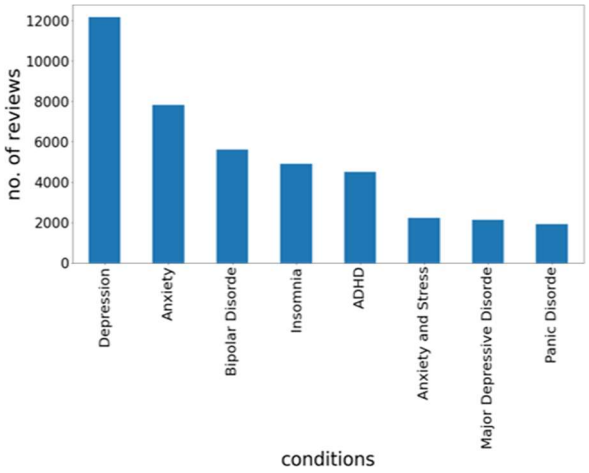


Fig. 2. Number of reviews per condition

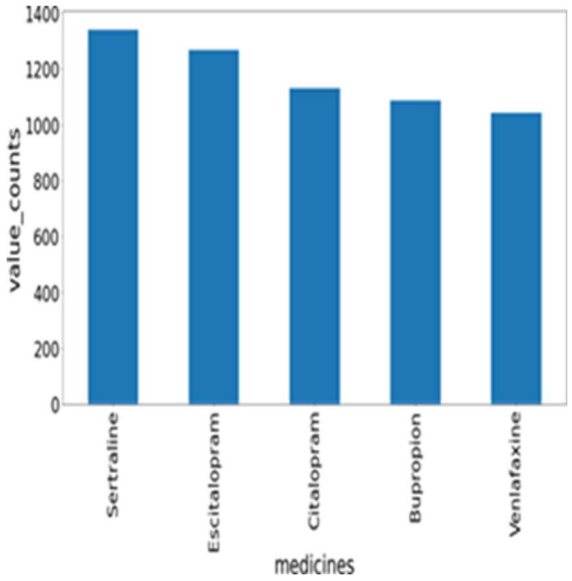


Fig. 3. Popular drugs based on counts

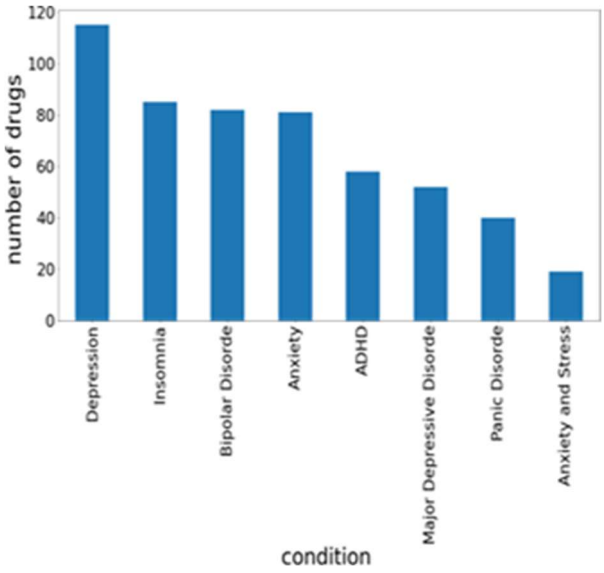


Fig. 4. Number of drugs per condition

Table 2. Mean rating and mean polarity of drugs

Sl. No	Drug name	Condition	Mean rating	Mean polarity
1	Bupropion	depression	7.445	0.097
2	Sertraline	depression	7.497	.066
3	Venlafaxine	depression	6.8	0.053
4	Pristiq	depression	7.218	0.094
5	Desvenlafaxi ne	depression	7.274	0.092
6	Citalopram	depression	7.666	0.081
7	Escitalopram	depression	7.843	0.082
8	Lexapro	depression	7.808	0.081
9	Cymbalta	depression	6.572	0.061

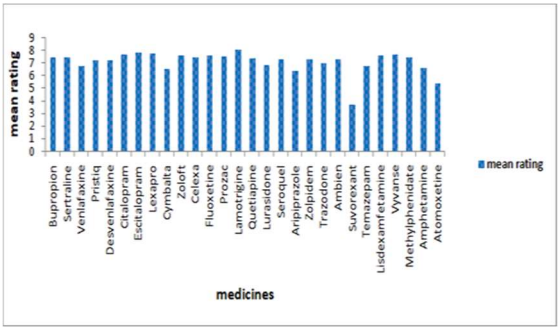


Fig. 5. Mean rating of drugs

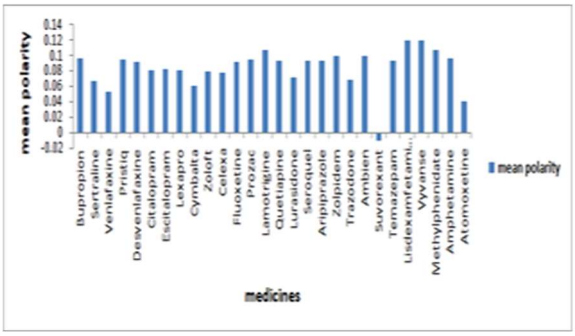


Fig. 6. Mean Polarity of Drugs

Twenty-eight drugs were compared based on their polarity and rating and the best medicine was obtained.

4.2 Results and Discussion

In this section, EDA is performed and results are analyzed.

Exploratory Data Analysis (EDA)

The filtered data set contains six attributes with eight numbers of conditions.

The number of medicines used for those conditions is 299. Table 1 contains the top three reviews on the basis of usefulness count. The users found two most popular drugs useful and are Sertraline and Zoloft. In this case, condition is depression and useful Count are 1291 and 949 respectively.

We drew a scatter plot on rating verses useful Count where the medicine Sertraline and Zoloft which is having 10 rating has highest useful count as 1291.

Some users also have given those medicines 9 rating where useful count is 949 which is clearly understood from the scatter plot as shown in Fig. 1.

The bar graph as shown in Fig. 2 shows the number of reviews per condition. This graph shows highest number of reviews in depression condition and lowest number of reviews in panic disorder condition.

The graph shows top five most popular drugs on the basis of their usefulness count where we found Sertraline, Escitalopram, Citalopram, Bupropion, Venlafaxine are top five most popular drugs and is shown in Fig. 3.

From Fig. 4, it is clear that Depression has highest number of drugs that is 115 and the condition Anxiety and stress has lowest number of drugs such as 19.

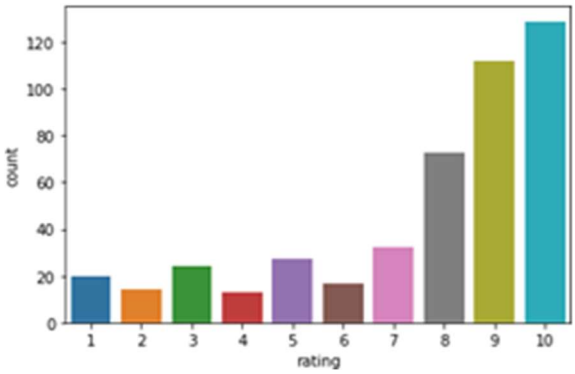


Fig. 7. Rating vs Count of Lisdexamfetamine

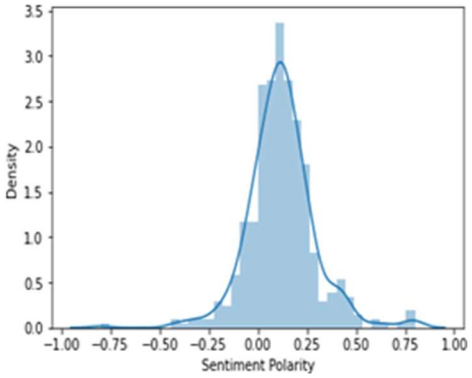


Fig. 8. Sentiment polarity vs. Density of Lisdexamfetamine

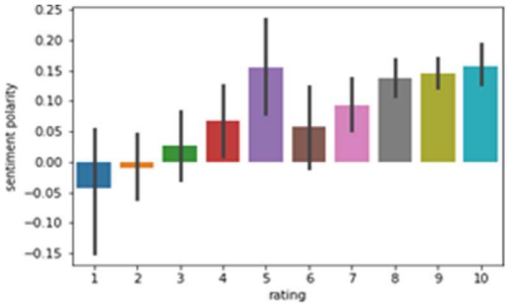


Fig. 9. Rating vs. sentiment polarity of Lisdexamfetamine

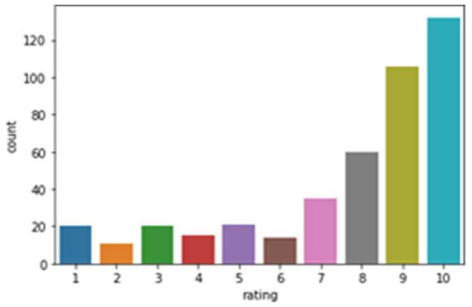


Fig. 10. Rating vs. Count of *Vyvanse*

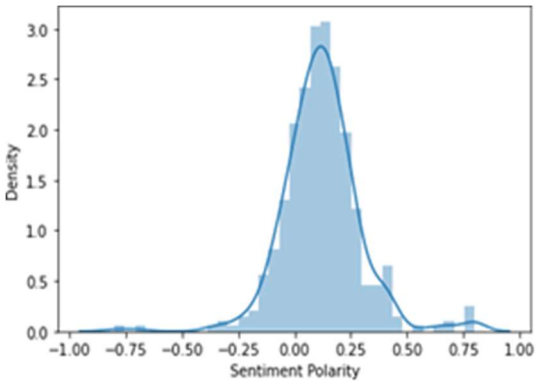


Fig. 11. Sentiment polarity vs. Density of Vyvanse

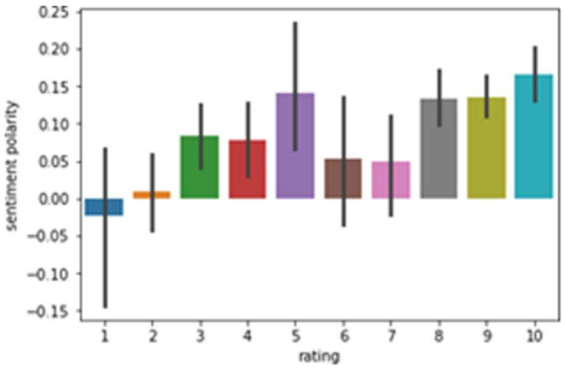


Fig. 12. Rating vs. Sentiment polarity of Vyvanse

Similar kind of drugs may be used for all the conditions. This is shown in Fig. 4.

Sentiment Analysis

Social media data are useful based on healthcare, disease diagnosis and so on. Sentiment analysis is the way to facilitate analysis of information obtained from social media and gives benefit to same kind of users [13]. Self-reported patient data can be obtained from social media that gives positive impact on other patients [14, 16]. On the basis of 8 conditions we have filtered the subset from drug review data set. Many drugs are suggested for each and every conditions. All the

conditions are psychiatric conditions according to which popular drugs are selected and preprocessed the number of reviews [10].

We have performed sentiment analysis using textblob module of python for text classification as positive, negative and neutral.

Then analysis is performed to define polarity numerically, where polarity ranges from -1 to +1. We have taken around 28 medicines on the basis of number of reviews. We have considered those medicines when the number of reviews must exceed 100.All the 28 drugs related to at least 1 condition and at most 8 conditions because one drug can be used to treat several conditions. We have compared the drugs on the basis of their mean rating and mean polarity.

The mean rating of 28 medicines represented in Fig. 5 shows that according to customer rating.

Sentiment analysis according to user reviews obtained from our experimental study is clearly mentioned in Fig. 6. It defines polarity in the range -1 to +1. In other words, it defines positive or negative polarity. According to graphical representation Lisdexamfetamine, Vyvanse, and Lamotrigine are having corresponding polarity values 0.1204, 0.12, and 0.1075 respectively.

Lisdexamfetamin and Vyvanse, both are used to treat the condition Attention deficit hyper activity (ADHD) and Lamotrigine is used to treat the condition Bipolar disorder. From the results it was found that all three medicines are having nearly same positive polarity which is widely accepted by the customers. Suvorexant is the medicine used for treatment of the condition.

Lamotrigine is the best medicine which has mean rating of 8.113 which is mainly used for treatment of bipolar disorder. Polarity of this medicine is 0.107. Customers given lowest rating to the medicine. Suvorexant used to treat Insomnia that is 3.761.So customers are not appreciating this medicine properly, so it needs to be improved.

Insomnia which is having lowest polarity value of -0.0108. This shows negative polarity which is not appreciated by customers and needs to be improved.

The detailed analysis and description of all the three medicines are given below.

The bar graph as shown in Fig. 7 shows rating verses count of Lisdexamfetamine and it shows that maximum customers have given 10 star rating

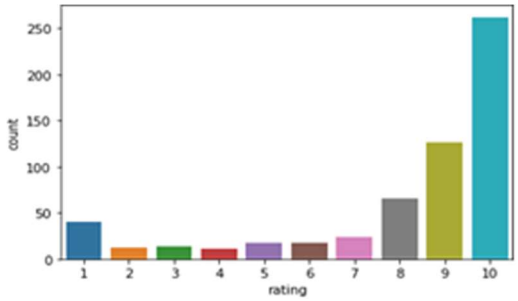


Fig. 13. Rating vs. Count of Lamotrigine

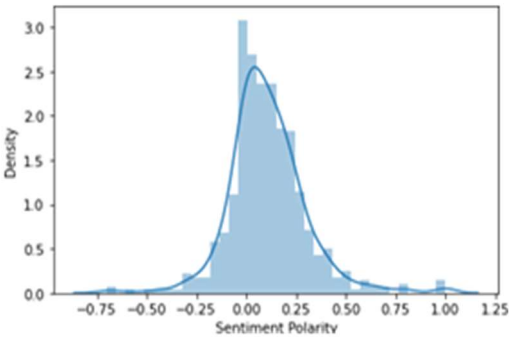


Fig. 14. Sentiment Polarity vs. Density of Lamotrigine

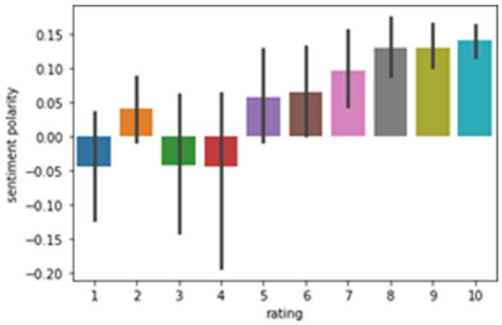


Fig. 15. Rating vs. Sentiment polarity of Lamotrigine

followed by 9, 8, and 7 rating respectively. The histogram as shown in Fig. 8 shows data distribution as sentiment polarity verses density.

The frequency distribution is higher in case of positive polarity and lower in case of negative polarity, so we can assume that users of this medicine have positive sentiment towards it. Fig. 8 shows that the highest frequency lies between 0 and 0.25. Fig. 9 shows rating verses sentiment polarity of Lisdexamfetamine. It is found that most of the customers have given rating 5 where polarity is highest. Many customers have also given 8, 9

and 10 rating where polarity also increases accordingly.

Vyvanse is same kind of medicine as Lisdexamfetamine with negligible difference in number of reviews. Fig. 10, 11, and 12 shows the features of Vyvanse.

The histogram as shown in Fig. 14 shows sentiment polarity verses density where in spite of some existing negative comments there are much more positive reviews of customers who have used this medicine. So in this case positive polarity is found to be higher and hence the product can be considered as reliable.

Lamotrigine sentiment analysis proves it as the 2nd best suggested medicine in our drug review analysis. The graph as shown in Fig. 13 shows rating verses count of Lamotrigine, where users have given highest 10 star rating to this product followed by 9 and 8. So the number of positive reviews is more for Lamotrigine.

From Fig. 15, it is found that many users have given 10 star rating followed by 9, 8 and 7. Sentiment polarity is highest in case of rating 8. It also shows that rating 9 and 10 has more positive polarity.

5 Conclusion and Future Work

In this particular research, EDA and SA is applied on the drug review dataset obtained from UCI machine learning repository. On the basis of eight considered conditions, dataset was compiled because the customers had given more reviews on those conditions.

In EDA, the total number of medicines used for all eight conditions, number of reviews per condition, five most popular drugs on the basis of usefulness count, number of drugs per each condition etc. are described where as in SA, 28 drugs were compared on the basis of rating and polarity.

The drugs such as Lisdexamfetamine, Vyvanse, and Lamotrigine are found to the best drugs in the view of customers as per their rating and positive polarity. Suvorexant was found to be the drug having negative polarity and least rating. Other than eight considered conditions there are also several other conditions and drug classes present in the data set and we can categorize the drugs into groups and can also analyze the drug review dataset for different conditions.

References

1. **Cavalcanti, D. C., Prudêncio, R. B. C. (2017).** Unsupervised aspect term extraction in online drugs reviews. 30th International Florida Artificial Intelligence Research Society Conference (FLAIRS), pp. 38–43.
2. **Hiremath, B. N., Patil, M. M. (2020).** Enhancing optimized personalized therapy in clinical decision support system using natural language processing. Journal of King Saud University-Computer and Information Sciences, Vol. 34, No. 6, pp. 2840–2848. DOI: 10.1016/j.jksuci.2020.03.006.
3. **Das, S., Kumar-Mahata, S., Das, A., Deb, K. (2021).** Disease prediction from drug information using machine learning. American Journal of Electronics & Communication, Vol. 1, No. 4, pp. 16–21. DOI: 10.15864/ajec.1403.
4. **Vijayaraghavan, S., Basu, D. (2020).** Sentiment analysis in drug reviews using supervised machine learning algorithms, arXiv:2003.11643. DOI:10.48550/.arXiv.2003.11643.
5. **Shiju, A., He, Z. (2021).** Classifying drug ratings using user reviews with transformer-based language models. MedRxiv. DOI: 10.1101/2021.04.15.21255573.
6. **Compagner, C., Lester, C., Dorsch, M. (2021).** Sentiment analysis of online reviews for selective serotonin reuptake inhibitors and serotonin–norepinephrine reuptake inhibitors. Pharmacy, Vol. 9, No. 1. DOI: 10.3390/pharmacy9010027.
7. **Gräßer, F., Kallumadi, S., Malberg, H., Zaunseder, S. (2018)**. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. Proceedings of the International Conference on Digital Health (DH), pp. 121–125. DOI: 10.1145/3194658.3194677.
8. **Patil, P. (2021).** What is exploratory analysis? Towards data science (TDS).
9. **Zolnoori, M., Fung, K. W., Patrick, T. B., Fontelo, P., Kharrazi, H., Faiola, A., Shah, N.**

D., Wu, Y. S. S., Eldredge, C. E., Luo, J. (2019). The PsyTAR dataset: From patients generated narratives to a corpus of adverse drug events and effectiveness of psychiatric medications. Data in brief, Vol. 24. DOI: 10.1016/j.dib.2019.103838.

10. **Cavalcanti, D., Prudêncio, R. (2017).** Aspect-based opinion mining in drug reviews. In: **Oliveira, E., Gama, J., Vale, Z., Lopes-Cardoso, H., eds.**, Progress in Artificial Intelligence (EPIA), Lecture Notes in Computer Science, Springer Cham, Vol. 10423, pp. 815–827. DOI: 10.1007/978-3-319-65340-2_66.
11. **Cohen, J. (1960).** A coefficient of agreement for nominal scales. Educational and psychological measurement, Vol. 20, No. 1, pp. 37–46. DOI: 10.1177/00131644600 2000104.
12. **Denecke, K. (2015).** Sentiment analysis from medical texts. Health Web Science: Social Media Data for Healthcare, Springer, Cham, pp. 83–98. DOI: 10.1007/978-3-319-20582-3_10.
13. **Gopalakrishnan, V., Ramaswamy, C. (2017).** Patient opinion mining to analyze drugs satisfaction using supervised learning. Journal of applied research and technology, Vol. 15, No. 4, pp. 311–319. DOI: 10.1016/j.jart. 2017.02.005.
14. **Korkontzelos, I., Nikfarjam, A., Shardlow, M., Sarker, A., Ananiadou, S., Gonzalez, G. H. (2016).** Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. Journal of biomedical informatics, Vol. 62, pp. 148–158. DOI: 10.1016/j.jbi.2016.06.007.
15. **Leaman, R., Wojtulewicz, L., Sullivan, R., Skariah, A., Yang, J., Gonzalez, G. (2010).** Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. Workshop on biomedical natural language processing, Association for Computational Linguistics (BioNLP), pp. 117–125.
16. **Mishra, A., Malviya, A., Aggarwal, S. (2015).** Towards automatic pharmacovigilance: analysing patient reviews and sentiment on oncological drugs. IEEE International Conference on Data Mining Workshop (ICDMW), pp. 1402–1409. DOI: 10.1109/ICDMW.2015.230.
17. **UCI (2021).** UCI Machine Learning Repository. UCI Center for Machine Learning and Intelligent Systems.
18. **Srikanth, K., Murthy, N. V. E. S., Prasad-Reddy, P. V. G. D. (2021).** Sentiment classification on online retailer reviews. 3rd International Conference on Communications and Cyber Physical Engineering, Lecture Notes in Electrical Engineering, Vol. 698, pp. 1557–1563. DOI: 10.1007/978-981-15-7961-5_140.
19. **Mishra, S. (2021).** Drug review sentiment analysis using boosting algorithms. International Journal of Trend in Scientific Research and Development (IJTSRD), Vol. 5, No. 4, pp. 937–941.

Article received on 01/12/2021; accepted on 03/03/2022.
Corresponding author is Chhabi Rani Panigrahi.

ISSN 2007-9737

Cybersecurity and Internet of Things. Outlook for this Decade

Jairo Eduardo Márquez Díaz

Universidad de Cundinamarca,
Colombia

jemarquez@ucundinamarca.edu.co

Abstract. The Internet of Things is one of the technologies with the greatest incursion and expansion in the services market, making it attractive to cyberattacks due to its various vulnerabilities, both in its protocols and in its implementation. This brings with its aspects that the industry and users must take into account to minimize the risk of suffering various types of attacks, compromising sensitive information in the process. In this sense, a study on the advantages and disadvantages of the IoT is shown, focusing on the protection of information based on its present flaws, which will eventually have to be taken into account for future developments that involve data management and administration through devices. smart. The methodology used is based on theoretical and quasi-experimental research represented in the skills and experience in ethical hacking applied to the corporate environment. In this way, the most representative flaws in terms of cybersecurity related to the IoT are exposed, so that they are attended by companies and personnel responsible for their security.

Keywords. Advanced persistent threats, botnet, cyberattacks, distributed denial of services, ransomware.

1 Introduction

The internet of things (IoT) refers to the set of electronic devices connected to the internet that are permanently recording and sharing data. Under this scenario, any electronic device (bioelectronic or nanoelectronic) including household appliances, office equipment, machinery, accessories attached to clothing and personal items that share data among themselves or a central, belong to the world of IoT.

For this new decade, the technology represented in the Internet of Things (IoT) with its various variants such as: Internet of Vehicles (IoV), Internet of energy (IoE), industrial IoT, artificial intelligence in IoT (IAoT), Internet of Things-Grid (IoT-G), Internet of Robotized Things (IoRT), IoT on the Battlefield (IoTotBF) [1], Internet of Wearables Things (IoWT), Internet of Medical Things (IoMT), etc., van to set the pace of societies in various contexts, whose objective will be framed in improving the quality of life of people, industry, cities and the environment through increased connectivity, navigability, monitoring, interaction and ubiquity, either in an environment urban as rural.

This brings with its new challenges in terms of security and regulations that not only guarantee the proper use of technologies, but also the algorithms that control them, in this particular case artificial intelligence (AI) through developments based on deep learning and machine learning [2].

The demand for new technologies, applications and IoT solutions in this new decade will be marked by digital health care and control, as well as the development of intelligent environments (which involve the labor field, recreation, transportation, home and study among others) that allow remote monitoring of any variable that implies improving a service in favor of people's well-being.

This requires high-speed network technologies equipped with new connectivity protocols that guarantee a higher speed rate than the current one, with low latency and protection against cyberattacks.

In addition to the above, connectivity will not only be represented in 5G, WiFi (including its latest variants such as WiFi 6) or LiFi technologies, but also in low-orbit satellite communication.

This represents a challenge, based on the fact that the space around the Earth is being increasingly saturated with satellites, which is why the market for services is expected to increase for this decade.

For example, the Starlink company with its plan to have 12,000 satellites, provides high-speed connectivity services worldwide with its scarce 1,000 satellites at the time of writing this article, in addition to other satellite fleets from several countries that intend to expand this service by placing hundreds of thousands of these artifacts into orbit.

With this in mind, it is clear that the societies of the 21st century are going to be permanently connected anywhere in the world, with multiple services according to the needs of each user and industry. Although it should be noted that this connectivity may be expanded to other worlds before half a century, such as the Moon and Mars, as planned by large aerospace industries.

2 Internet of Things and Vulnerabilities

The IoT is increasingly present in our daily lives, either at home or at work, being essential for monitoring variables such as temperature, humidity, flow rate, pressure, electrical conductivity, pH, measurement of heavy metals. in the air, access control, security, driving, purchase of items, vital signs, active biological pollutants, etc., all using the wireless internet as a means of communication.

This implies that the IoT presents specific functions for capturing data from the environment, which are then processed and stored to later analyze the convergent information for decision-making. This process demands the use of distributed networks adjusted to universal standards as those of each manufacturer, for which the devices make use of sensors and / or actuators in order to communicate with each other.

Another aspect to take into account about the IoT is that it is constantly evolving, with a moderate bandwidth demand that promises to increase with the incorporation of machine learning algorithms, expanding its services, giving way to the so-called Analytics on the Edge [3].

Also, the IoT being scalable allows several of its data capture and recording processes to be optimized; This affects the use of energy efficiently (remembering that they are wireless devices that work with batteries).In the same way, the IoT uses modern Web technologies for its connectivity and data transfer in real time, either to a local server or a central one in the cloud (servers/platform), so that its applications are extended to different fields using multiplatform mobile applications on a recurring basis.

Regarding the wireless connectivity of IoT devices, it is carried out through wide coverage networks such as LPWAN that include Sigfox, LORA and NB-IoT networks, where each one is characterized by working under different modulations and bandwidths. according to the own needs on which they work in each country [4].

At the security level, there is a great diversity of problems that IoT technology presents in terms of manipulating its operating environment, either through hardware or software, as summarized below:

1 Vulnerable wired and wireless network services due to factors such as: weak passwords that allow easy cracking, insecure password recovery systems, buffer overflow, outdated firmware with active back doors, API (*application programming interface*) problems in the devices and backend of manufacturers and third parties, DoS, DDoS, account locks and credential management, malware injection and replay and brute force attacks, unencrypted or improperly encrypted services, weaknesses in UDP services and protocols of UPnP (*Universal Plug and Play*) communication, inadequate payload verification and message integrity among many others.
2 Human factor: other weaknesses are attributed to the updating and authentication mechanisms whose responsibility on the part of the network administrator is critical. In this sense, failures are detected in updates that do not appear encrypted or with write permissions, or authentication is not enabled for firmware and patches, etc.
3 In the framework of authentication, there are variants that many administrators take as one.

For example, between IoT device to device, IoT device to mobile application, IoT device to cloud, mobile application to cloud, and web application to cloud. With any failure in any of these authentications, the entire network is compromised in terms of its privacy, being exposed to disclose the location and data of the user(s).

Now, as the IoT is in continuous expansion in the services market, it has also demonstrated its importance in the social, health and industrial framework, especially when there are already strong synergies with disciplines such as Big Data, artificial intelligence (AI), Cloud computing, artificial vision as a service (CVaaS), Edge computing, perimeter computing and blockchain among others, which have been laying the foundations for the development and consolidation of technologies such as Smart Systems (which involves Smart energy, Smart home, Smart buildings, Smart Cities, Smart transport, Smart Health, and Smart industry) and the IoT of Wearables (IoWR).

In the same way, the relationship between IoT and AI is growing, being incorporated into various devices, where household appliances are no exception, thus seeking to improve energy efficiency and the security of the data that circulates both through the devices and through the network to which they are connected to the home, building or industry.

Under this scenario, a problem to overcome consists of the incorporation of deep learning programs in the diversity microcontrollers of IoT devices, in particular memory chips, whose capacity is limited, hence the data is currently sent to the cloud that, of course, can be vulnerable to cyberattacks, if certain information security policies are not complied with. An advance in this direction is the TibyNAS algorithm, which as stated [5] "generates compact neural networks with the best possible performance for a given microcontroller, without unnecessary parameters", which is combined with the MCUNet system in charge of image classification locally, thereby reducing the risk of information theft.

This type of advance is crucial for future technologies, based on the fact that the number of household appliances, wearables and IoT devices implemented in people, homes, buildings, industry, transport and cities in general, is growing day by day, which is expected. exceed one billion devices, where limitations such as memory and processing capacity will be quickly overcome; This will demand a large amount of data storage and processing resources, therefore, Big Data in conjunction with disciplines of AI and data engineering will contribute their own in this regard.

3 Vulnerabilities in Protocols

The protocols that govern the IoT have been for many years one of many critical security weaknesses, which have demanded to be addressed in order to guarantee a standardized intra- and inter-device communication that has been partially solved. However, there are non-standardized protocols related to the IoT such as Zigbee, Z-wave, XBee, bluetooth, WiFi and LoRa among others, which work on the open-source stacks of TCP/IP protocols, which in turn present vulnerabilities in each structure since its creation, starting in most cases due to memory corruption. These flaws allow different types of malware, DDoS attacks and DNS record injection to be executed that expose the information to a cyberattack.

It is important to note that the TCP/IP model presents the entire structure on which a set of specific network protocols allows any equipment or nodes to communicate end-to-end, under specific addressing for both transmission-reception and routing between other technical aspects. The vulnerabilities of the TCP/IP protocols are exploited to carry out DDoS-type attacks according to the layer, so [6] classify them as follows:

a. Application: in this layer the most common attacks are: HTTP / HTTPS Flooding, FTP Flooding, Telnet DDoS, Mail Bombs, SQL Slammer and DNS Flood.
b. Transport: the attacks are of a volumetric type, understood in the sense that it is aimed at destroying networks, denying or consuming their resources until the server collapses. The most common DDoS attacks are: SYN Flooding and UDP Flooding and TCP Null Flooding.
c. Internet: attacks occur in this layer due to vulnerabilities inherent to the design of the

TCP/IP protocols. The most common attacks are Smurf (compromises the ICMP protocol), Fraggle, TearDrop and ICMP Flooding.

d. Access: Attacks exploit weaknesses in the network layer and its protocols. The most common DDoS attacks are: VLAN hopping, MAC Flooding, DHCP Attack, and ARP Spoofing.

Other vulnerabilities directly related to IoT systems are open-source TCP/IP stacks, which are not owned by a single company, these are: PicoTCP, uIP, FNET and Nut / Net, which are present in IPv6 protocols, DNS, mDNS, TCP, ICMP and LLMNR, all of them related to the communication of devices connected to the internet. A particular example related to TCP are DDoS attacks through Microsoft's Remote Desktop Protocol (RPD), taking advantage of UDP port 3389; Although a set of privileges is required, this type of attack cannot be ruled out even with patches and possible unauthorized access within a network.

As a complement to the above, there are vulnerabilities inherent to the operating systems and the architecture that the current Internet supports, which are connected to servers, computer equipment, IoT devices, access controls, etc., which is a real problem for the time to know the operational characteristics of the firmware and connectivity hardware of these elements to establish if they are at risk or not, added to the presence of bad practices in software development.

For example, there are online resources, specialized software or a simple script (https://github.com/Forescout/project-memoria-detector) that allow probing the ICMP protocol, TCP option signatures and handling of their flags to detect vulnerabilities and take corrective actions in order to prevent future attacks.

To finish this section, the Thread protocol [7] has recently been proposed, which is supported by the major industries of IoT technology through the Thread Group, designed to establish secure wireless IP connectivity without the need for concentrators or hub thanks to the use of IPv6 and 6LoWPAN standards. The Thread protocol uses encryption mechanisms to guarantee secure communication, even via Bluetooth.

In addition, this protocol guarantees the transparency of connectivity between devices, expanding a network if necessary, taking into account low energy consumption. With this in mind, as IoT devices and coverage increase, the problem of saturation and range of action is resolved respectively, in addition to the fact that the network adapts in case a device fails.

Therefore, it is expected that this protocol will be adopted by the entire industry, minimizing the compatibility problem of IoT technologies present today.

4 Geopolitics and IoT

The COVID-19 pandemic has taught society great lessons and one of them is that you cannot let your guard down in the face of an epidemic that can spread rapidly throughout the world. Mandatory confinement brought to the fore the fragility of the health sector in treating conditions, detecting and treating chronic diseases, due in part to the fact that clinical procedures were planned to be developed in hospital facilities and laboratories.

The IoT proposal based on the above, was to increase the number of devices implemented at home, work and even the patient's body, in such a way that monitoring is carried out in situ, controlling the patient's health status, minimizing risks associated with a disease getting out of control. With this new monitoring perspective, the industry and health companies have begun to massively introduce the IoT in their facilities and homes of workers and patients respectively, seeking to improve their safety and care. In this sense, it seeks to improve the efficiency of the resources used within a company, for example, using intelligent lighting systems, intelligent energy and environmental control, security systems and monitoring in areas of low and high traffic among others, which guarantee the well-being of the worker.

However, although the interest in using the IoT has increased, so has the risk of compromising sensitive information to third parties, due to the vulnerabilities inherent to this technology, added to the geopolitical instabilities that have opened a gap to cybercrime in recent years for carry out attacks of various types.

It is evident that technological ubiquity and dependence on it draws attention to cybercrime, especially under the global economic and geopolitical uncertainty that apparently will remain in this way for some time to come.

Thus, advanced persistent threats (APT) [8], ransonware and DDoS attacks will be the common denominator for the next few years.

These cyberattacks have various connotations: economic, personal, corporate, political and geopolitical, just to name a few. In the particular case of geopolitics, it is marked by new technical and technological innovations that seek that particular target (diplomats, governmental and non-governmental organizations, health centers, research centers, universities, etc.) fall into digital traps, downloading files corrupt or malicious attachments in order to steal and / or hijack your systems.

In the case of cyberattacks directed at government platforms, defense and technology companies and critical infrastructures (understood as: hospitals, transport and energy sectors, roads, bridges, tunnels, airports, seaports, public services, buildings, etc.), they seek control your computer networks through the creation of chains of infection, phishing and use of legitimate services, making it almost impossible to take corrective action in this regard.

The problem with this type of cyberattack is that its source points to Russian, Chinese, Iranian and North Korean organizations (a fairly complete list can be consulted in this regard SINCE 2016 in CSIS) [9]. In this type of scenario, there are other cyber-piracy groups from countries such as Vietnam [10, 11], which, through their social networks, spread all kinds of malware and phishing with political objectives and the interests of the Vietnamese government, for example, property theft. intellectual and cryptocurrency mining. These types of cyberattacks seek the collection of confidential business information that is sold to the competition. The modus operandi is the sending of messages to the holders of emails with directing to false pages, this implies the use of techniques such as social engineering, spear-phishing [12] and pharming [13] among others.

It should be noted that techniques to bypass the security of a physical and logical system are permanently refined, for example, dropping PE (*Portable Executable*) binaries to load advanced malware, combined with basic or low-tech techniques.

The goal is to control the victim's operating system in such a way that it is not easy to reinstall it or even replace the hard drive. Another way to increase the success rate of attacks using phishing-type malware is through the implantation of emails directly to the entrance of the victim's mailbox, using tools such as Email Appeder. With this system, email security is circumvented, because the incoming email credentials are valid, so once approved it connects to the victim's email accounts through the IMAP (*Internet Message Access Protocol*) protocol, which is responsible for receiving messages from a mail server. Once this step is completed, the cybercriminal customizes the messages so that they are credible and the victim accepts them, opens and enters personal or corporate information. The problem with this type of attack is that it is new, with a high degree of effectiveness, so the risk of being attacked under this advanced phishing scheme should not be underestimated, especially in companies and industry in general.

Another type of malware recently discovered, shows the new generation of computer worms that IoT systems and operating systems will have to deal with in the coming years, called Gitpaste-12. This type of malware used GitHub and Pastebin to store the code of its components and host 12 different attack modules to attack different vulnerabilities. With these characteristics, it has the ability to propagate progressively in a corporate network emulating what a botnet would do, but internally, compromising devices such as routers, firmware and operating systems, using exploits, which then proceeded to execute a dynamic script with in order to download and run the other components of Gitpaste-12, which were constantly updated and at the same time disabled the security protocols.

It also includes the rules of firewall devices, software for monitoring and prevention of attacks, the apparmor module (this module belongs to the Linux kernel that allows restricting some processes of certain programs as an administrator) and commands related to access security. Cloud. This implies that this type of threat aims to have access and control of the infrastructure that connects and

manages cloud computing and, therefore, the data of all IoT devices and other connected systems.

To make matters worse for the victim, this worm runs a cryptominer, the objective of which is to hijack the idle processing of the network and extract cryptocurrencies for other fraudulent attacks, either to other external services such as the victim's clients.

This type of attack is perverse, since it not only has access to all the victim's data, but also uses its own infrastructure to attack others, preventing the network administrator from collecting information on those processes that are running by blocking certain instructions, such as: readdir, tcpdump, sudo, openssl, /proc, etc. Finally, the Gitpaste-12 worm at the time of discovery contained a library that downloaded and executed Pastebin files that contained more malicious code.

What this type of malware hints at is that they are sophisticated programs designed to deal with new developments at the security level of operating systems and firmware of devices connected to a network, selectively attacking the IP addresses contained within a network. random range of classless interdomain routing CIDR (classless interdomain routing), execution of scripts to open certain ports such as 30004 related to the Transmission Control Protocol (TCP) and port 30005 related to the bidirectional SOAP / HTTP protocol, in charge communication between router devices or network switches, as well as automatic configuration servers.

In conclusion, the emergence of new worms with botnet characteristics will increase in the coming years (for example, the Golang worm), with the mitigation that they will be combined with other intrusive techniques such as crypto mining to attack servers with Windows operating systems and Linux, cloud systems with exploits that link ransomware, APT and DDoS with all their variants [14].

Governments and industry have begun to take action on the matter, however, not only cybercrime is taking a step forward, but also organizations sponsored by the State itself, contracted to carry out targeted attacks on organizations and industry from other nations [15, 16].

5 Ransomware Attacks

Ransomware is a type of cyberattack characterized by hijacking information from a system by encrypting it, and then charging the victim for its ransom with a fixed term, through payment by means of electronic currency such as bitcoin.

By not acceding to these claims, cybercriminals proceed to delete the information, auction it or publish it on filtering sites on the darknet so that other criminals can appropriate it and thereby perpetuate the scam. A recent method of pressuring the victim to pay when she refuses is by harassing and coercing her through intimidation that ranges from disclosing exfiltrated information to threatening the lives of employees and their families. This cyberattack panorama has shown a constant evolution in the last decade, both in the form of attack and in the means of pressure exerted by the payment to release the release, as evidenced by the most known ransomware-type malware worldwide such as They are: Wannacry, Bad rabbit, Peya, Spora, Reveton, Doxware, Locky, Zcrypt, Goldeneye, Cryptowall, Emotet, jigsaw and Marozka, among many others [17, 18, 19].

In 2020, apart from the pandemic that health and research centers, schools, universities, the government sector and non-profit charities had to deal with, ransomware-type attacks were added. The reason for these cyberattacks was that, by hijacking a critical computer network for these institutions, the probabilities that they will pay the ransom to recover their services are high.

Although initially the ransomware groups promised not to attack these institutions, the ease of obtaining money from information hijacking was and continues to be high. In this context, there were specific cases of “altruistic” actions where cybercriminals donated part of the loot to charities and non-profit organizations, which, of course, does not exempt them from their crime and, therefore, this money was confiscated by the authorities.

Humanitarian reasons do not apply to this type of cyberattack, "if you allow others to access your information, you pay for it", it does not matter if the lives of patients are compromised, there are no scruples, as evidenced in different hospitals

around the world [20, 21], where countries such as the United States [22], Great Britain [23], France, Asia, Europe [24, 25] and the Middle East have been the hardest hit.

Other reasons why the health sector is attacked lie in the fact that its IT infrastructure is weak or with management and administration processes that range from basic to non-existent.

Most clinical devices are networked, such as CT scanners, monitors, radiodiagnostic equipment, etc., which act as weak link points by transmitting data in an insecure way. What is critical about this situation is that millions of patient data are compromised, whose information flows online that can be accessible to those who know how to search them.

In general, disruptive ransomware campaigns are regularly created by cybercriminals, aimed at exploiting weaknesses in certain systems, for example, credentials associated with databases, in particular MySQL and PostgreSQL. The disturbing thing about this situation is that more and more systems are compromised and the worst of the case as [26] points out “as a reminder and warning to those who do not pay for the ransom, more than 250,000 databases of 83,000 MySQL servers and 77 terabytes of leaked data”.

Although the health sector is mentioned as a target for ransomware, it is not the only one, since sectors such as pharmaceuticals, finance, education, transportation and even cybersecurity and technology-based companies are lucrative sources for organized criminal groups.

To carry out a ransomware attack on this type of infrastructure, techniques such as phishing and advanced software such as Ryuk and TrickBot [27, 28] are often used, characterized to collect credentials and filter specific data. Similarly, ransomware has been combined with advanced persistent threat systems (APT) [8] to enhance damage to the interior of a system, either through monitoring the information traffic that circulates through it and stealing confidential data. to sell them to the competition or kidnap them for later payment.

One aspect to reflect on this issue are the attacks on the industry that depend directly on the use of controllers or PLCs for their production processes that, although they present security protocols, are not exempt from being hijacked by these devices with the corresponding consequences, where APTs and now ransomware become the Swiss army knife to carry out targeted attacks with irreparable damage to a logical as well as a physical system.

Attacks of this type are often selective and require time and planning; This is partly due to the fact that a reasonable amount of time is required to know the environment in which the victim moves and then to escalate privileges in the system to capture the greatest amount of information. In the particular case of the IoT, it allows permanent monitoring of what is done within an organization or in the worst case in a city, to have access to its monitoring devices. In this particular case, smart cities in the coming years will increase the technologies related to the IoT, which will not only be aimed at monitoring critical infrastructures, security, transport and health care, but also at the early detection of viral vectors with a view to minimize pandemic risks.

How can you deal with this kind of problem? The answer is to improve the physical and logical protocols and information security, which, of course, is not easy, either because human error is permanently present or because of the progressive advances in cyberattacks, of which no one takes for granted. found out until it's too late. This is where artificial intelligence comes into play, assuming a leading role in automatic decision-making for defense and attack, which will be essential for the coming years. Similarly, if it happens that the system has been compromised, it is advisable not to pay for the demands of cyber attackers. In this way, when the monetary supply is cut, it does not have the objective of hijacking a system, although this is easier said than done, because there are various commercial, corporate, personal, financial and political motives and interests that lead to pay and even shut up and deny that ever there was such an attack and pay for it.

6 Botnet and DDoS

Botnets or zombie networks are defined as a set of computer networks infected by malware, which allows the execution of their inactive processing, in order to increase the computational power to

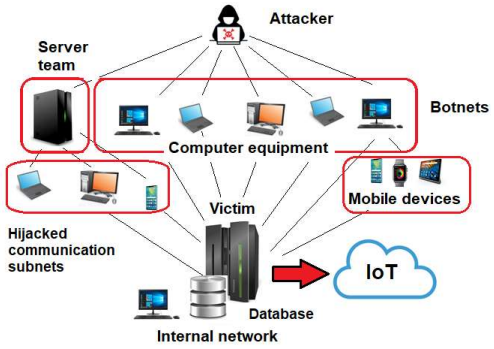


Fig. 1. Representation of a DDoS distributed denial of service attack, where botnets are responsible for spreading malware to other networks and computers by infecting them, then attacking the victim with hundreds of thousands of requests

attack other systems by brute force or through the techniques Denial of Service (DoS) and Distributed Denial of Service DDoS attack. The basic difference between DoS and DDoS lies in the number of infected computers that progressively and repeatedly make requests to a victim system.

This implies that this type of attack demands a set of computer equipment networks infected with malicious software from different sources that facilitate its control, allowing the sending of spam and spreading other types of malwares to continue to progressively infect other networks.

Regarding a DDoS-class attack, [29] points out that: "it consists of a massive attack that seeks to congest the server of a target or consume the entire Internet outgoing bandwidth of the victim's organization." Either of the two attacks makes a network useless, making it collapse for the benefit of the attacker, giving way to infecting and scaling the system, taking over the equipment and information that are available there.

Figure 1 shows in a general way the hierarchical structure of a DDoS and Bonet attack, where the attacker uses a set of previously infected networks, servers and computers, which act as a zombie network or bots in charge of distributing malware to the victim through requests, in order to carry out the DDoS-type attack.

However, this type of attack does not exclude the involvement of any vulnerable mobile technology available, connected to a network such as smartphones, tablets, wearables and of course the IoT.

An inherent characteristic of botnets is that their level of attack is continually technified and diversified, making it almost impossible to eradicate them, an example of this is the InterPlanetary Storm botnet, which detects and evades the security systems of computer networks, whose system operating can be Mac or Android. Another type of botnet is FritzFrog [30] that uses peer to peer (P2) communication to attack SSH servers and Hoaxcalls, facilitating large-scale attacks. This scenario predicts what industry and governments will have to deal with in this decade.

The particularity of botnets is that they are designed and/or rented at the service of the highest bidder to carry out multiple criminal activities that include: DDoS attacks, command execution, sabotage and industrial espionage among others. Any vulnerability that a network system presents will be used by cybercrime for unauthorized access. Examples of the risk that IoT devices expose to Botnet and DDoS attacks are related to the activation of unnecessary or insecure network services that facilitate unauthorized access and control of any service, violating confidentiality, integrity, authentication and/or availability of the information.

There are risks associated with interfaces in charge of managing proprietary or third-party devices, for example, mobile applications, data repositories in the cloud and even the corporate website and the backend APIs (typical of application programming). All these flaws lead to vulnerabilities such as the implementation of weak encryption (or the absence of it) on the data that circulates through the network, as well as the absence of input/output filters.

Other flaws found in IoT devices that can be exploited for unauthorized access are: outdated firmware that lead to the lack of management of encryption processes in transit and validation of updates without appropriate mechanisms for this; use of insecure or outdated software components and libraries; inappropriate use of personal information stored on a device whose degree of security is questionable in addition to the absence of formal permission or informed consent; absence of data encryption and access control.

All these failures converge to the lack of management and administration of IoT devices

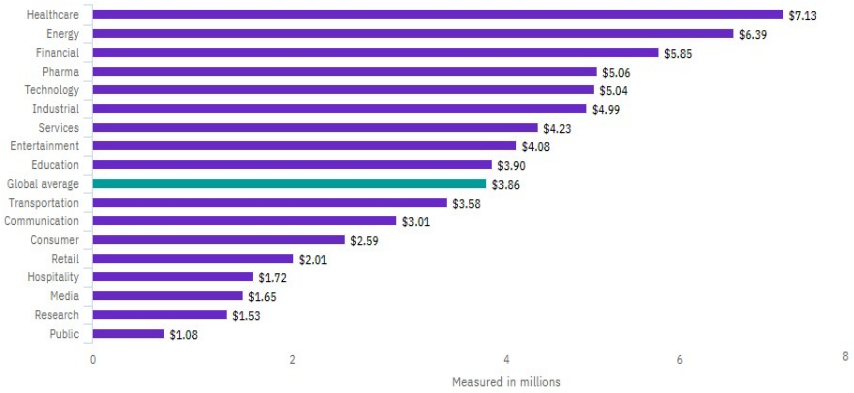


Fig. 2. Average total cost of a data breach by sector [33]

attributed to human errors, which do not comply with the corporate information security standards and/or policies dictated by national and international entities such as the ISO 27000 family of standards. [31, 32].

7 Discussion

The risk of increasing cyberattacks in the coming years is high, this is due to the simple fact that there will be millions of servers around the world that will be compromised due to security failures, where the IoT with its different variants present greater vulnerabilities for the industry and society in general, whose dependence on technology grows day by day. Likewise, the medium-term attack methods are designed to counteract standard security tools and forensic examinations, which entail new advanced malware models that in most could have integrated algorithms based on artificial intelligence (AI).

In this sense, both the way of attack and defense of computer systems must evolve to new levels that will demand significant resources, especially for governments, industry and technology-based companies.

One aspect to take into account is the direct and indirect cost of data breaches, which can take years to correct within an organization. In fact, data leakage and security incidents can lead to the disappearance of an organization, either due to the payment of ransom for the hijacked information, loss of customers and businesses due to bad publicity, system inactivity, detection time and contention of a cyberattack, loss of share value (for those companies that are listed on the stock exchange), demands that this entails due to the exposure of the personal data of thousands or millions of users and non-compliance with the regulations in this regard; that carries regulatory fines that can run into the millions of dollars.

For example, a report by IBM and the Ponemon Institute [33] showed that the average cost of a data breach in 2020 was $ 3.86 million on average. This situation shows that the sector that was hit the hardest was the health sector with a value of more than 7 million dollars, followed by the energy and financial sectors, as can be seen in figure 2.

Based on these facts, it is clear that for the next few years the outlook will not change much, this in part due to the COVID-19 pandemic, which acted as a trigger for cyberattacks to be focused on the health and corporate sectors. However, although the vaccine already exists, preventive isolation and remote work from home will continue for some time to come [34, 35] and, consequently, cyberattacks will be concentrated in these sectors; This suggests that the costs of data breaches may be increased if no action is taken on the matter with regard to cybersecurity.

For example, at the end of 2020, cyberattacks were accentuated in agencies in charge of issuing authorizations for several vaccines against the coronavirus, in particular the European Medicines Agency (EMA). This poses a serious cybersecurity problem for years to come, not just for vaccines being sold on the black market, but for any other essential medicine, where cybercrime has found a near inexhaustible source of profit for profit.

Another aspect to consider are those cyberattacks aimed at stealing the login in the most commercial browsers; this in order to distribute malware aimed at fraud and credential theft. Although current browsers have a high level of security, they are not exempt from advanced cyberattacks that seek to modify DLLs or insert polymorphic malware in cookies and pop-up pages, among others, which are expected to evolve this decade.

Most of the failures mentioned can be solved through corrective actions within a corporate network, such as: firmware update, installing patches issued directly by the provider, encrypting devices with their own passwords and not leaving the factory default one, two-factor authentication (*password + code or 2FA key*), disabling non-essential services of protocols such as IPv6 and IPv4 +, configuration failures in the web interface, configuration of devices to work under internal DNS servers, monitoring of packet traffic in the network for network abnormalities, strong encryption and access controls.

Cyberattacks evolve and therefore, the measures and protection of information must be improved, which is not unnecessary to remember, it is the most important asset of any organization.

There are solutions on the market that minimize the risk of cyberattacks, such as Microsoft's Azure Defender for IoT, which integrates third-party information technology security tools, in such a way that it allows it to work with different devices from recognized IoT providers. Another solution proposed by Amazon Web Services is AWSIoT and AWSIoT Core for public and private networks of low power and wide area LoRaWAN (Low Power Wide Area Network), designed to improve connectivity and security in IoT devices connected to the AWS cloud.

In the case when the crime has been committed and a ransom is requested for the information, it is advisable not to pay and notify the authorities, since, in doing so, what is achieved is to perpetuate the intrusions and maintain these criminal actions that are every more lucrative, due to the fact that it is paid to prevent confidential information from being published, which in many cases, despite being paid, is published on filtering sites combined with the so-called double extortion.

These two modalities are recent; the first is used as additional pressure on the victims to pay, publishing information on those companies or government agencies that have refused to pay, and the second consists of publishing some data on the darknet so that the victim can pay and see that they are talking seriously. In this regard, there are nations that prohibit the payment of these ransoms under penalty of heavy sanctions. Employee training and awareness is a critical part of a company, since it is enough for a single employee to violate safety standards and / or policies to compromise the entire system. Now, in the event that the cyberattack has been completed, it is advisable to establish contingency plans for technical incidents and commercial recovery, which are supposed to have been previously designed to deal with this type of scenario.

Based on the above, various government agencies and the technology industry are looking for plausible solutions that make it possible to nullify or at least minimize the negative impact of cyberattacks and cyber espionage on IoT systems.

For example, the ETSI (*European Telecommunications Standards Institute*) is a European body that launched the security standard ETSI TS 103 645 [36]; which includes data protection and security in household appliances and consumer devices such as smart cameras, access controls, wearables and consumer systems that include IoT gateways, base stations and hubs, portable devices, home automation systems, connected gateways, door lock and window sensors.

The objective of this standard and others under construction are aimed at unifying criteria that facilitate both organizations and nations to keep a strict control of the IoT devices that come out and circulate in the market. Although common agreements need to be defined at the global level that allow defining and assigning responsibilities and ownership in matters of security and data management, it is a matter of time before they reach an agreement, this in part due to the boom in the development and implementation of new technologies. IoT for the next few years.

On attacks focused on the capture of information stored in the cloud, it is the holy grail of cybercrime, because they would have control of all the information of an organization and, therefore,

all its assets would be compromised, with a devastating impact on the services provided, ranging from the extortionate payment of large sums for releasing the information, to the periodic payment of sensitive information from both the company and its clients, even using various subterfuges to apply them to the latter.

However, despite concerns at the security level, for the present decade the increase in networks composed of intelligent IoT devices and variants thereof will be even greater, involving various emerging technologies to expand their services, such as those where not only they manage data packages, but rather energy packages through the different nodes of the network, who will be in charge of calculating the most optimal route to their destination. This new economic environment opens up new business opportunities for the services industry, energy distribution and smart monitoring of cities, homes and people, and unfortunately new opportunities for cybercriminals and political destabilizing cyber groups, not to take action on the matter for part of the competent authorities.

7 Conclusions

Information security for the next few years will be increasingly compromised by continuous advances in computer systems and advanced algorithms. This has serious implications on the risk of compromising sensitive data to criminal organizations or states interested in profiting from the vulnerabilities of others, or destabilizing the economy of their counterparts. The truth of all this is that companies must be better prepared to deal with this type of scenario. The rule is simple, a company that does not invest in cybersecurity is doomed to disappear. It goes without saying that the regulations and fines for breach of data protection are leading the industry and service companies to take this issue seriously, because there is not only the pecuniary punishment, but also the reputation and potential lawsuits to which it is expose for not complying with the law in this regard.

In the case of IoT, things are not going well, because the above applies equally to companies that trade and use IoT devices.

It is advisable to take into account the use of encryption in an expansive way, AES-XTS [37] block encryption for Flash drives and secure boot based on the RSA algorithm and/or variants, automate security thereby minimizing human risk, establish business continuity plans and decoy teams, permanently train employees, perform online and offline data backups, use of blockchain as a system to monitor transaction and data processing, among other aspects with a view to minimizing risk.

It is undeniable that the security factor is fundamental and its relevance in any communication system cannot be overlooked, which will demand new developments in hardware as well as in software; so much so that the collection of personal data by IoT devices will increase in the coming years, forcing the adoption of new encryption techniques such as homomorphic cryptography [38] and protocols that involve confidential computing with security level 4 [39], guaranteeing the user that their data is safe.Also, the incorporation of applications related to artificial intelligence (IAoT) is contemplated, ranging from predictive learning, through interactive audio and voice systems to monitoring and in situ human-device interaction, for example, autonomous vehicles, civil drones and military, clinical monitoring, logistics and traceability among others

For this decade, the modalities of malware and cyberattacks will continue to evolve, so it is necessary to prepare for it. Ransomware campaigns will become increasingly aggressive, with higher ransom demands, although the attacks are concentrated in organizations, it does not imply that an ordinary citizen is exempt from it, it all depends on the degree of interest of the cybercriminals or who I hired them. Coordinated attacks can be more effective and lucrative, therefore, it is advisable not to lower your guard and be vigilant, not only the personnel in charge of the systems and networks, but also of each employee, since not only corporate information is being compromised but the information of the staff and even their families.

It is important to anticipate what is coming for this decade in terms of emerging technologies such as 5G networks (including 6G), whose implementation has begun and has also begun to

show weaknesses in terms of security, such as user location and theft of data, opening countless opportunities to hijack, scale a system and steal information on a massive scale through attacks on certain protocols and DDoS and APT attacks. It is worth mentioning in this regard that, with services focused on the mobile consumer, security, trust, convenience and ubiquity are factors to consider under current standards and future communication technologies.

To conclude, with the current geopolitical dynamics that in the future show that the tensions between the great superpowers will increase even more, the ransomware, APT and DDoS attacks will be focused on attacking critical infrastructures and industry of a nation.

In this sense, the industrial sector and governments must anticipate this type of attacks, requiring seeing data security as an investment that demands ideal technical and technological resources, as well as designing and implementing incident response plans, enforcing regulations in cybersecurity among many other aspects.

References

1. **Márquez, D. J. (2019).** Riesgos y vulnerabilidades de la denegación de servicio distribuidos en internet de las cosas. Revista de Bioética y Derecho No. 46, pp. 85–100. DOI:10.1344/rbd2019.0.27068.
2. **Lin, J., Chen, W., Lin, Y., Cohn, J., Gan, C., Han, S. (2020).** MCUNet: Tiny deep learning on IoT devices. Advances in Neural Information Processing Systems 33 NeurIPS´20, DOI: 10.48550/arXiv.2007. 10319.
3. **Harth, N., Anagnostopoulos, C., Pezaros, D. (2018).** Predictive intelligence to the edge: Impact on edge analytics. Evolving Systems Vol. 9, pp. 95–118. DOI: 10.1007/s12530-017-9190-z.
4. **Mekki, K., Bajic, E., Chaxel, F., Meyer, F. (2018).** A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express. DOI: 10.1016/j.icte.2017.12.005.
5. **Ackerman, D. (2020).** System brings deep learning to “internet of things” devices. MIT News on campus and around the Word. https://news.mit.edu/2020/iot-deep-learning-1113
6. **Acharya, S., Tiwari, N. (2016).** Survey of DDoS attacks based on TCP/IP protocol vulnerabilities. IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 18, No. 3, pp. 68-76. DOI: 10.9790/0661-1803046876.
7. **Sistu, S., Liu, Q., Ozcelebi, T., Dijk, E., Zotti, T. (2019).** Performance evaluation of thread protocol based wireless mesh networks for lighting systems. International Symposium on Networks, Computers and Communications (ISNCC), Istanbul, Turkey, pp. 1–8. DOI: 10.1109/ISNCC.2019.8909109.
8. **Márquez, D. J. E. (2017).** Armas cibernéticas. inteligencia artificial para el desarrollo de virus informáticos letales**.** Revista Ing.USBMed, Vol. 8, No. 2, pp. 48-57. DOI: 10.21500/20275846. 2955
9. **CSIS (2021).** https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents
10. **Luoung, H. T., Phan, H. D., Chu, D. V., Nguyen, V. Q., Le, K. T., Hoang, T. L. (2019).** Understanding Cybercrimes in Vietnam: from leading-point provisions to legislative system and law enforcement. International Journal of Cyber Criminology, Vol. 13, No. 2, pp. 290–308. DOI: 10.5281/zenodo.3700724.
11. **Baezner, M. (2018).** Hotspot analysis: use of cybertools in regional tensions in Southeast Asia. Zurich: Center for Security Studies (CSS). Cyber operations in the gray zone, 27.
12. **Bullee, J. W., Montoya, L., Junger, M., Hartel, P. (2017).** Spear phishing in organisations explained. Information and Computer Security, Vol. 25, No. 5, pp. 593–613. DOI: 10.1108/ICS-03-2017-0009.
13. **Ortiz, C. N. J. (2019).** Normativa Legal sobre Delitos Informáticos en Ecuador. Revista Científica Hallazgos21, Vol. 4, No. 1, pp. 100–111.
14. **Márquez, D. J. E. (2020).** Internet of things and distributed denial of service as risk factors in information security. Chapter 19 Bioethics in Medicine and Society, DOI: 10.5772/ intechopen.94516.

15. **Associated Press. (2021).** Suspected Russian hack fuels New US action on cybersecurity. https://www.voanews.com/usa/suspected-russian-hack-fuels-new-us-action-cybersecurity
16. **Sanger, D. E., Perlroth, N. (2020).** U.S. to accuse China of trying to hack vaccine data, as virus redirects cyberattacks. https://www.nytimes.com/2020/05/10/us/politics/coronavirus-china-cyber-hacking.html.
17. **Connolly, Y. L., Wall, D. S. (2019).** The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures. Computers & Security, No. 87, 101568. DOI: 10.1016/j.cose.2019.101568.
18. **Maurya, A. K., Kumar, N., Agrawal, A., Khan, R. A. (2018).** Ransomware: Evolution, Target and Safety Measures. International Journal of Computer Sciences and Engineering, Vol. 6, No. 1, pp. 80–85. DOI: 10.26438/ijcse/v6i1.8085.
19. **Brewer, R. (2016).** Ransomware attacks: detection, prevention and cure. Network Security, Vol. 2016, No. 9, pp. 5–9. DOI: 10.1016/s1353-4858(16)30086-1.
20. **Harkins, M., Freed, A. (2018).** The Ransomware Assault on the Healthcare Sector. Journal of Law & Cyber Warfare, Vol. 6, No. 2, pp. 148-164.
21. **Collier, R. (2017).** NHS ransomware attack spreads worldwide. CMAJ: Canadian Medical Association journal (journal de l'Association medicale canadienne), Vol. 189, No. 22, E786–E787. DOI:10.1503/cmaj.1095434.
22. **Branch, L.E., Eller, W.S., Bias, T.K., McCawley, M.A., Myers, D.J., Gerber, B.J., Bassler, J.R., (2019).** Trends in Malware Attacks against United States Healthcare Organizations, 2016-2017. Global Biosecurity, Vol. 1, No. 1, pp. 15–27. DOI: 10.31646/gbio.7.
23. **Argaw, S. T., Troncoso-Pastoriza, J. R., Lacey, D., Florin, M.-V., Calcavecchia, F., Anderson, D., Flahault, A. (2020).** Cybersecurity of hospitals: Discussing the challenges and working towards mitigating the risks. BMC Medical Informatics and Decision Making, Vol. 20, No. 1. DOI: 10.1186/s12911-020-01161-7.
24. **Yeo, J., Vander Ende, R. (2017).** Cyber evolution. En Route to Strengthening Resilience in Asia-Pacific. FireEye.
25. **European Commission. (2020).** Cybersecurity. Our digital anchor a European perspective. Publications Office of the European Union.
26. **Johnson, D. B. (2020).** New ransomware campaign exploits weak MySQL credentials to lock thousands of databases.
27. **Unterfingher, V. (2020).** Ryuk ransomware – untangling a convoluted malware narrative.
28. **Gittins, Z., Soltys, M. (2020).** Malware persistence mechanisms. 24th international conference on knowledge-based and intelligent information & engineering systems. Procedia Computer Science Vol. 176, pp. 88–97. DOI: 10.1016/j.procs.2020.08.010.
29. **Astudillo, B. K. (2019).** Hacking ético. Tercera edición, Ed. Ra-ma.
30. **Anonymous news (2020).** Massive new botnet discovered. Computer Fraud & Security, Vol. 2020, No. 9, p. 3. DOI: 10.1016/S1361-3723(20)30092-0.
31. **Baena, G. R., Mendoza, M. R., Joel, C. E. (2019).** Importancia de la norma ISO/EIC 27000 en la implementación de un sistema de gestión de la seguridad de la información. Revista contribuciones a la Economía, pp. 1– 13.
32. **MinTIC (2016).** Guía para la Implementación de Seguridad de la Información en una MIPYME (Norma No. 1.2).
33. **IBM Security (2020).** Informe sobre el coste de una brecha de datos. Madrid, España, IBM, España, SA.
34. **International Labour Organization (2020).** An employers' guide on working from home in response to the outbreak of COVID-19. Geneva: International Labour Office. www.ilo.org/publns
35. **Clifford, S., Quilty, B. J., Russell, T. W., Liu, Y Desmond-Chan, Y. W., Pearson, C. A. B., Eggo, R. M., Endo, A., Flasche, S., Edmunds, W. J. (2020).** Estrategias para reducir el riesgo de reintroducción del SARS-

CoV-2 de viajeros internacionales. MedRxiv´20. DOI: 10.1101/2020.07.24.20161281.

36. **ETSI (2020).** CYBER; cyber security for consumer internet of things: baseline requirements. ETSI EN 303 645 V2.1.1. European Standard.
37. **Luo, C., Fei, Y., Ding, A., Closas, P. (2019).** Comprehensive side-channel power analysis of XTS-AES. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 38, No. 12, pp. 2191–2200. DOI: 10.1109/TCAD.2018.2878171.
38. **Zhao, E, M., Geng, Y. (2019).** Homomorphic encryption technology for cloud computing. Procedia Computer Science, Vol. 154, pp. 73–83. DOI: 10.1016/j.procs.2019.06.012.
39. **FIPS 140-3 (2019).** Federal information processing standards publication (Supersedes FIPS PUB 140-2). Security requirements for cryptographic modules. Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8900. DOI: 10.6028/NIST.FIPS.140-3.
40. **Turjman, F. A. (ed.) (2019).** Artificial intelligence in IoT. Transactions on Computational Science and Computational Intelligence. Springer.

Article received on 01/04/2021; accepted on 23/07/2022.
Corresponding author is Jairo Eduardo Márquez Díaz.

ISSN 2007-9737

Tunisian Dialect Agglutination Processing with Finite Transducers

Roua Torjmen[1], Kais Haddar[2]

[1] University of Sfax,
Faculty of Economics and Management of Sfax,
Tunisia

[2] University of Sfax,
Faculty of Sciences of Sfax,
Tunisia

rouatorjmen@gmail.com, kais.haddar@yahoo.fr

Abstract. The agglutination is a widespread phenomenon in the formation of words in the Tunisian dialect. So, it requires a treatment at the morphological level. In this context, we present our way of dealing with this phenomenon for the different grammatical categories in Tunisian dialect. The proposed method is based on the construction of morphological grammars using a set of finite state transducers. These transducers offer great flexibility in the construction of morphological grammars and allow their maintenance and reuse. Our goal is to create a set of linguistic resources allowing the treatment of the phenomenon of agglutination in the Tunisian dialect. The NooJ linguistic platform with new technologies makes it possible to elaborate and experiment our constructed resources. The obtained results are ambitious and highlight our proposed method.

Keywords. Agglutination phenomenon, finite transducer, morphological grammar, Tunisian dialect.

1 Introduction

The agglutination is a widespread phenomenon in the Tunisian Dialect (TD). This reason reveals the importance of this linguistic phenomenon and makes its treatment a necessity. The construction of morphological grammars by relying on a set of finite transducers solves the treatment of agglutination and facilitates lexical analysis. Furthermore, finite transducers provide flexibility for maintenance and reuse.

In addition, the treatment of agglutination via a morphological analyzer allows the recognition of TD words and their different parts. Besides, this morphological analyzer can be integrated in many applications such as automatic annotation of TD corpus, POS-Tagging, TD speech synthesis and automatic translation from TD to Modern Standard Arabic (MSA) and vice versa.

The processing of TD raises other issues, such as the lack of standard spelling because it is never taught in educational institutions. In addition, there are dialectal differences from one city to another. This diversity creates the appearance of non-existent letters in MSA and the existence of words of different origins: French, Maltese, Turkish, etc.

Moreover, the lack of linguistic resources in language platforms like NooJ for TD exacerbates the problems of building robust tools and applications. In this paper, our principal objective is to build morphological grammars based on finite transducers that allow the processing of agglutination in TD.

In order to achieve this goal, we carry out a linguistic study on the phenomenon of agglutination. Subsequently, we need to implement a set of linguistic resources using the NooJ linguistic platform. The present paper is divided into six sections.

Table 1. IOC and DOC in Tunisian Dialect

Persons	DOC	IOC
1st singular person	ني *'nii'* (me)	لي *'lii'* (to me)
1st plural person	نا *'naa'* (us)	لنا *'lnaa'* (to us)
2nd singular person	ك *'k'* (you)	لك *'lik'* (to you)
2nd plural person	كم *'kum'* (you)	لكم *'lkum'* (to you)
3rd masculine singular person	هو / و / ه *'huu'* (him/it)	لو *'luu'* (to him/it)
3rd feminine singular person	ها *'haa'* (her)	لها *'lhaa'* (to her)
3rd plural person	هم *'hum'* (them)	لهم *'lhum'* (to them)

In the second section, we present related work dealing with the morphological analyzer for the Arabic dialects and MSA. In the third section, we exhibit our in-depth linguistic study. In the fourth section, we explain our linguistic resources related to the phenomenon of agglutination by explaining the designed dictionary and grammars.

In the fifth section, we experiment and evaluate our constructed grammars and dictionary. Finally, our paper ends with a conclusion and some perspectives.

2 Related Work

The agglutination treatment is done by the construction of morphological analyzers. In what follows, we introduce some works dedicated to MSA and Arabic dialects. Numerous works deal morphologically with MSA such as the Buckwalter Arabic Morphological Analyzer (BAMA) [1], the Standard Arabic Morphological Analyzer (SAMA) [9], the morpho-syntactic analyzer Alkhalil [8], the tool for morphological analysis and disambiguation MADAMIRA [10] and the Arabic morpho-syntactic analyzer using the NooJ linguistic platform [4, 7].

On the one hand, other works consider the approach that treats Arabic dialects using tools designed for MSA. The work of [12] which leans on SAMA and BAMA analyzers to recognize the prefixes and suffixes of the Egyptian dialect. Besides, the authors [5] have developed an Algerian morphological analyzer based on the BAMA and Al-Khalil.

Moreover, the Analyzer for Dialectal Arabic Morphology (ADAM) [13] allows the morphological processing of three dialects (Egyptian, Levantine and Iraqi). These three dialects have many similar morphological characteristics such the negation verbs, propositions and indirect object complements.

On the other hand, other works have chosen another approach that works directly on the dialect. Among these works, the authors [11] have created an a Moroccan dialect electronic dictionary (MDED) in order to develop a Moroccan morphological analyzer.

Besides, the authors [2] have proposed a machine learning method to extract Egyptian morphological lexicons from morphologically annotated corpora, such as inflection classes and associated lemmas.

Concerning TD, the authors [3] are interested in the creation of a morphological analyzer using the Morphological Analyzer and GEnerator for Arabic Dialects (MAGEAD).

This analyzer only processes verbs. The authors [6] have suggested a TD morphological analyzer using aebWordNet, Tunisian lexical dictionary and twenty two predicate rules. This created system does not deal with the standardized TD.

In addition, the authors [15, 16] have sought to create a morphological analyzer processing TD using the NooJ linguistic platform. In fact, the second approach generally gives better morphological analyzers than the first approach in terms of quality because it requires handwritten rules. Thus, this paper is based on [15, 16].

Table 2. Noun suffix in Tunisian dialect

Persons	Noun suffix
1st singular person	ي *'ii'* (my)
1st plural person	نا *'naa'* (our)
2nd singular person	ك *'k'* (your)
2nd plural person	كم *'kum'* (your)
3rd masculine singular person	هو / و / ه *'huu'* (his/its)
3rd feminine singular person	ها *'haa'* (her)
3rd plural person	هم *'hum'* (their)

```
ال,PREF+NW
هَ,DEM+NW
هَذَا,DEM+FLX=DEM
عَ,PREP+NW
عْلَى,PREP
كْ,NSUFF+2+s+NW
كُمْ,NSUFF+2+p+NW
مَاكْ,V+FLX=VERBE2
بِشْكْلَةْ,N+FLX=MFP5
تَجْرْبَةْ,N+FLX=MFP5
```

Fig. 1. Example of dictionary entries

3 Linguistic Study on Agglutination Phenomenon

TD is an agglutinative dialect. Indeed, the agglutination is the association of several grammatical categories in the same word. An agglutinated word has either proclitics, or enclitics, or both. The proclitics are located before the inflected or canonic form and the enclitics are after. In the following, we list the forms of agglutination at the level of verbs, nouns and particles.

3.1 Agglutinated Verbs

The verb is the most complicated grammatical category in terms of agglutination because it has multiple patterns. Regarding the proclitics of a verb, there is the conjunction (CONJ) which is frequently و *'wa'* (and), the adverb of interrogation (INTERR) ش *'ch'* (what) and the adverb (ADV) ما *'maa'*.

For example, the word وشماخرج *'wachmakharraj'* (and what does it take out) gathers all proclitics. Like proclitics, enclitics are found in TD verb. There are several types of enclitics: the adverb of interrogation which concerns yes/no questions (INTERR) شي *'chii'*, the adverb of negation (NEG) ش *'ch'* (not), direct object complements (DOC) and indirect object complements (IOC) as shown in Table 1.

Indeed, negation and interrogation adverbs cannot be together in the same verb. The TD verb structure is defined by the regular expression 1:

$$\text{CONJ? ADV? Verb DOC? IOC? (NEG|INTERR)?} \quad (1)$$

For example, the word وماوراوهوليش *'wmaawarrawhuuliich'* (they did not show it to me) is the longest structure for a Tunisian verb that represents an entire sentence. This word is composed of the conjunction و *'wa'* (and), the adverb ما *'maa'*, the verb وراو *'warraw'* (show), the DOC هو *'huu'* (it), the IOC لي *'lii'* (to me) and finally the negation adverb ش *'ch'* (not).

3.2 Agglutinated Nouns

Nouns have many agglutination patterns. Regarding the proclitics, definite nouns are preceded by the definite article (PREF) ال *'il'* (the). The prepositions (PREP) that appear before the nouns are as follows:

ب *'b'* (by), ل *'l'* (to), ك *'k'* (as), م *'m'* (from) is the abbreviation for من *'min'*, ع *''a'* (on) is the abbreviation for على *''alaa'* and ف *'fi'* (in) is the abbreviation for في *'fii'*.

The combination between the preposition ل *'l'* and the definite article ال *'il'* produces the proclitic لل *'lil'* (to the). The demonstrative pronoun ه *'ha'* (this) stands as a proclitic only before definite nouns.

Enclitics are found only for indefinite nouns in the form of an annexation compound. These enclitics are noun suffixes (NSUFF) as shown in Table 2. Formally, The TD definite noun structure is defined by the regular expression 2:

$$\text{CONJ? PREP? DEM? PREF? Definite\_Noun} \quad (2)$$

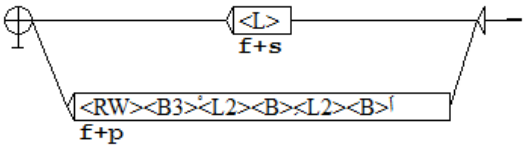


Fig. 2. Example of transducer for nouns

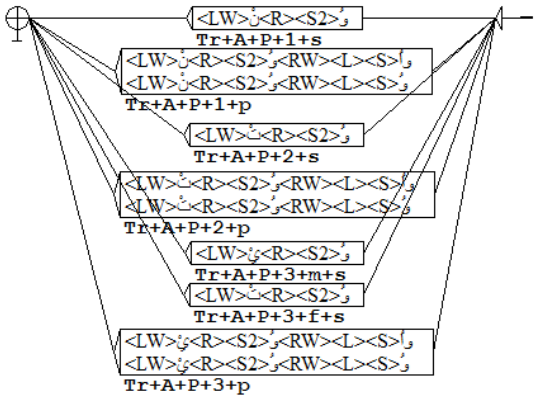


Fig. 3. Extract of transducer for Hollow verbs

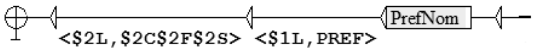


Fig. 4. Transducer for definite noun

In addition, the TD indefinite noun structure is defined by the regular expression 3:

$$\text{CONJ? PREP? Indefinite\_NounNSUFF?} \quad (3)$$

For example, the words وبالمشروع *'wibhalmachruu'a'* (and by this project) and وبمشروعهم *'wbimachruu'ahum'* (and by their project) are respectively the longest definite noun and indefinite noun structures in TD.

The first word is composed of the conjunction و *'wa'* (and), the preposition ب *'b'* (by), the demonstrative pronoun ه *'ha'* (this), the definite article ال *'il'* (the) and the noun مشروع *'machruu'a'* (project).

The second word is composed of the conjunction و *'wa'* (and), the preposition ب *'b'* (by), the noun مشروع *'machruu'a'* (project), the noun suffix هم *'hum'* (their).

3.3 Agglutinated Particles

Particles are composed of several grammatical categories. Most of them have the phenomenon of agglutination. First of all, prepositions also have proclitics and enclitics. The TD preposition structure is defined by the regular expression 4:

$$\text{CONJ? ADV? Preposition NSUFF? (NEG|INTERR)?} \quad (4)$$

For example, the word ومامعهاش *'wmaam'ahaach'* (And not with her) is the longest structure for a Tunisian preposition. This word is composed of the conjunction و *'wa'* (and), the adverb ما *'maa'*, the preposition مع *'m'a'* (with), the noun suffix ها *'haa'* (her), and finally the negation adverb ش *'ch'* (not). Furthermore, the TD personnel pronoun structure is defined by the regular expression 5:

$$\text{CONJ? ADV? Personal\_Pronoun (NEG|INTERR)?} \quad (5)$$

For example, the word وماهيش *'wmahich'* (She is not) is the longest structure for a Tunisian personal pronoun.

This word is composed of the conjunction و *'wa'* (and), the adverb the adverb ما *'maa'*, the personal pronoun هي *'hiya'* (she) and finally the negation adverb ش *'ch'* (not). In addition, the TD demonstrative pronoun structure is defined by the regular expression 6:

$$\text{CONJ? Prep? Demonstrative\_Pronoun} \quad (6)$$

For example, the word وبهذا *'wbihathaa'* (and by this) is the longest structure for a Tunisian demonstrative pronoun. This word is composed of the conjunction و *'wa'* (and), the preposition ب *'b'* (by) and the demonstrative pronoun هذا *'hathaa'* (this).

The conjunction و *'wa'* (and) marks a great presence in all grammatical categories in TD. In addition, the TD adverb ما *'maa'* is frequently found in words without or with the form of interrogation or negation.

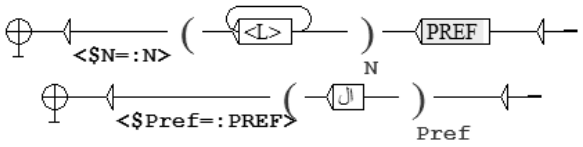


Fig. 5. PrefNom and Pref transducers

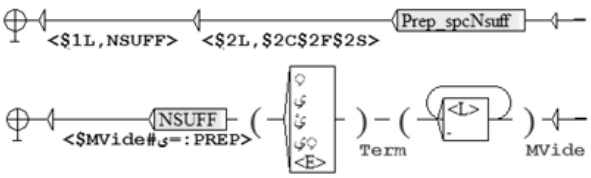


Fig. 6. Transducer for agglutinated preposition

```
Dictionary has been successfully compiled in file:
C:\Users\Lenovo\Documents\NooJ\ar\Lexical Analysis\barcha.nod
(342673/1190 states; 45294 infos; recognizes 169815 forms)
```

Fig. 7. Dictionary compilation result

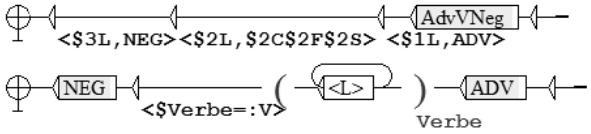


Fig. 8. Transducer for negation form of verbs

3.4 Effect of Agglutination on Certain Letters

The agglutination phenomenon has an effect on some letters at the end of the word. The correlated letter tā ة becomes ت. For example, the noun كرهبة *'karhbah'* (a car) after the agglutination becomes كرهبتها *'karhbatha'* (her car). Moreover, the letter shortened alif ى *'aa'* also undergoes a transformation and becomes the letter ي *'yi'*.

For example, the preposition على *''alaa'* (on) becomes عليه *''aliih'* (on it). Also, the letter hamzah ء becomes the letter yā hamzah ئ. For example, the noun اصدقاء *'asdikaa'* (friends) becomes اصدقائها *'asdikaiha'* (her friends).

4 Proposed Method

The method we propose to deal with the agglutination phenomenon starts with the automatic extraction of all non-repetitive words from the study corpus. Afterwards, the filtering phase allows to eliminate the same words which are written under different inflected forms.

Then, the choice of a canonical form allows to present the collected words. Finally, this canonical form is enriched by adding morphological, lexical and syntactic features. All these phases are established thanks to the dictionary, inflectional and morphological grammars of the linguistic platform NooJ [14].

4.1 Dictionary and Inflectional Grammars

The dictionary is a set of entries that consist of a canonical form, a lexical category and an inflectional grammar if necessary. Fig. 1 shows an example of dictionary entries. The dictionary entries presented in Fig. 1 contain different lexical categories such as definite article (PREF), demonstrative pronoun (DEM), preposition (PREP), noun suffix (NSUFF), verb (V) and noun (N).

Entries with the NW (non-word) code should not be analyzed as real words because they are either proclitic or enclitic. For example, the demonstrative pronoun هذا *'hathaa'* (this) is a real word while its abbreviation ه *'ha'* is not a real word but it is a proclitic.

Similarly, the preposition ع *''a'* (on) is not a real word while على *''alaa'* is real word. In fact, inflectional grammars (FLX) generate all the inflected forms of the dictionary entry. As mentioned in Fig. 1, a set of nouns uses an inflectional grammar called "MFP5".

This grammar is presented by a transducer in Fig. 2. This transducer is dedicated to feminine nouns having the scheme فعلة *'fa'alalah'* in the singular and transforming into the other scheme فعالل *'fa'alil'* in the plural. For example, the canonical form بسكلة *'bisklah'* (bike) remains the same in the first path and becomes بساكل *'bsaakil'* (bikes) in the second path.

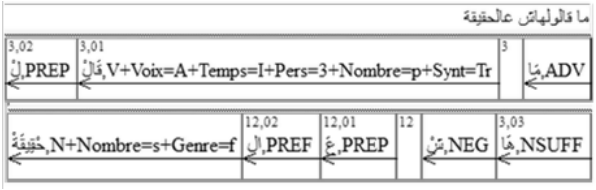


Fig. 9. Example of linguistic analysis

After	Seq.	Before
زعمة زعمة نسيت	وقلت	ق مسحت دموعي
انشالله بوها و امها	قلت	زيت يدي لربي و
يا بطية عرفت وحد	ماتقوللهاش	تتنرفز عليك كان
راهو هالديدويات ه	نقول	كوها عليه نعاود
و نأكد انو مانيش نا	نقولها	ا في مخي اخيرًا

Fig. 10. Result excerpt of <قال> location

Table 3. Summarizing the obtained metrics for all words

Corpus	Recall	Precision	F-measure
18680	0.89	0.95	0.91

In addition, the TD verb (V) in Fig. 1 is a hollow verb. Thus, it uses an inflectional grammar called "VERBE2" presented by a transducer in Fig. 3. Indeed, this transducer allows to generate hollow verbs whose root origin having the second letter is و *'w'* and transforming into the letter ا *'a'*.

For example, after conjugation in the present tense (P) with the third person masculine singular (3+m+s) of the Tunisian canonical form مات *'maat'* (to die), it becomes the conjugated verb يموت *'ymuut'* (he dies).

4.2 Morphological Grammars

To deal with the phenomenon of agglutination, we establish two morphological grammars based on a set of nested finite transducers.

For TD noun agglutination, we construct several transducers, among which the nested transducer shown in Fig. 4 solves the definite noun problem. Indeed, the nested transducer in Fig. 4 contains a subgraph called "PrefNom" and two nodes. The first one recognizes the first lemma ($1L) as a definite article (PREF) and the second node recognizes the category ($2C), the inflectional feature ($2F) and the semantic and syntactic feature of the second lemma ($2L). Fig. 5 shows the PrefNom transducer and its PREF subgraph. In the first transducer, the name is stored in the variable ($N) indicating that the loop <L> means a sequence of letters.

Thus, the contents of the variable ($N) are verified by a dictionary lookup. With the same variable principle, the second transducer recognizes the definite article ال *'il'* (the). For verbs, we also establish several transducers, for example, the set of transducers in Fig. 8 solves the negation form of verbs.

The first transducer is the main graph contains a subgraph called "AdvVNeg" and three nodes. The first one recognizes the first lemma ($1L) as an adverb (ADV). The second node is explained above. The third node recognizes the third lemma ($3L) as a negation particle (NEG). In addition, the second transducer recognizes the adverb and the negation particle by its two subgraphs called "ADV" and "NEG" respectively.

In addition, we dedicate a set of finite state transducers to deal with TD particles. For example, the two finite transducers illustrated in Fig. 6 treat a specific type of agglutination for prepositions explained below. The first transducer is the main graph that contains a subgraph called "Prep_spcNsuff" and two nodes.

This subgraph which is the second transducer is used to recognize prepositions that undergo a transformation from the shortened letter alif ى *'aa'* to the letter ي *'yi'*. This problem is solved by using two variables. The first variable ($MVide) stores the unchanged part and the second stores the changed part. Thus, the code ($MVide#ى= :PREP) adds the letter ى *'aa'* to the first variable and then checks its existence in the dictionary.

In addition, the subgraph called "NSUFF" recognizes the noun suffix in TD. In conclusion, we construct 95 finite state transducers to solve the different forms of the agglutination phenomenon for all grammatical categories in TD; among which 23 main transducers for agglutinated verbs, 13 main transducers for agglutinated nouns and 18 main transducers for agglutinated particles.

Table 4. Obtained results for all words

	Noun	Verb	Particle	Adjective	Total
Corpus	8450	3260	5740	1140	18680
Correct recognized word	7940	2820	4910	990	16660

5 Experimentation and Evaluation

To experiment with our constructed linguistic resources on the collected test corpus, we have implemented our lexical resources in the NooJ linguistic platform. In fact, the dictionary is edited and saved in the file "barcha.dic" which is extended by the file "barcha.nod" after compilation.

Up to now, our NooJ morphological analyzer generates, from 4422 entries, 169815 forms as presented in Fig. 7. Moreover, the morphological grammars allowing the resolution of agglutination are stored in the file "agglutination.nom" and are implemented by finite transducers.

As already indicated, to evaluate our resources, we have collected a corpus from Tunisian dialect novels and social networks such as Facebook and Twitter. The test corpus contains 3300 sentences and 18680 words. The evaluation of our NooJ prototype is based on the recognition of TD words. Thus, we used the known metrics: recall, precision and f-measure.

We obtain the following results presented in Table 3. More precisely, Table 4 shows the results obtained of the prototype application. It shows how well our grammars recognize nouns, verbs, particles and adjectives. Table 4 shows that our prototype recognizes 89% of the total words.

93% of nouns in the corpus are recognized. Some unrecognized nouns are compound proper names, city names or company names. Moreover, our prototype detects 86% of verbs, particles and adjectives in the corpus. In addition, a set of unrecognized words belong entirely to MSA.

For example, ستعودين *'sata'udina'* (you will come back) is not a Tunisian verb because the proclitic س *'sa'* (will) does not belong to the Tunisian dialect as well as the flexion of the verb. Another example, the adjective حائرين *'ha'iriin'* (worried) it is not considered as a Tunisian adjective because the correct writing is حايرين *'hayiriin'*.

Among the undetected words, these contain a repetitive series of letters such as برشاااا *'barchaaaaa'* (many). In fact, our prototype detects all demonstrative, relative and personal pronouns as well as interrogative adverbs. Moreover, agglutinated words are well recognized in different grammatical categories in TD.

For example, the linguistic analysis of the Tunisian sentence shown in Fig. 9: ما قالولهاش عالحقيقة *'maa kaaluulhaach 'alhkikah'* (they do not tell her about the truth) is as follows. We get that the word ما *'maa'* is recognized as an adverb (ADV).

Moreover, the recognized word قالولهاش *'kaaluulhaach'* (tell) is a verb (V) conjugated in the past tense (I) with the third person (3) plural (p) having recognized enclitics: ل *'l'* (to) as a preposition and ها *'haa'* (her) as a noun suffix which are an enclitic IOC and ش *'ch'* (not) as an adverb of negation (NEG).

Finally, the recognized word عالحقيقة *''alhkikah'* (about truth) is a singular (s) feminine (f) noun (N) that is preceded by the definite article (PREF) ال *'il'* (the) and also by the preposition (PREP) ع *''a'* (about).

Thanks to the NooJ linguistic platform, we can locate patterns in the test corpus and detect all the different morphological, inflected, and agglutinated forms of different grammatical categories. For example, to locate a specific verb, we simply write <قال> in the regular expression box.

An excerpt of the result of this location is shown in Fig. 10. Among the results, there are just inflected forms like the words قلت *'kult'* (I said), نقول *'nkuul'* (I say), and several agglutinated forms. Precisely, there are words with only proclitics like وقلت *'wkult'* (and I said), others with only enclitics like نقولها *'nkuulhaa'* (I said it).

The TD words in our dictionary have different origins like French and Turkish. The unrecognized words have a typographical error or are MSA words. We consider that the results obtained are ambitious. Moreover, they can be improved by increasing the coverage of the dictionary and by adding more morphological rules.

6 Conclusion and Perspectives

In the present paper, we have created a set of linguistic resources for TD in the NooJ language platform. These resources that deal with the agglutination phenomenon are realized through a set of nested finite transducers and are based on a deep linguistic study.

All these resources allow us to construct a NooJ prototype. Moreover, we have demonstrated the efficiency of our NooJ prototype. Thus, the evaluation is performed on a set of sentences belonging to the test corpus. The obtained results are ambitious and show that several agglutinated words can be detected and resolved.

As perspectives, we will increase the coverage of our dictionaries. Furthermore, we will improve our grammars by adding morphological rules that recognize other linguistic phenomena.

References

1. **Buckwalter, T. (2004).** Buckwalter Arabic morphological analyzer: version 2.0.
2. **Eskander, R., Habash, N., Rambow, O. (2013).** Automatic extraction of morphological lexicons from morphologically annotated corpora. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1032–1043.
3. **Hamdi, A., Boujelbane, R., Habash, N., Nasr, A. (2013).** Un système de traduction de verbes entre arabe standard et arabe dialectal par analyse morphologique profonde, pp. 396–406.
4. **Hammouda, N., Torjmen, R., Haddar, K. (2018).** Transducer cascade to parse Arabic corpora. **Silberztein, M., Atigui, F., Kornyshova, E., Métais, E., Meziane, F.**, editors, Natural Language Processing and Information Systems. NLDB´18, volume 10859, pp. 230–237.
5. **Harrat, S., Meftouh, K., Abbas, M., Smaïli, K. (2014).** Building resources for Algerian Arabic dialects. pp. 2123–2127.
6. **Karmani, N. B. M., Soussou, H., Alimi, A. M. (2016).** Intelligent tunisian arabic morphological analyzer. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1–8.
7. **Mesfar, S. (2008).** Analyse morpho-syntaxique automatique et reconnaissance des entités nommées en arabe standard.
8. **Mohamed, B., Azzeddin, e. M., Mohamed Ould, A., Abdelhak, L., Abderrahim, B. (2017).** Alkhalil morpho sys 2: A robust Arabic morpho-syntactic analyzer. Journal of King Saud University - Computer and Information Sciences, Vol. 29, No. 2, pp. 141–146.
9. **Mohamed, M., Graff, D., Bouziri, G., Krouna, S., Bies, A., Kulick, S. (2010).** LDC standard Arabic morphological analyzer (SAMA) version 3.1. Linguistic Data Consortium, LDC catalog number LDC2009E73.
10. **Pasha, A., Elbadrashiny, M., Diab, M., Elkholy, A., Eskandar, R., Habash, N., Pooleery, M., Rambow, O., Roth, R. (2014).** MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. Proceedings of the 9th International Conference on Language Resources and Evaluation, pp. 1094–1101.
11. **Ridouane, T., Bouzoubaa, K. (2014).** Building a Moroccan dialect electronic dictionnary (MDED). pp. 216–221.
12. **Salloum, W., Habash, N. (2013).** Dialectal Arabic to English machine translation: Pivoting through modern standard Arabic. pp. 348–358.
13. **Salloum, W., Habash, N. (2014).** ADAM: Analyzer for dialectal Arabic morphology. Journal of King Saud University - Computer and Information Sciences, Vol. 26, No. 4, pp. 372–378.
14. **Silberztein, M. (2005).** NooJ's dictionaries. Proceedings of LTC, Vol. 5, pp. 21–23.
15. **Torjmen, R., Haddar, K. (2019).** Construction of morphological grammars for the Tunisian dialect. Formalizing Natural Languages with NooJ 2018 and Its Natural Language Processing Applications, Springer International Publishing, pp. 62–74.

16. Zribi, I., Ellouze, M., Belguith, L., Blache, P. (2017). Morphological analysis of Tunisian dialect. International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining, Vol. 11107, pp. 180–187.

Article received on 21/02/2019; accepted on 04/01/2021.
Corresponding author is Roua Torjmen.

ISSN 2007-9737

A Study on Stochastic Variational Inference for Topic Modeling with Word Embeddings

Kana Ozaki, Ichiro Kobayashie

Ochanomizu University,
Japan

{ozaki.kana, koba}@is.ocha.ac.jp

Abstract. Probabilistic topic models based on Latent Dirichlet Allocation (LDA) is widely used to extract latent topics from document collections. In recent years, a number of extended topic models have been proposed, especially Gaussian LDA (G-LDA) has attracted a lot of attention. G-LDA integrates topic modeling with word embeddings by replacing discrete topic distributions over words with multivariate Gaussian distributions on the word embedding space. This can reflect semantic information into topics. In this paper, we use G-LDA for our base topic model and apply Stochastic Variational Inference (SVI), an efficient inference algorithm, to estimate topics. Through experiments, we could extract the topics with high coherence in practical time.

Keywords. Topic model, latent Dirichlet allocation, word embeddings, stochastic variational inference.

1 Introduction

Probabilistic topic models such as Latent Dirichlet Allocation (LDA) [2], are widely used to uncover hidden topics within text corpus. In LDA, each document may be viewed as a mixture of latent topics where each topic is a distribution over words. With statistical inference algorithms, LDA reveals latent topics using document-level word co-occurrence.

In recent years, a number of extended topic models have been proposed, especially Gaussain LDA(G-LDA) [3] that integrates LDA with word embeddings has gained much attention. G-LDA uses Gaussian distribution as the topic distribution over words.

Furthermore, Batmanghelich et al. [1] proposed spherical Hierarchical Dirichlet Process (sHDP) which use the von Mises-Fisher distribution as the topic distribution to model the density of words over unit sphere. They used the Hierarchical Dirichlet Process (HDP) for their base topic model and apply Stochastic Variational Inference (SVI) [5] for efficient inference.

They showed that sHDP is able to exploit the semantic structures of word embeddings and flexibly discovers the number of topics. Hu et al. [7] proposed Latent Concept Topic Model (LCTM) which introduces latent concepts to G-LDA. LCTM models each topic as a distribution over latent concepts, where each concept is a localized Gaussian distribution in word emgedding space.

They reported that LCTM is well suited for extracting topics from short texts with diverse vocaburaly such as tweets. Xun et al. [15] proposed a correlated topic model using word embeddings. Their model enables us to exploit the additional word-level correlation information in word embeddings and directly models topic correlation in the continuous word embedding space.

Nguyen et al. [12] proposed Latent Feature LDA (LF-LDA) which integrates word embeddings into LDA by replacing the topic-word Dirichlet multinomial component with a mixture of a Dirichlet multinomial component and a word embedding component. They compared the performance of LF-LDA to vanilla LDA on topic coherence, document clustering and document classification evaluations and showed that LF-LDA improves both topic-to-word mapping and document-topic assignments compared to vanilla LDA, especially

on datasets with few or short documents. Kumar et al. [14] presented an unsupervised topic model for short texts that performs soft clustering over word embedding space.

They modeled the low-dimensional semantic vector space represented by word embeddings using Gaussian mixture models (GMMs) whose components capture the notion of latent topics. Their proposed framework outperforms vanilla LDA on short texts through both subjective and objective evaluation, and showed its usefulness in learning topics and classifying short texts on Twitter data for several foreign languages.

Zhao et al. [17] proposed a focused topic model where how a topic focuses on words is informed by word embeddings. Their models are able to discover more informed and focused topics with more representative words, leading to better modelling accuracy and topic quality. Moody [10] proposed a model, called lda2vec, which learns dense word vectors jointly with Dirichlet distributed latent document-level mixtures of topic vectors.

His method is simple to incorporate into existing automatic differentiation frameworks and allows for unsupervised document representations geared for use by scientists while simultaneously learning word vectors and the linear relationships between them. Yao et al. [16] proposed Knowledge Graph Embedding LDA (KGE-LDA), which combines topic model and knowledge graph embeddings.

KGE-LDA models document level word co-occurrence with knowledge encoded by entity vectors learned from external knowledge graphs and can extract more coherent topics and better topic representation. In this paper, we use G-LDA as our base topic model. Compared with vanilla LDA, G-LDA produces higher Pointwise Mutual Information (PMI) in each topic because it has semantic information of words as prior knowledge.

In addition, because G-LDA operates on the continuous vector space, it can handle out of vocabulary (OOV) words in held-out documents whereas the conventional LDA cannot. On the other hand, the cost for estimating the posterior probability distribution for latent topics in word embedding space is costly because of dealing with the high dimensional information of words. So, it is unpractical to use the methods which take much time to estimate the posterior probability distribution such as Gibbs sampling. To reduce the cost for estimating the posterior probability distribution, G-LDA utilizes Cholesky decomposition of covariance matrix and applies Alias Sampling [8] for that.

In a similar case of dealing with high dimensional data, it is also difficult to estimate latent topics in massive documents using sampling methods. To deal with this problem, Hoffman et al. [4] developed online Variational Bayes (VB) for LDA. Their model is handily applied to massive and streaming document collections.

Their proposed method, online variational Bayes, becomes well known as "Stochastic Variational Inference" [13, 5]. Referring to their approach, in this paper, we propose a method to efficiently estimate latent topics in the high dimensional space of word embeddings by adopting SVI (Stochastic Variational Inference).

2 LDA and Gaussian LDA

2.1 Latent Dirichlet Allocation (LDA)

LDA [2] is a probabilistic generative model of document collections. In LDA, each topic has a multinomial distribution β over a fixed vocabulary and each document has a multinomial distribution θ over K topics.

Distributions β and θ are designed to be sampled from the conjugate Dirichlet priors parameterized by η and α, respectively. Suppose that D and N_d denote the number of documents and words in dth document, respectively. The generative process is as follows:

1. for $k = 1$ to K
   a) Choose topic $\boldsymbol{\beta}_k \sim \mathrm{Dir}(\boldsymbol{\eta})$
2. for each document d in corpus D
   a) Draw topic distribution $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$
   b) For each word index n from 1 to N_d
      a) Draw a topic $z_n \sim \mathrm{Categorical}(\boldsymbol{\theta}_{\mathrm{d}})$
      b) Draw a word $w_n \sim \mathrm{Categorical}(\boldsymbol{\beta}_{\mathrm{z_n}})$.

The graphical model for LDA is shown in the left side of Figure 1.

2.2 Gaussian LDA (G-LDA)

Hu et al. [6] proposed a new method to model the latent topic in the task of audio retrieval, in which each topic is directly characterized by Gaussian distribution over audio features.

Das et al. [3] presented an approach for accounting for semantic regularities in language, which integrates the model proposed by Hu et al. [6] with word embeddings. They use word2vec [9], to generate skip-gram word embeddings from unlabeled corpus.

In this model, they characterize each topic k as a multivariate Gaussian distribution with mean μ_k and covariance Σ_k in an M-dimentional embedding space, and concurrently replaces the Dirichlet priors with the conjugate Normal Inverse Wishart (NIW) priors on Gaussian topics.

Because the observations are no longer discrete values but continuous vectors, word vectors are sampled from continuous topic distributions. They reported that G-LDA produced higher PMI score than conventional LDA as the result of the experiment, which means topical coherence was improved.

Because G-LDA uses continual distributions as the topic distributions over words, it can assign latent topics to OOV words without training the model again, whereas the original LDA cannot deal with those words. The generative process is as follows:

1. for $k = 1$ to K
   a) Draw topic covariance $\mathbf{\Sigma}_k \sim \mathcal{W}^{-1}(\mathbf{\Psi}, \mu)$
   b) Draw topic mean $\boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{\beta}\mathbf{\Sigma}_k)$
2. for each document d in corpus D
   a) Draw topic distribution $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$
   b) for each word index n from 1 to N_d
      a) Draw a topic $z_n \sim \mathrm{Categorical}(\boldsymbol{\theta}_\mathrm{d})$
      b) Draw $\boldsymbol{v}_{d,n} \sim \mathcal{N}(\boldsymbol{\mu}_{z_n}, \mathbf{\Sigma}_{z_n})$.

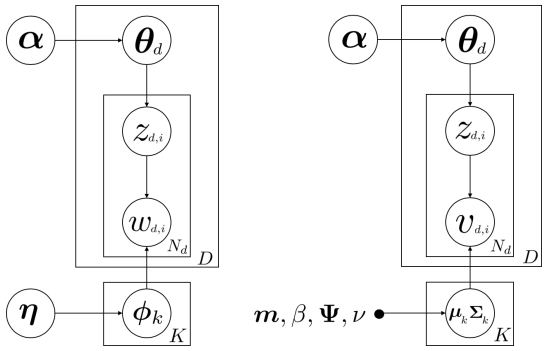


Fig. 1. Grphical representations of LDA (left) and Gaussian LDA (right).

Although $\boldsymbol{\theta}_d$ represents topic distributions of dth document as the traditional LDA does, $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ represent the mean and the covariance of the multivariate Gaussian distribution, respectively. Besides, $\boldsymbol{v}_{d,n}$ represents word vector. The graphical model for G-LDA is shown in the right side of Figure 1.

3 Posterior Inference with SVI

Sampling method such as Gibbs sampler is widely used to perform approximate inference in topic modeling. Although Gibbs sampler has an advantage for easy implementation, it takes much time to estimate a posterior distribution.

Hence, we employ an efficient inference algorithm based on VB, i.e., Stochastic Variational Inference (SVI) [5], to estimate the posterior probability distributions of the latent variables. SVI is an efficient algorithm for large datasets because it can sequentially process batches of documents.

With VB inference, the true posterior probability distribution is approximated by a simpler distribution $q(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\mu}, \mathbf{\Sigma})$, which is indexed by a set of free parameters $\boldsymbol{\theta}, \boldsymbol{\mu}$ and $\mathbf{\Sigma}$.

These parameters are optimized to maximize the Evidence Lower BOund (ELBO) , a lower bound

on the logarithm of the marginal probability of the observations $\log p(\boldsymbol{v})$:

$$\begin{aligned}\log p(\boldsymbol{v} \mid \boldsymbol{\alpha},\ \boldsymbol{\zeta}) \geq L(\boldsymbol{v},\ \boldsymbol{\phi},\ \boldsymbol{\gamma},\ \boldsymbol{\zeta})\\ \triangleq \mathbb{E}_q[\log p(\boldsymbol{v},\ \boldsymbol{z},\ \boldsymbol{\theta}\ ,\ \boldsymbol{\mu},\ \boldsymbol{\Sigma} \mid \boldsymbol{\alpha},\ \boldsymbol{\zeta})] -\\ \mathbb{E}_q[\log q(\boldsymbol{z},\ \boldsymbol{\theta},\ \boldsymbol{\mu},\ \boldsymbol{\Sigma})].\end{aligned} \tag{1}$$

Based on the assumption that variables are independent in the mean-field family, approximate distribution q is fully factorized as follows:

$$q(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = q(\boldsymbol{z})q(\boldsymbol{\theta})q(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{2}$$

Let ϕ be the parameter for the latent variables z, γ be the parameter for the distribution θ over topics and $\boldsymbol{\zeta} = (\boldsymbol{m}, \beta, \boldsymbol{\Psi}, \nu)$ be the parameter of the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of the topic distribution over word types. Factorized distributions of q are:

$$q(z_{di} = k) = \phi_{dw_{di}k}, \tag{3}$$

$$q(\theta_d) = \mathrm{Dir}(\theta_{\mathrm{d}}|\gamma_{\mathrm{d}}), \tag{4}$$

$$q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = NIW(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k|\boldsymbol{m}_k, \beta_k, \boldsymbol{\Psi}_k, \nu_k), \tag{5}$$

$$\gamma_{dk} = \alpha + \sum_w n_{dw}\phi_{dwk}, \tag{6}$$

$$\phi_{dwk} \propto \exp\{\mathbb{E}_q[\log \theta_{dk}] + \mathbb{E}_q[\log N(\boldsymbol{v}_{dw}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]\}. \tag{7}$$

SVI needs not analyze the whole data set before improving the global variational parameters and can apply new data which is constantly arriving, while VB requires a full pass through the entire corpus at each iteration. $q(\boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k)$ is the object for sequential learning, while $q(\boldsymbol{z}_d)$ and $q(\boldsymbol{\theta}_d)$ are optimized at each iteration.

Thus, we apply the stochastic natural gradient descent to update the parameters $\boldsymbol{\zeta} = (\boldsymbol{m}, \beta, \boldsymbol{\Psi}, \nu)$. At dth document containing n_d words, we optimize ϕ_d and γ_d, holding $\boldsymbol{\zeta}$ fixed. Next, we calculate intermediate global parameters $\boldsymbol{\zeta}^* = (\boldsymbol{m}^*, \beta^*, \boldsymbol{\Psi}^*, \nu^*)$ as follows:

$${\beta_k}^* = \beta + D\sum_w n_{dw}\phi_{dwk}, \tag{8}$$

$$\nu_k^* = \nu + D\sum_w n_{dw}\phi_{dwk}, \tag{9}$$

$$\boldsymbol{m}_k^* = \frac{\beta\boldsymbol{m} + D\sum_w n_{dw}\phi_{dwk}\bar{\boldsymbol{v}}_k}{\beta_k^*}, \tag{10}$$

$$\boldsymbol{\Psi}_k^* = \boldsymbol{\Psi} + \boldsymbol{C}_k + \frac{\beta D\sum_w n_{dw}\phi_{dwk}}{\beta_k^*}(\bar{\boldsymbol{v}}_k - \boldsymbol{m})(\bar{\boldsymbol{v}}_k - \boldsymbol{m})^{\mathrm{T}}. \tag{11}$$

Here:

$$\bar{\boldsymbol{v}}_k = \frac{\sum_w n_{dw}\phi_{dwk}\boldsymbol{v}_{dw}}{\sum_w n_{dw}\phi_{dwk}}, \tag{12}$$

$$\boldsymbol{C}_k = D\sum_w n_{dw}\phi_{dwk}(\boldsymbol{v}_{dw} - \bar{\boldsymbol{v}}_k)(\boldsymbol{v}_{dw} - \bar{\boldsymbol{v}}_k)^{\mathrm{T}}. \tag{13}$$

D denotes the number of corpus, which means that $\boldsymbol{\zeta}$ is optimized if the entire corpus consisted of the singe document n_d repeated D times. By this operation, it becomes possible to update the parameters $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\zeta}$ at each iteration without whole documents, so that it can analyze massive document collections, including those arriving in a stream. We then update $\boldsymbol{\zeta}$ using a weighted average of its previous value and the estimated $\boldsymbol{\zeta}^*$. The update is:

$$\boldsymbol{\zeta} = (1 - \rho_d)\boldsymbol{\zeta} + \rho_d\boldsymbol{\zeta}^*. \tag{14}$$

The weight for $\boldsymbol{\zeta}^*$ is given by $\rho_d \triangleq (\tau_0 + d)^{-\kappa}$, where $\kappa \in (0.5, 1]$ controls the rate at which old values of $\boldsymbol{\zeta}$ are forgotten and $\tau_0 \geq 0$ slows down in the early iterations of the algorithm. The expectations under q of $\log \theta_{dk}$ and $\log N(\boldsymbol{v}_{dw}|\boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k)$ are:

$$\mathbb{E}_q[\log \theta_{dk}] = \Psi(\gamma_{dk}) - \Psi(\sum_{i=1}^{K}\gamma_{di}), \tag{15}$$

$$\begin{aligned}\mathbb{E}_q[\log N(\boldsymbol{v}_{dw}|\boldsymbol{m}\boldsymbol{u}_k, \boldsymbol{\Sigma}_k)] = -\frac{1}{2}\boldsymbol{v}_{dw}^{\mathrm{T}}\langle\boldsymbol{\Sigma}_k^{-1}\rangle\boldsymbol{v}_{dw}\\ +\boldsymbol{v}_{dw}^{\mathrm{T}}\langle\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\rangle - \frac{1}{2}\langle\log|\boldsymbol{\Sigma}_k|\rangle.\end{aligned} \tag{16}$$

where, Ψ and $\langle\ \rangle$ denote the digamma function and the expectation, respectively. Algorithm 1 presents the full algorithm of SVI for Gaussian LDA.

4 Experiments

We construct a model which integrates SVI to word vector topic model following Algorithm 1 and conduct the experiment of topic extraction. In this paper, we evaluate whether our model is able to find coherent and meaningful topics compared with the conventional LDA.

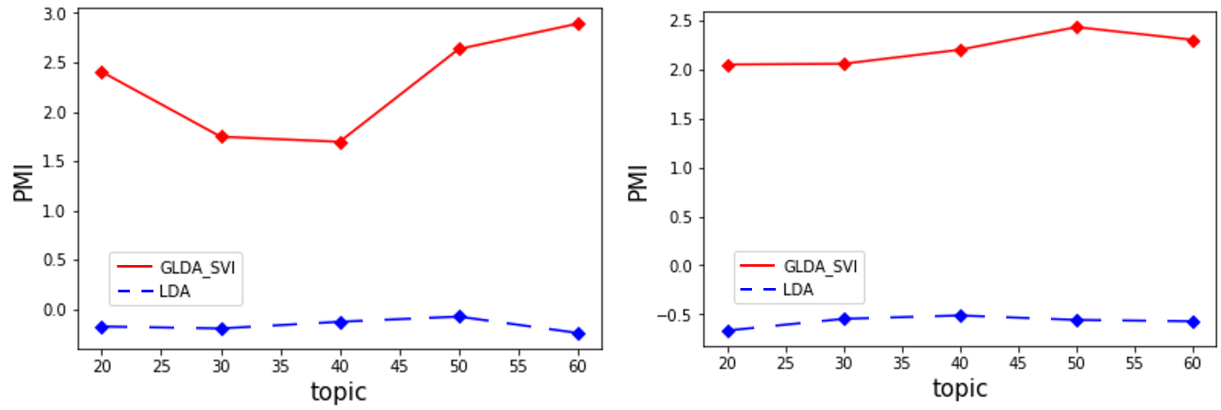


Fig. 2. PMI performance of the top 10 words on 20Newsgroups (left) and NIPS (right)

Table 1. Top 10 words of some topics from our model and multinomial LDA on 20Newsgroups for K = 40 and PMI score

Gaussian LDA topics									
cie	geophysics	manning	authenticicity	beasts	ton	acts	disasters	provoke	normals
informatik	astrophysics	neely	veracity	creatures	tons	exercising	disaster	provocation	histograms
nos	physics	carney	credence	demons	gallon	coercion	hazards	futile	gaussian
gn	meteorology	brady	assertions	monsters	mv	act	catastrophic	suppress	linear
nr	astronomy	wilkins	inaccuracies	eleves	cargo	enforcing	devastation	resorting	symmetric
sta	geophysical	brett	particulars	spirits	cruiser	collective	dangers	threatening	histogram
vy	geology	seaver	texttual	unicorns	pound	proscribed	pollution	aggression	vectors
gl	astrophysical	reggie	merits	denizens	pounds	regulating	destruction	urge	inverse
cs	chemistry	ryan	substantiate	magical	corvettes	initiating	impacts	inflict	graphs
ger	microbiology	wade	refute	gods	guns	involving	destructive	expose	variables
6.6429	6.2844	5.3070	5.0646	4.3270	3.6760	3.1671	2.8486	2.7723	2.7408
Multinomial LDA topics									
drive	ax	subject	data	south	la	supreme	writes	key	goverment
disease	max	lines	doctors	book	goal	bell	article	code	law
hard	a86	server	teams	lds	game	at&t	organization	package	gun
scsi	0d	organization	block	published	cal	zoology	senate	window	clinton
drives	1t	spacecraft	system	adl	period	subject	subject	data	congress
disk	giz	spencer	spave	armenian	bd	covenant	dod	information	clipper
subject	3t	program	output	books	roy	suggesting	lines	anonymous	key
daughter	cx	space	pool	documents	55.0	lines	income	ftp	clayton
unit	bh	software	resources	isubject	its	off	deficit	program	federal
organization	kt	graphic	bits	information	season	origins	year	source	constitution
2.3514	2.2500	1.3700	1.1216	1.0528	0.8338	0.7092	0.4531	0.4501	0.4355

4.1 Experimental Setting

We perform experiments on two different text corpora: 18846 documents from 20Newsgroups[1] and 1740 documents from the NIPS[2]. We utilize 50-dimensional word embeddings trained on text from Wikipedia using word2vec and run out the model with various number of topics ($K = 20 \sim 60$). The document distribution over topics θ is designed to be sampled from the conjugate Dirichlet prior parameterized by $\alpha = 1/K$. In equation (7), we set parameters $\tau_0 \in \{1, 4, 16, 64, 256, 1024\}$ and $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ then we set the batch size according to the number of documents: $S \in \{4, 16, 64, 256, 1024\}$ for

[1] http://qwone.com/~jason/20Newsgroups/
[2] https://cs.nyu.edu/~roweis/data.html

Table 2. Top 10 words of some topics from our model and multinomial LDA on NIPS for K = 40 and PMI score

Gaussian LDA topics									
topological	ginzburg	mitsubishi	negation	m.s	generalize	vx	gcs	behaviors	describes
projective	goldmann	vw	predicate	ms	analytically	xf	dcs	behaviours	describing
subspaces	jelinek	gm	disjunction	m/s	generalizations	vf	tss	behavior	interprets
symplectic	kolmogorov	motors	predicates	bd	intuitively	vz	rbp	behaviour	discusses
homotopy	markov	flat	propositional	tat	generalizing	r4	sdh	biases	illustrates
topology	pinks	dyna	priori	dd	computable	xr	modulators	arousal	relates
euclidian	christof	integra	reflexive	stm	theretic	rx	signalling	behavioural	identifies
integrable	koenig	combi	duality	bs	solvable	tlx	mds	behaviorally	characterizes
subspace	engel	gt	categorical	lond	generalization	t5	analysers	attentional	demonstrates
affine	lippmann	suzuki	imperfect	bm	observable	spec	bss	predisposition	observes
11.3463	9.4716	6.8211	6.7072	4.8832	4.4388	4.2489	3.6088	3.4125	3.3666
Multinomial LDA topics									
model	network	learning	network	neural	network	network	learning	function	network
figure	model	network	algorithm	networks	networks	neural	neural	network	model
neural	input	figure	neural	model	neural	funtion	figure	model	input
learning	learning	data	learning	input	input	input	network	neural	learning
input	neural	units	training	data	learning	learning	data	training	data
network	networks	model	input	learning	data	model	input	learning	system
output	output	input	output	function	training	networks	training	set	training
number	function	set	networks	figure	output	figure	function	algorithm	neural
function	data	neural	set	units	number	output	model	data	function
data	figure	output	function	output	set	training	output	figure	output
0.4945	0.4302	0.3506	0.3232	0.2280	0.1759	-0.0473	-0.1412	-0.1784	-0.2415

Algorithm 1 SVI for Gaussian LDA

Define $\rho_d \triangleq (\tau_0 + d)^{-\kappa}$
Initialize $\boldsymbol{m}, \beta, \boldsymbol{\Psi}, \nu$ randomly.
for $d = 0$ to ∞ **do**
  Estep:
  Initialize $\gamma_{dk} = 1$ (The constant 1 is arbitrary.)
  repeat
    Set $\phi_{dwk} \propto \exp\{\mathbb{E}_q[\log \theta_{dk}]$ + $\mathbb{E}_q[\log N(\boldsymbol{v}_{dw}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]\}$
    Set $\gamma_{dk} = \alpha + \sum_w n_{dw}\phi_{dwk}$
  until $\frac{1}{K}\sum_k |change\ \ in\ \ \gamma_{dk}| < 0.00001$
  Mstep:
  Compute $\boldsymbol{\zeta}_k^*$ with Eq.(5)
  Set $\boldsymbol{\zeta} = (1 - \rho_d)\boldsymbol{\zeta} + \rho_t \boldsymbol{\zeta}^*$
end for

20Newsgroups-dataset, $S \in \{4, 10, 16\}$ for NIPS -dataset. Our model implementation is in Python[3]. In the experiments, we used the conventional LDA as a baseline model. The hyper parameters α and η in Dirichlet distribution are $1/K$ and 0.01, respectively.

[3] https://github.com/KanaOzaki/SVI_GLDA

4.2 Evaluation

We use PMI score to evaluate the quality of topics learnt by our models as well as it is used to evaluate the ability of G-LDA [3]. Newman et al. [11] showed that PMI has relatively good agreement with human scoring.

We use a reference corpus of documents from Wikipedia and use co-occurence statistics over pairs of words (w_i, w_j) in the same document. The PMI score of topic k is computed by:

$$PMI(k) = \frac{1}{\binom{N}{2}} \sum_{j=2}^{N} \sum_{i=1}^{j-1} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}. \qquad (17)$$

We use the average of the score of top 10 words of each topic. A higher PMI score implies a more coherent topic as it means the topic words usually co-occur in the same document.

4.3 Result

The experimental results of PMI on 20Newsgroups and NIPS-datasets are shown in Figure 2. We plot the average of the PMI scores for the top 10 words

in each topic, the result of 20Newsgroups with the parameters of $S = 16,\ \kappa = 1.0,$ and $\tau_0 = 1024$, and NIPS with those of $S = 4,\ \kappa = 1.0,$ and $\tau_0 = 1024$.

It is clearly seen that our model outperforms the conventional LDA in terms of PMI score. Some examples of top topic words are listed in Table 1 and Table 2. The parameter settings is the same as above and we present the top 10 topics in descending order.

In the last line of the tables, we present the PMI score for 10 topics for both our model and the traditional LDA. We see that the topics of our model seems more coherent than the baseline model.

In addition, our model is able to capture several intuitive topics in the corpus such as natural science, mythology and cargo in Table 1, mathematics and car in Table 2. In particular, our model discovered the collection of human names, which was not captured by traditional LDA.

5 Conclusions and Future Work

Traditional topic models do not account for semantic regularities in language such as contextual relation of words as expressed in word embedding space. Therefore, G-LDA integrates the conventional topic model with word embeddings.

However, dealing with high dimensional data such as word vectors in embedding space requires costly computation. So, G-LDA employs faster sampling using Cholesky decomposition of covariance matrix and Alias Sampling.

On the other hand, Stochastic Variational Inference is much faster inference method than Markov chain Monte Carlo (MCMC) sampler such as Gibbs sampling and can deal with enormous dataset. Hence, we draw attention to SVI with expectation that SVI is also effective to handle high dimensional data.

In this paper, we have proposed to apply efficient inference algorithm based on SVI to the topic model with word embeddings. As a qualitative analysis, we have verified the coherence in the extracted latent topics through the experiments and confirmed that our model is able to extract meaningful topics as G-LDA is.

In the future work, we will observe perplexity convergence to evaluate the inference speed and the soundness of our model.

References

1. **Batmanghelich, K., Saeedi, A., Narasimhan, K., Gershman, S. (2016).** Nonparametric spherical topic modeling with word embeddings.
2. **Blei, D. M., Ng, A. Y., Jordan, M. I. (2003).** Latent Dirichlet allocation. J. Mach. Learn. Res., Vol. 3, pp. 993–1022.
3. **Das, R., Zaheer, M., Dyer, C. (2015).** Gaussian LDA for topic models with word embeddings. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, pp. 795–804.
4. **Hoffman, M., Bach, F., Blei, D. (2010).** Online learning for latent Dirichlet allocation. Advances in Neural Information Processing Systems, volume 23, Curran Associates, Inc.
5. **Hoffman, M., Blei, D. M., Wang, C., Paisley, J. (2012).** Stochastic variational inference.
6. **Hu, P., Liu, W. J., Jiang, W., Yang, Z. (2012).** Latent topic model based on Gaussian-LDA for audio retrieval. pp. 556–563.
7. **Hu, W., Tsujii, J. (2016).** A latent concept topic model for robust topic inference using word embeddings. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, pp. 380–386.
8. **Li, A. Q., Ahmed, A., Ravi, S., Smola, A. J. (2014).** Reducing the sampling complexity of topic models. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, pp. 891–900.
9. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality.
10. **Moody, C. E. (2016).** Mixing Dirichlet topic models and word embeddings to make lda2vec.

11. **Newman, D., Karimi, S., Cavedon, L. (2011).** External evaluation of topic models. ADCS 2009 - Proceedings of the Fourteenth Australasian Document Computing Symposium.
12. **Nguyen, D. Q., Billingsley, R., Du, L., Johnson, M. (2018).** Improving topic models with latent feature word representations.
13. **Paisley, J., Blei, D., Jordan, M. (2012).** Variational Bayesian inference with stochastic search.
14. **Rangarajan Sridhar, V. K. (2015).** Unsupervised topic modeling for short texts using distributed representations of words. Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Association for Computational Linguistics, pp. 192–200.
15. **Xun, G., Li, Y., Zhao, W. X., Gao, J., Zhang, A. (2017).** A correlated topic model using word embeddings. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 4207–4213.
16. **Yao, L., Zhang, Y., Wei, B., Jin, Z., Zhang, R., Zhang, Y., Chen, Q. (2017).** Incorporating knowledge graph embeddings into topic modeling. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, No. 1.
17. **Zhao, H., Du, L., Buntine, W. (2017).** A word embeddings informed focused topic model. Proceedings of the Ninth Asian Conference on Machine Learning, volume 77, PMLR, pp. 423–438.

Article received on 15/02/2018; accepted on 11/01/2020 .
Corresponding author is Kana Ozaki.

ISSN 2007-9737

Improving Question Analysis for Arabic Question Answering in the Medical Domain

Sondes Dardour, Hela Fehri, Kais Haddar

University of Sfax, MIRACL Laboratory,
Tunisia

{hela.fehri, kais.haddar}@yahoo.fr, dardour.sondes@yahoo.com

Abstract. Question analysis is a basic module in a question answering (QA) system, and its quality affects the performance of QA system. In this paper, we address the problem of Arabic question analysis in the medical domain where several specific challenges are met. The major challenging issue in processing Arabic medical question is the need for ambiguity resolution. Nevertheless, this issue has not been well studied in related works. Our question analysis uses dictionaries and transducers to analyze any medical question, factoid or complex. This module detects important elements of the question, including: different words in the question that identify what the user wants to ask for, and the nature of the expected answer. To identify well these elements a step of disambiguation is applied. Then, the words used in the question will be extended by adding new words that connect semantically to those in the question. Experimentation of the question analysis module of our Arabic medical question answering system show interesting results.

Keywords. Question answering, Arabic, disambiguation, medical domain, dictionary, transducer.

1 Introduction

Nowadays, due to the continuous exponential growth of information produced in the medical domain, and due to the important impact of such information upon research and upon real world applications, there is a particularly great and growing demand for Question Answering (QA) systems that can effectively and efficiently aid users in their medical information search [1].

QA system takes a question posted in natural language instead of a set of key-words, analyzes and understands the meaning of the question, and then provides the exact answer from a set of knowledge resources [2]. The QA system consists of three main processing modules, namely, question processing, passages retrieval processing, and answer processing. A question processing is the primary and basic source through which a search process is directed for answer. Therefore, an accurate and careful analysis to the question is required. Thus, question processing is the most fundamental module in any QA system, and the performance of its results significantly impacts on the following modules of information retrieval and answer extraction.

To our knowledge, proposed Arabic medical QA systems are so limited either in terms of their performance as well as in terms of the types of questions they are designed to answer. Moreover, the most attention in Arabic has been paid to answering factoid questions, in which the answer is a single word or a short phrase [3].

Ambiguity is a common phenomenon in human natural language. In QA, ambiguity is a critical challenge in extracting what the user looking for in his question. Therefore, ambiguity can cause confusion in interpretation of the question, and then impacts negatively the performance of the QA system.

In this paper, we propose a new approach to handle medical questions (factoid and complex questions) for the Arabic language. Moreover, our approach overcomes the ambiguity in the question processing module, an issue that has not been appropriately addressed in the field of Arabic QA.

The remainder of this paper is structured as follows. Section 2 presents the related works. Ambiguity problems are presented in section 3. Section 4 describes our approach. Section 5 deals with the experimentation carried out to evaluate the efficiency of our question analysis module. Finally, section 6 draws the main contributions and proposes further perspectives.

2 Related Works

The problem of answering questions formulated in natural language has been studied in the field of Information Retrieval (IR) since the mid-1990s [4]. However, unlike IR, the QA system returns simple and precise answer to a natural language question instead of a large number of documents [5, 6] . As we mentioned, the QA system is composed of three modules: question analysis, passage or document retrieval and answer extraction. Different QA systems may use different implementation for each module [7, 8]. In this section, we focus on some studies for the question analysis module.

Until now, very little effort was directed toward the development of QA system for the medical domain in the Arabic language, compared to other languages such as French and English. This is mainly attributed to the particularities of the medical domain and the language (see Section 3). The situation is further aggravated by the lack of linguistic resources and Natural Language Processing (NLP) tools that is available for Arabic [9, 10].

In an effort to achieve a better question analysis, [2] analyzed the question to extract type and category of desired answer whether it is a place, a quantity, a name or a date, which makes the answer extraction easier.

[10] analyzed Arabic questions by formulating the query, extracting the expected answer type, the question focus and the question keywords. The focus is the noun phrase of the question which the user wants to ask about. For instance, if the user's question is "What is the capital of Canada?" then the question focus is "Canada" and the keyword is "capital" and the expected answer type is a named entity for a location.

[11] analyzed the question by:

— Tokenization and normalization.

— Determining answer type by question words (When, What...)

— Named entity recognition.

— Focus determination by extracting the main named entity.

— Keywords extraction.

— Removing stop words using the Khoja stop list.

— Query expansion using the Arabic dictionary of synonyms. Named entities are not expanded to avoid ambiguity.

— Stemming by Khoja's Stemmer and named entities are not stemmed.

— Query generation of keywords into a boolean formula.

[3] made six steps to process Why-questions. They tokenized the question, then normalized it, then removed stop words (optional step). After that, they applied khoja's stemmer to obtain the root of each non-stop word in the question. Then, they used the extracted keywords to formulate and generate the query. Finally, they extended the list of keywords by including synonyms and words that share the same root.

The systems of [12] and [13] are developed for the medical domain in Arabic language. These systems analyze only factoid questions by extracting the topic and the focus of the question, and extracting named entities. The system of [12] classifies the questions into organization, location, person, viruses, diseases, treatment.

We can confirm, from literature reviews, that most Arabic QA systems ensure analysis of factoid questions. Nevertheless, there are few studies that have addressed the problem of answering complex questions. In addition, there are few works that have integrated semantic analysis and treated the medical field in the Arabic language, which makes the development of a new Arabic QA system is crucial.

3 Ambiguity

A study of different questions showed us the existence of several linguistic phenomena which can cause ambiguities in the question processing. Indeed, if we solve these problems, the errors will be so minimal and our system will be more relevant compared to existing Arabic QA systems.

3.1 Specific Arabic Difficulties

Arabic specific difficulties consist in its richness that needs special processing, which makes regular NLP systems, designed for other languages, unable to process it. One of the Arabic-specific difficulties is the lack of diacritics (i.e. kasra, fatha, damma), which leads to more ambiguous situations than any other language. This issue can be explained through the question من الذي قتل في أوغندا؟ (Who was killed in Uganda?).

The lack of diacritics in verb قتل (to kill) presents at least two cases for the question processing:

— قُتِلَ qutila which means that the question is "Who was killed in Uganda?", so قتل in this question means (was killed).

— قَتَلَ qatala which means that the question is "Who did kill in Uganda?", so قتل in this question means (kill).

Arabic language morphology is challenging when compared to other languages. This is because Arabic is a highly agglutinative and derivational language where a word token can replace a whole sentence in other languages. For example, for the question أيمكننا منع الجلطة؟ (Do we can prevent the clot?), the sentence "Do we can" can be expressed in one Arabic word أيمكننا which includes the verb يمكن (can), the prefix أ (do) and the pronoun نا (we). Therefore, extracting keywords from an Arabic question will be more complex than any other language. Furthermore, in a question like من الطبيبان اللذان منحا جائزة نوبل في الطب لعلاج السرطان؟ (Who are the two scientists who won the Nobel Prize in medicine for cancer treatment?), the user looks for the name of two persons (i.e. James P. Allison and Tasuku Honjo). In English, the system catches this user require through the word "two". In Arabic QA, this keyword is embedded in the word الطبيبان Alt abiybaAni (two scientists) thanks to the suffix ان Ani. Actually, the question above is just an example; the morphology of an Arabic word may contain multiple information (basic POS, number, gender, etc.) which are important for each module of Arabic QA. Unlike English and most Latin-based languages, Arabic does not have capital letters which makes Named Entity Recognition (NER) harder [14].

3.2 Specific Difficulties of Medical Domain

Apart from ambiguity in Arabic language, ambiguity also appears in medical terms. We observed that the more ambiguous terms are diseases names. For example, the term القمل means both an insect and a dermatological disease. This issue can be explained through the question ماهو القمل (What is a louse?); such system can extract the following answers:

1 القمل هو نوع من الحشرات الضارة التي تتغذى على دم الإنسان
(Louse is a kind of harmful insect that feeds on human blood)
Definition of an insect.

2 هو مرض يصيب فروة الرأس القمل
(Louse is a disease that affects the scalp)
Definition of a disease.

In fact, to extract the right answer, the system must understand the context. For example, in (2), the keyword مرض (disease) indicate that it is a definition of the disease القمل (Louse). Furthermore, in open-domain, the nature of the expected answer is known from the interrogative pronouns. For instance, in a When-question متى اكتشفت امريكا؟ (When America discovered?), the nature of the expected answer is a time. Nevertheless, in medical-domain, a When-question can indicate an age, a condition

Table 1. Ambiguity of the question متى (When)

Question	Translation	Question type	Expected answer
متى يتكون قلب الجنين؟	When is the fetus heart developed?	Wh-question	Age
متى يجب زيارة طبيب نفسي؟	When should visit a psychiatrist?	Wh-question	Condition
متى اكتشف مرض الزهايمر؟	When did Alzheimer's disease be discovered?	Wh-question	Time

Table 2. Collected questions

Question type	What	When	Where	Who	How many/much	How	Why
Number	85	53	42	38	45	39	48

or a time. Table 1 gives an example. To extract the correct answer, we must define a sequence of keywords which define the question and disambiguate it in the sense that it indicates what the question is looking for.

4 Proposed Method

The challenges discussed in the previous section make clear the need for new method to deal with Arabic medical QA. In addition, the most of previous studies are based on a superficial analysis of factoid questions (i.e. where, when, how much/many, who and what).

The originality of our approach lies in the disambiguation and the semantic analysis of factoid and complex questions (i.e. why and how to). In our proposal, the question analysis module is based on five steps as illustrated in Fig 1: Corpus study, Named Entity Recognition (NER), Stop word removal, Disambiguation, and Question Expansion (QE). In the first step, questions are gathered and studied to define the disambiguation patterns.

These patterns are transformed into transducers to process any type of medical question in Arabic language. Questions will be processed by the parallel steps (NER, Stop word removal, and disambiguation) using dictionaries, syntactic grammars, and morphological grammar in order to get some useful information. Finally, the last step will extend the extracted keywords.

4.1 Corpus Study

The need to have an Arabic corpus is a necessity for processing Arabic QA systems. Indeed, the questions are gathered from several sources, namely, discussion forums, frequently asked questions (FAQ) and some questions translated from Text REtrieval Conference (TREC). Currently, we collected 350 questions which contain seven categories (see Table 2). The questions are then subjected to an analysis step.

According to our study, we identify 158 question disambiguation patterns. Table 3 shows some patterns of the question متى (When). These patterns will be transformed into transducers to parse the questions.

4.2 Named Entity Recognition (NER)

The previous studies emphasize that the NER is important for all the QA system components. Indeed, the integration of a NER step will definitely boost our system performance because the answer of a factoid question is a named entity.

In our case, we developed our own NER tool especially formulated for our proposal. This step is based on dictionaries and transducers. We have considered five categories:

— Organ: Names of medical organs.

— Location: Names of location.

— Disease: Names of diseases, sickness, illness.

Table 3. Some question disambiguation patterns of the question متى (When)

Question	Expected answer
When<Verb>Fetus| Child |Infant?	Age
When <Verb><Noun>Child| Fetus | Infant?	Age
When <Verb><Noun><Condition>	Condition
When <Verb>< Trigger ><Virus>	Time
When <Verb><Virus>	Time

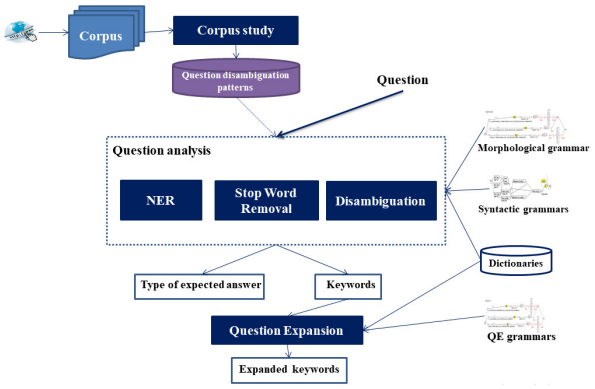


Fig. 1. Proposed method for question analysis module

```
ضَرَبَ,V+Tr+Correct+Salem+FLX=V_daraba1+DRV=N_daraba1:FlxDRV
نَـزَلَ,V+Tr+Correct+Salem+FLX=V_daraba1+DRV=N_daraba1:FlxDRV
وَلَـدَ,V+Tr+Sens_generer+FLX=V_allama4+DRV=N_allama4:Flx1
أَوْلَـدَ,V+Tr+Sens_naitre+FLX=V_akrama5+DRV=N_akrama5:FlxDRV
```

Fig. 2. Extract of dictionary

— Virus: Names of medical viruses.

— Treatment: Names of Treatments.

4.3 Stop Words Removal

This step removes the conjunctions, prepositions and interrogative pronouns. After removing the stop words, the important terms in the question will be remaining. In our proposal, the stop words are eliminated from the outputs of the syntactic transducers (see Fig 3).

4.4 Disambiguation

Our system is based on dictionaries and transducers. These resources allow us to disambiguate ambiguous words and the nature of the expected answer (Problems mentioned in the previous section).

4.4.1 Word Sense Disambiguation (WSD)

WSD process is required in application such as a QA application [15]. Some ambiguous words which have a different sense influence negatively the extraction of the correct answer. Let's take the following questions as an example:

1 متى يولد الدماغ الطاقة؟
(When does the brain generate electricity?)

2 متى يولد الجنين الذي يعاني من تشوهات خلقية؟
(When a baby who has congenital anomalies is born?)

As shown above, the verb يولد have the sense of "generate" in the question (1) and the sense of "born" in the question (2). To resolve this problem, as shown in our dictionary in Fig 2, each ambiguous word is associated with semantic feature to identify the sense of the entry (sens-generer, sens-naitre). This feature is used in the syntactic transducers (see Fig 3).

The WSD process allows also our system to define the correct stem. For instance, the stem of يولد ywld in question (1) is وَلَّدَ wal ada and in the question (2) is أَوْلَدَ >awolada.

4.4.2 Disambiguation of the Nature of the Expected Answer

For a reliable disambiguation, each defined pattern in the corpus study step is transformed into transducers. The identification of the nature of the expected answer is related to the focus of the question. For example, the transducer of Fig 3 describes the paths of the pattern "When<Verb>Fetus| Child |Infant?" (see Table 3). This transducer can analyze a question like متى يحبو الطفل؟ (When the child crawls?). The focus in this example is طفل (child), so the nature of the expected answer is "Age".

Fig. 3. Transducer of pattern When<Verb>Fetus| Child |Infant?

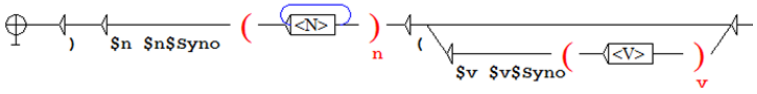


Fig. 4. QE transducer to extract synonyms

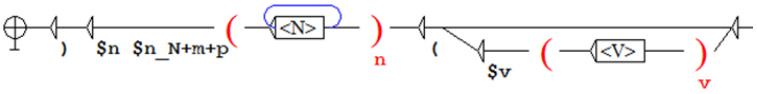


Fig. 5. QE transducer to extract different forms

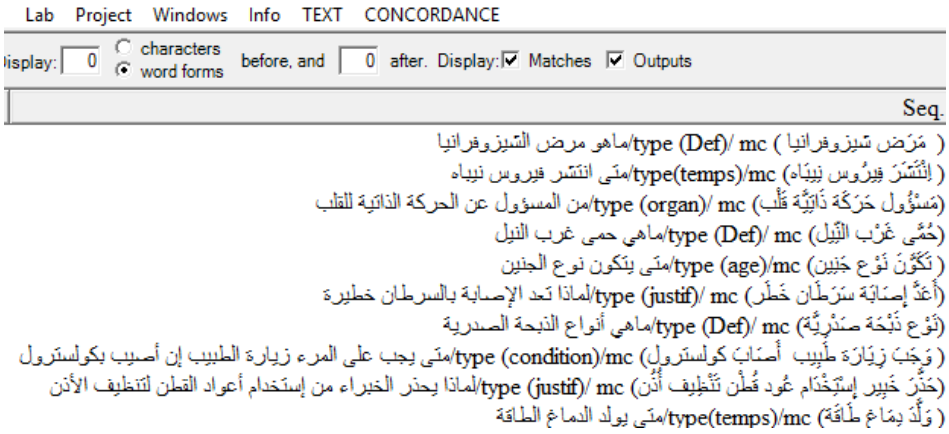


Fig. 6. Extract of concordance table

4.5 Question Expansion

Extraction only original question keyword is proved to have some limitations. To get rid of these limitations, we need to define the meaning the user looking for.

Therefore, in question expansion (QE), we extend the list of the exact words of the user's question by adding new words that connect semantically to those in the question. Since the documents may not contain the terms that the user used in his question, expanding question will increase the chance of getting the answer [16].

In the previous works, QE is achieved using Arabic WordNet [1]. In our dictionaries, the feature Syno (for synonyms) is used to expand questions. This feature is called in the QE transducer as shown in Fig 4.

After processing the question ما هو الاستخدام المناسب لواقي الشمس للرضيع؟ (What is the appropriate use of sunscreen for a baby?) with the previous steps, the transducer of Fig 4 can extract the synonym of استخدام <isotixdaAm which is استعمال <isotiEomaAl, the synonym of

[1] http://globalwordnet.org/arabic-wordnet/

مناسب munaAsib which is ملائم mulaAim and the synonym of رضيع raDiyEo which is طفل Tifl.

Expanding question can be applied also in order to overcome the situations where the Passage Retrieval (PR) module eliminates relevant passages containing other forms of the question keywords. The idea now is adding other forms of the keywords that share the same root (see Fig 5).

Let's continue with the same question ما هو الاستخدام المناسب لواقي الشمس للرضيع؟.

Thanks to QE process, the PR module can extract not only the passages that contain the keyword رضيع raDiyEo but also its broken plural form رضع ruDãEo. This process is applied also to extracted synonyms. Therefore, we consider each keyword with its synonym and its different forms since the QE would theoretically generate all these terms.

The expanded list of terms extracted from the question will be sent to the PR module to extract the passages that may contain the answer.

5 Experimentation and Evaluation

In our proposal, linguistic resources are built with the linguistic platform NooJ [17]. We conduct a set of experimentation to evaluate the performance of our question analysis module. Therefore, we exploit a test corpus which contains 399 questions. For each question type (ما "What", متى "When",أين "Where", من "Who", كم "How many/much", كيف"How", لماذا"Why") a set of 57 questions is used.

The results of applying the transducer that extracts the type of the expected answer and keywords are illustrated in Fig 6. This transducer allows the NER, stop words removal, and disambiguation. Then, the keywords are expended by the QE transducers.

After applying the analysis on the test corpus using our linguistic resources, we obtain the results illustrated in Table 4.

Table 4 shows that the disambiguation process enhances the F-Measure by 28%. It is then concluded that by reducing ambiguity, especially when processing the medical domain in the Arabic language, the obtained results will be increased.

Table 4. Summarizing the measure values

Method	Without disambiguation	With disambiguation
Precision	66%	93%
Recall	58%	87%
F-Measure	61%	89%

Errors are often due to the problem in writing some Arabic letters such as the letter ا "A" which can also be writing like أ > or آ | or إ <. For example, in some question, we can find the word "inflammation" written like التهاب AlotihaAbo or إلتهاب <ilotihaAbo. To resolve this problem, we need to rewrite the question by unifying all variants of a letter into a single form. Furthermore, the presented errors in the question analysis are due to dictionaries' coverage that must be improved and the complexity of some questions that requires special handling techniques.

6 Conclusion

In the present paper, we have developed a question analysis module (QAM) for our system to analyze an Arabic medical question.

Our QAM is mainly concerned with the identification of four factors, namely, keywords extraction, disambiguation, question expansion, and nature of the expected answer extraction. This analysis of question allows extracting all the necessary information that will be used as inputs for the other QA components. Our proposed method achieves satisfactory results.

In the future work, we seek to add a pre-processing to normalize the question. We also seek to improve our linguistic resources by adding new terms in the dictionaries.

References

1. **Athenikos, S. J., Han, H. (2010).** Biomedical question answering: A survey. Computer methods and programs in biomedicine, Vol. 99, No. 1, pp. 1–24. DOI: 10.1016/j.cmpb.2009.10.003.

2. **Hammo, B., Abuleil, S., Lytinen, S., Evens, M. (2004).** Experimenting with a question answering system for the Arabic language. Computers and the Humanities, Vol. 38, No. 4, pp. 397—415. DOI:10.1007/s10579-004-1917-3.

3. **Azmi, A. M., Alshenaifi, N. A. (2017)**. Lemaza: An Arabic why-question answering system. Natural Language Engineering, Vol. 23, No. 6, pp. 877–903. DOI: 10.1017/S1351324917000304

4. **Verberne, S. (2010)**. In Search of the Why. PhD Thesis, University of Nijmegen, The Netherlands.

5. **Kanaan, G., Hammouri, A., Al-Shalabi, R., Swalha, M. (2009).** A new question answering system for the Arabic language. American Journal of Applied Sciences, Vol. 6, No. 4, pp. 797–825.

6. **Trigui, O., Belguith, L. H., Rosso, P. (2010).** DefArabicQA: Arabic definition question answering system. Workshop on Language Resources and Human Language Technologies for Semitic Languages, 7th LREC, Valletta, Malta, pp. 40–45.

7. **Benajiba, Y., Rosso, P., Gómez-Soriano, J. M. (2007).** Adapting the JIRS passage retrieval system to the Arabic language. Lecture Notes in Computer Science, vol 4394, Springer, Berlin, Heidelberg. pp. 530–541. DOI: 10.1007/978-3-540-70939-8_47.

8. **Ezzeldin, A. M., Shaheen, M. (2012).** A survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends. Proceedings of the 13th International Arab Conference on Information Technology (ACIT´12), pp. 1–8.

9. **Abouenour, L., Bouzoubaa, K., Rosso, P. (2008).** Improving Q/A using Arabic Wordnet. Proceedings of the 2008 International Arab Conference on Information Technology (ACIT'2008), Tunisia, December.

10. **Brini, W., Ellouze, M., Mesfar, S., Belguith, L. H. (2009).** An Arabic question-answering system for factoid questions. IEEE International Conference on Natural Language Processing and Knowledge Engineering. NLP-KE´09, pp. 1-—7. DOI: 10.1109/NLPKE.2009.5313730.

11. **Abdelbaki, H., Shaheen, M., Badawy, O. (2011).** ARQA high performance Arabic question answering system. Proceedings of Arabic Language Technology International Conference (ALTIC).

12. **Bessaies, E., Mesfar, S., Ghzela, H. B. (2018).** Processing medical binary questions in standard Arabic using nooJ. Lecture Notes in Computer Science, Vol. 10859, Springer, Cham. DOI: 10.1007/978-3-319-91947-8_19.

13. **Ennasri, I., Dardour, S., Fehri, H., Haddar, K. (2017).** Question-response system using the nooJ linguistic platform. In: **Mbarki, S., Mourchid, M., Silberztein, M., eds.** Formalizing Natural Languages with NooJ and Its Natural Language Processing Applications. NooJ´17. Communications in Computer and Information Science, vol 811. Springer, Cham. DOI: 10.1007/978-3-319-73420-0_16

14. **Shaheen, M., Ezzeldin, A. M. (2014)** Arabic question answering: systems, resources, tools, and future trends. Arabian Journal for Science and Engineering, Vol. 39, No. 6, pp. 4541–4564.

15. **Ferrandez, S., Roger, S., Ferrández, A., Aguilar, A., López-Moreno, P. (2006).** A new proposal of Word Sense Disambiguation for nouns on a Question Answering System. Advances in Natural Language Processing. Research in Computing Science, Vol. 18, pp. 83–92.

16. **Al-Chalabi, H., Ray, S., Shaalan, K. (2015).** Semantic based query expansion for Arabic question answering systems. Arabic Computational Linguistics (ACLing´15), First International Conference on IEEE, pp. 127–132. DOI: 10.1109/ACLing.2015.25.

17. **Silberztein, M. (2018).** Using linguistic resources to evaluate the quality of annotated corpora. Proceedings of the First Workshop on Linguistic Resources for Natural Language, pp. 2–11.

Article received on 20/02/2019; accepted on 09/01/2021.
Corresponding author is Sondes Dardour.

ISSN 2007-9737

Probabilistic Error Detection Model for Knowledge Graph Refinement

Manuela Nayantara Jeyaraj, Srinath Perera,
Malith Jayasinghe, Nadheesh Jihan

WSO2, Mountain View,
United States

manuela.n.jeyaraj@mytudublin.ie

Abstract. Knowledge graphs are widely used in information queries. They are built using triples from knowledge bases, which are extracted with varying accuracy levels. Accuracy plays a key role in a knowledge graph, and knowledge graph construction uses several techniques to refine and remove any inaccurate triples. There are many algorithms that have been employed to refine triples while constructing knowledge graphs. These techniques use the information about triples and their connections to identify erroneous triples. However, these techniques lack in effective correspondence to human evaluations. Hence, this paper proposes a machine learning approach to identify inaccurate triples that correspond to actual human evaluations by injecting supervision through a subset of crowd-sourced human evaluation of triples. Our model uses the probabilistic soft logic's soft truth values and an empirical feature, the fact strength, that we derived based on the triples. We evaluated the model using the NELL and YAGO datasets and observed an improvement of 12.56% and 5.39% in their respective precision. In addition, we achieved an average improvement of 4.44% with the F1 scores, representing a better prediction accuracy. The inclusion of the fact strength augmented the modeling precision by an average of 2.13% and provided a higher calibration. Hence, the primary contribution of this paper is the proposal of a model that effectively identifies erroneous triples, aligning with high correspondence to actual human judgment.

Keywords. Information extraction, knowledge graph, machine learning, probabilistic soft logic.

1 Introduction

A knowledge extraction pipeline takes in data, converts it to a knowledge base, and finally provides the outcome of knowledge extraction as a Knowledge Graph. A single link or an edge in a knowledge graph is the relationship that connects a subject to its object.

The subject, object and the relationship together are known as the triple. Knowledge base is a collection of triples while knowledge graph adds missing connections and confidence measurements to those connections. We extract triples from various sources such as free text, database and knowledge bases, using NLP techniques such as part-of-speech tagging, tokenizing, stemming, and so on.

These extracted triples have different levels of accuracy. If inaccurate information is incorporated into the knowledge graph, queries based on that graph can return erroneous responses. Hence, this is a current concern with regard to knowledge graphs.

In order to completely identify the accuracy of knowledge graphs, the most trivial method is to perform a complete manual check on all the facts used for the graph. Yet this is rather expensive and exhaustive. Hence, most of the existing solutions resolve to automated methods and pre-process the knowledge bases for erroneous triples [23].

These techniques known in literature, measure the accuracy of triples, considering their neighboring triples. They can be based on heuristics, building vector space models and computing word scores based on the tf-idf weights of vectors [20], or averaging the ontology coverage based on the frequency classes [29] (further explained in section 6). However, these automated methods have not satisfactorily addressed the

inaccurate facts and the lack of correspondence to actual human evaluations.

Therefore, this paper leverages the advantages of automated methods and the correspondence to human judgment from manual methods, to propose a semi-approach for evaluating the accuracy of a large set of triples based on the human evaluation of a subset of triples.

We use machine learning to verify the correctness of the triples based on a set of features; subject, object, predicate and the probabilistic soft truth confidence values. We further improve our technique based on a novel, empirical feature, which we term as the "fact strength". We use human evaluated data as the target variable for training and use the model to gauge the accuracy of triples.

Accordingly, as the primary contribution of this paper, we identify machine learning as a suitable candidate for further refining the knowledge bases or knowledge graphs based on a partially evaluated dataset. We propose classification models to identify the erroneous triples that correspond to actual human evaluation. We evaluated the models using the Never-Ending-Language-Learner (NELL)[1] and YAGO[2] datasets, and observed a 12.56% and 5.39% of improvement in the precision, respectively.

In addition, we achieved an average improvement of 4.44% in the F1 scores, representing a better prediction accuracy. Introducing the fact strength as a feature, provided an average positive augmentation of 2.13% in the precision to achieve the above improvement. Thus, this model addresses the use-case of effectively removing erroneous triples from knowledge graphs.

This paper is outlined with further details on the background, proposed solution, evaluation, discussion of results and the related work, with regard to addressing our contribution.

[1]The NELL knowledge base : http://rtw.ml.cmu.edu/rtw/

[2]The YAGO dataset: https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

2 Background

2.1 Knowledge Graphs

Since the age of the internet, the retrieval and storage of information has become a vital part of all activities that are being conducted both online as well as offline. As such, a massive amount of data is being generated by seconds. According to Forbes, 2.5 quintillion bytes of data are being generated every single day[3]. In the past, databases were considered a sufficient store for information [18].

Further, as the amount of content grew exponentially, data warehouses came into context [13]. Consequently, the multivariate nature of content being produced, called forth the need for more sophisticated methods to retrieve data from these content and create a more generalized storage unit, ergo the concept of knowledge bases emerged [11]. Knowledge bases are built on an ontology based storage of information or so-called 'facts' [10] and consists of 2 major components : the interface engine and the knowledge repository. The interface engine serves as a search engine to browse for information stored in the repository.

Searching for facts is enabled through classified ontologies. Based on the sense in which ontologies are used, their definition varies [15]. Considering the Knowledge Engineering domain, we adhere to Gruber's definition of an ontology [14] : A representational identification of a vocabulary, for a particular domain. As the knowledge base learns its facts from various sources, it classifies the learned facts under ontologies.

Such knowledge bases that have been of primal use to mankind are explicated here. (1) Freebase [4], contained 1.9 billion triples or learned-facts[4], before being deprecated, as Meta-web, the developer of Freebase, sold Freebase to Google in 2010. It harnessed its data from the semantic web [3] and Wikipedia articles. (2) DBpedia [2] extracts information from Wikipedia and builds structured

[3]Amount of data generated on a daily basis: https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/\#6ef79f2760ba\

[4]Data Dumps of Free base : https://developers.google.com/freebase/

facts that can be queried upon. (3) YAGO [30] does not directly scour Wikipedia articles.

Alternatively, it leverages the category pages available in Wikipedia[5]. YAGO represents the learned facts in the form of the Web Ontology Language (OWL) [24]. OWL uses the Resource Description Framework Schema (RDFS) [1], which only describes the relationship between facts using unrefined semantics. Hence, YAGO was developed as a refinement of RDFS. (4) The Never Ending Language Learner (NELL), which was developed at the Carnegie Mellon University, adopts the key difference between human-learning and machine-learning, continuously learning facts since 2010. Due to this learning process, NELL accumulates both correct as well as incorrect facts/beliefs.

An example of an accumulated fact in NELL is: *"Astoria is a city that lies on the river Columbia (river)"*. In this case, "*Astoria*" is the subject that relates to the object, i.e., "*Columbia*", through the predicate or relationship, "*lies on*". Hence, this relationship connects 2 entities: the subject and object, to form a triple, which is a fact learned within the knowledge base. Currently NELL has accrued 50 million such candidate beliefs.

Notwithstanding, knowledge bases lack in the sense of inter-connectivity between entities. Though a single fact's entities: the subject and object, are connected, the way in which the rest of the entities connect with one another cannot be directly observed in plain, flat knowledge bases, consequently giving rise to Knowledge Graphs.

"Knowledge Graph" was a term coined by Google as it introduced Google's Knowledge Graph in 2012 [9]. Knowledge graphs are built of triples where, the relationships or predicates form the edges between various entities, assembling a massive network of interconnected entities thus, providing more context into how different entities interact and maintain a relationship. The edges, which are the relationships in knowledge graphs, are inferred based on statistical relational learning (SRL). Probabilistic models in SRL are used to compute confidences to justify how far these inferred relationship edges hold in the graph. One such probabilistic model, that we use in our solution, is the Probabilistic Soft Logic (PSL) [7].

[5]Information regarding the detailing of a category, time-stamps and sort keys can be fetched by querying in the api

2.2 Probabilistic Soft Logic

The probabilistic soft logic is a statistical relational learning framework that infers a soft truth value which serves as a confidence for each fact based on the joint probabilistic reasoning in its relational domain [17].

It uses first order logic and a weight learning of rules as it projects the most probable explanation for inferences in the form of a convex optimization [6] and estimates a soft truth value which ranges between [0,1] as opposed to a restricted 0 or 1. PSL is best resorted to, in computing how far a fact or belief holds. It defines a set of general rules such as transitive, commutative, associative, etc. PSL rules are of the following form:

$$w : R_1(A, B) \wedge R_2(B, C) \rightarrow R_3(A, C), \qquad (1)$$

where R_1, R_2 *and* R_3 are the relationships, *A, B* and *C* are entities, and *w* is the weight of the rule. As we apply constants or real-world entities from the facts, onto the rule, they become ground rules. And this process is appropriately known as *"grounding"* as shown in (2):

$$0.9 : livesIn(Claire, Paris) \wedge spouse(Claire, Blake) \rightarrow livesIn(Blake, Paris). \qquad (2)$$

This states that if *Clair* lives in *Paris* and *Clair* is the spouse of *Blake*, then it implies that *Blake* could also be living in *Paris*. This rule holds with a weight of 0.9 with the belief that spouses are more likely to live in the same place. Here, *livesIn(Claire, Paris)*, *spouse (Claire, Blake)* and *livesIn(Blake, Paris)* are considered as atoms, *x*.

In the Knowledge Engineering community, these atoms can be addressed as triples as well. Here, the implied atom, which is on the right side, becomes the head of the relationship and the ones on the left are the body. Some of these triples can be known triples, with previously observed soft

truth values and the others may be unknown triples whose soft truth values are previously unknown:

$$\begin{aligned} \alpha_x = \tau, x \in \mathbb{S}^+, \\ \alpha_x = p(x_e), x \in \mathbb{S}^-. \end{aligned} \tag{3}$$

where α_x is the soft truth value of the atom *x*, $\mathbb{S}^+$ is the set of known triples, $\mathbb{S}^-$ is the set of unknown triples, τ is the confidence of the known triple *x*, and $p(x_e)$ is the conditional probability of atom *x* for its embedded soft truth values.

When 2 entities e_1 and e_2 are in a relationship *r*, as an atom *x*, and their soft truth value is previously unobserved, then we derive their initial soft truth value using $p(x_e)$ which is the conditional probability represented by $p(e_1—e_2)$ as shown in (4):

$$p(e_1|e_2) = \frac{count_{facts}(e_1, e_2)}{\Sigma_{r \in R}\, count_{facts}(r, e_2)}. \tag{4}$$

Here, *R* is the set of all the relationships in the domain. This is how we derive the soft truth values for the previously unknown atoms/facts of a ground rule [22]. Once, the soft truth values for the atoms are inferred, the logical connectives: $\wedge$, $\vee$ and $\neg$ of ground rules need to be relaxed using a normalization by Lukasiewicz t-norms [7]:

$$\begin{aligned} a \overset{\sim}{\wedge} b = max\{0, a + b - 1\}, \\ a \overset{\sim}{\vee} b = min\{1, a + b\}, \\ \overset{\sim}{\neg} a = 1 - a. \end{aligned} \tag{5}$$

This normalization is performed to enumerate an aggregated soft truth value for the entire body of the atom. With a soft truth value for the body, r_{body}, deduced using the Lukasiewicz normalization in (5), and a soft truth value for the head, r_{head}, obtained using (4), PSL models a statistical relationship based inference using the following concept.

If r is a rule in PSL such that $r_{body} \rightarrow r_{head}$ and given an inference *I* that is grounded to *r*, *r* is satisfied only if $I(r_{body}) \leq I(r_{head})$. If the rule is satisfied, the distance to satisfaction (*d*), will be 0. On the other hand, if it fails to be satisfied, the degree to which it deviates from satisfaction will be captured by the distance to satisfaction for that Inference, $d(I)$, as shown in (6):

$$d(I) = \ max\{0,\ I(r_{body}) - I(r_{head})\}. \tag{6}$$

Furthermore, given the set of ground atoms, a distribution will be built as the probability density function, $f(I)$ [17]:

$$\begin{aligned} f(I) = \frac{1}{Z} exp[-\sum_{r \in R} \lambda_r (d(I))^p]; \\ Z = \int_I exp[-\sum_{r \in R} \lambda_r (d(I))^p]. \end{aligned} \tag{7}$$

In (7), R is the finite set of all defined rules, Z is the discrete Markov random field normalization constant for a continuous value, λ_r is the assigned weight for the rule r, $d(I)$ is the distance to satisfaction for the inference I, and p indicates the loss function with either '1' which supports interpretations that fully satisfy a single rule, willing to suffer a higher distance to satisfaction for the other rules or '2' which is a quadratic loss function that attempts to support all the rules to some extent.

Here, the optimal distribution will be the conclusively inferred, soft truth value of the implied fact, i.e, the head atom, based on the statistical relationship between entities. As such, all the facts are grounded as ground rules such that they become the implied relationship or head atom in the rule, and the optimal distribution's probability will be rendered as that fact's soft truth value or confidence from each distribution generated based on prior computations.

These soft truth values are mere indications of how far the system is confident in its relational inference. But we do not have the means to clearly classify a fact as true or false according to human judgment by solely using these confidences. Hence, PSL constructs knowledge graphs with an accumulation of all these facts, along with their inferred confidence scores. This can build a noisy graph with inaccurately inferred facts.

Since there have been a substantial amount of work conducted with regard to missing data (explained further in the Related Work), in this paper, we attempt to address the removal of false data from massive knowledge graphs using

PSL and a small subset of human evaluated fact truths. The core difference between previous models that relate to error detection in knowledge graphs and our model is that, the validity of the fact is modeled with correspondence to actual human judgment, instead of solely relying on system generated confidences. And we discover a pattern to restrictively label facts as true or false, using a machine learning approach. The next section explains our solution model that was developed to achieve the above same.

3 A Model to Address Erroneous Triples in Knowledge Graphs

In order to identify the erroneous triples, we propose a supervised machine learning approach based on classification techniques to predict the accuracy of a triple. Our initial experiments were based on a basic features-set; the triples' subject, predicate, object, PSL's soft truth values and the human evaluations.

Since the subject, predicate and object are words, we encode to obtain their tf-idf as the feature to the models. However, the PSL soft truth value is directly taken as a feature to incorporate the confidence of each fact.

Later we extend our models by adding the fact strength as a feature. As the dependent variable to the classifiers, we use the human-evaluated score for each fact during the training. For supervision, we need this human evaluation.

Buhrmester et al. have shown the use of crowd-sourcing to generate human evaluated task sets for assessing the performance of various knowledge graph identification tasks [8]. Hence, we propose using crowd-sourcing to evaluate a randomly selected subset of triples from the complete knowledge base, using those triples for training the models.

Moreover, we extend our solution model to introduce a novel empirical feature to quantify the importance of each triple. This feature was a derivative of the number of relationships or interaction between the subject-predicate-object triple. We term this as the ***fact strength***, φ, for referential purposes. Equation (8) shows how we compute φ:

Table 1. Sample dataset of the triple fields

Fact id	ontology : subject	predicate	ontology : object
1	person:leonardHofstader	livesIn	place:california
2	person:sheldonCooper	isFriendOf	person:leonardHofstader
3	person:sheldonCooper	isSpouseOf	person:amyFarrahFowler
4	person:sheldonCooper	worksWith	person:leonardHofstader
5	person:amyFarrahFowler	livesIn	place:california
6	person:leonardHofstader	isFriendOf	person:AmyFarrahFowler

$$\varphi = \frac{(\eta_{spd} \times \eta_{sd}) + (\eta_{opd} \times \eta_{od})}{\eta_{so}}. \tag{8}$$

For each fact or belief *b* in the dataset, such that, $b \in \mathbb{B}$, where $\mathbb{B}$ is the finite set of all beliefs within the domain, we compute η_x. The following explicates the variations of η_x used in equation (8):

$\eta_{spd} \rightarrow$ Number of times the `ontology of the subject` and the `predicate` appear in a fact within the dataset.
$\eta_{sd} \rightarrow$ Number of times the `subject` and the `predicate` appear in a fact within the dataset.
$\eta_{opd} \rightarrow$ Number of times the `ontology of the object` and the `predicate` appear in a fact within the dataset.
$\eta_{od} \rightarrow$ Number of times the `object` and the `predicate` appear in a fact within the dataset.
$\eta_{so} \rightarrow$ Number of times the `subject` and the `object` appear in a fact within the dataset.

For example, consider we have a dataset with 6 facts as shown in Table 1. However, in reality there are millions of facts.

According to the sample dataset in Table 1:
η_{spd} = 2, since the subject's ontology, *person*, and the predicate, `livesIn`, occur twice throughout the dataset.
η_{sd} = 1, since the subject, *leonardHofstader* and predicate, *livesIn*, occurs only once throughout the dataset.
η_{opd} = 2, since the object's ontology, `place`, and the predicate, *livesIn*, occur twice throughout the dataset.
η_{od} = 2, since the object, *california*, and the predicate, *livesIn*, occur twice throughout the dataset.
η_{so} = 2, since the subject, *leonardHofstader*,

and the object, *california*, occur twice throughout the dataset.

Thus, the fact strength will be computed as:

$$\varphi = \frac{(2 \times 1) + (2 \times 2)}{2}, \\ \varphi = 3. \tag{9}$$

Hence, the fact strength for the first fact in the dataset will be 3. This is an abstract example of how the derived, empirical feature φ is calculated for each and every fact. Nevertheless, in large datasets such as the ones we have used to train and test, the fact strength can range to a greater value such as 200 and above. However, the fact strength will always be a positive value as $\varphi \geq 1$.

We extend our models to adopt the fact strength (computed using the complete knowledge base) as a feature apart from the initial set of features.

During the preparation of the dataset, we computed the PSL-stv and the fact strength for the dataset, keeping the crowd-sourced predictions as the target variable. The inclusion of the fact strength showed an improvement in the precision, F1 and recall of the predictions, varying based on the type of classifier used.

4 Evaluation

For the training process, we used the NELL dataset which is a universal standard when it comes to knowledge base and knowledge graph experiments. The main intent of the NELL dataset is to expand its knowledge base by iteratively learning facts. It uses the facts that it had learned through previous experience to generate newly learned candidate beliefs/facts. Also, our selection of NELL was based on the consideration that NELL has been constantly learning facts and, therefore is up-to-date on the facts that it had learned, without being deprecated. With any other dataset, the time-value of information will have to be considered for the temporal validity of information. NELL also accumulates its facts from a wide array of sources, with confidences based on the sources' reliability as understood from previously learned facts. We extracted a subset of NELL that consists of around 2000 triples, along with their crowd-sourced evaluations.

Another dataset that we opted for evaluation purposes, is the YAGO dataset which is often used as the test set along with the NELL test dataset. YAGO fetches its facts from Wikipedia as well as WordNet. The classified labeling of the entities and the properties of YAGO facts, enables the easy access to its ontologies. In addition to that, scouring Wikipedia articles through their category pages allows YAGO to incorporate an extensive knowledge of a domain, based on its sub domains. Being triple based, YAGO bears reference to the time and location of the source from which its facts were derived. Hence, we adhere to the NELL and YAGO datasets based on their aforementioned advantageous features. As such, we extract around 1400 triples from the YAGO dataset for our experiments.

The crowd-sourced evaluations for both the datasets were obtained using the Amazon Mechanical Turk[6]. Consequently, the accumulated crowd-sourced outcomes are rendered in the form of 1 and 0 for each fact, respectively indicating a fact being true or false according to human evaluators. These results were more sought after as the evaluators were of diverse demographics [8].

As performed in [12] and [19], we used the 70:30 split on the NELL dataset to proceed with the training.

Since this prediction is a classification problem, we chose the Support Vector Classifier, Stochastic Gradient Descent and Random Forest Classifier to identify the optimal classifier.

We use the rbf kernel as a default for the classifiers as it is a stable kernel that is invariable to translations. In the case of evaluating $M(a, b)$, the rbf kernel will compute the same value, that is, $M(a, b)$ for a translation such as $M(a + x, b + x)$, where x is a constant vector for the dimension, such that it aligns with the inputs. The rbf kernel achieves this invariability by observing the difference between the computed vectors. Further, we maintain the default hyper parameters across all the classifiers.

[6] Amazon Mechanical Turk: https://www.mturk.com/

4.1 Experiments

Initially we investigate the effect of the basic features (tf-idf of subject, predicate and object, PSL soft truth values, and the human evaluations) to identify the erroneous facts with the aforementioned classifiers. The evaluation was performed on both the NELL and YAGO datasets. We have presented these observations in the first part of this section.

4.1.1 Experiment 1

After training the model on the training dataset from NELL, we evaluated the test dataset of NELL using the trained model. These results are depicted in Figure 1.

The Baseline here, is the performance index values for the triples, based on solely their soft truth values. For example, if we assume that all the triples with a soft truth value less than or equal to 0.1 are false, and all those with a soft truth value greater than 0.1 are true, and evaluate our prediction accuracy against actual human evaluations, then the performance index values obtained for the precision, recall and f1 scores will be considered the baseline for the threshold value of 0.1.

According to [26], we chose the threshold of $\theta \geq 0$ for the PSL soft truth values θ, based on the consideration that this is the threshold that maximized the F1 score for our PSL evaluations. Hence, our baseline for the precision is 0.828, recall is 0.910 and f1 is 0.867.Comparing against this baseline, we evaluated to see if the classifiers showed a proven improvement.

Please note that the performance index values displayed in the table in Figure 1 are rounded off to 3 decimal points, while the graph adheres to the original values.

According to Figure 1, all the classifiers show an improvement in the precision, whereas the Random Forest Classifier obtains the optimal precision of 0.927, as opposed to the baseline of 0.828. This indicates a 11.96% of improvement in the precision index for the classifier. However, the test accuracy was evaluated in terms of the improvement achieved for the F1 score. From a baseline of 0.867 to an augmented 0.920, the Random Forest Classifier displays a 6.11% improvement in the prediction accuracy for the NELL dataset. However, the recall drops for the support vector classifier and the stochastic gradient descent compared to the baseline. The Random Forest Classifier achieves slightly better recall and significantly outperforms the baseline in terms of the F1 score.

4.1.2 Experiment 2

We applied the trained models on the YAGO dataset. The performance index values rendered for this experiment are shown in Figure 2.

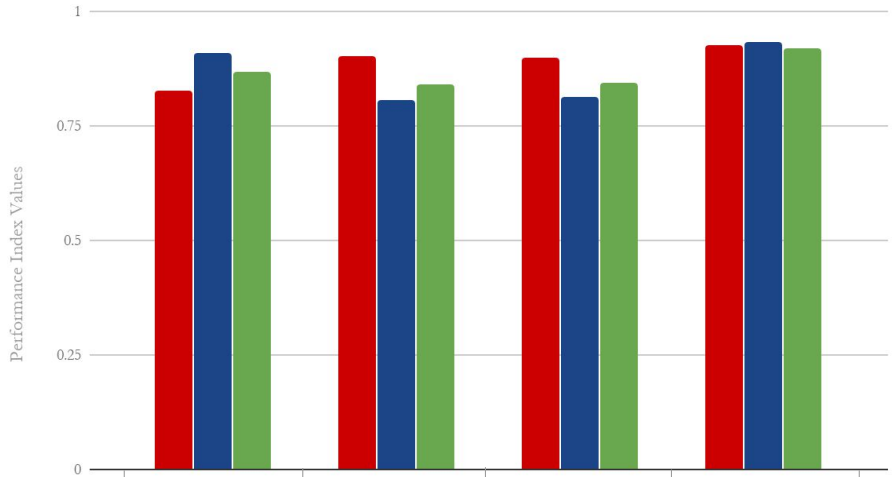
According to this, the Stochastic Gradient Descent holds a slightly higher precision of 0.945, compared to the the random forest classifier that showed a precision of 0.943. Hence, the stochastic gradient descent and the random forest classifier showed respective precision improvements of 1.94% and 1.73%.

Here, we achieve an F1 score of 0.957 from a baseline of 0.944, with the random forest classifier which renders a 1.38% improvement in the test accuracy, in terms of the F1 score.

Considering the above two experiments, with the sole feature-set of the triples (subject, predicate and object), the PSL confidences, and the human evaluations, we were able to achieve an average precision improvement of 6.85%. Consequently, we were able to arrive at an observation that the random forest classifier displayed an enhanced test accuracy with an average improvement of 3.75%.

During these experiments, the Random Forest Classifier performed with the optimal precision and descent recall values.

Furthermore, we extend the experiments to assess the accuracy gain of the models with the feature that we introduce, the fact strength. The following part of this section will illustrate the observed performance of the models after incorporating the fact strength in assessing erroneous triples.



	Baseline	Support Vector Classifier	Stochastic Gradient Descent	Random Forest Classifier
Precision	0.828	0.901	0.899	0.927
Recall	0.910	0.807	0.813	0.933
F1	0.867	0.840	0.844	0.920

Fig. 1. Performance Index Values for the NELL dataset(experiment 1)

4.1.3 Experiment 3

Here, we added the derived feature, that is, the fact strength, during the training process and performed the above same for the NELL dataset, now with an extended feature-set (subject, predicate, object, PSL soft truth values, fact strength and the human evaluations). This is shown in Figure 3.

Compared to the results obtained in Figure 1, the new models, shown in Figure 3, perform better, with the Random Forest Classifier achieving a precision of 0.932. Comparing this against our baseline showed that the precision had increased by 12.56%.

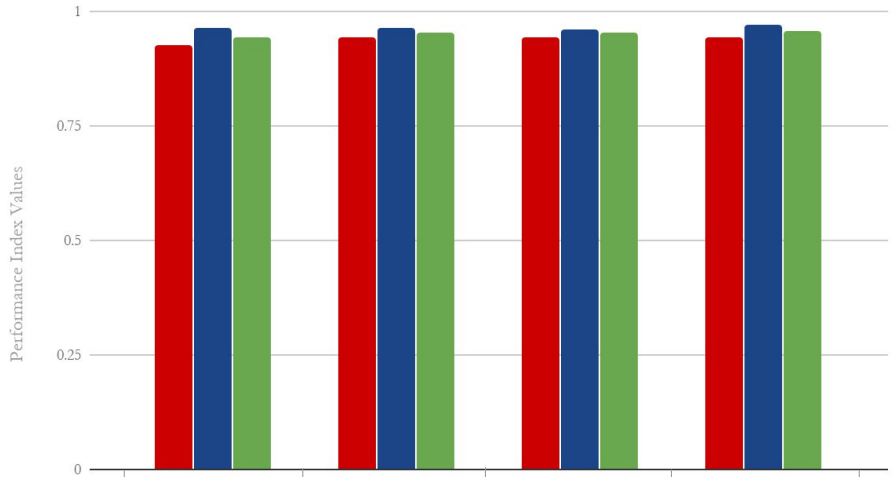
This indicates that the derived fact strength positively influences the classification based on the precision and provides a better recall of 0.936 as well.

Subsequently, we analyzed the F1 score for the datasets in order to garner a complete picture of the classifiers' accuracy. As such, with the introduction of the fact strength, the random forest classifier produced an F1 score of 0.922 from its baseline of 0.867, giving a test accuracy improvement of 6.34%.

Solely considering the F1 score to verify the impact of the fact strength, we observed only a trivial raise of 0.23%.

However, viewed together with the enhancements in the precision and recall, we can safely derive that the random forest classifier displays a better test accuracy on the whole. Furthermore, we observed a significant drop in the stochastic gradient descent, where its F1 score plummeted to 0.300.

With the precision at a recognizable 0.908, the recall of 0.256 suggested that the stochastic gradient descent was biased towards the false negatives and misinterprets more of the the false facts. Hence, as a result, the F1 depreciated as well.



	Baseline	Support Vector Classifier	Stochastic Gradient Descent	Random Forest Classifier
Precision	0.927	0.943	0.945	0.943
Recall	0.963	0.965	0.961	0.971
F1	0.944	0.954	0.953	0.957

Fig. 2. Performance Index Values for the YAGO dataset (experiment 2)

4.1.4 Experiment 4

We conducted the above same experiment on the YAGO dataset with the extended set of features, including the computed fact strength, and the results are depicted in Figure 4.

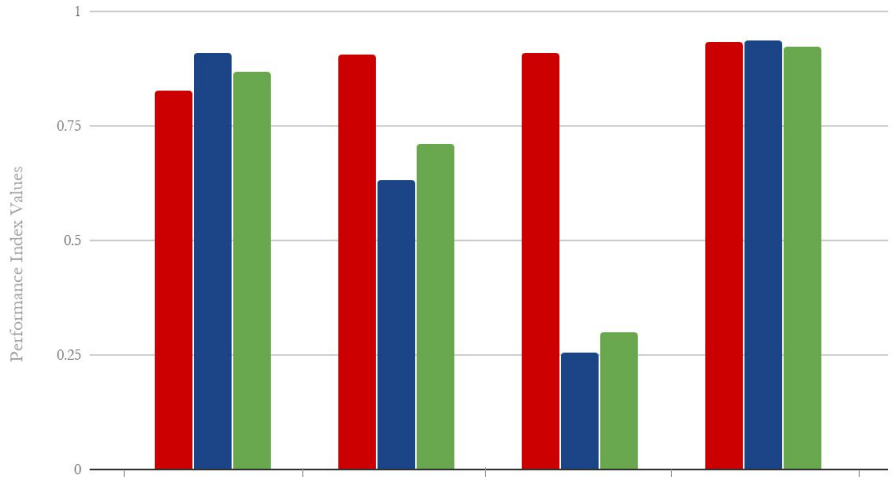
Experimenting the same on the YAGO dataset had a much better response to the fact strength feature as it augmented the precision by 3.61%. Also, the trivial improvement in the recall suggested that the true positive rate had increased. With most of the facts in this dataset being evaluated as true facts through human judgment, we were able to obtain an improved high F1 score of 0.968 as opposed to the baseline 0.944, through the Random forest classifier. All the classifiers gain an improvement over the precision baseline of 0.927 and F1 baseline of 0.944. This illustrated a test accuracy improvement of 2.54% for the random forest classifier.

Hence, with the introduction of the fact strength, we were able to garner an average prediction accuracy improvement of 4.44% for the random forest classifier. This test also proved that the random forest classifier produces a high precision and recall, deriving a reasonable F1 score as identified in the previous experiments as well.

4.2 Classifier Calibration

In addition to the above performance index metrics, we proceeded to evaluate the calibration of the classifiers[7]. The classifier calibration models the classifiers' alignment as opposed to the best or ideal calibration for the actual predictions. The predicted truth value against the fraction of positives, plots the graph in the form of $y = mx + c$. As such, Figure 5 was rendered and is shown here.

[7]Calibration of Classifiers: https://scikit-learn.org/stable/modules/calibration.html



	Baseline	Support Vector Classifier	Stochastic Gradient Descent	Random Forest Classifier
Precision	0.828	0.905	0.908	0.932
Recall	0.910	0.632	0.256	0.936
F1	0.867	0.710	0.300	0.922

Fig. 3. Performance Index Values for the NELL dataset (experiment 3)

5 Discussion

Initially we present the discussion of the results on the NELL and YAGO datasets with only the subject, predicate, object, PSL soft truth values and the human evaluations, excluding the fact strength.

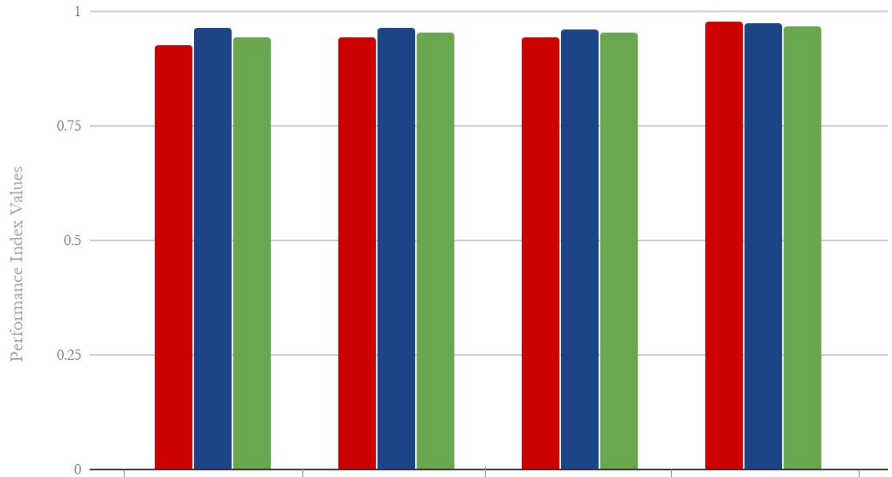
Based on the experiments, we support the claim that, our approach can be generalized to different types of dataset domains, by applying the NELL trained dataset on the YAGO dataset and still achieving remarkable Precision and F1 score improvements as proved in the second and fourth experiments.

Further, our evaluation proves that the addition of the empirical fact strength feature improves the models' performance index values as shown in the third and fourth experiments. Hence, we identify the fact strength as an enhanced feature in modelling the strength or weight of a fact in relation to its entire domain.

Reverting back to our contribution, adding a level of supervision in the form of actual human evaluations, increases the prediction accuracy in identifying the inaccurate triples as proven through the average Precision and F1 scores as evidenced in these experiments.

Secondly, the addition of a small set of human evaluated triples, to evaluate a larger unseen dataset performs effectively, especially for the random forest classifier. This claim of ours is supplemented throughout the experiment, with observed improvement in the precision, recall, and F1 scores for all the experiments, when compared against their respective baselines, where actual human evaluation based supervision is not included.

We were able to accompany our claims for all the experiments, with the Random forest classifier. Though the Stochastic Gradient Descent showed occasionally reasonable performance index values, they fluctuate throughout the



	Baseline	Support Vector Classifier	Stochastic Gradient Descent	Random Forest Classifier
Precision	0.927	0.943	0.945	0.977
Recall	0.963	0.965	0.961	0.976
F1	0.944	0.954	0.953	0.968

Fig. 4. Performance Index Values for the YAGO dataset (experiment 4)

experiments. Hence, we resorted to the random forest classifier that constantly illustrates the expected behaviour, with high performance index values and average improvements.

Solely based on these experiment results, we saw the best performance of the random forest classifier.

However, in order to verify its performance, in terms of calibration, we analyzed the classifier calibration as shown in Figure 5. Here, it is clearly observable that the Random Forest Classifier (RFC) has the optimal calibration with its actual prediction, as it lies as close to the ideal, perfectly calibrated $y = mx$ line, while the calibration for the Stochastic Gradient Descent (SGD) and Support Vector Classifier (SVC) deviate further away from the perfect calibration.

Hence, based on the performance index values (precision, recall and f1), and the classifier calibration, we safely arrive at the conclusion that the Random Forest Classifier, trained on a part of the NELL dataset, with a feature set of triples (subject, predicate, object), computed PSL soft truth values, derived fact strength, and crowd-sourced human evaluations, can be used to effectively predict the most probable human evaluation/truths for any larger datasets of triples.

Using these evaluations, we then propose on dropping the facts or triples, predicted as false by the classifier and building the knowledge graph with the facts/triples predicted to be true. This will be an effective pre-process in removing erroneous data from knowledge graphs.

Hence, the knowledge graph will be more pure and accurate in the sense of holding true facts that correspond to actual human judgment. The application of this model also proposes use cases in information extraction systems, where data can be represented in the form of triples.

A constraint that we identified with this model is the specificity of the feature set, in the form of triples, in order to compute the PSL soft truth

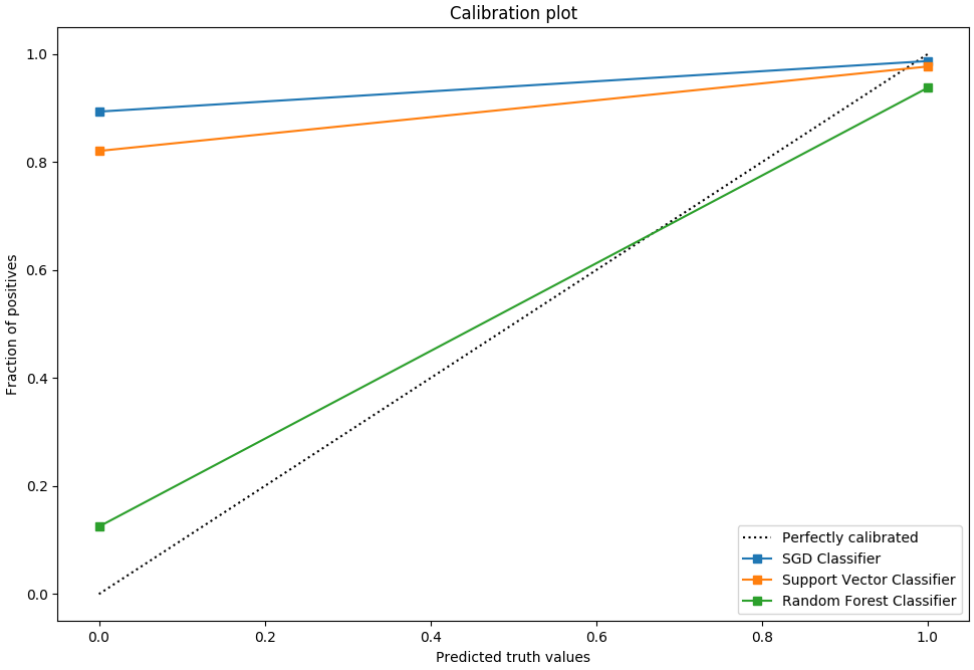


Fig. 5. Classifier Calibration Plot

values. Thus, our future research will be directed towards the generalization of this solution model such that it is applicable across omni-various fact/data formats.

6 Related Work

There are various forms of knowledge stores or representation methods as described in the Background section. We were able to observe that the evolution of these methods correlated to the amount and complexity of the data being generated with time.

Since we identify knowledge graphs as the optimal knowledge representation method, we looked into the constraints in knowledge graphs. As such, there are 2 major issues that need to be addressed [25]. The first concerns data completion, where certain information may be missing from knowledge graphs and may not be available when queried for. The second issue is the presence of erroneous data which will lead to the return of incorrect responses, or responses that do not correlate with human judgment, when the user queries the knowledge graph.

There has been extensive research with regard to data completion in knowledge graph whereas the removal of erroneous data is still an area with on-going research prospects. Previous work by Lin et al. [21] has discussed about handling knowledge graph completion by learning entities and relation embeddings.

This was set as an extension of the TransE [5] and TransH [31] models that compute entity and relational embeddings as a translation from one entity to the other in a relationship. This model was called TransR.

In the TransE and TransH models, vector embeddings are learned and encrusted within the same space implying that the entities and relationships are set in the same vector space $\mathbb{R}^k$. But entities have variations, and relations model for the variations of the entities.

Hence, the TransR model proposed by Lin et al. identifies separate vector spaces for the entities and relations for each triple (h, r, t). The entities

will be embedded as $h, t \in \mathbb{R}^k$ and the relation will be embedded as $r \in \mathbb{R}^d$.

Another method in knowledge graph completion includes the Adaptive Sparse Transfer Matrix where each entity and relation is encoded numerically and triplet classification and link prediction tasks are performed to complete the graph [16]. Extending from the previous work on the TransE, TransH and TransR models, Ji et al. identified the heterogeneity of and imbalance in the entities and relations in knowledge graphs. Their SparseTrans(share) model resolves to sparse transfer matrices in place of transfer matrices.

When both ends of a relation, the entities, are set in the same transfer matrix, the relations connect a number of entity pairs, determining the degrees of the sparse transfer matrices. This addresses the heterogeneous nature of the entities and relations. The SparseTrans(separate) model deals with the imbalance, which occurs as the number of entities on either side of a relation vary in number. For this, it uses 2 separate sparse matrices, one for each entity and evaluates the degrees based on the number of entities in each space.

A project by Shi and Weninger, aimed on filling missing data in knowledge graphs by using a shared variable neural network model that learns joint embeddings of the knowledge graph's entities and edges with trivial changes to the standard loss function [28].The scalability of the model to massive knowledge graphs was a major concern that generated Shi et al.'s model, projE.

Also, the knowledge that the other models were found to use, were based on pre-trained embeddings. So projE considered the task of knowledge completion as a ranking task. Based on the ranking priority, the candidates are directed into 2 separate input embedding spaces using a combination bias as shown in (10):

$$e \oplus r = D_e e + D_r r + b_c. \tag{10}$$

Here, D_e and D_r are the entity and relation weights that are represented as $m \times m$ diagonal matrices and b_c is a the connective combination bias. This computes the embedding function using f and c which are activation functions as shown in equation (11). Here, $W^c \in R^{s \times k}$, such that s is the number of candidate entities:

$$h(e, r) = g(W^c f(e \oplus r) + b_p). \tag{11}$$

Considering the removal of erroneous data from knowledge graphs, Ryu et al. proposed an erroneous relation elimination method that removes the erroneous data from knowledge graphs. This rests on the concept that entities within a semantic relationship are represented by the same node [27]. Therefore, a single representative entity will be selected to represent each semantically similar relation. Consequently, error detection is performed based on relational weights and predefined limitations or conditions.

The Deep Fact Validation is yet another method that addresses this issue by providing users with brief extracts of web pages and a confidence score for the facts based on the sources from which they were retrieved [20]. This paved way to eliminate facts with relatively low confidence values, assuming a direct proportionality to their sources' accuracy.

The Probabilistic Soft Logic is a statistical relational framework that computes confidences in the form of soft truth values for facts or triples. As discussed in the Background section, we harness PSL, to infer confidence scores about the triples in order to use them as a feature. We garner the baseline performance index values for our experiments based on [26]. According to Pujara et al., the PSL threshold value that determines the baseline of the evaluations, is the threshold that gives the optimal f1 score. As such, the baseline performance index values for the precision, recall and f1 scores are 0.828, 0.910 and 0.867, respectively.

So far, all of these refinement techniques adhere to simply evaluating the validity of facts based on their sources, their semantic similarities, relational embeddings, all of which are completely automated and computed based on the dataset. However, the previous work does not incorporate human evaluation while measuring the accuracy of the triples in the knowledge graph. In contrast, we propose a machine learning based approach to incorporate human evaluation to achieve significant improvement over the baseline.

7 Conclusion

Knowledge Graphs are a sleek way to represent mass amount of data and model the relationship between various entities. However, they are not always complete or accurate. Hence, our solution model proposes a method to address the removal of inaccurate or erroneous facts from knowledge graphs. We consider the validity of the fact being accurate as a correspondence to actual human evaluations. Manually processing the knowledge graph facts can be expensive and extensive.

Thus, we use a machine learning approach, along with the probabilistic soft truth values computed using PSL, and an empirically derived feature, the fact strength, to train a model on a subset of human evaluated triples. Evaluating the trained model on unseen datasets rendered a precision improvement of 12.56% and 5.39% which achieved an average precision gain of 8.98%.

We were also able to achieve an average improvement of 4.44% for the prediction accuracy, in terms of the F1 score. Therefore, the inclusion of the fact strength as a training feature here, showed an average amplification of 2.13% in the precision as opposed to the model that did not use the fact strength.

Hence, the primary contribution of this paper is the proposal of a machine learning approach injected with a sufficient level of supervision through a subset of human evaluated fact truths and a probabilistic inference of fact confidences. These predictions can be used to eliminate inaccurate edges or relations in the knowledge graph, thus, refining it and addressing the erroneous data issue in knowledge graphs. Also, we intend on exploiting the capabilities of Bayesian statistics to compute the probabilistic confidences for the facts in our future work.

References

1. **Allemang, D., Hendler, J. (2011).** Semantic web for the working ontologist: effective modeling in RDFS and OWL.
2. **Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. (2007).** Dbpedia: A nucleus for a web of open data. In The semantic web. Springer, pp. 722–735.
3. **Berners-Lee, T., Hendler, J., Lassila, O. (2001).** The semantic web. Scientific american, Vol. 284, No. 5, pp. 34–43.
4. **Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. (2008).** Freebase: a collaboratively created graph database for structuring human knowledge. Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp. 1247–1250.
5. **Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O. (2013).** Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, pp. 2787–2795.
6. **Boyd, S., Vandenberghe, L. (2004).** Convex optimization. Cambridge university press.
7. **Brocheler, M., Mihalkova, L., Getoor, L. (2012).** Probabilistic similarity logic. arXiv preprint arXiv:1203.3469.
8. **Buhrmester, M., Kwang, T., Gosling, S. D. (2011).** Amazon's mechanical Turk: A new source of inexpensive, yet high-quality, data? Perspectives on psychological science, Vol. 6, No. 1, pp. 3–5.
9. **Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W. (2014).** Knowledge vault: A web-scale approach to probabilistic knowledge fusion. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 601–610.
10. **Fensel, D. (2001).** Ontologies. In Ontologies. Springer, pp. 11–18.
11. **Giaretta, P., Guarino, N. (1995).** Ontologies and knowledge bases towards a terminological clarification. Towards very large knowledge bases: knowledge building & knowledge sharing, Vol. 25, No. 32, pp. 307–317.
12. **Godbole, S., Sarawagi, S. (2004).** Discriminative methods for multi-labeled classification. Pacific-Asia conference on knowledge discovery and data mining, Springer, pp. 22–30.
13. **Golfarelli, M., Maio, D., Rizzi, S. (1998).** The dimensional fact model: A conceptual

model for data warehouses. International Journal of Cooperative Information Systems, Vol. 7, No. 02n03, pp. 215–247.

14. **Gruber, T. R. (1993).** A translation approach to portable ontology specifications. Knowledge acquisition, Vol. 5, No. 2, pp. 199–220.
15. **Guarino, N., Oberle, D., Staab, S. (2009).** What is an ontology? In Handbook on ontologies. Springer, pp. 1–17.
16. **Ji, G., Liu, K., He, S., Zhao, J. (2016).** Knowledge graph completion with adaptive sparse transfer matrix. AAAI, pp. 985–991.
17. **Kimmig, A., Bach, S., Broecheler, M., Huang, B., Getoor, L. (2012).** A short introduction to probabilistic soft logic. Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications, pp. 1–4.
18. **Korfhage, R. R. (2008).** Information storage and retrieval.
19. **Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T. (2011).** Hmdb: a large video database for human motion recognition. Computer Vision (ICCV´11) IEEE International Conference on, IEEE, pp. 2556–2563.
20. **Lehmann, J., Gerber, D., Morsey, M., Ngomo, A.-C. N. (2012).** Defacto-deep fact validation. International Semantic Web Conference, Springer, pp. 312–327.
21. **Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. (2015).** Learning entity and relation embeddings for knowledge graph completion. AAAI, volume 15, pp. 2181–2187.
22. **Liu, S., Liu, K., He, S., Zhao, J. (2016).** A probabilistic soft logic based approach to exploiting latent and global information in event classification. AAAI, pp. 2993–2999.
23. **Ma, Y., Gao, H., Wu, T., Qi, G. (2014).** Learning disjointness axioms with association rule mining and its application to inconsistency detection of linked data. Chinese Semantic Web and Web Science Conference, Springer, pp. 29–41.
24. **McGuinness, D. L., Van Harmelen, F. (2004).** Owl web ontology language overview. W3C recommendation, Vol. 10, No. 10, pp. 2004.
25. **Paulheim, H. (2017).** Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web, Vol. 8, No. 3, pp. 489–508.
26. **Pujara, J., Miao, H., Getoor, L., Cohen, W. (2013).** Knowledge graph identification. International Semantic Web Conference, Springer, pp. 542–557.
27. **Ryu, P. M., Jang, M. G., Kim, H., Hwang, Y., Lim, S., Heo, J., Lee, C. H., Oh, H. J., Lee, C., Choi, M., others (2013).** Apparatus and method for knowledge graph stabilization. US Patent 8,407,253.
28. **Shi, B., Weninger, T. (2017).** Proje: Embedding projection for knowledge graph completion. AAAI, volume 17, pp. 1236–1242.
29. **Spyns, P. (2005).** EvaLexon: Assessing triples mined from texts. STAR, Vol. 9, pp. 09.
30. **Suchanek, F. M., Kasneci, G., Weikum, G. (2007).** Yago: a core of semantic knowledge. Proceedings of the 16th international conference on World Wide Web, ACM, pp. 697–706.
31. **Wang, Z., Zhang, J., Feng, J., Chen, Z. (2014).** Knowledge graph embedding by translating on hyperplanes. Proceedings of the AAAI Conference on Artificial Intelligence, volume 28, pp. 1112–1119.

Article received on 27/02/2019; accepted on 17/01/2021.
Corresponding author is Manuela Nayantara Jeyaraj.

ISSN 2007-9737

An Automatic Text Summarization: A Systematic Review

Vishwa Patel, Nasseh Tabrizi

East Carolina University,
USA

patelvi17@students.ecu.edu, tabrizim@ecu.edu

Abstract. The 21st century has become a century of information overload, where in fact information related to even one topic (due to its huge volume) takes a lot of time to manually summarize it into few lines. Thus, in order to address this problem, Automatic Text Summarization methods have been developed. Generally, there are two approaches that are currently being used in practice: Extractive and Abstractive methods. In this paper, we have reviewed papers from IEEE and ACM libraries those related to Automatic Text Summarization for the English language.

Keywords. Automatic text summarization, extractive, abstractive, review paper.

1 Introduction

Nowadays, the Internet has become one of the important sources of information. We surf the Internet, to find some information related to a certain topic. But search engines often return an excessive amount of information for the end user to read increasing the need for Automatic Text Summarization (ATS), increased in this modern era. The ATS will not only save time but also provide important insight into a piece of information.

Many years ago scientists started working on ATS but the peak of interest in it started from the year 2000. In the early 21st century, new technologies emerged in the field of Natural Language Processing (NLP) to enhance the capabilities of ATS. ATS falls under NLP and Machine Learning (ML).

The formal definition of ATS is mentioned in this book [11] “Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)”.

A general approach to write summary of a person is that read the whole text first and then try to express the idea with either using same words or sentences in the document or rewrite it.

In either case the most salient idea is captured and expressed. The basic objective of ATS is to create summaries which are as good as human summaries. There are various other methods to extract summaries, but these are the 2 main methods: Extractive and Abstractive Text Summarization [3].

Extractive Text Summarization extracts main keywords or phrases or sentences from the document, combine them and include them in the final summary. Sometimes, the summary generated by this approach can be grammatically erroneous.

While Abstractive Text Summarization totally focuses on generating phrases and/or sentences from scratch (i.e. paraphrasing the sentences in original documents) in order to maintain the key concept alive in summary.

Here the summary generated is free of any grammatical errors, which is an advantage compared to Extractive method.

It is important to note that the summaries generated by this approach are more similar to summaries generated by human (Similar means that the whole document’s idea is rewritten using a different set of words). Implementing Abstractive Text Summarization is more difficult compared to Extractive approach.

ISSN 2007-9737

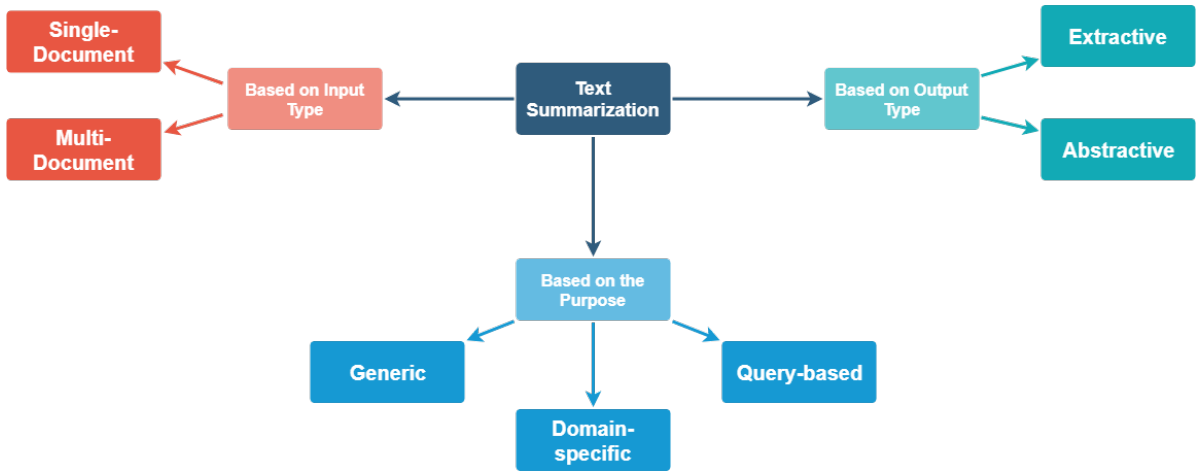


Fig. 1. Types of text summarization

2 Background

Since the idea of ATS emerged years ago, the research got accelerated when tools for NLP and ML with Text Classification, Question-Answers etc became available. Some of the advantages of ATS [3] are listed as follows:

— Reduces time for reading the document.

— Makes selection process easier while searching for documents or research papers.

— Summaries generated by ATS are less biased compared to humans.

— Personalized summaries are more useful in question-answering systems as they provide personalized information.

ATS has many applications in almost all fields, for example summarizing news. It is also useful in medical field, where long medical history of a patient can be summarized in few words which saves lot of time and also helps doctors to understand patient's condition easily.

Authors of the book [11] provide the daily useful applications of ATS which are described as follows:

— Headlines (from around the world).

— Outlines (notes for students).

— Minutes (of the meeting).

— Previews (of movies).

— Synopses (soap opera listings).

— Reviews (of a book, CD, movie, etc.).

— Digests (TV guide).

— Biography (resumes, obituaries).

— Abridgments (Shakespeare for children).

— bulletins (weather forecasts/stock market reports).

— Sound bites (politicians on a current issue).

— Histories (chronologies of salient events).

When the Internet resurfaced in 2000, data started to expand. The following two evaluation programs (conferences related to Text) were established by National Institute of Standard and Technology (NIST), Document Understanding Conferences (DUC) [18] and Text Analysis Conference (TAC) [13] in the USA. They both provide data related to summaries.

3 Types of Text Summarization

Text Summarization can be classified into many different categories. Figure 1 illustrates the different types of Text Summarization [3].

3.1 Based on Output Type

There are 2 types of Text Summarization based on Output type[3]:

— Extractive Text Summarization.

— Abstractive Text Summarization.

3.1.1 Extractive Text Summarization

Extractive Text Summarization where important sentences are selected from the given document and then are included in the summary. Most of the text summarization tools available are Extractive in nature. Below mentioned are some of the online Tools available:

— TextSummarization.

— Resoomer.

— Text Compactor.

— SummarizeBot.

The full list of ATS tools available at [16]. The process of analyzing the document is fairly straight forward.

First the information (text) is pre-processing step where all words are converted into either lower-or-upper-case letters, stop words are eliminated and remaining words are converted into their root forms. Next step is to extract different features based on which next step will be decided. Some of these features are:

— Length of the Sentence.

— The frequency of the word.

— The most appearing word in sentence.

— Number of characters in sentence.

Now, based on these features, using sentence scoring all the sentences will be placed either in descending or ascending order. In the last step, the sentences which have the highest value will be selected for the summary.

Figure 2 represents the steps of the process of Extractive Summarization. Authors in [9] paper have implemented ATS for multi-document, which contains headings for each document.

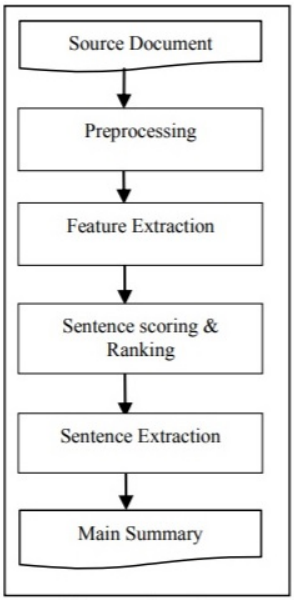


Fig. 2. Steps of extractive text summarization

3.1.2 Abstractive Text Summarization

Abstractive Text Summarization, tries to mimic human summary by generating new phrases or sentences in order to offer a more coherent summary to the user. This approach sounds more appealing because it is the same approach that any human use in order to summarize the given text.

The drawback of this approach is that its practical implementation is more challenging compared to Extractive approach. Thus, most of the tools and research have focused on Extractive approach. Recently, researchers are using deep learning models for Abstractive approaches and are achieving good results.

These approaches are inspired by Machine Translation problem. The authors [2] presented the attentional Recurrent Neural Network (RNN) encoder-decoder model which was used in Machine Translation, which produced excellent performance. Researchers then formed Text Summarization problem as a Sequence – to – Sequence Learning.

Other authors [12] have used the same model ”Encoder-Decoder Sequence-to-Sequence

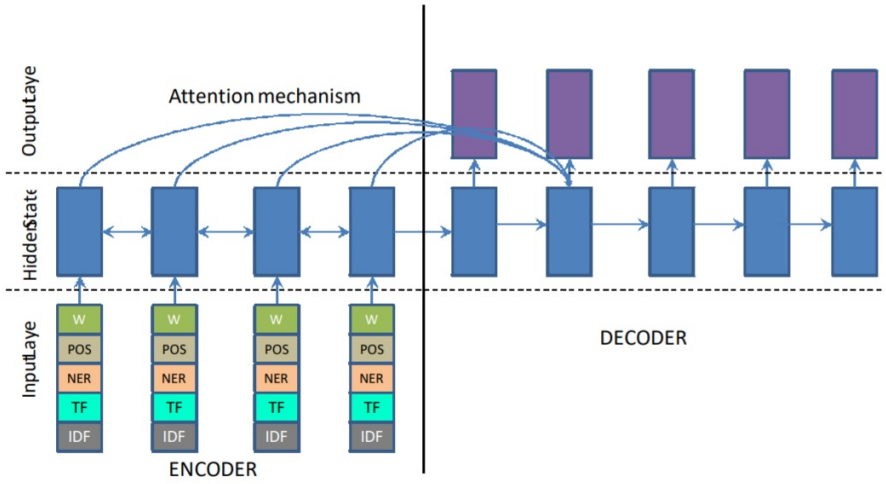


Fig. 3. Architecture of encoder-decoder sequence-to-sequence RNN

RNN" and used it in order to obtain a summary which is an Abstractive method. Figure 3 is an "Encoder-Decoder Sequence-to-Sequence RNN" architecture.

3.2 Based on Input Type

There are 2 types of Text Summarization based on Input types [3]:

— Single-document.

— Multi-document.

3.2.1 Single-Document

This type of text summarization is very useful for summarizing the single document. It is useful for summarizing short articles or single pdf, or word document. Many of the early summarization systems dealt with single document summarization.

Text Summarizer [17] is an online tool which summarizes a single document, it takes URL or text as input and summarize the input document into several sentences.

It will accept the input text and will find the most important sentences and will include them into the final summary.

3.2.2 Multi-Document

It gives summary which is generated from multiple text documents. If multiple documents on related to specific news, or articles are given to multi-document text document then it will be able to create a concise overview of the important events.

This type of ATS is useful when user needs to reduce overall unnecessary information, because these multiple documents of articles about the same events can contain several sentences that are repeated.

3.3 Based on Purpose

There are 3 text summarization techniques based on its purpose [3]:

— Generic.

— Domain specific.

— Query based.

3.3.1 Generic

This type of text summarization is general in application, where it does not make any assumption regarding the domain of article or the content of the text. It treats all the inputs as equal.

For example, generating headlines of news articles, generating a summary of news, summarizing a person's biography, or summarizing sound-bites of politicians, celebrities, entrepreneurs etc. Most of the work that has been done in ATS field, is related to generic text summarization.

The authors [15] has developed ATS which summarizes news articles. These articles can consist of news from different categories. Any ATS which is not explicitly designed for a specific domain or topic, falls under this category.

3.3.2 Domain-Specific

This type of text summarization is different than generic type, domain means the topic of the text. Domain-specific means that the model uses domain-specific knowledge along with the input text, it helps in producing a more accurate summary.

An example of this could be a text summarization model which uses a heart(cardiology) related knowledge, or computer science related knowledge. The main benefit here is that domain-specific knowledge helps model to understand the context of the text and can extract more important sentences which are related to the field.

3.3.3 Query Based

Query based means that user gives the query to text summarization tool which then retrieves information related to that query. This type of tool is mainly used for natural language question-answers. The goal here is to extract personalized summary based on user needs.

For an instance, if a article is related to "John", "Car" and user wants to extract summary which is related towards "john" then ATS will retrieve summary which is related to it.

4 Current Research in Automatic Text Summarization

The authors [8] presented an ATS system which summarizes Wikipedia articles using an Extractive Approach. They first perform preprocessing step, where the text is tokenized, porter-stemming is applied, and 10 different features are extracted (f1 - f10) and given as the input to neural network with one hidden layer and one output layer. Output scores ranges from 0-1.

This score is proportional to the importance of the sentence. These scores are then used to generate summary. Windows Word 2007 is used to generate summary for the same article. Summary generated from Microsoft Word 2007 is referred to as "Reference Summary".

Both summaries (Reference Summary and System Generated Summary) is then used to evaluate model performance, and precision, recall and f1-score are calculated. Model performs best if it uses the only f9 feature with f-1 score of 0.223. Similarly, f7 has lowest f1-score of 0.055.

Other authors [6], presented a 4 dimensional graph model for ATS. Graph models show the relationship between the sentences in the text, which is valuable for ATS tasks. They used the TextRank algorithm to evaluate in the context of Extractive Text Summarization.

They used CNN dataset for evaluation. Their model improves the TextRank algorithm overall(better precision, recall and f-measure) by improving 34.87% in relation to the similarity model. Here is the list of 4 dimensions which were used to create the graph:

— **Similarity**, It measures the overlapping content between pairs of sentences. If it exceeds a threshold score which is selected by the user, then edge between the sentence pair is created.

— **Semantic Similarity**, It employs ontology conceptual relations such as synonyms, hyponym and hypernym. Then sentences must first be represented as vectors with words and the semantic similarity scores for each pair of words using WordNet must be calculated.

— **Coreference resolution** It is the process by which they identified the noun that was referring to the same entity. There are 3 forms of coreference: named, nominal or pronominal.

— **Discourse Relations** It is used to highlight the relevant relationships in the text.

The authors [1] proposed a Query-oriented ATS using Sentence Extraction technique. First the input text is pre-processed (Tokenization, Stop Words Removal, Stemming and POS tagging), then 11 features were extracted from the input text.

The first set of features are used to identify informative sentences and the second set of appropriate features will help to extract the query relevant sentences. Based on those features, each sentence was scored, and used DUC-2007 dataset for training and evaluation purpose.

Min-Yuh Day and Chao-Yu Chen [4] proposed an AI approach for ATS. They have developed ATS with 3 different models: Statistical, Machine Learning and Deep Learning Models. They used Essay titles and abstracts as their dataset.

Using the Essay abstracts as input, it is inputted into all 3 models, and a headline for essay is generated by all 3 models. Then all 3 generated title summaries are evaluated by using ROUGE evaluation metric, and then best fitting title is selected.

The authors in [5] published an article presents an unsupervised extractive approach based on graphs. This method constructs an undirected weighted graph from the original text by adding a vertex for each sentence and calculates a weighted edge between each pair of sentences that are based on a similarity/dissimilarity criterion.

A ranking algorithm is applied and most important sentences based on their corresponding rank are identified. They used DUC-2002 dataset for their analysis. Results are then evaluated using ROUGE-1 using different distance measures like LSA, TextRank, Correlation, Cosine, Euclidean etc.

Other authors [7] introduced an ATS which is based on an unsupervised graph based ranking model. This model builds a graph by collecting words and their lexical relationships from the document. They collect a subset of high rank and low rank words. Sentences are extracted based on how many high rank words are present in it.

Collecting such sentences leads to generating a summary of the document. Here authors have focused on using ATS for people who are visually challenged or visual loss. They have tested the proposed system on NIPS(Neural Information Processing System) Dataset. They have focused on Single Document Summarization.

5 Research Methodologies

The purpose of this study is to investigate the trends in which the Automatic Text Summarization (ATS) for English language have progressed by doing research on published articles and to gain intuition on the current direction of ATS.

The first step was researching for ”Automatic Text Summarization” from different databases. There were 160+ papers published in our selected databases. As our goal is to study the current trends in ATS field, so the research was limited to past 7 years (2012-2019), and omitted any type of book and early available articles.

Therefore, we started our search in 2 databases: *IEEE*, *ACM*, using this query: ”Automatic Text Summarization” (with quotes). Among all the related papers, some papers were not related to our topic, for instance, some paper had built ATS for different languages like Arabic, Hindi etc.

Since our focus is on ATS for the English Language, these papers were excluded from our research. For writing information about ATS, we have used several articles and some research papers.

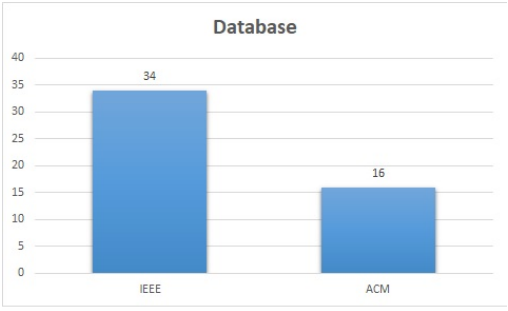


Fig. 4. Distribution of papers by database

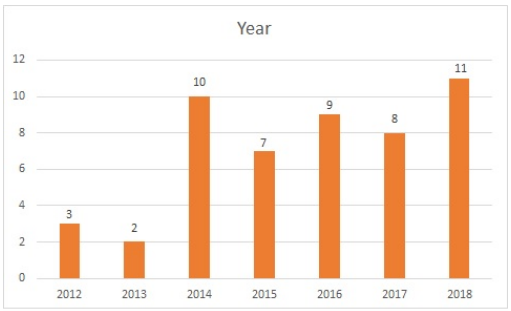


Fig. 5. Distribution of papers by year of publication

6 Classification of Papers

The selected papers from 2 different databases were classified them in different categories. The details are described below:

6.1 Distribution by Database

We have collected 50 research papers from 2 different databases: *IEEE* and *ACM*. 34 papers were found from IEEE (68%) and, 16 papers from ACM (32%) see *Figure-4*. All the research papers are conference paper which is related to our topic ”Automatic Text Summarization”.

6.2 Distribution by Publication Year

We found 160+papers while firing query related to our topic. We restricted our search to the past 7 years. As we can see from *Figure-5* that initially in 2012-2013 there were not much research being conducted, while from 2014 there was an increase in published papers. From the data we can clearly say there were 2 years having the highest number of publications, 11 publications in 2018, while 10 publications in 2014.

6.3 Distribution by Type of ATS

6.3.1 Based on Output Type

As mentioned earlier in the paper, there are 2 types of ATS based on Output type: Extractive Text Summarization and Abstractive Text Summarization.

We have increased one more category as ”Hybrid” where researchers have used both Text Summarization technique and combined them to generate summaries. We classified our papers based on these types, Figure 6 shows that among the papers, 46 Extractive type, 2 Abstractive and 2 Hybrid.

6.3.2 Based on Input Type

As, there are 2 types of Text Summarization based on Input type: Single and Multi Text Summarizations. Figure 7 represents that 46 of papers were based on Single Document ATS, while 4 were based on Multi Document ATS. Most of the researchers focus on summarizing single input document.

7 Dataset

Among all papers, DUC[18] dataset was most popular among the research studies. DUC offers single document articles with handwritten summaries. These summaries are also referred to as ”Gold Summary”, which is used to compare the resultant summary obtained by Text Summarization. DUC has many different datasets according to year wise, starting from 2001-2007.

The second most popular dataset is CNN dataset, which consists of news and/or articles from CNN website. There are other datasets which were used in papers which consists of (but not limited to) Gigaword, Elsevier Articles, Opinonis and Daily Mail.

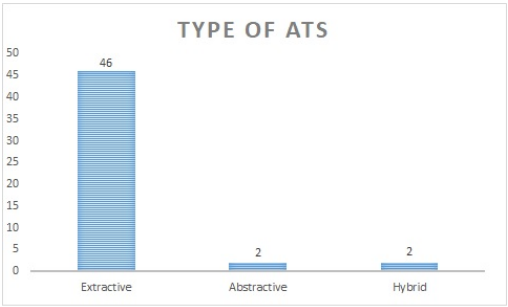


Fig. 6. Distribution of papers by output type of text summarization

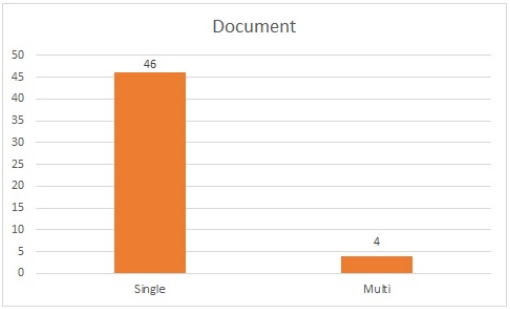


Fig. 7. Distribution of papers by input type of text summarization

8 Evaluation Technique

Main evaluation methodology used is ROUGE evaluation [10]. ROUGE stands for Recall-Oriented Understudy. It is a set of metrics for evaluating automatic test summarization of text and also for machine translation.

Basically, it compares 2 different types of summaries, Automatically Produced Summary by ATS and Set of Reference Summary (which is typically produced by humans). Another evaluation methodology used for evaluation is F-1 measure, where Precision and Recall is calculated.

F1 score or *F-measure* is used to calculate the accuracy of a certain system. F1 score calculation uses both, Precision (p) and Recall (r). Precision is the fraction of the summary that is correct.

Recall is the fraction of the correct (model) summary that is outputted. Some papers, however, did not use any evaluation metrics to check the accuracy.

Precision in the Context of ROUGE, we are actually measuring how much of the ATS summary was actually relevant or needed? While Recall means that how much of the reference summary is the ATS summary recovering or capturing?

$$\text{Precision} = \frac{\text{number\_of\_overlapping\_words}}{\text{total\_words\_in\_system\_summary}}, \quad (1)$$

$$\text{Recall} = \frac{\text{number\_of\_overlapping\_words}}{\text{total\_words\_in\_reference\_summary}}. \quad (2)$$

Besides Precision and Recall, there are 3 other evaluation metrices:

— ROUGE-N.

— ROUGE-L.

— ROUGE-S.

8.1 ROUGE-N

This ROUGE package [10] is used to measure unigrams, bi-grams, trigrams and higher order n-grams overlap. For example,ROUGE-1 refers to unigrams, whereas ROUGE-2 refers to bigrams, ROUGE-3 as trigrams etc.

They use both summaries, system summary and reference summary to calculate overlap of unigrams, or bigrams or trigrams or any high order n-grams [14].

8.2 ROUGE-L

This measures [10] longest matching sequence of words using LCS. The advantage of using LCS is that it does not require consecutive matches but in sequence matches that reflects sentence level word order.

It automatically includes the longest in-sequence common n-grams, that's why there is no need of defining predefined n-gram length [14].

8.3 ROUGE-S

Skip-gram measures the overlap of word pairs that can have maximum n gaps between words. For example, skip-bigram measures the overlap of word pairs that can have a maximum of two gaps between words [14, 10].

9 Conclusion

In this paper, we have researched 50 papers from IEEE and ACM databases in Automatic Text Summarization. We described different type of ATS based on input, output and purpose. Current research studies are also discussed in the paper.

We distributed the collected papers in various categories like by year, input type, output type and database and discussed various databases used by researchers. Finally, the most used for evaluation metric, ROUGE is explained along with its different metrics.

Acknowledgment

This work is supported partially by the National Science Foundation.

References

1. **Afsharizadeh, M., Ebrahimpour-Komleh, H., Bagheri, A. (2018).** Query-oriented text summarization using sentence extraction technique. 2018 4th International Conference on Web Research (ICWR), IEEE, pp. 128–132.
2. **Bahdanau, D., Cho, K., Bengio, Y. (2014).** Neural machine translation by jointly learning to align and translate.
3. **Chauhan, K. (2018).** Unsupervised text summarization using sentence embeddings.
4. **Day, M. Y., Chen, C. Y. (2018).** Artificial intelligence for automatic text summarization. 2018 IEEE International Conference on Information Reuse and Integration (IRI), pp. 478–484.
5. **de la Peña Sarracén, G. L., Rosso, P. (2018).** Automatic text summarization based on betweenness centrality. Proceedings of the 5th Spanish Conference on Information Retrieval, pp. 11.
6. **Ferreira, R., Freitas, F., Cabral, L. d. S., Lins, R. D., Lima, R., França, G., Simskez, S. J., Favaro, L. (2013).** A four dimension graph model for automatic text summarization. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), volume 1, pp. 389–396.
7. **Hamid, F., Tarau, P. (2014).** Text summarization as an assistive technology. Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments, Association for Computing Machinery, pp. 60.
8. **Hingu, D., Shah, D., Udmale, S. (2015).** Automatic text summarization of Wikipedia articles view document. 2015 International Conference on Communication, Information & Computing Technology, IEEE, pp. 1–4.
9. **Krishnaveni, P., Balasundaram, S. R. (2017).** Automatic text summarization by local scoring and ranking for improving coherence. 2017 International Conference on Computing Methodologies and Communication, pp. 59–64.
10. **Lin, C. Y. (2004).** ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out, Association for Computational Linguistics, pp. 74–81.
11. **Mani, I., Maybury, M. T. (1999).** Advances in automatic text summarization. MIT press.
12. **Nallapati, R., Zhou, B., dos Santos, C. N., Gulcehre, C., Xiang, B. (2016).** Abstractive text summarization using sequence-to-sequence RNNs and beyond.
13. **NIST (2018).** Text Analysis Conference.
14. **RxNLP (2018).** ROUGE Evaluation Metrics.
15. **Sethi, P., Sonawane, S., Khanwalker, S., Keskar, R. B. (2017).** Automatic text summarization of news articles. 2017 International Conference on Big Data, IoT and Data Science (BID), pp. 23–29.
16. **Softsonic (2018).** List of Online Automatic Test Summarization Tools.
17. **Text Summarizer (2018).** Manual of Text Summarization.
18. **Voorhees, E. (2002).** Document understanding conferences website.

Article received on 25/02/2019; accepted on 15/01/2021.
Corresponding author is Nasseh Tabrizi.

ISSN 2007-9737

Modelación matemática para zapatas combinadas de correa en esquina apoyadas sobre el terreno: Parte 1

María Azucena Moreno Hernandez, Arnulfo Luévanos Rojas,
Sandra López Chavarría, Manuel Medina Elizondo

Universidad Autónoma de Coahuila,
Instituto de Investigaciones Multidisciplinaria,
México

azucena.moreno@gmail.com, {arnulfol_2007, sandylopez5}@hotmail.com,
drmanuelmedina@yahoo.com.mx

Resumen. Este trabajo de investigación muestra un modelo matemático para zapatas combinadas de correa en esquina apoyadas sobre el terreno para obtener la superficie mínima en planta que soportan una carga concentrada y dos momentos ortogonales en cada columna. El trabajo considera un diagrama de presión que varía linealmente. El modelo tradicional considera una presión uniforme del suelo sobre las tres zapatas (una de esquina y dos de borde) y aplicando las tres ecuaciones de equilibrio estático (ΣF, ΣM_x y ΣM_y) se obtiene la solución para las reacciones de las tres zapatas, la presión se considera uniforme porque la reacción del suelo se aplica en el centro de cada zapata. Cuatro ejemplos numéricos se muestran para zapatas combinadas de correa en esquina, y cada ejemplo presenta tres casos diferentes. Este modelo es más general porque se puede aplicar a zapatas combinadas de esquina, simplemente considerando los anchos en dirección X de las zapatas 1 y 3 iguales, y los anchos en dirección Y de las zapatas 1 y 2 iguales.

Palabras clave. Modelación matemática, zapatas combinadas de correa en esquina, superficie mínima, área óptima.

Mathematical Modeling for Corner Strap Combined Footings Resting on the Ground: Part 1

Abstract. This research work shows a mathematical model for corner strap combined footings supported on the ground to obtain the minimum surface in plan that supports a concentrated load and two orthogonal moments in each column. The work considers a pressure diagram that varies linearly. The traditional model considers a uniform pressure of the soil on the three footings (one corner and two edge) and applying the three equations of static equilibrium (ΣF, ΣM_x and ΣM_y) the solution for the reactions of the three footings is obtained, the pressure is considered uniform because the soil reaction is applied in the center of each footing. Four numerical examples are shown for corner strap combined footings, and each example presents three different cases. This model is more general because it can be applied to corner combined footings, simply considering the widths in the X direction of the footings 1 and 3 equal, and the widths in the Y direction of the footings 1 and 2 equal.

Keywords. Mathematical modeling, corner strap combined footings, minimum surface, optimum area.

1. Introducción

Los cimientos son los elementos estructurales que soportan columnas y muros y transfieren las cargas al suelo subyacente sin exceder la capacidad de carga del suelo por debajo de la estructura. Si las cargas se van a transmitir correctamente, los cimientos deben diseñarse para superar un gran asentamiento o rotación, reducir el asentamiento diferencial y ofrecer la seguridad necesaria contra deslizamientos y vuelcos.

Los tipos de cimentaciones se dividen en:

1 Cimentaciones poco profundas o superficiales.
2 Cimentaciones profundas.

Las cimentaciones poco profundas o superficiales de concreto reforzado se clasifican en:

1 Zapata individual o zapata aislada.
2 Zapatas combinadas.
3 Zapatas corridas.
4 Zapatas de correa.
5 Losas de cimentación o balsas.

Las cimentaciones profundas se dividen en:

1 Pilotes de cimentación.
2 Pozos perforados o cimentación de cajón.

Las zapatas aisladas soportan columnas individuales, y pueden ser cuadradas, circulares o rectangulares en planta.

Las zapatas combinadas soportan dos o más columnas, y pueden ser rectangulares, trapezoidales, en forma de "T" y en forma de "L" (esquinera).

Las zapatas corridas soportan muros.

Las zapatas de correa es un caso especial de zapatas combinadas, es decir, dos zapatas aisladas unidas por una viga de liga.

Las losas de cimentación o balsas soportan todas las columnas del edificio.

El caso especial para solucionar la cimentación de esquina puede ser una zapata combinada de esquina del tipo L o zapatas aisladas (una de esquina y dos de bode) unidas por vigas de liga como se muestra en la Figura 1.

Los modelos optimizados para zapatas aisladas rectangulares, cuadradas y circulares (cimentaciones poco profundas o superficiales) tomando en cuenta la distribución de la presión del suelo como lineal han utilizado diversos softwares para obtener la solución [1-17].

Los modelos de optimización para zapatas combinadas rectangulares, trapezoidales, en forma de T, de tira o correa y de esquina descansando sobre el terreno utilizan técnicas de optimización para resolver el problema [18-25].

Los artículos más cercanos al tema de optimización para zapatas combinadas de correa en esquina son: López-Chavarría *et al.* [19] desarrollaron un modelo óptimo para obtener el área mínima de zapatas combinadas de esquina, esta contribución puede ser útil cuando las zapatas tienden a traslaparse una sobre otra, pero cuando las zapatas no se traslapan se deben de usar vigas de correa o de liga (zapatas combinadas de correa en esquina), es decir, las zapatas se unen mediante vigas. Aguilera-Mancilla *et al.* [24] muestran un modelo matemático para obtener el área mínima de las zapatas combinadas de correa para dos columnas (una columna de frontera y la otra interior), por lo tanto, no se puede usar para obtener la superficie óptima para zapatas combinadas de correa en esquina.

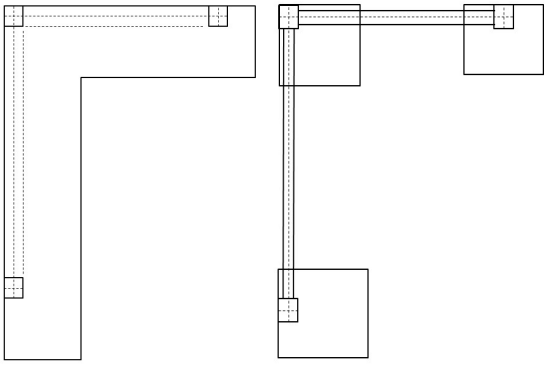


Fig. 1. Cimentaciones de esquina

Este trabajo de investigación presenta un modelo óptimo para zapatas combinadas de correa en esquina para obtener el área optimas o superficie mínima en planta de contacto sobre el terreno que soporta una carga concentrada y momentos alrededor de los ejes "X" e "Y" en cada columna.

El modelo propuesto en este trabajo considera que la presión tiene una variación lineal. El modelo tradicional considera una presión uniforme del suelo sobre la zapata de esquina y las dos zapatas de borde y aplicando las tres ecuaciones de equilibrio estático (ΣF, ΣM_x y ΣM_y) se obtiene la solución para las reacciones de las tres zapatas, la presión se considera uniforme porque la reacción del suelo se aplica en el centro de cada zapata.

Cuatro ejemplos numéricos se muestran para zapatas combinadas de correa en esquina: Ejemplo 1: Lados libres en las direcciones X e Y. Ejemplo 2: Lado restringido en la dirección X y libre en la dirección Y. Ejemplo 3: Lado restringido en la dirección Y y libre en la dirección

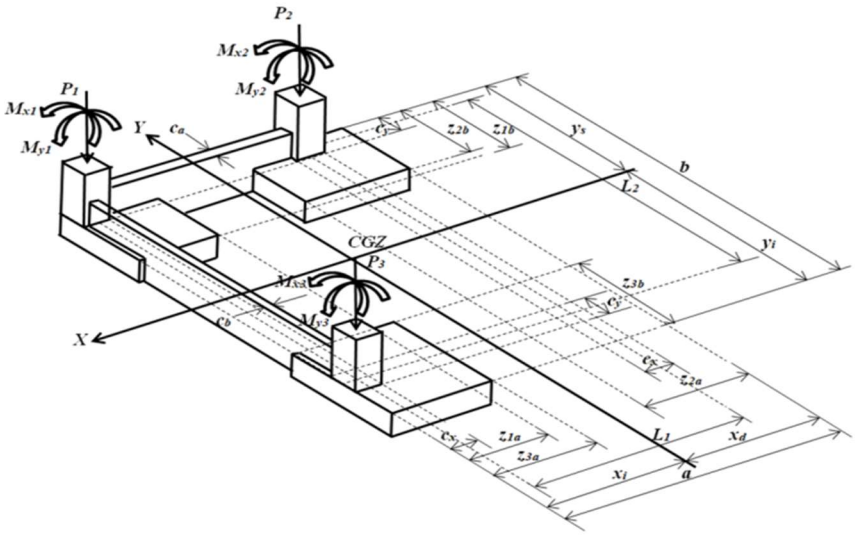


Fig. 2. Vista isométrica de la zapata combinada de correa en esquina

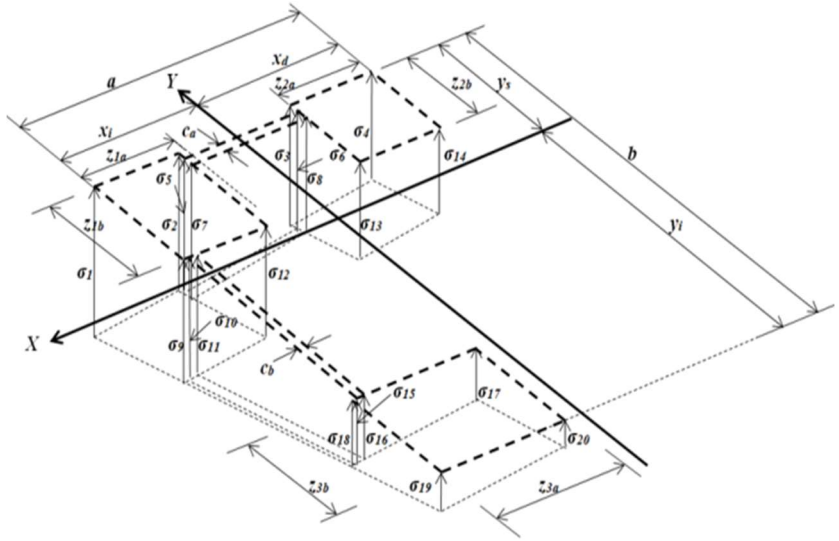


Fig. 3. Diagrama de presiones debajo de la zapata combinada de correa en esquina

X. Ejemplo 4: Lado restringido en las direcciones X e Y.

Cada ejemplo presenta tres casos diferentes: Caso 1 toma en cuenta $z_{1a} = z_{1b}$, $z_{2a} = z_{2b}$ y $z_{3a} = z_{3b}$, es decir, las zapatas tienen diferentes dimensiones y son cuadradas. Caso 2 considera $z_{1a}/M_{y1} = z_{1b}/M_{x1}$, $z_{2a}/M_{y2} = z_{2b}/M_{x2}$ y $z_{3a}/M_{y3} = z_{3b}/M_{x3}$, es decir, las zapatas están relacionadas con los momentos. Caso 3 considera $z_{1a} = z_{1b}$, $z_{2a} = z_{2b}$, $z_{3a} = z_{3b}$, $z_{1a} = z_{2a}$ y $z_{1a} = z_{3a}$, es decir, las zapatas tienen las mismas dimensiones y son cuadradas.

2. Formulación matemática del modelo

La ecuación generalizada para las zapatas sometidas a una carga axial y dos momentos ortogonales (flexión biaxial) es:

$$\sigma = \frac{P}{A} + \frac{M_x y}{I_x} + \frac{M_y x}{I_y}. \quad (1)$$

dónde: σ = esfuerzo generado por el suelo en cualquier parte de la zapata (presión del suelo), A = área en planta de la zapata (superficie de contacto sobre el suelo), P = carga concentrada sobre la zapata, M_x = momento alrededor del eje "X", M_y = momento alrededor del eje "Y", x = distancia paralela al eje "X" medida a partir del eje "Y" al punto de análisis, y = distancia paralela al eje "Y" medida a partir del eje "X" al punto de análisis, I_y = momento de inercia sobre el eje "Y" de la cimentación e I_x = momento de inercia sobre el eje "X" de la cimentación.

La Figura 2 muestra una zapata combinada de correa en esquina que soporta tres columnas rectangulares sometidas a una carga axial y dos momentos ortogonales (flexión biaxial) debido a cada columna.

La Figura 3 muestra el diagrama de presiones generadas por el suelo para una zapata combinada de correa en esquina.

Los esfuerzos generados en cada vértice de la zapata combinada de correa en esquina por la ecuación (1) se obtienen:

$$\sigma_1 = \frac{R}{A} + \frac{M_{xT} y_s}{I_x} + \frac{M_{yT} x_i}{I_y}, \quad (2)$$

$$\sigma_2 = \frac{R}{A} + \frac{M_{xT} y_s}{I_x} + \frac{M_{yT}(x_i - z_{1a})}{I_y}, \quad (3)$$

$$\sigma_3 = \frac{R}{A} + \frac{M_{xT} y_s}{I_x} + \frac{M_{yT}(x_i - a + z_{2a})}{I_y}, \quad (4)$$

$$\sigma_4 = \frac{R}{A} + \frac{M_{xT} y_s}{I_x} + \frac{M_{yT}(x_i - a)}{I_y}, \quad (5)$$

$$\sigma_5 = \frac{R}{A} + \frac{M_{xT}(y_s - c_y/2 + c_a/2)}{I_x}, + \frac{M_{yT}(x_i - z_{1a})}{I_y}, \quad (6)$$

$$\sigma_6 = \frac{R}{A} + \frac{M_{xT}(y_s - c_y/2 + c_a/2)}{I_x} + \frac{M_{yT}(x_i - a + z_{2a}),}{I_y}, \quad (7)$$

$$\sigma_7 = \frac{R}{A} + \frac{M_{xT}(y_s - c_y/2 - c_a/2)}{I_x} + \frac{M_{yT}(x_i - z_{1a})}{I_y}, \quad (8)$$

$$\sigma_8 = \frac{R}{A} + \frac{M_{xT}(y_s - c_y/2 - c_a/2)}{I_x} + \frac{M_{yT}(x_i - a + z_{2a})}{I_y}, \quad (9)$$

$$\sigma_9 = \frac{R}{A} + \frac{M_{xT}(y_s - z_{1b})}{I_x} + \frac{M_{yT} x_i}{I_y}, \quad (10)$$

$$\sigma_{10} = \frac{R}{A} + \frac{M_{xT}(y_s - z_{1b})}{I_x} + \frac{M_{yT}(x_i - c_x/2 + c_b/2)}{I_y}, \quad (11)$$

$$\sigma_{11} = \frac{R}{A} + \frac{M_{xT}(y_s - z_{1b})}{I_x} + \frac{M_{yT}(x_i - c_x/2 - c_b/2)}{I_y}, \quad (12)$$

$$\sigma_{12} = \frac{R}{A} + \frac{M_{xT}(y_s - z_{1b})}{I_x} + \frac{M_{yT}(x_i - z_{1a})}{I_y}, \quad (13)$$

$$\sigma_{13} = \frac{R}{A} + \frac{M_{xT}(y_s - z_{2b})}{I_x} + \frac{M_{yT}(x_i - a + z_{2a})}{I_y}, \quad (14)$$

$$\sigma_{14} = \frac{R}{A} + \frac{M_{xT}(y_s - z_{2b})}{I_x} + \frac{M_{yT}(x_i - a)}{I_y}, \quad (15)$$

$$\sigma_{15} = \frac{R}{A} + \frac{M_{xT}(y_s - b + z_{3b})}{I_x} + \frac{M_{yT}(x_i - c_x/2 + c_b/2)}{I_y}, \quad (16)$$

$$\sigma_{16} = \frac{R}{A} + \frac{M_{xT}(y_s - b + z_{3b})}{I_x} + \frac{M_{yT}(x_i - c_x/2 - c_b/2)}{I_y}, \quad (17)$$

$$\sigma_{17} = \frac{R}{A} + \frac{M_{xT}(y_s - b + z_{3b})}{I_x} + \frac{M_{yT}(x_i - z_{3a})}{I_y}, \quad (18)$$

$$\sigma_{18} = \frac{R}{A} + \frac{M_{xT}(y_s - b + z_{3b})}{I_x} + \frac{M_{yT} x_i}{I_y}, \quad (19)$$

$$\sigma_{19} = \frac{R}{A} + \frac{M_{xT}(y_s - b)}{I_x} + \frac{M_{yT} x_i}{I_y}, \quad (20)$$

$$\sigma_{20} = \frac{R}{A} + \frac{M_{xT}(y_s - b)}{I_x} + \frac{M_{yT}(x_i - z_{3a})}{I_y}. \quad (21)$$

Aquí R = fuerza resultante, M_{xT} = momento resultante alrededor del eje "X" y M_{yT} = momento resultante alrededor del eje "Y", estos se obtienen por las ecuaciones siguientes:

$$R = P_1 + P_2 + P_3, \quad (22)$$

$$M_{xT} = M_{x1} + M_{x2} + M_{x3} + R\left(y_s - \frac{c_y}{2}\right) - P_3 L_2, \quad (23)$$

$$M_{yT} = M_{y1} + M_{y2} + M_{y3} + R\left(x_i - \frac{c_x}{2}\right) - P_2 L_1. \quad (24)$$

Las propiedades geométricas de la zapata combinada de correa en esquina (vista en planta) se obtienen por las ecuaciones siguientes:

$$:A = z_{1a}z_{1b} + z_{2a}z_{2b} + z_{3a}z_{3b} + (a - z_{1a} - z_{2a})c_a + (b - z_{1b} - z_{3b})c_b, \quad (25)$$

$$y_s = \frac{[(z_{1a}-c_b)z_{1b}{}^2+(a-z_{1a}-z_{2a})c_a c_y+z_{2a}z_{2b}{}^2+c_b(b-z_{3b})^2+z_{3a}z_{3b}(2b-z_{3b})]}{\{2[z_{1a}z_{1b}+z_{2a}z_{2b}+z_{3a}z_{3b}+(a-z_{1a}-z_{2a})c_a+(b-z_{1b}-z_{3b})c_b]\}}, \quad (26)$$

$$y_i = b - y_s, \quad (27)$$

$$x_i = \frac{[(z_{1b}-c_a)z_{1a}{}^2+(b-z_{1b}-z_{3b})c_b c_x+z_{2a}z_{2b}(2a-z_{2a})+c_a(a-z_{2a})^2+z_{3a}{}^2 z_{3b}]}{\{2[z_{1a}z_{1b}+z_{2a}z_{2b}+z_{3a}z_{3b}+(a-z_{1a}-z_{2a})c_a+(b-z_{1b}-z_{3b})c_b]\}}, \quad (28)$$

$$x_d = a - x_i, \quad (29)$$

$$I_x = \frac{z_{1a}z_{1b}{}^3}{12} + z_{1a}z_{1b}\left(y_s - \frac{z_{1b}}{2}\right)^2 + \frac{z_{2a}z_{2b}{}^3}{12} + z_{2a}z_{2b}\left(y_s - \frac{z_{2b}}{2}\right)^2 + \frac{z_{3a}z_{3b}{}^3}{12} + z_{3a}z_{3b}\left(b - y_s - \frac{z_{3b}}{2}\right)^2 + \frac{(a-z_{1a}-z_{2a})c_a{}^3}{12} + (a - z_{1a} - z_{2a})c_a\left(y_s - \frac{c_y}{2}\right)^2 + \frac{c_b(b-z_{1b}-z_{3b})^3}{12} + c_b(b - z_{1b} - z_{3b})\left(\frac{b+z_{1b}-z_{3b}}{2} - y_s\right)^2, \quad (30)$$

$$I_y = \frac{z_{1b}z_{1a}{}^3}{12} + z_{1b}z_{1a}\left(x_i - \frac{z_{1a}}{2}\right)^2 + \frac{z_{2b}z_{2a}{}^3}{12} + z_{2b}z_{2a}\left(a - x_i - \frac{z_{2a}}{2}\right)^2 + \frac{z_{3b}z_{3a}{}^3}{12} + z_{3b}z_{3a}\left(x_i - \frac{z_{3a}}{2}\right)^2 + \frac{(b-z_{1b}-z_{3b})c_b{}^3}{12} + (b - z_{1b} - z_{3b})c_b\left(x_i - \frac{c_x}{2}\right)^2 + \frac{c_a(a-z_{1a}-z_{2a})^3}{12} + c_a(a - z_{1a} - z_{2a})\left(\frac{a+z_{1a}-z_{2a}}{2} - x_i\right)^2. \quad (31)$$

Las restricciones para las condiciones de frontera son:

Caso 1: Lados libres en las direcciones X e Y:

$$\frac{c_x}{2} + L_1 + \frac{z_{2a}}{2} = a. \quad (32)$$

$$\frac{c_y}{2} + L_2 + \frac{z_{3b}}{2} = b. \quad (33)$$

Caso 2: Lado restringido en la dirección X y libre en la dirección Y:

$$\frac{c_x}{2} + L_1 + \frac{c_x}{2} = a, \quad (34)$$

$$\frac{c_y}{2} + L_2 + \frac{z_{3b}}{2} = b. \quad (35)$$

Caso 3: Lado libre en la dirección X y restringido en la dirección Y:

$$\frac{c_x}{2} + L_1 + \frac{z_{2a}}{2} = a, \quad (36)$$

$$\frac{c_y}{2} + L_2 + \frac{c_y}{2} = b. \quad (37)$$

Caso 4: Lados restringidos en las direcciones X e Y:

$$\frac{c_x}{2} + L_1 + \frac{c_x}{2} = a, \quad (38)$$

$$\frac{c_y}{2} + L_2 + \frac{c_y}{2} = b. \quad (39)$$

Ahora la función objetivo para la superficie mínima de contacto con el suelo se muestra en la ecuación (25).

Las funciones de restricción para obtener las dimensiones de las zapatas y las vigas de liga en planta se obtienen por las ecuaciones (2) a (24) y (26) a (31), y además las ecuaciones (32) y (33) (caso 1), o las ecuaciones (34) y (35) (caso 2), o las ecuaciones (36) y (37) (caso 3), o las ecuaciones (38) y (39) (caso 4).

Nota: los esfuerzos deben ser menores o iguales a la capacidad de carga admisible disponible del suelo "σ_{ad}" y mayores o iguales a cero. Este modelo se puede usar para obtener el área mínima para zapatas combinadas de esquina, substituyendo $z_{1b} = c_a = z_{2b}$ y $z_{1a} = c_b = z_{3a}$.

3. Aplicación numérica

Cuatro ejemplos numéricos se presentan para obtener la superficie mínima de las zapatas combinadas de correa en esquina (Ejemplo 1: Lados libres en las direcciones X e Y, es decir, $c_x/2 + L_1 + z_{2a}/2 = a$ y $c_y/2 + L_2 + z_{3b}/2 = b$. Ejemplo 2: Lado restringido en la dirección X y libre en la dirección Y, es decir, $c_x/2 + L_1 + c_x/2 = a$ y $c_y/2 + L_2 + z_{3b}/2 = b$. Ejemplo 3: Lado libre en la dirección X y restringido en la dirección Y, es decir, $c_x/2 + L_1 + z_{2a}/2 = a$ y $c_y/2 + L_2 + c_y/2 = b$. Ejemplo 4: Lados restringidos en las direcciones X e Y, es decir, $c_x/2 + L_1 + c_x/2 = a$ y $c_y/2 + L_2 + c_y/2 = b$), y para cada ejemplo se presentan tres casos (Caso 1 toma en cuenta $z_{1a} = z_{1b}$, $z_{2a} = z_{2b}$ y $z_{3a} = z_{3b}$, es decir, todas las zapatas tienen diferentes dimensiones y son cuadradas cada una. Caso 2 considera $z_{1a}/M_{y1} = z_{1b}/M_{x1}$, $z_{2a}/M_{y2} = z_{2b}/M_{x2}$ y $z_{3a}/M_{y3} = z_{3b}/M_{x3}$, es decir, las zapatas están relacionadas con los momentos. Caso 3 considera $z_{1a} = z_{1b}$, $z_{2a} = z_{2b}$, $z_{3a} = z_{3b}$, $z_{1a} = z_{2a}$ y $z_{1a} = z_{3a}$, es decir, todas las zapatas tienen las mismas dimensiones y son cuadradas).

Tabla 1. Ejemplo 1. Esquina columna 1

	Caso 1				Caso 2				Caso 3			
σ_{ad}	250	210	170	130	250	210	170	130	250	210	170	130
I_x	99.95	113.17	128.47	143.33	104.05	119.32	138.08	159.29	123.39	144.77	187.32	294.51
I_y	148.81	172.65	203.05	240.26	143.85	165.23	191.57	221.74	162.64	191.19	248.25	392.08
M_{xT}	0	0	0	0	0	0	0	0	715.98	1067.78	1659.46	2804.21
M_{yT}	0	0	0	0	0	0	0	0	-551.45	-226.15	328.25	1423.08
a	9.17	9.26	9.37	9.49	9.34	9.44	9.55	9.66	9.21	9.32	9.52	9.96
b	8.07	8.13	8.21	8.26	7.92	7.98	8.05	8.12	8.21	8.32	8.52	8.96
x_i	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.26	3.36	3.54	3.88
x_d	5.74	5.83	5.94	6.06	5.91	6.01	6.12	6.23	5.95	5.96	5.98	6.08
y_s	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.86	2.97	3.16	3.51
y_i	5.43	5.49	5.57	5.62	5.28	5.34	5.41	5.48	5.35	5.35	5.36	5.45
z_{1a}	1.72	2.11	2.63	3.40	1.99	2.44	3.03	3.93	2.02	2.24	2.65	3.53
z_{1b}	1.72	2.11	2.63	3.40	1.49	1.83	2.28	2.94	2.02	2.24	2.65	3.53
z_{2a}	1.95	2.13	2.34	2.58	2.28	2.48	2.70	2.92	2.02	2.24	2.65	3.53
z_{2b}	1.95	2.13	2.34	2.58	1.63	1.77	1.93	2.09	2.02	2.24	2.65	3.53
z_{3a}	1.73	1.87	2.01	2.13	2.15	2.34	2.55	2.77	2.02	2.24	2.65	3.53
z_{3b}	1.73	1.87	2.01	2.13	1.43	1.56	1.70	1.84	2.02	2.24	2.65	3.53
σ_1	250	210	170	130	250	210	170	130	218.78	198.97	170	130
σ_2	250	210	170	130	250	210	170	130	225.62	201.63	166.50	117.19
σ_3	250	210	170	130	250	210	170	130	243.17	207.35	160.91	106.64
σ_4	250	210	170	130	250	210	170	130	250	210	157.41	93.83
σ_5	250	210	170	130	250	210	170	130	225.33	201.26	166.05	116.72
σ_6	250	210	170	130	250	210	170	130	242.87	206.98	160.47	106.17
σ_7	250	210	170	130	250	210	170	130	223.59	199.04	163.40	113.86
σ_8	250	210	170	130	250	210	170	130	241.13	204.77	157.81	103.31
σ_9	250	210	170	130	250	210	170	130	207.08	182.44	146.53	96.40
σ_{10}	250	210	170	130	250	210	170	130	207.25	182.50	146.46	96.22
σ_{11}	250	210	170	130	250	210	170	130	208.26	182.86	146.06	95.13
σ_{12}	250	210	170	130	250	210	170	130	213.91	185.09	143.02	83.59
σ_{13}	250	210	170	130	250	210	170	130	231.46	190.82	137.44	73.04
σ_{14}	250	210	170	130	250	210	170	130	238.30	193.47	133.93	60.23
σ_{15}	250	210	170	130	250	210	170	130	183.02	154.19	117.89	78.06
σ_{16}	250	210	170	130	250	210	170	130	184.04	154.55	117.49	76.97
σ_{17}	250	210	170	130	250	210	170	130	189.69	156.79	114.45	65.44
σ_{18}	250	210	170	130	250	210	170	130	182.85	154.14	117.95	78.24
σ_{19}	250	210	170	130	250	210	170	130	171.15	137.60	94.48	44.64
σ_{20}	250	210	170	130	250	210	170	130	177.99	140.25	90.98	31.83
A_{min}	12.80	15.24	18.82	24.62	12.80	15.24	18.82	24.62	15.01	17.68	23.30	38.81

Los datos a considerar para la superficie mínima de las zapatas combinadas de correa en esquina son: $c_x = c_y = 0.40$ m, $c_a = c_b = 0.30$ m, $L_1 = 8.00$ m, $L_2 = 7.00$ m, $P_1 = 600$ kN, $P_2 = 1400$ kN, $P_3 = 1200$ kN, $M_{x1} = 150$ kN-m, $M_{x2} = 250$ kN-m, $M_{x3} = 200$ kN-m, $M_{y1} = 200$ kN-m, $M_{y2} = 350$ kN-m, $M_{y3} = 300$ kN-m son iguales en los cuatro ejemplos y en los tres casos, y en cada caso se presentan cuatro tipos de capacidad de carga admisible disponible del suelo de "σ_{ad} = 250, 210, 170, 130 kN/m^2". La función objetivo (superficie mínima) se obtiene por la ecuación (25), y las funciones de restricción se obtienen por las ecuaciones (2) a (24) y (26) a (31).

Tabla 2. Ejemplo 2. Esquina columna 1 y 2

	Caso 1				Caso 2				Caso 3			
σ_{ad}	250	210	170	130	250	210	170	130	250	210	170	130
I_x	96.82	109.63	124.81	141.15	100.74	115.69	134.59	158.31	145.50	173.24	213.65	278.70
I_y	127.30	144.77	166.53	193.64	118.60	133.35	151.31	173.30	148.30	169.47	197.36	236.13
M_{xT}	0	0	0	0	0	0	0	0	1253.61	1652.95	2154.10	2834.06
M_{yT}	0	0	0	0	0	0	0	0	-1216.10	-985.82	-702.46	-327.19
a	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40
b	8.05	8.12	8.19	8.26	7.91	7.97	8.04	8.12	8.34	8.47	8.66	9.92
x_i	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.05	3.13	3.21	3.33
x_d	4.97	4.97	4.97	4.97	4.97	4.97	4.97	4.97	5.35	5.27	5.19	5.07
y_s	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.64	3.03	3.15	3.31	3.52
y_i	5.41	5.48	5.55	5.62	5.27	5.33	5.40	5.48	5.31	5.32	5.35	5.40
z_{1a}	1.56	1.92	2.39	3.09	1.73	2.14	2.66	3.43	2.27	2.55	2.91	3.45
z_{1b}	1.56	1.92	2.39	3.09	1.30	1.61	2.00	2.58	2.27	2.55	2.91	3.45
z_{2a}	2.15	2.37	2.65	2.99	2.58	2.84	3.17	3.57	2.27	2.55	2.91	3.45
z_{2b}	2.15	2.37	2.65	2.99	1.85	2.03	2.26	2.55	2.27	2.55	2.91	3.45
z_{3a}	1.71	1.84	1.99	2.12	2.12	2.31	2.52	2.77	2.27	2.55	2.91	3.45
z_{3b}	1.71	1.84	1.99	2.12	1.41	1.54	1.68	1.85	2.27	2.55	2.91	3.45
σ_1	250	210	170	130	250	210	170	130	181.12	161.14	140.10	118.36
σ_2	250	210	170	130	250	210	170	130	199.74	175.94	150.47	123.14
σ_3	250	210	170	130	250	210	170	130	231.38	195.19	159.63	125.22
σ_4	250	210	170	130	250	210	170	130	250	210	170	130
σ_5	250	210	170	130	250	210	170	130	199.31	175.47	149.97	122.63
σ_6	250	210	170	130	250	210	170	130	230.94	194.72	159.13	124.71
σ_7	250	210	170	130	250	210	170	130	196.73	172.60	146.94	119.58
σ_8	250	210	170	130	250	210	170	130	228.36	191.85	156.10	121.66
σ_9	250	210	170	130	250	210	170	130	161.55	136.85	110.73	83.30
σ_{10}	250	210	170	130	250	210	170	130	161.96	137.14	110.91	83.37
σ_{11}	250	210	170	130	250	210	170	130	164.42	138.89	111.98	83.79
σ_{12}	250	210	170	130	250	210	170	130	180.17	151.66	121.10	88.08
σ_{13}	250	210	170	130	250	210	170	130	211.81	170.91	130.26	90.17
σ_{14}	250	210	170	130	250	210	170	130	230.43	185.71	140.63	94.94
σ_{15}	250	210	170	130	250	210	170	130	129.28	104.87	82.37	62.74
σ_{16}	250	210	170	130	250	210	170	130	131.74	106.62	83.44	63.16
σ_{17}	250	210	170	130	250	210	170	130	147.49	119.39	92.56	67.45
σ_{18}	250	210	170	130	250	210	170	130	128.87	104.58	82.19	62.67
σ_{19}	250	210	170	130	250	210	170	130	109.30	80.30	52.82	27.62
σ_{20}	250	210	170	130	250	210	170	130	127.92	95.10	63.19	32.39
A_{min}	12.80	15.24	18.82	24.62	12.80	15.24	18.82	24.62	17.77	21.44	27.08	36.72

Además las ecuaciones (32) y (33) para el ejemplo 1, o las ecuaciones (34) y (35) para el ejemplo 2, o las ecuaciones (36) y (37) para el ejemplo 3, o las ecuaciones (38) y (39) para el ejemplo 4. Las superficies mínimas para las zapatas combinadas de correa en esquina se obtienen usando el software MAPLE-15, y los resultados para el ejemplo 1 (ver. Tabla 1), para el ejemplo 2 (ver. Tabla 2), para el ejemplo 3 (ver. Tabla 3), y para el ejemplo 4 (ver. Tabla 4).

Tabla 3. Ejemplo 3. Esquina columna 1 y 3

	Caso 1				Caso 2				Caso 3			
σ_{ad}	250	210	170	130	250	210	170	130	250	210	170	130
I_x	87.49	97.71	109.62	122.55	93.97	106.33	121.39	138.98	89.75	99.50	116.63	155.00
I_y	146.39	170.24	201.33	241.67	141.45	162.62	189.12	221.03	150.15	172.15	214.63	331.08
M_{xT}	0	0	0	0	0	0	0	0	-245.84	-47.44	270.12	901.99
M_{yT}	0	0	0	0	0	0	0	0	-520.32	-248.21	205.74	1184.39
a	9.17	9.26	9.37	9.50	9.33	9.43	9.54	9.67	9.17	9.26	9.42	9.80
b	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40
x_i	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.27	3.36	3.50	3.80
x_d	5.74	5.83	5.94	6.07	5.90	6.00	6.11	6.24	5.90	5.90	5.92	6.00
y_s	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.56	2.62	2.72	2.92
y_i	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.84	4.78	4.68	4.48
z_{1a}	1.59	1.96	2.44	3.18	1.88	2.32	2.88	3.73	1.93	2.11	2.44	3.21
z_{1b}	1.59	1.96	2.44	3.18	1.41	1.74	2.16	2.80	1.93	2.11	2.44	3.21
z_{2a}	1.94	2.12	2.33	2.59	2.27	2.46	2.69	2.93	1.93	2.11	2.44	3.21
z_{2b}	1.94	2.12	2.33	2.59	1.62	1.76	1.92	2.09	1.93	2.11	2.44	3.21
z_{3a}	1.92	2.09	2.29	2.47	2.32	2.54	2.80	3.09	1.93	2.11	2.44	3.21
z_{3b}	1.92	2.09	2.29	2.47	1.55	1.70	1.87	2.07	1.93	2.11	2.44	3.21
σ_1	250	210	170	130	250	210	170	130	212.95	195.65	170	130
σ_2	250	210	170	130	250	210	170	130	219.64	198.69	167.66	118.52
σ_3	250	210	170	130	250	210	170	130	238.02	205.95	163.31	106.40
σ_4	250	210	170	130	250	210	170	130	244.71	208.99	160.97	94.93
σ_5	250	210	170	130	250	210	170	130	219.78	198.72	167.55	118.23
σ_6	250	210	170	130	250	210	170	130	238.16	205.97	163.19	106.11
σ_7	250	210	170	130	250	210	170	130	220.60	198.86	166.85	116.49
σ_8	250	210	170	130	250	210	170	130	238.98	206.11	162.50	104.37
σ_9	250	210	170	130	250	210	170	130	218.24	196.65	164.35	111.33
σ_{10}	250	210	170	130	250	210	170	130	218.41	196.73	164.30	111.15
σ_{11}	250	210	170	130	250	210	170	130	219.45	197.16	164.02	110.08
σ_{12}	250	210	170	130	250	210	170	130	224.93	199.70	162.02	99.85
σ_{13}	250	210	170	130	250	210	170	130	243.31	206.95	157.66	87.73
σ_{14}	250	210	170	130	250	210	170	130	250	210	155.32	76.26
σ_{15}	250	210	170	130	250	210	170	130	228.10	198.24	158.46	105.42
σ_{16}	250	210	170	130	250	210	170	130	229.14	198.67	158.17	104.36
σ_{17}	250	210	170	130	250	210	170	130	234.62	201.21	156.17	94.13
σ_{18}	250	210	170	130	250	210	170	130	227.93	198.17	158.51	105.61
σ_{19}	250	210	170	130	250	210	170	130	233.22	199.17	152.86	86.94
σ_{20}	250	210	170	130	250	210	170	130	239.91	202.22	150.52	75.46
A_{min}	12.80	15.24	18.82	24.62	12.80	15.24	18.82	24.62	13.83	15.86	19.96	32.19

4. Resultados

La Tabla 1 muestra lo siguiente (Ejemplo 1):

Caso 1: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a y b aumentan, la posición del centro de gravedad x_i e y_s son los mismos y x_d e y_i aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 2: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos

Tabla 4. Ejemplo 4: Esquina columna 1, 2 y 3

	Caso 1				Caso 2				Caso 3			
σ_{ad}	250	210	170	130	250	210	170	130	250	210	170	130
I_x	84.87	94.74	106.40	119.72	91.08	103.15	118.16	136.98	99.49	112.16	128.55	151.17
I_y	125.24	142.64	164.65	193.04	116.66	131.19	149.05	171.35	136.25	154.91	179.51	213.97
M_{xT}	0	0	0	0	0	0	0	0	110.55	350.54	737.35	1006.79
M_{yT}	0	0	0	0	0	0	0	0	-1183.27	-969.59	-710.34	-369.97
a	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40	8.40
b	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40	7.40
x_i	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.43	3.06	3.13	3.21	3.32
x_d	4.97	4.97	4.97	4.97	4.97	4.97	4.97	4.97	5.34	5.27	5.19	5.08
y_s	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.64	2.67	2.75	2.84	2.95
y_i	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.76	4.73	4.65	4.56	4.45
z_{1a}	1.42	1.77	2.21	2.87	1.62	2.02	2.52	3.24	2.14	2.39	2.72	3.19
z_{1b}	1.42	1.77	2.21	2.87	1.22	1.51	1.89	2.43	2.14	2.39	2.72	3.19
z_{2a}	2.13	2.35	2.63	3.00	2.56	2.82	3.14	3.56	2.14	2.39	2.72	3.19
z_{2b}	2.13	2.35	2.63	3.00	1.83	2.02	2.25	2.54	2.14	2.39	2.72	3.19
z_{3a}	1.88	2.05	2.24	2.45	2.28	2.50	2.76	3.08	2.14	2.39	2.72	3.19
z_{3b}	1.88	2.05	2.24	2.45	1.52	1.67	1.84	2.05	2.14	2.39	2.72	3.19
σ_1	250	210	170	130	250	210	170	130	177.05	157.42	136.76	115.48
σ_2	250	210	170	130	250	210	170	130	195.66	172.38	147.51	121.00
σ_3	250	210	170	130	250	210	170	130	231.39	195.05	159.25	124.48
σ_4	250	210	170	130	250	210	170	130	250	210	170	130
σ_5	250	210	170	130	250	210	170	130	195.60	172.22	147.27	120.67
σ_6	250	210	170	130	250	210	170	130	231.33	194.89	159.00	124.14
σ_7	250	210	170	130	250	210	170	130	195.27	171.28	145.78	118.67
σ_8	250	210	170	130	250	210	170	130	231.00	193.95	157.51	122.15
σ_9	250	210	170	130	250	210	170	130	174.67	149.96	123.29	94.20
σ_{10}	250	210	170	130	250	210	170	130	175.10	150.27	123.49	94.29
σ_{11}	250	210	170	130	250	210	170	130	177.71	152.15	124.67	94.81
σ_{12}	250	210	170	130	250	210	170	130	193.28	164.91	134.04	99.73
σ_{13}	250	210	170	130	250	210	170	130	229.01	187.58	145.77	103.21
σ_{14}	250	210	170	130	250	210	170	130	247.62	202.53	156.53	108.73
σ_{15}	250	210	170	130	250	210	170	130	171.64	142.08	113.74	87.55
σ_{16}	250	210	170	130	250	210	170	130	174.25	143.95	114.94	88.07
σ_{17}	250	210	170	130	250	210	170	130	189.82	156.72	124.30	92.99
σ_{18}	250	210	170	130	250	210	170	130	171.21	141.76	113.54	87.46
σ_{19}	250	210	170	130	250	210	170	130	168.83	135.00	100.07	66.19
σ_{20}	250	210	170	130	250	210	170	130	187.44	149.25	110.82	71.71
A_{min}	12.80	15.24	18.82	24.62	12.80	15.24	18.82	24.62	15.94	19.00	23.63	31.51

resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a y b aumentan, la posición del centro de gravedad x_i e y_s son los mismos y x_d e y_i aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 3: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} aumentan, la dimensión a y b aumentan, la posición del centro de gravedad x_i, x_d, y_s e y_i aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, el esfuerzo máximo permitido para σ_{ad} = 250 kN/m^2 se alcanzan en σ_4 = 250 kN/m^2 y el esfuerzo mínimo σ_{19} = 171.15 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 210 kN/m^2 se alcanzan en σ_4 = 210 kN/m^2 y el esfuerzo mínimo σ_{19} = 137.60 kN/m^2,

el esfuerzo máximo permitido para σ_{ad} = 170 kN/m^2 se alcanzan en σ_1 = 170 kN/m^2 y el esfuerzo mínimo σ_{20} = 90.98 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 130 kN/m^2 se alcanzan en σ_1 = 130 kN/m^2 y el esfuerzo mínimo σ_{20} = 31.83 kN/m^2, las áreas mínimas aumentan.

La Tabla 2 muestra lo siguiente (Ejemplo 2):

Caso 1: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a es constante e igual a 8.40 m y b aumentan, la posición del centro de gravedad x_i, x_d e y_s son los mismos e y_i aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 2: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a es constante e igual a 8.40 m y b aumentan, la posición del centro de gravedad x_i, x_d e y_s son los mismos e y_i aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 3: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} aumentan, la dimensión a es constante e igual a 8.40 m y b aumentan, la posición del centro de gravedad x_i, y_s e y_i aumentan y x_d disminuye, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, el esfuerzo máximo permitido para σ_{ad} = 250 kN/m^2 se alcanzan en σ_4 = 250 kN/m^2 y el esfuerzo mínimo σ_{19} = 109.30 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 210 kN/m^2 se alcanzan en σ_4 = 210 kN/m^2 y el esfuerzo mínimo σ_{19} = 80.30 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 170 kN/m^2 se alcanzan en σ_4 = 170 kN/m^2 y el esfuerzo mínimo σ_{19} = 52.82 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 130 kN/m^2 se alcanzan en σ_4 = 130 kN/m^2 y el esfuerzo mínimo σ_{19} = 27.62 kN/m^2, las áreas mínimas aumentan.

La Tabla 3 muestra lo siguiente (Ejemplo 3):

Caso 1: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a aumentan y b es constante e igual a 7.40 m, la posición del centro de gravedad x_i, y_s e y_i son los mismos y x_d aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 2: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a aumentan y b es constante e igual a 7.40 m, la posición del centro de gravedad x_i, y_s e y_i son los mismos y x_d aumentan, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 3: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} aumentan, las dimensiones a aumentan y b es constante e igual a 7.40 m, la posición del centro de gravedad x_i, x_d e y_s aumentan e y_i disminuye, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, el esfuerzo máximo permitido para σ_{ad} = 250 kN/m^2 se alcanzan en σ_{14} = 250 kN/m^2 y el esfuerzo mínimo σ_1 = 212.95 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 210 kN/m^2 se alcanzan en σ_{14} = 210 kN/m^2 y el esfuerzo mínimo σ_1 = 195.65 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 170 kN/m^2 se alcanzan en σ_1 = 170 kN/m^2 y el esfuerzo mínimo σ_{20} = 150.52 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 130 kN/m^2 se alcanzan en σ_1 = 130 kN/m^2 y el esfuerzo mínimo σ_{20} = 75.46 kN/m^2, las áreas mínimas aumentan.

La Tabla 4 muestra lo siguiente (Ejemplo 4):

Caso 1: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a es constante e igual a 8.40 m y b es constante e igual a 7.40 m, la posición del centro de gravedad x_i, x_d, y_s e y_i son los mismos, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 2: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} son constantes e iguales a cero, las dimensiones a es constante e igual a 8.40 m y b es constante e igual a 7.40 m, la

posición del centro de gravedad x_i, x_d, y_s e y_i son los mismos, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, los esfuerzos en cada vértice de la zapata alcanzan el máximo permitido y es igual a σ_{ad}, las áreas mínimas aumentan.

Caso 3: Cuando σ_{ad} disminuye, los momentos de inercia I_x e I_y aumentan, los momentos resultantes M_{xT} y M_{yT} aumentan, las dimensiones a es constante e igual a 8.40 m y b es constante e igual a 7.40 m, la posición del centro de gravedad x_i, e y_s aumentan y x_d e y_i disminuye, los lados de las zapatas z_{1a}, z_{1b}, z_{2a}, z_{2b}, z_{3a} y z_{3b} aumentan, el esfuerzo máximo permitido para σ_{ad} = 250 kN/m^2 se alcanzan en σ_4 = 250 kN/m^2 y el esfuerzo mínimo σ_{19} = 168.83 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 210 kN/m^2 se alcanzan en σ_4 = 210 kN/m^2 y el esfuerzo mínimo σ_{19} = 135.00 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 170 kN/m^2 se alcanzan en σ_4 = 170 kN/m^2 y el esfuerzo mínimo σ_{19} = 100.07 kN/m^2, el esfuerzo máximo permitido para σ_{ad} = 130 kN/m^2 se alcanzan en σ_4 = 130 kN/m^2 y el esfuerzo mínimo σ_{19} = 66.19 kN/m^2, las áreas mínimas aumentan.

La Figura 4 muestra el área de la zapata 1 para cada ejemplo y en cada caso para cada capacidad de carga admisible disponible del suelo. El área menor se presenta en el ejemplo 4, caso 1 y σ_{ad} = 250 kN/m^2. El área mayor se presenta en el ejemplo 1, caso 3 y σ_{ad} = 130 kN/m^2.

La Figura 5 muestra el área de la zapata 2 para cada ejemplo y en cada caso para cada capacidad de carga admisible disponible del suelo. El área menor se presenta en el ejemplo 3, caso 2 y σ_{ad} = 250 kN/m^2. El área mayor se presenta en el ejemplo 1, caso 3 y σ_{ad} = 130 kN/m^2.

La Figura 6 muestra el área de la zapata 3 para cada ejemplo y en cada caso para cada capacidad de carga admisible disponible del suelo. El área menor se presenta en el ejemplo 2, caso 2 y σ_{ad} = 250 kN/m^2. El área mayor se presenta en el ejemplo 1, caso 3 y σ_{ad} = 130 kN/m^2.

La Figura 7 muestra el área total mínima de toda la cimentación para cada ejemplo y en cada caso para cada capacidad de carga admisible disponible del suelo. El área menor se presenta en los ejemplos 1, 2, 3 y 4, casos 1 y 2 y σ_{ad} =

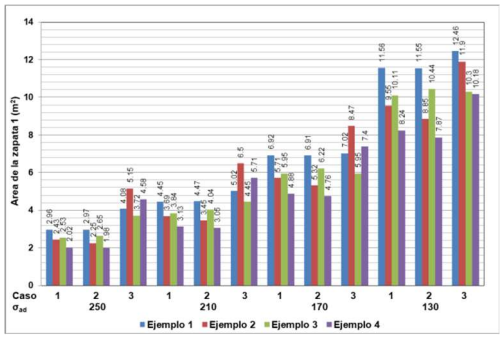


Fig. 4. Áreas de la zapata 1

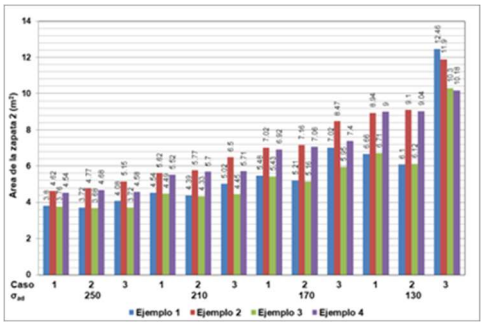


Fig. 5. Áreas de la zapata 2

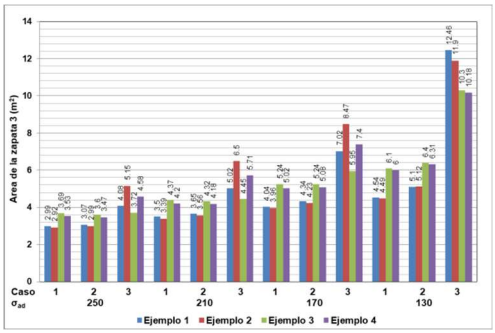


Fig. 6. Áreas de la zapata 3

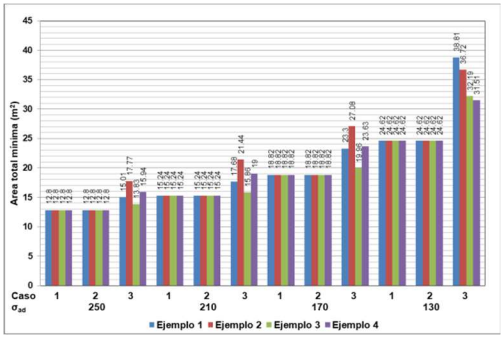


Fig. 7. Áreas totales mínimas de las cimentaciones

250 kN/m^2. El área mayor se presenta en el ejemplo 1, caso 3 y σ_{ad} = 130 kN/m^2. También se observa claramente que en los cuatro ejemplos para los casos 1 y 2, para cada capacidad de carga admisible disponible del suelo son los mismos.

5. Conclusiones

La cimentación de una construcción o edificación es la parte principal para transmitir las cargas de columna o muros al terreno debajo de la estructura. El nuevo modelo presentado en este artículo produce resultados que tienen una precisión sin precedentes para todos los problemas de ingeniería de cimentaciones. La parte principal de esta investigación es obtener la superficie mínima para las zapatas combinadas de correa en esquina apoyados sobre el terreno utilizando las técnicas de optimización.

Este estudio asume que el suelo de soporte es elástico y las zapatas son perfectamente rígidas, que cumplen con las ecuaciones de la flexión biaxial, es decir, la presión del suelo sobre la zapata varia linealmente.

El modelo propuesto presentado en este trabajo para encontrar la superficie mínima en planta para las zapatas combinadas de correa en esquina bajo una carga concéntrica y dos momentos ortogonales en cada columna, también se puede usar para los otros casos: 1) Zapatas bajo una carga concéntrica en cada columna, es decir, todos los momentos son cero; 2) Zapatas bajo una carga concéntrica y un momento en una dirección en cada columna, es decir, los momentos alrededor del eje X o Y son cero.

Las principales conclusiones son:

1 La metodología presentada en este trabajo de investigación es más precisa que cualquier otra metodología.
2 El modelo propuesto para obtener la superficie mínima en planta para zapatas combinadas de correa en esquina se puede usar para el modelo de zapatas combinadas de esquina propuesto por López-Chavarría *et al.* [11].
3 El modelo propuesto se puede utilizar para encontrar la superficie mínima en planta de zapatas combinadas de correa en esquina para dos, tres y cuatro líneas de propiedad de lados restringidos (ver Tablas 1 a 4).
4 Cuando los momentos resultantes de "M_{xT}" y "M_{yT}" son iguales a cero, esto significa que la fuerza resultante de todas las cargas y momentos se ubica en el centro de gravedad de la zapata, es decir, los esfuerzos generados por el suelo son los mismos e igual a la capacidad de carga admisible disponible del suelo (ver Tablas 1 a 4, casos 1 y 2).
5 El modelo propuesto es congruente con los resultados obtenidos sobre los esfuerzos generados por el suelo, porque cuando se presenta el esfuerzo máximo en un punto, y en el punto opuesto se presenta el esfuerzo mínimo, es decir, $\sigma_{max} = \sigma_4$ y $\sigma_{min} = \sigma_{19}$, $\sigma_{max} = \sigma_1$ y $\sigma_{min} = \sigma_{20}$ (ejemplo 1, caso 3), $\sigma_{max} = \sigma_4$ y $\sigma_{min} = \sigma_{19}$ (ejemplo 2, caso 3), $\sigma_{max} = \sigma_{14}$ y $\sigma_{min} = \sigma_1$, $\sigma_{max} = \sigma_1$ y $\sigma_{min} = \sigma_{20}$ (ejemplo 3, caso 3), $\sigma_{max} = \sigma_4$ y $\sigma_{min} = \sigma_{19}$ (ejemplo 4, caso 3).

Por lo tanto, el modelo propuesto en este trabajo de investigación para obtener la superficie mínima de las zapatas combinadas de correa en esquina se puede aplicar a zapatas combinadas de esquina, simplemente considerando los anchos en dirección X de las zapatas 1 y 3 iguales, y los anchos en dirección Y de las zapatas 1 y 2 iguales.

Las sugerencias para los siguientes trabajos de investigación pueden ser:

1 Superficie mínima de zapatas combinadas de correa en esquina apoyadas sobre suelos totalmente arcillosos (suelos cohesivos) o suelos totalmente arenosos (suelos granulares), es decir, el diagrama de presión del suelo sobre la zapata es parabólico.
2 Superficie mínima de cimentaciones completas para una edificación usando zapatas combinadas de correa.
3 Superficie mínima de cimentaciones completas para una edificación usando una losa de cimentación.

Agradecimientos

La investigación descrita en este trabajo fue financiada por el Instituto de Investigaciones Multidisciplinarias de la Facultad de Contaduría y Administración de la Universidad Autónoma de Coahuila. Los autores también agradecen a los revisores y al editor por los comentarios y sugerencias para mejorar la presentación. El estudiante de doctorado María Azucena Moreno Hernandez (CVU: 934750) agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico.

Referencias

1. **Chagoyén, E., Negrín, A., Cabrera, M., López, L., Padrón, N. (2009).** Diseño óptimo de cimentaciones superficiales rectangulares. Formulación. Revista de la Construcción, Vol. 8, No. 2, pp. 60–71.
2. **Hassaan, G. A. (2014).** Optimal design of machinery shallow foundations with silt soils. International Journal of Mechanical Engineering (IJME), Vol. 4, No. 3, pp. 11–24.
3. **Yeh, J. P., Huang, K. H. (2017).** Effects of strengths of steel and concrete, eccentricity and bar size on the optimization of eccentrically loaded footings. Transactions on Machine Learning and Artificial Intelligence, Vol. 5, No. 5, pp. 87–97. DOI: 10.14738/tmlai.55.3592.
4. **Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M. (2017).** Optimal design for rectangular isolated footings using the real soil pressure. Ingeniería e Investigación, Vol. 37, No. 2, pp. 25–33. DOI: 10.15446/ing. investig.v37n2.61447.
5. **Jelušič, P., Žlender, B. (2018).** Optimal design of pad footing based on MINLP optimization. Soils and Foundations, Vol. 58, No. 2, pp. 277–289. DOI: 10.1016/j.sandf. 2018.02.002.
6. **Malapur, M. M., Cholappanavar, P., Fernandes, R. J. (2018).** Optimization of RC column and footings using genetic algorithm. International Research Journal of Engineering and Technology (IRJET), Vol. 5, No. 8, pp. 546–552.
7. **Rawat, S., Mittal, R. K. (2018).** Optimization of eccentrically loaded reinforced-concrete isolated footings. Practice Periodical on Structural Design and Construction, Vol. 23, No. 2. DOI: 10.1061/(ASCE)SC.1943-5576.0000366.
8. **López-Chavarría, S., Luévanos-Rojas, A., Medina-Elizondo, M., Sandoval-Rivas, R., Velázquez-Santillán, F. (2019).** Optimal design for the reinforced concrete circular isolated footings. Advances in Computational Design, Vol. 4, No. 3, pp. 273–294. DOI: 10.12989/acd.2019.4.3.273.
9. **Al-Ansari, M. S. (2013).** Structural cost of optimized reinforced concrete isolated footing. International Journal of Civil and Environmental Engineering, Vol. 7, No. 4, pp. 290–297. DOI: 10.5281/zenodo.1080444.
10. **Chaudhuri, P., Maity, D. (2020).** Cost optimization of rectangular RC footing using GA and UPSO. Soft Computing, Vol. 24, No. 2, pp. 709–721. DOI: 10.1007/s00500-019-04437-x.
11. **Ray, R., Kumar, D., Samui, P., Roy, L. B., Goh, A. T. C., Zhang, W. (2021).** Application of soft computing techniques for shallow foundation reliability in geotechnical engineering. Geoscience Frontiers, Vol. 12, No. 1, pp. 375–383. DOI: 10.1016/j.gsf. 2020.05.003.
12. **Solorzano, G., Plevris, V. (2020).** Optimum design of rc footings with genetic algorithms according to aci 318-19. Buildings, Vol. 10, No. 6, pp. 1–17. DOI: 10.3390/buildings 10060110.
13. **Khajuria, K., Singh, B., Singla, S. (2020).** Design optimization of RC column footings under axial load of beam and rooftop surfaces. International Journal of Scientific & Technology Research, Vol. 9, No. 2, pp. 5683–5688.
14. **El-Kady, M. S., Badrawi, E. F. (2017).** Performance of isolated and folded footings. Journal of Computational Design and Engineering, Vol. 4, No. 2, 150–157. DOI: 10.1016/j.jcde.2016.09.001.

15. **Gandomi, A. H., Kashani, A. R. (2018).** Construction cost minimization of shallow foundation using recent swarm intelligence techniques. IEEE Transactions on Industrial Informatics, Vol. 14, No. 3, pp. 1099–1106. DOI: 10.1109/TII.2017.2776132.
16. **Kashani, A. R., Gandomi, M., Camp, C. V., Gandomi, A. H. (2020).** Optimum design of shallow foundation using evolutionary algorithms. Soft Computing, Vol. 24, pp. 6809–6833. DOI: 10.1007/s00500-019-04316-5.
17. **Das, M. R., Mohanty, M., Das, S. K. (2021).** Multi-objective optimum design of geosynthetic reinforced soil foundation using genetic algorithm. In: **Samui, P., Kumari, S., Makarov, V., Kurup, P., eds.**, Modeling in Geotechnical Engineering, chapter 8. Academic Press, pp. 151–164. DOI: 10.1016/B978-0-12-821205-9.00018-6.
18. **Hui, L., Zhuoyi, C., Mingji, Z. (2015).** Genetic algorithm application on optimal design of strip foundation. The Open Cybernetics & Systemics Journal, Vol. 9, No. 1, pp. 335–339. DOI: 10.2174/1874110X 01509010335.
19. **López-Chavarría, S., Luévanos-Rojas, A., Medina-Elizondo, M. (2017).** Optimal dimensioning for the corner combined footings. Advances in Computational Design, Vol. 2, No. 2, pp. 169–183. DOI: 10.12989/acd.2017.2.2.169.
20. **Ranpura, N. K., Areakar, V., Patel, V. (2021).** Optimum design of combined rectangular RCC footing using GA. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Vol. 7, No. 2, pp. 202–210. DOI: 10.48175/IJARSCT-1679.
21. **Velázquez-Santillán, F., Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M., Sandoval-Rivas, R. (2018).** Numerical experimentation for the optimal design for reinforced concrete rectangular combined footings. Advances in Computational Design, Vol. 3, No. 1, pp. 49–69. DOI: 10.12989/acd.2018.3.1.049.
22. **Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M. (2018).** A new model for T-shaped combined footings Part I: Optimal dimensioning. Geomechanics and Engineering, Vol. 14, No. 1, pp. 51–60. DOI: 10.12989/gae.2018.14.1.051.
23. **Stanley, E. U. (2021).** Optimization of isolated and combined pad foundation using computer aided application of finite element approach. Global Journal of Engineering and Technology Advances, Vol. 6, No. 3, pp. 24–41. DOI: 10.30574/gjeta.2021.6.3.0030.
24. **Aguilera-Mancilla, G., Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M. (2019).** Modeling for the strap combined footings part I: optimal dimensioning. Steel and Composite Structures, Vol. 30, No. 2, pp. 97–108. DOI: 10.12989/scs.2019.30.2.097.
25. **Pasillas-Orona, A. I., Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M., Aguilera-Mancilla, G. (2020).** Un modelo optimizado para zapatas combinadas trapezoidales apoyadas sobre el terreno: Superficie óptima. Acta Universitaria, Vol. 30, No. e2973. DOI: 10.15174/au.2020.2973.

Article received on 17/11/2021; accepted on 25/07/2022.
Corresponding author is Arnulfo Luévanos Rojas

ISSN 2007-9737

Development of a Normalized Hadith Narrator Encyclopedia with TEI

Hajer Maraoui[1], Kais Haddar[2], Laurent Romary[3]

[1] University of Tunis El Manar,
Faculty of Sciences of Tunis, MIRACL Laboratory,
Tunisia

[2] University of Sfax,
Faculty of Science of Sfax, MIRACL Laboratory,
Tunisia

[3] Inria, Team ALMAnaCH,
Germany

hajer.maraoui@fst.utm.tn, kais.haddar@yahoo.fr, laurent.romary@inria.fr

Abstract. The investigation in the narrator list of prophetic tradition (hadith) is considered an important task for different hadith sciences such as the biography of narrators. In fact, the authenticity of hadith is intensely related to the chain of narrators how transmitted the story. In addition, this science is interested essentially in analyzing the hadith narrator profile. Indeed, having a standardized encyclopedia of hadith narrators can help researchers to explore and manipulate the various hadith documents and can simplify different analyzes. For this reason, we aim to develop a standardized hadith narrator encyclopedia with Text Encoding Initiative (TEI) language. To achieve this, we propose a TEI model for the hadith narrator properties. To experiment our TEI model, firstly, we construct a corpus of articles about hadith narrators from Wikipedia. Secondly, we use a system allowing the named entity recognition in relation with narrator data. Thirdly, we perform a post-processing to complete the corpus TEI annotation. Fourthly, we generate the hadith narrator encyclopedia based on our TEI model. The obtained result is encouraging despite some problems related to exceptional cases.

Keywords. Hadith narrator encyclopedia, narrator data standardization, TEI model, ANER system, Wikipedia database.

1 Introduction

To build knowledge used by Hadith sciences, the researchers collect the required information from different resources. The Internet is on the top of these resources via Islamic websites and free encyclopedia. The majority of questions under the theme of hadith are directed at its components Isnad or Sanad (the chain of narrators) and Matn (the narration text).

Besides, more interrogations are focusing on the chain of narrators of the hadith text, cited in the Isnad. From the hadith science perception, examining the chain of narrators is considered a main task for different disciplines such as the science of the biography of narrators (*al-jarḥ wa al-ta'dīl,* discrediting and accrediting) which proceed several aspects like hadith authentication and classification [1].

It is also substantial for the individual searcher who looks for the information related with hadith transmitters from the first generation of narrators: the "Sahaba" (or companions) of the Prophet

Muhammed (peace upon him) to their descendants. This data can be extended to cover more details concerning the narrator properties and relationships. In fact, this data needs to be collected in one standardized database to support the interoperability of the information.

From this perspective, we aim to create a normalized encyclopedia containing all the relative information of the hadith narrators, in order to prepare it to be used by different Islamic disciplines.

We use the Text Encoding Initiative (TEI guidelines) [2] to achieve a standardized formalism of the data. Besides, we use the free resources of Wikipedia to collect the required information about the hadith narrators. We follow a symbolic approach method to realize the hadith narrator encyclopedia.

This method starts with a TEI model development in order to encode the hadith narrator data. Then, we construct a corpus including hadith narrator properties. To create the hadith narrator encyclopedia, we annotate in TEI the constructed corpus using a named entity extraction tool [3]. Finally, we develop a prototype to generate the normalized narrator encyclopedia. The developed prototype experiments the proposed TEI model.

This paper is composed of six sections. Section 2 presents an overview on the related works. Section 3 gives details about our proposed TEI model for the data encoding of hadith narrator. Then, section 4 introduces our system for the construction of a normalized hadith encyclopedia. Section 5 presents the system evaluation step. We cloture our paper with a conclusion and perspectives in section 6.

2 Related Works

The term hadith is an Arabic word means report. Hadith mentions the speech or the action of the Prophet Muhammad (peace upon him) transmitted first orally between his companions and among the next generations of hadith narrators [4]. After that, the scientists of hadith traveled and collected hadiths to write it down in corpora [4].

The well-known corpus, Sahih al-Bukhari, is one of the six major hadith collections of Sunni Islam. Muhammad al-Bukhari, the author of this corpus, had collected 600000 hadiths of which he only considered 7275 ones as authentic in his work [5]. In next subsections, we present a summary on the Isnad of the hadith and some related works.

2.1 Overview on Hadith Isnad

The hadith text is characterized with two main components: the Matn /المتن/ and the Isnad /إسناد/ (or Sanad /سند/). The Matn contains the text of the narration and the Isnad states the list of narrators who transmitted the hadith.

The Isnad supports the hadith authenticity. In fact, the hadith sciences proved that Isnad is essential to verify whether a hadith is sound or not. The authentication method follows a careful examination of the chain of transmission to find out if the temporal and spatial links between the narrators are possible and to judge the reliability of the reporter.

Which also means that any weakness in the Isnad conclude that the relative Matn is rejected [1, 4-6]. Hadith processing was the topic of many projects focused on several fields of research (hadith ontology, linguistic analyzing, etc.). The following subsection presents some related works that experiment the Isnad treatment.

2.2 Related Works

Many researchers used several NLP methods and techniques to create different software and websites that helps to determine hadith judging and classification. For this field of research, we mention the following examples.

The research that carried out by [7] developed a prototype to build database to encode hadith and narrators with XML format. This work is based on a database that manages the hadiths of “Sahih Al-Boukari” book and the narrator information extracted from “Tahdheeb Altahdeeb” book.

This prototype identified the chain of narrators and treated the collected information with HPSG formalism. The prototype offered a graphical user interface allowing the access and the visualization of the list of narrators and hadiths from the treated books.

Table 1. Summary table of the person head element

Element	Description	
<listPerson**>**	(List of persons) represents a list or group of identifiable individuals where each one is marked up in a <person> element.	
<Person>	Includes all the information about a definite person.	
	Attribute	Description
	xml:id	Specifies one unique identifier for the <person> element.
	role	Mentions an additional information about the person like his occupation, or his social rank.
	sex	Indicates the gender of the person.
	age	Indicates the age of the person.

Table 2. Summery table of the TEI elements for the personal information

Category	Element	Description	
Name phrase of the person	<persName>	(Personal name) contains a part or full name of a person.	
	<surname>	Contains the family name of the mentioned person.	
	<forename>	Contains a forename of the mentioned person.	
		Attribute**:** *type*	Describes the type of the forename using a significant term.
		Attribute**:** *Sort*	Identifies the order of the forename comparing to other forenames.
	<roleName>	Contains the society rank or official title of the cited person.	
	<addName>	(Additional name) contains an additional name for the person.	
	<nameLink>	(Name link) contains a connecting phrase or link occurs in the phrase of the person name such as "*de*" or "*بن*".	
Dates related to the person	<birth>	Contains the birth details of the person like the date and place.	
		Attribute**:** *when*	Redefines the birth date in a standard form
	<death>	It is similar to birth element but contains the death details.	
		Attribute: *when*	Redefines the death date in a standard form
	<floruit>	Define the period during which a person lived.	
		Attribute: *notBefore*	Represents the initial date for the lived period in a standard form.
		Attribute: *notAfter*	Represents the ending date for the lived period in a standard form.
	<date>	Represent a date value in any format.	

The study conducted by [8] proposed a system allowing the recognition of hadith narrator chain and their classification. This system presented the extracted sequences of the narrator names in a graphical format as a network. The edges in this graph presented the transmission links between narrators.

The study performed by [9] putted forward an Isnad ontology to support the hadith authentication. The authors extended the terms of the ontology to cover more semantic relations and properties of hadith narrators to process the Isnad judging. The authors experimented an evaluation step with DL-Queries and hadith text examples.

Research that performed by [10] proposed an annotated narrator graph extractor (ANGE) from hadith text and biography books.

This technique is based on different NLP technologies like graph algorithms, cross-document reconciliation, finite state machines and morphological features.

According to the different works focused on the Isnad part, the chain of narrators is a key part that needs to be extracted and annotated for the authentication process of the hadith.

However, only few studies, frequently in ontology, extended the narrators presentation to extract their properties and the semantic relations between them.

Yet, in the one hand, according to our knowledge, no study has concentrated on applying a standardization formalism like TEI to normalize the extracted information. In the other hand, there is more details in the person properties and the relationships that can be mentioned to define the narrator historical profile.

3 TEI Modelling of Hadith Narrator Data

One of the benefits of applying TEI normalization on any type of data is the flexibility of TEI modelling. TEI covers large data categories and classes and allows the restructuring of the elements without losing the specification of each component [11]. We took advantage from these features to propose a TEI encoding model formed for hadith narrator.

For the final encyclopedia, this model represents the primer unit encoding which refers to a hadith reporter. Therefore, to create the required TEI model, we start with the selection of the elements from the core module that coordinates with the person data encoding.

Then, we form the structural design of the final model. This section contains, in a first part, an overview on the collection of the elements dedicated for the named entities and person properties description. Then, in a second part, we present our proposed TEI model.

3.1 Encoding the Person Properties with TEI

TEI guidelines [2] recommend a full module to encode the named entities in the text [12]. Person name is one of the most detailed entities in this module. Moreover, TEI presented different elements to encode other types of named entities that indicate an information about that person such as birth date, place of residence, etc. Indeed, the main TEI element to represent all information related to a person is <person> [12].

Table 1 illuminates this element and some attributes connected with it. The <person> element is the main component to define a person. Besides, the <listPerson> element can be used to enumerate a list of persons as precise index or bibliography.

The <person> element is characterized with a number of attributes that can serve to present some personal information like “xml:id” for the personal identifier, or “age”, “sex” and “role” for farther specifications. The rest of the data related to the person can be observed as two categories: personal and social information.

Table 2 summarizes TEI elements for the encoding of the personal information. The <persName> element can distinguish a name phrase of a person in the text. It is possible to include one or more name component (like forename, surname, added name, etc.) in the corresponding elements imbricated inside <persName> element.

It is also the case for the <birth> and the <death> elements which contain all the information about the dates in <date> elements and their places in <placeName> elements (defined bellow in Table 3). Moreover, these elements can have the “when” attribute to present a standard form of the date value. The second category of data is the social information. Table 3 summarizes TEI elements for this category.

TEI provides several elements to markup different information about the language skills, the residence and the social identity of a person. Additionally, information about relationships between the people also makes a part in the TEI encoding for the person data. Table 4 summarizes the two main TEI elements for this information.

Table 3. Summary table of the TEI elements for the social information

Category	Element	Description	
Language competences of the person	<langKnowledge>	(Language knowledge) presents the linguistic knowledge of the person in <langKnow> elements.	
	<langKnow>	(Language known) indicates one linguistic competence	
		Attribute: *tag*	Provides a practical tag for the relative language.
		Attribute: *level*	Precise the level of knowledge of the language
Places related to the person	<residence>	Refer to the places of residence of the person.	
	<placeName>	Contains an absolute or relative place name.	
		This element can have the attributes: *notBefore and notAfter* (see Table 2)	
	<country>	Specifies the geo-political unit, such as the nation, country, commonwealth, etc.	
		Attribute: *Key*	Identify a meaningful value defined externally to identify the named entity.
	<settlement>	Contains a single geo-political or administrative unit, such as a city, town, or village.	
		This element can have the attribute: *type* (see table 2).	
Social identity of the person	<nationality>	Describe of a person nationality or citizenship.	
	<education>	Describe of the educational experience.	
	<affiliation>	Present the person affiliation.	
	<faith>	Specifies the faith, religion of a person.	

The element <listRelation> englobes a sequence of <relation> element, to express relation links between mentioned persons. In fact, the <relation> element performs the main unit of the relationship encoding. It can come with the attributes “type” and “name” to specify the classification and the designation of the relationship. As well, it can contain the attributes “active”, “passive” and “mutual” to specify whether the links are conjoint or not.

Here, only one of the attributes “active” and “mutual” can be supplied. The attributes “passive” and “active” can only supplied together. This constraint is not enforced for all schema language. For example, the following personal relation illustrates the link between the father “UrwaBinAz-Zubair” and his son “HishamBinUrwa”:

```
<relation type="personal" name="FatherOf"
active = "#UrwaBinAz-Zubair"
passive="#HishamBinUrwa" cert="high"/>
```

To achieve the TEI modelling of the hadith narrator data, we select the adaptable TEI elements from TEI module for encoding persons, dates, places and personal relationships [11]. The proposed TEI model is presented in the next section.

3.2 Proposed TEI Model for the Hadith Narrator Data

For the representation of the hadith narrators in an encyclopedic structure, we start with the basic elements that include the unit sequence. Then, each component is presented in the element that englobes all the relative information.

Fig. 1 illustrates the basic elements model that we adapt with the data provided for the hadith reporter in the Islamic resources and Arabic literature. The representation begins by encoding the properties of the encyclopedia in the <teiHeader> element. Then, it uses the <body>

Table 4. Summery table for the TEI elements for the relationships between persons

TEI element	Description	
<listRelation>	Provides information about relationships either identified amongst people, places, and organizations, informally as prose or as formally expressed relation links.	
<relation>	(Relationship) describes any kind of relationship or linkage amongst a specified group of places, events, persons, objects or other items.	
	Attribute	Description
	type	Provides a unique identifier for the element bearing the attribute
	name	Supplies a name for the kind of relationship of which this is an instance.
	active	Identifies the 'active' participants in a non-mutual relationship, or all the participants in a mutual one.
	passive	Identifies the 'passive' participants in a non-mutual relationship.
	mutual	Supplies a list of participants amongst all of whom the relationship holds equally.
	cert	(Certainty) signifies the degree of certainty associated with the object pointed to by the certainty element.

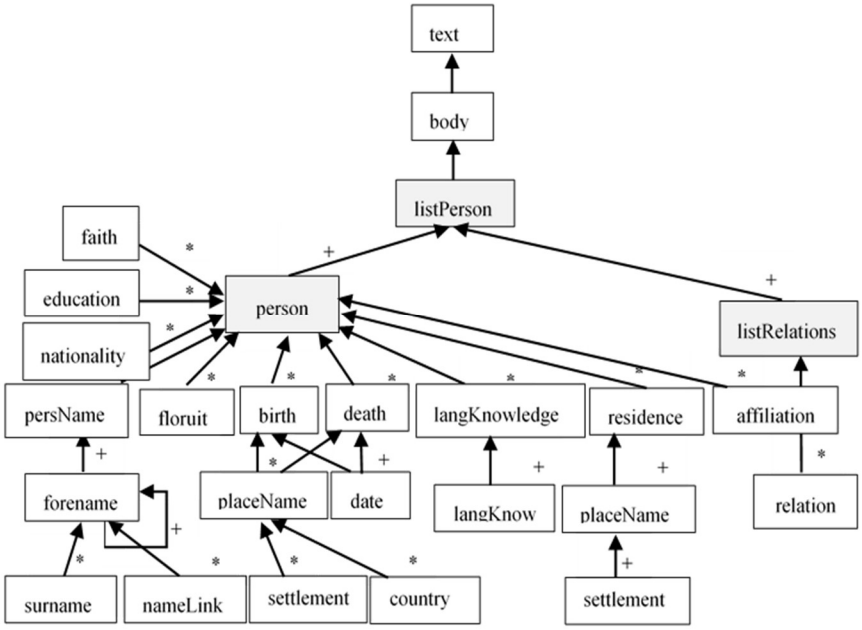


Fig. 1. Extraction from TEI encoding model for the details of hadith narrator

element including in the <text> element. The <body> element contains the main <listPerson> component that covers the entire list of narrators and its information.

This element contains a sequence of <person> element and a <listRelation> element that represents respectively the information about hadith narrators and the relationships between them. The <person> element has an "xml:id" attribute to assign a unique identifier for each narrator, which will then be used for narrator identification in the relationship list. Moreover, this identifier will be the connector link used in the Isnad encoding of each hadith.

In addition, the <person> element includes all the personal and the social properties of the hadith

narrator. The personal properties such as the full name, the birth date, the death date and their places are encoded respectively in <persName>, <birth> and <death> elements and the corresponding place names in <placeName> element.

The <floruit> element contains the period that the narrator lived according to the birth and the death dates. The social properties such as the nationality, the residence, the education, the religion, the affiliation and the spoken languages are described respectively in <nationality>, <residence>, <education>, <faith>, <affiliation> and <langKnowledge> elements.

4 Elaborated Method for the Construction of a Hadith Narrator Encyclopedia

To achieve the final hadith narrator encyclopedia, we start with a conceptual study of a creation system for the encyclopedia. The realization of this system follows a symbolic approach. It based on a method composed of three main phases. Fig. 2 illustrates the general architecture of the system.

The method allowing the construction of the encyclopedia is composed of three successive stages. In this method, the first step starts with a connection to the Wikipedia resources to provide all the data relative to the narrator profile. This procedure is the step of collecting the articles about the hadith narrators. We rely on this database for the reason that it provides all the authentic information about hadith narrators. In addition, Wikipedia offers the free accessibility to the required information. Also, through its Kiwix platform, it allows the backup of resources as .txt files. For this reason, we use the open-source version of the Wikipedia database for Arabic language.

The second phase performs a named entity recognition step. This step is based on a system proposed by [3]. This system generates the extracted named entities in XML output and borders each detected component with TEI encoding. In addition, we perform a post-processing to extend the data recognition process for the rest of the social information and the relationships. The obtained XML files are used as an input of the next phase.

The third step is for the TEI encoding process of the collected narrator properties. In this step, we developed a tool for the automatic encoding of the data. This tool applies the TEI encoding to generate the base of the narrator encyclopedia. This encoding procedure follows our specific TEI model for hadith narrator properties.

5 Experimentation and Evaluation

For the implementation, we use some tools and programs. Indeed, for the manipulation of the TEI files, we use Oxygen XML Editor. Besides, we develop the prototype using JAVA language and the JDOM Library.

The output of our system is the encyclopedia of narrators encoded in TEI. The following TEI code illustrates an example of generated data encoding for the hadith narrator: "الزبير بْنِ العوام" (Az-ZubairBinAl-Awwam).

```
<person xml:id="Az-ZubairBinAl-Awwam"
role="hadithNarrator" sex="1" age =
"Adult">
   <persName xml:lang="ara">
      <forename>الزبير</forename>
      <forename sort="1" type = "nasab">
          <nameLink>بْنِ</nameLink>
      <forename>العوام</forename></forename>
      <forename sort="1" type =
"nisba">الأسدي</forename>
      <surname>القريشي</surname></persName>
   <birth when="0594">
      <date when="0594">594 AD - 28
BH</date>
      <placeName>
          <settlement
type="city">Makkah</settlement>
          <country key="KSA">The Arabian
Peninsula</country>
      </placeName></birth>
   <death when="0656">
      <date when="0656">656 AD - 38
AH</date>
      <placeName><settlement
type="city">Bassorah</settlement>
         <country key="Ir">Iraq</country>
      </placeName></death>
   <faith>Islam</faith>
   <langKnowledge>
```

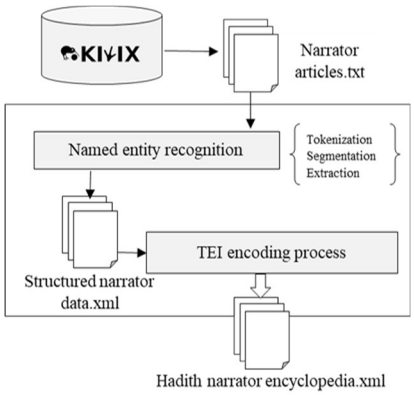


Fig. 2. Developed method for the construction of hadith narrator encyclopedia encoded in TEI

Table 5. Result obtained by the system

Hadith narrator Information	Encoded correctly	Encoded incorrectly
642 article of hadith narrators from Wikipedia	96%	4%

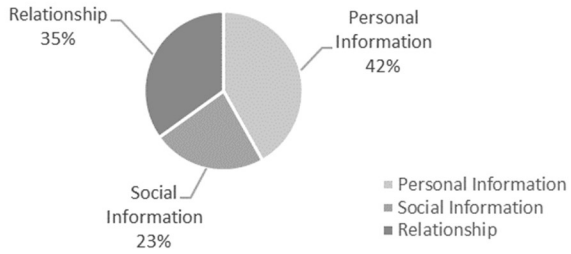


Fig. 3. Information coverage inside the hadith narrator encyclopedia

Table 6. Summary table of the precision, recall and F-score

Precision	Recall	F-measure
0.96	0.96	0.96

```
      <langKnown tag="ar" level="H">
Arabic</langKnown>
   </langKnowledge>
   <nationality>Arabian
Peninsula</nationality>
   <residence notBefore="0584"
notAfter="0656">
      <placeName><settlement
type="city">Makkah</settlement>
      </placeName></residence>
   <affiliation>Politician</affiliation>
   <education>Islamic
legislation</education>
   <floruit notBefore="0584"
notAfter="0656"/></person>
```

After the sequence of <person> elements, the system encodes all the relationships detected between the narrators in the <listRelations> element.

These relationships refer to all personal and social connections between different people (such as "father of", "mother of", "son of", "uncle of", etc.). The following TEI code presents an extraction of

the generated relationships encoded in the <relation> sub-elements:

```
<listRelation>
<relation type="personal" name="SpouseOf"
mutual="#Az-ZubairBinAl-Awwam
#AsmaBintAbuBakr" cert="high"/>
<relation type="personal" name="FatherOf"
active="#Az-ZubairBinAl-Awwam"
passive="#UrwaBinAz-Zubair" cert="high"/>
<relation type="personal" name="MotherOf"
active = "#AsmaBintAbuBakr"
passive="UrwaBinAz-Zubair" cert="high"/>
<relation type="personal" name="SonOf"
active="#UrwaBinAz-Zubair" passive="#Az-
ZubairBinAl-Awwam #AsmaBintAbuBakr"
cert="high"/>
<relation type="personal" name="UncleOf"
active="#AbdullahBinAz-Zubair"
passive="#HishamBinUrwa" cert="high"/>
…</listRelation>
```

To evaluate our system, we use Kiwix platform of Wikipedia to collect the articles about the hadith narrators from the list of the prophet companions. This database provides us with information about 642 hadith narrators. Therefore, we obtain an article for each narrator.

As a result, the system generates the hadith narrator encyclopedia encoded with TEI, representing all the collected data. Table 5 illustrates the obtained results.

The system succeeded to generate a total correct encoding of the information of 616 hadith narrators while respecting the TEI model. This number is equal to 96% of the treated data. However, 4% of the information is not detected or incorrectly encoded.

This is related with some particular forms of the information inside the hadith narrator articles, or a wrong detection or encoding of the information related with some ambiguous parts.

Regarding the results, the coverage of the information in the hadith narrator encyclopedia is divided on three themes: personal information, social information and relationships. Fig. 3 presents their measurements.

The majority of information is about the hadith transmitter personal information (full name, birth and death) with 42% of the total information. Then, the relationships cover a 35% of the encoded data. The last 23% of the data is for the social information (affiliation, language, education, etc.). We measure the quality of our work with the values of precision, recall and F-measure presented in Table 6.

The calculation provides an identical precision and recall equal to 0.96. Consequently, F-measure as well is equal to 0.96. Therefore, these measurements prove that the obtained results are encouraging.

6 Conclusion and Perspectives

The normalization of the data collection of hadith narrator in one organized encyclopedia can support the manipulation of the information and reduce the difficulties of deep studies.

To realize our main objective, we based on TEI standardization. In this work, we started with an overview on the hadith chain of narrators in the Isnad. Then, we continued with a study on the TEI module for encoding person properties. After that, we proposed an adaptable TEI model to encode the information about hadith narrator.

Then, we followed a symbolic approach to accomplish the creation of the hadith narrator encyclopedia. The first phase was to collect the data from Arabic Wikipedia. The second phase was to extract the required information including a system of named entity recognition.

The final phase was to encode the collection of data with the proposed TEI model. Then, we tested our system with 642 articles about hadith narrators from Wikipedia. As mentioned, the obtained measures showed that the results were encouraging. This normalized encyclopedia can be used in many other applications.

As perspectives, we want to select more information about hadith narrators from other authentic resources to enrich the data coverage in the encyclopedia.

Moreover, we want to expand our TEI model to cover more details related to hadith narrators by integrating other specifications.

In addition, we want to improve our system to resolve the problems. Besides that, we want to use the normalized encyclopedia to test a developed system for analyzing hadith text.

References

1. **Alrfaai, S. (2004).** Scholars' attention to the chain of narrators and the science of *Jarh* and modification, and its impact on preserving the Prophet's Sunnah, "*Einayat aleulama' bialaisinad waeilm aljurh waltaedil w 'athar dhalik fi hifz alssnt alnabawia*". AlMadina Almonawara.
2. **Burnard, L., Sperberg-McQeen, C. M. (2016).** TEI P5: guidelines for electronic text encoding and interchange. Text Encoding Initiative Consortium, Version 3.0.0, revision 89ba24e.
3. **Mesmia, F. B., Friburger, N., Haddar, K., Maurel, D. (2015).** Arabic named entity recognition process using Transducer Cascade and Arabic Wikipedia. Proceedings of Recent Advances in Natural Language Processing, pp. 48–54.
4. **Abu-Zaho, M. (1984).** Hadith and Muhaddithin, "*Alhadith walmuhdithuna*". House of Arab Thought, Riyadh, KSA, Vol. 1.
5. **Alaskalani, A. (2001).** Fath Al-Bari with the explanation of Sahih Al-Bukhari, "*Fath albari bisharh sahih al-Bukharii*". KSA, Vol. 1.
6. **Alaskalani, A. (2008).** Gilding Approximation Refinement, "*Taqrib altahdhib*". Society AlRisala, Bayrout, Labunan.
7. **Najeeb, M. A. (2016).** XML Database for hadith and narrators. American Journal of Applied Sciences, Vol. 13, No. 1, pp. 55–63. DOI: 10.3844/ajassp.2016.55.63.
8. **Siddiqui, M. A., Saleh, M. E. S., Bagais, A. A. (2014).** Extraction and visualization of the chain of narrators from hadiths using named entity recognition and classification. International Journal of Computational Linguistics Research, Vol. 5, No. 1, pp. 14–25.
9. **Baraka, R. S., Dalloul, Y. (2014).** Building hadith ontology to support the authenticity of Isnad. International Journal on Islamic Applications in Computer Science and Technology, Vol.2, No. 1, pp. 25–39.
10. **Zaraket, F., Makhlouta, J. (2012).** Arabic Cross-Document NLP for the Hadith and Biography Literature. 25th International Florida Artificial Intelligence Research Society Conference, pp. 256–261.
11. **Burnard, L., Sperberg-McQeen, C. M. (1996).** La TEI simplifiée : Une introduction au codage des textes électroniques en vue de leur échange. Cahiers GUTenberg, No. 24, pp. 23–151.
12. **Dufournau, N., Demonet, M. L., Uetani, T. (2008).** Manuel d'encodage XML-TEI Renaissance et temps modernes Imprimés-manuscrits. Version Beta, UMR, Vol. 6576.
13. **Maraoui, H., Haddar, K., Romary, L. (2017).** Encoding prototype of Al-Hadith Al-Shareef in TEI. Communications in Computer and Information Science, ICALP Conference, Vol 782, pp. 1–13. DOI: 10.1007/978-3-319-73500-9_16.

Article received on 09/01/2019; accepted on 02/04/2021.
Corresponding author is Hajer Maraoui.

ISSN 2007-9737

What's Your Style? Automatic Genre Identification with Neural Network

Andrea Dömötör[1,3], Tibor Kákonyi[1], Zijian Győző Yang[1,2]

[1] MTA-PPKE Hungarian Language Technology Research Group, Hungary

[2] Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, Hungary

[3] Pázmány Péter Catholic University, Faculty of Humanities and Social Studies, Hungary

{yang.zijian.gyozo, domotor.andrea}@itk.ppke.hu
kakonyi.tibor@hallgato.ppke.hu

Abstract. Genre identification is an important task in natural language processing that can be useful for many practical and research purposes. The challenge of this task is that genre is not a homogeneous and unequivocal property of the texts and it is often hard to separate from the topic. In this paper we compare the performance of two different automatic genre identification methods. We classified six text types: literary, academic, legal, press, spoken and personal. In one part of our research we did experiments with traditional machine learning methods using linguistic, n-gram and error features. In the other part we tested the same task with a word embedding based neural network. In this part we did experiments with different training data (words only, POS-tags only, words and POS-tags etc.). Our results revealed that neural network is a suitable method for this task while traditional machine learning showed significantly lower performance. We gained high (around 70%) accuracy with our word embedding based method. The results of the different text categories seemed to depend on the stylistic properties of the studied genres.

Keywords. Genre identification, text classification, machine learning, neural networks, word embedding, stylistics

1 Introduction

Automatic genre identification is an application of computational stylistics which originates from the idea that the different text types have different lexical and grammatical features.

While the term *genre* can be interpreted in several ways (see overview in [2]), modern definitions usually mention the communicative purpose (function), content and form as the main distinctive properties of genres. In this study we concentrate on the last characteristic: the structural and lexical features of the different text types (form). This decision is in line with both our methods and motivation. On the one hand, we used linguistic properties (words, lemmas and POS-tags) as training data in our experiments.

On the other hand, our purpose of building an automatic genre identification system is also linguistically motivated. We expect this system to support the creation of genre-specific (sub)corpora which can be useful for corpus linguistics and stylistic studies. Genre identification may also help other natural language processing systems (for

example, rule-based parsers) by allowing the use of genre-specific rules.

The traditional genre identification methods are based on the selection of features ([5][9]). These can be either surface features like function words, genre-specific words, word length or sentence complexity; structural features, for example parts of speech or verb tenses or presentation and other features, such as token types or links. The classification algorithms used for this task also vary in the literature from decision trees, through naive Bayes and regression models to neural networks and clustering.

In this paper we compare two methods on the same training data set. In one part of this study, we did experiments with a classification model based on feature extraction. In the other part we used deep neural networks and word embedding. The peculiarity of our work is that it is sentence-based, while other studies of genre identification usually use bigger text units. However, not all of them. [6] for instance, actually searches the minimal unit necessary to identify genre.

The reason we chose this type of task is, on one hand, that the style of web pages may not be homogeneous. For this reason it is important to be able to deal with smaller text units in order to build genre-specific corpora. On the other hand, as we mentioned before, the study also has the purpose to enable genre identification for natural language processing tools and corpus linguists. In these cases it can be necessary to identify the genre of a one-sentence input or research data.

2 Training Data

The training data was extracted from the Hungarian Gigaword Corpus (HGC) [8]. The corpus contains 187.6 million tokens of lemmatized and morphologically annotated texts from different genres. The analysis of the corpus was realized with the Humor morphological analyzer tool [10] which is a reversible, string-based, unification approach for lemmatizing and disambiguation.

Our training data was provided by the press, literary, academic, legal, personal and spoken language subcorpora. The press subcorpus contains texts from news webpages. This adds up the major part of the whole HGC. The literary subcorpus is a processed collection of digitally available texts of Hungarian literature. The academic texts originate from a Hungarian digital library. The legal subcorpus contains texts of laws, decrees and parliamentary records. The personal subcorpus is built of web forum conversations. These texts are usually below standard and often noisy. Finally, the spoken language corpus consists of transcriptions of radio programmes.

We queried 300 thousand random sentences from each type. The training data elaborated of these sentences contains the original words, lemmas and POS-tags. We used all these three characteristics for our experiments because we presume that genres have both particular lexical and structural characteristics.

Vocabulary is an obvious distinctive feature of text types. Table 1 shows the most frequent trigrams of the different genres (not taken into consideration punctuation marks and conjunctions). As it can be seen, the categories are more or less recognizable from their common collocations, however there are similarities due to similar topics (legal, press, spoken) or to the generality of the genre's vocabulary (personal, literary). As Hungarian is a morphologically rich language, it seems adequate to use both full word forms and lemmas.

The relevance of POS-tags is demonstrated in Table 2 which shows the relative frequency of personal pronouns in each text type. These data reveal that press and academic texts show strong preference to the third person[1], while the second person is sightly more prominent in personal and literary texts compared to the other genres. These characteristics are expected to cause significant differences in the distribution of (verbal) POS-tags.

We created 5 different kinds of training and test corpus. These contain the following information:

— Full word forms (original text).

— Lemmas.

— POS-tags.

[1] This stands for legal texts as well, if we take into consideration that the formal *you* (*ön*) in Hungarian also takes the third person.

Table 1. Most frequent trigrams of text types

Personal
nem csak a – 'not only the'
az a baj – 'the problem is'
a mai napon – 'this day'
még akkor is – 'even if'
még mindig nem – 'still not'
Legal
megadom a szót – 'I give the floor'
az Európai Unió – 'the European Union'
a módosító javaslatot – 'the amendment'
köszönöm a szót – 'thank you for the floor'
nem fogadta el – 'has not accepted'
Literary
ez volt a – 'this was the'
ha nem is – 'even if not'
még akkor is – 'even if'
még mindig nem – 'still not'
nem is tudom – 'I don't know'
Spoken
én azt gondolom – 'I think'
az Európai Unió – 'the European Union'
jó reggelt kívánok – 'good morning'
jó napot kívánok – 'good afternoon'
az Európai Bizottság – 'the European Commitee'
Press
az Egyesült Államok – the United States
az Európai Unió – the European Union
a tervek szerint – 'according to plans'
a múlt héten – 'last week'
az Európai Bizottság – 'the European Commitee'
Academic
a második világháború – 'the second world war'
a 19. század – 'the 19th century'
részt vett a – 'took part in'
volt az első – 'was the first'
a 20. század – 'the 20th century'

— Full word forms and lemmas.

— Full word forms and POS-tags.

The combined types are necessary to distinguish homographs. The two missing types (lemmas and POS-tags; full word forms, lemmas and POS-tags) are redundant, because the combinations of full word forms and POS-tags, lemmas and POS-tags and full word forms, lemmas and POS-tags equally determine the word unambiguously.

We used the texts as they appeared in the corpus, no preprocessing steps or normalization was applied. In our judgment quality issues can play a significant role in genre identification, for instance, the omission of accented characters or punctuation marks is a characteristic of informal texts. The only intervention to the corpus data was the filtering of duplications and some noise (like html tags or meta data).

3 Methods and Experiments

3.1 Traditional Machine Learning Method

In one part of our research we did experiments to build a classification model using traditional machine learning. For this task we tested various classification methods and the Random Forest algorithm gained the best results, thus in this paper we show only the results of our Random Forest model (RFM).

To build the RFM, we used the PiRate system [12]. We implemented 37 different kinds of features. According to the functionality, we can separate these features into the following categories:

— Linguistic features:

  – Percentage of nouns, verbs, pronouns, adverbs, adjectives, conjunctions, pronouns, determiners, preverbs, numerals, interjections in the sentence.
  – Ratio of number of nouns and verbs in the sentence.
  – Ratio of number of nouns and adjectives in the sentence.
  – Ratio of number of verbs and preverbs in the sentence.
  – Ratio of number of nouns and determiners in the sentence.
  – Rumber of tokens.
  – Average word length in the sentence.

Table 2. Relative frequency of pronouns in different genres

	én ('I')	te (you.sg) (informal)	ön (you.sg) (formal)	ő ('he/she')	mi ('we')	ti (you.pl)	ők ('they')
Personal	33.3%	15.1%	3.0%	23.2%	11.2%	3.9%	10.4%
Legal	28.5%	0.6%	26.0%	19.7%	16.2%	0.2%	8.7%
Literary	32.9%	10.7%	1.2%	32.1%	10.6%	1.5%	11.0%
Spoken	30.2%	1.6%	9.1%	26.0%	18.3%	0.4%	14.5%
Press	14.1%	2.8%	3.2%	41.9%	16.3%	0.5%	22.3%
Academic	11.1%	3.5%	0.9%	53.3%	8.4%	0.9%	21.8%

— n-gram features:
  – Sentence LM probability.
  – Sentence LM perplexity.
  – LM probability of lemmas and POS tags of the sentence.
  – LM perplexity of lemmas and POS tags of the sentence.

— Neural network features:
  – 1-gram, 2-gram and 3-gram perplexity.

— Error features:
  – Percentage of accented words in the sentence.
  – Percentage of unknown words in the sentence.
  – Percentage of punctuation marks in the sentence.

The training of the n-gram models (for the n-gram features) was effectuated with the SRILM [11] toolkit. As n-gram training corpus we used a subcorpus of the HGC that contains 98500 lemmatized and POS-tagged sentences.

For the training of the neural network language model we used a subcorpus of the Pázmány Corpus [3] that contains 1 million sentences. The language model was built with an RNN architecture with GRUs (Gated recurrent unit). We also used a Hungarian word embedding model [7] for word representation.

3.2 Word Embedding Masked Neural Network Method

In the other part of our research we made experiments using fastText, which is a state-of-the-art, neural network based library for word embedding [4] and text classification [1] developed by Facebook Artificial Intelligence Research.

For text classification fastText uses a linear classifier based on supervised learning, it needs labeled corpora as training and validation sets. During the training fastText builds an embedding model where labeled sentences and labels are represented as vectors in a way that a sentence is really close to its associated labels in the vector space.

An initial sentence vector is the average of embedding vectors of words inside the sentence. (An advantageous ability of fastText is that it does not work simply with words but with n-gram features, hence it is able to handle some partial information about the local word order.) The sentence vector is fed into a linear classifier and softmax function is used to calculate the probability distribution over labels. fastText uses stochastic gradient descent algorithm to maximize the probability of the correct label belonging to the sentence.

In our experiment each sentence of the corpus had one label that marked which style that piece of text belongs to. We trained models for all five kinds of the corpus with 27 different parameter sets that are generated as combinations of the following values:

— Number of epochs (number of times fastText sees a training example): 5, 27 or 50.

— Learning rate (degree of the model's change after processing an example): 0.1, 0.5 or 1.0.

— Maximal length of word n-grams: 1, 3 or 5.

(Only the model giving the best results is mentioned for each corpus variety in the Results section.)

4 Results and Evaluation

Table 3 shows the accuracy results of our experiments. First, comparing the performance of the different training corpus types in the method described in chapter 3.2 it can be seen that, as was expected, the only POS-tag version gave the lowest results, 52% in average and 56.6% best case (0.1 learning rate, 5 epochs, 5-grams). Nevertheless, these results are still remarkable, taking into consideration how limited information the model had, and it still performed far above random. This means that the studied genres do have unique structural properties and the difference between them is not only lexical or thematic. As for the other four types of subcorpora, we have almost the same results and they also share the best parameter set (0.1 learning rate, 5 epochs, 3-grams). It seems, contrarily to the assumptions, that full morphology does not contribute much to the lexical-based genre identification: the model works just the same with lemmas only as with full word forms and POS-tags. In all four cases we got a fair (around 70% best case) result. Other observation worth to mention is that increasing the number of epochs did not prove to increase the performance.

Table 3 also shows that our Random Forest Model performed significantly below fastText, even if the latter only had the POS-tags as input data. These results demonstrate that word embedding methods are much more powerful for this task than traditional machine learning. The relative inefficiency of the Random Forest Model is, however, not that surprising if we consider that the majority of the features used in this method is not sensitive to the vocabulary.

Table 3. Accuracy results of the word embedding based and the random forest method

	Average accuracy	Best accuracy	Best n-gram parameter
Words	68.5%	70.7%	3
Lemmas	68.3%	70.7%	3
Words + POS-tags	68.2%	70.3%	3
Words + lemmas	68.0%	70.1%	3
POS-tags	52.0%	56.6%	5
RFM	-	43.2%	-

We also measured precision, recall and F-score by category (Table 4). In this case the four subcorpus types that contained words or lemmas still performed almost the same in the word embedding based method, for this reason we only show the results of the models using words and POS-tags.

As seen, fastText's full word form measurement gave the best result for legal texts with an F-score over 80%. Apparently, this is the genre with the most characteristic vocabulary, which is presumably related to its thematic boundedness. Literary, academic and spoken texts also achieved high results with this method. The relatively low performance of the personal type can be attributed to the low quality of this subcorpus. This assumption is even more plausible considering the fT-POS results. The difficulty of identifying this kind of texts by POS-tags can be caused by the significant number of erroneous tags (which occur frequently in this subcorpus due to the omission of accents, typos, abbreviations etc.).

The fT-POS results follow the same order but the numbers are lower in proportion (except for the extremely low recall of the personal type).

The majority of Random Forest Model's results does not even reach the f-measurse of word embedding with POS-tags, except in case of personal texts. The scores gained by the traditional machine learning method are generally low. The highest f-measure (50.8%) belongs to the literary genre but this result is still lower than the worst score of fT-word.

To detect the common faults we made a confusion matrix of the fastText-word experiment (Table 5). The personal type is often confused with the literary. The reason may be that both genres are quite liberal in terms of text composition.

Table 4. Precision and recall results by category

		fT-word	fT-POS	RFM
Legal	Precision	82.47%	62.01%	45.9%
	Recall	79.52%	65.82%	40.2%
	F-Measure	80.96%	63.86%	42.9%
Literary	Precision	72.08%	57.35%	44.9%
	Recall	79.00%	67.01 %	58.4%
	F-Measure	75.38%	61.80%	50.8%
Academic	Precision	74.38%	59.44%	45.2%
	Recall	71.75%	61.55%	53.6%
	F-Measure	73.04%	60.48%	49%
Spoken	Precision	69.97%	54.57%	36.1%
	Recall	72.23%	55.81%	37%
	F-Measure	71.08%	55.19%	36.5%
Press	Precision	56.47%	43.46%	36.8%
	Recall	57.58%	41.53%	33.4%
	F-Measure	57.02%	42.48%	35%
Personal	Precision	57.72%	53.16%	52.6%
	Recall	48.00%	25.17%	37%
	F-Measure	52.41%	34.16%	43.3%

Table 5. Confusion matrix

	Personal	Legal	Literary	Spoken	Press	Academic
Personal	57.72%	3.88%	15.41%	7.10%	9.40%	6.50%
Legal	1.94%	82.47%	1.94%	4.95%	4.92%	3.79%
Literary	7.92%	2.30%	72.08%	4.87%	5.92%	6.90%
Spoken	4.55%	6.73%	4.95%	69.97%	10.27%	3.52%
Press	8.79%	6.25%	5.18%	12.79%	56.47%	10.53%
Academic	3.87%	3.64%	6.59%	2.92%	8.59%	74.38%

The spoken texts seem to be related with the press genre.

This can be explained with the similarity of their topics. As we mentioned before, the spoken subcorpus consists of transcriptions of radio programmes which often contain news and public topics.

The relatively high number of confusions between the press and academic genres may be explicable with the observation shown in Table 2 that these text types typically prefer the third person.

Finally, it should be mentioned that the task of genre identification by definition does not assume 100% accuracy, as genre is not a unequivocal property of texts. Any genre can contain neutral sentences which have no distinctive stylistic characteristics. Therefore, 70% accuracy on sentence level can be considered significant.

5 Conclusion

In this paper we compared the results of a traditional machine learning and a word embedding based method in the task of automatic genre identification. For both methods we used corpora that contained lexical and grammatical information, namely words, lemmas and POS-tags.

According to our results, the word embedding method is much more powerful for this task. The performance of the neural network based system far surpassed the traditional machine learning algorithms. With word embedding we achieved promising results (around 70% accuracy).

Our experiments provided other interesting findings as well. The word embedding measurements revealed that using the POS-tags only can be more effective than expected. This suggests that genres have specific structural characteristics which allow to identify them without lexical or topic-related features.

Other observation of linguistic interest is that we got the same result when using full word forms and lemmas despite that Hungarian is an agglutinative language which means that a lemma can have varied word forms.

As for genre-related results, we found that legal, literary and academic texts are easier to identify than the other three examined genres (spoken, press, personal). It seems that these text types have more representative lexical and structural characteristics than the others. It is also important to mention that the spoken and personal language types represent greater variation in topics which makes the lexical-based genre identification harder.

Finally, it is to be mentioned that the traditional machine learning methods are more language-dependent than word embedding. The feature set of our machine learning model contains features that are specific for Hungarian (like the number of accented characters). Other languages may need different features. However, the word embedding method can be used to any language without modifications.

References

1. **Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2016).** Enriching word vectors with subword information. CoRR.
2. **Clark, M., Ruthven, I., O'Brian Holt, P. (2009).** The evolution of genre in wikipedia. Journal for Language Technology and Computational Linguistics, Vol. 24, No. 1, pp. 1–22.
3. **Endrédy, I., Prószéky, G. (2016).** A pázmány korpusz. Nyelvtudományi Közlemények, , No. 112, pp. 191–206.
4. **Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2016).** Bag of tricks for efficient text classification. CoRR.
5. **Lustrek, M. (2007).** Overview of Automatic Genre Identification. Jožef Stefan Institute, Department of Intelligent Systems, Ljubljana, Slovenia.
6. **McCarthy, P. M., Myers, J. C., Briner, S. W., Graesser, A. C., McNamara, D. S. (2009).** A psychological and computational study of sub-sentential genre recognition. Journal for Language Technology and Computational Linguistics, Vol. 24, No. 1, pp. 23–56.
7. **Novák, A., Novák, B. (2018).** Magyar szóbeágyazási modellek kézi kiértékelése. XIV. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2018), Szegedi Tudományegyetem, Szeged, Hungary.
8. **Oravecz, C., Váradi, T., Sass, B. (2014).** The Hungarian Gigaword Corpus. **Calzolari, N., et al.**, editors, Proceedings of the 9th International Conference on Language Resources and Evaluation, ELRA, Reykjavik, Iceland.
9. **Petrenz, P., Webber, B. L. (2011).** Stable classification of text genres. Computational Linguistics, Vol. 37, No. 2, pp. 385–393.
10. **Prószéky, G., Tihanyi, L. (1996).** Humor – a morphological system for corpus analysis. Proceedings of the first TELRI seminar in Tihany, Budapest, Hungary, pp. 149–158.
11. **Stolcke, A. (2002).** Srilm – an extensible language modeling toolkit. in proceedings of the 7th international conference on spoken language processing (ICSLP 2002, pp. 901–904.
12. **Yang, Z. G., Laki, L. J. (2017).** Pirate: A task-oriented monolingual quality estimation system. International Journal of Computational Linguistics and Applications, Vol. 8.

Article received on 18/02/2018; accepted on 20/01/2020.
Corresponding author is Andrea Dömötör.

ISSN 2007-9737

Metaphor Interpretation Using Word Embeddings

Kfir Bar[1], Nachum Dershowitz[2], Lena Dankin[2]

[1] The College of Management, School of Computer Science, Israel

[2] Tel Aviv University, School of Computer Science, Israel

{nachum,lenadank}@tau.ac.il, kfirb@colman.ac.il

Abstract. We suggest a model for metaphor interpretation using word embeddings trained over a relatively large corpus. Our system handles nominal metaphors, like *time is money*. It generates a ranked list of potential interpretations of given metaphors. Candidate meanings are drawn from collocations of the topic (*time*) and vehicle (*money*) components, automatically extracted from a dependency-parsed corpus. We explore adding candidates derived from word association norms (common human responses to cues). Our ranking procedure considers similarity between candidate interpretations and metaphor components, measured in a semantic vector space. Lastly, a clustering algorithm removes semantically related duplicates, thereby allowing other candidate interpretations to attain higher rank. We evaluate using different sets of annotated metaphors, with encouraging preliminary results.

Keywords. Metaphor interpretation, word embeddings.

Epigraph: *Writing about metaphor is dancing with your conceptual clothes off, the innards of your language exposed by equipment more powerful than anything operated by the TSA. Still, one would be a rabbit not to do it in a world where metaphor is now top dog, at least among revived rhetorical devices with philosophical appeal.*

Carlin Romano, "What's a Metaphor For?", *The Chronicle of Higher Education* (July 3, 2011)

1 Introduction

Metaphor is pervasive in language and thought [4]. Based on a quantitative analysis, Krennmayr [7] found that even in academic papers almost every fifth word is part of a metaphorical concept, broadly construed.

Already Aristotle analyzed and wrote about the use of metaphor. "Metaphor", he says in the *Poetics*, "consists in giving the thing a name that belongs to something else." *The sunset of life* is one of his examples. In *Rhetoric* he explains: "A simile is also a metaphor; for there is little difference: when the poet says, 'He rushed as a lion,' it is a simile, but 'The lion rushed' would be metaphor ['lion' referring to a human hero]; since both are brave."

Following a definition provided by [13], A *metaphor* is a rhetorical figure, which is a peculiar expression of a sentiment different from the ordinary way. A *simile* is that figure by which we compare one object with another. In other words, a metaphor is a simile without any formal comparison [13]. Other examples of simile are: *as sweet as pie* (nominal) and *eat like a bird* (verbal); the expressions *life is a roller coaster*, *time is money*, and *you are my sunshine*, are nominal metaphors.

Aristotle heaps praise on metaphor:

> To be a master of metaphor is the greatest thing by far. It is the one thing that cannot be learnt from others, and it is also a sign of genius.
>
> Metaphor especially has clarity and sweetness and strangeness.
>
> Words which make us learn something are most pleasant. ... It is metaphor,

> therefore that above all produces this effect; for when Homer calls old age stubble, he teaches and informs us through the genus; for both have lost their bloom.

Lakoff and Johnson [10] claim that the human conceptual system is extremely metaphorical in nature. They talk about that *metaphorical concepts* are being defined in terms of *nonmetaphorical concepts*. They explain that nonmetaphorical concepts are those that emerge directly from experience and can be defined in their own terms. Therefore, metaphorical concepts are composed of their own terms as well as terms of other concepts. By way of example, they mention the metaphor *time is money*; *money* is a limited resource, and limited resources are valuable. Therefore, *time* is valuable.

Generally speaking, they argue that most of the metaphorical concepts are abstract (e.g. *time*, *emotions*, *ideas*), and that they are usually described metaphorically by concrete objects (e.g. *food*, *physical objects*). Ortony et al. [18] add that when using a metaphor the writer's goal is to convey only the metaphorical concept.

Metaphors are often used for expressing emotions, as a tool for visualizing concepts. A *broken heart* describes a sad feeling caused by someone or something; it is not meant literally. It creates an image of a heart that is broken into pieces for conveying an extreme feeling of sadness. In [14], it was shown that metaphors carry significantly more emotions than do literal expressions. This is one of the reasons for metaphor being a useful device in creative expression. For example, it allows a writer to describe a concept that is difficult to explain directly through a creative emotional imagery. In [8], *image metaphors* are defined as metaphors that map conventional mental images onto other conventional images with similar characteristics, as for example, describing a politician as a "bulldozer". This opens up many possibilities for creativity in writing.

A specific metaphor sometimes has an ambiguous interpretation. For example, when we say *memory is a river*, both *fluid* and *long* might be considered acceptable interpretations [21]. It has been shown in experiments that sentential context, too, may affect the meaning of the metaphor [18]. The emotional characteristic of metaphor increases the level of ambiguity, as people might interpret emotions in multiple ways.

The rhetorician, I. A. Richards [20], decomposes a metaphor into two main components: the *tenor* and the *vehicle*. The tenor, or *topic*, is that which is being described by potential meanings, referred to as *properties*, of the vehicle. There are several metaphorical syntactic constructions. Similarly to other works on this topic, we focus on Noun-Noun constructions, that is, metaphors of the form *Noun* is [a] *Noun*; *time is money*, for example. The first noun is the topic and the second, the vehicle. This type of metaphor is known as *nominal*. Noun-Noun constructions may extend beyond two nouns. For example, Albert Einstein once said: "All religions, arts, and sciences are branches of the same tree", suggesting that the three topics are related.

The meaning of a metaphor may be related more to the topic, the vehicle, or to both in the same level. For example, when one says that *Joe is a chicken*, the meaning is usually described as being *afraid*, which is more closely related to the vehicle *chicken* than to the topic *Joe*. On the other hand, Bob Dylan said in an interview on 1965, "Chaos is a friend of mine", a metaphor that can be interpreted as something *chaotic*, which is more related to the topic.

We describe a system that is designed for interpreting nominal metaphors, given without context. Similarly to previous works, we exploit a large corpus of text documents for semantically describing words and properties using a mathematical device. We use a word-embedding representation for calculating similarity between a candidate interpretation and the topic and the vehicle, so as to rank candidates based on a semantic score. As a final step, we automatically cluster results and keep only the best interpretations out of each cluster.

To summarize this paper's contributions:

1. We provide a new and improved dataset.

2. We extend previous works in this field using a richer semantic model for interpreting metaphors, and obtain competitive results.
3. We show that clustering and filtering the results to leave only the best in each cluster improves performance.
4. We show that using word associations as interpretation candidates, combined with collocations, improves performance, as do topic interpretations.
5. We suggest some additional metrics for evaluation, such as mean reciprocal rank and mean average precision. And we use word senses (WordNet synonyms) for matching.

The next section cites some related work. Our contributions and the results of experiments are described in the following two sections. Some conclusions are drawn in the final section.

2 Related Work

Different tasks relate to automatic metaphor processing. One is about automatically identifying metaphor in running text, that is, tagging words as being part of a metaphor or not. Many studies handle this. For example, Turney et al. [24] automatically tag words in a given context as either *literal* or *metaphorical*, by training a supervised classifier. They focus on features that measure the level of abstractness of the word's context. They were able to show state-of-the-art performance on a dataset of adjective-noun metaphors (e.g. *sweet child*). For more information on metaphor identification, we refer the reader to [22], a recent review of metaphor processing systems.

Before that, [1] presented a system for identifying literal and nonliteral usages of verbs, focusing on identifying metaphorical meaning, through statistical word-sense disambiguation and clustering algorithms. At a high level, they use a small set of manually sense-annotated sentences. Given a verb with its sentential context, they calculate the similarity between the input sentence and the annotated set, and decide on the sense that mostly occur within the most similar annotated sentences.

Neuman et al. [16] extended previous work [24], covering metaphors formed of only concrete concepts, by identifying *selectional preference* violations. A selection preference is a concept presented in [12], claiming that words mentioned literally in a sentence, usually co-occur with word that belong to a *selected* semantic concept. They treated a violation of this idea as an indication for the nonliteral class.

The computational task in which we are more interested, is *interpretation*, interpreting a given metaphor. This very challenging task has garnered interest over the past few years. *Metaphor Magnet* [25], allows users to enter a metaphor or simile, potentially augmented with sentiment polarity (e.g. $+/-$); for example, *life is a* $+$*game*, including a plus sign for *game*, indicating a positive sentiment. Using sentiment this way allows users to provide some information about the context.

To interpret a metaphor, the system expands the topic and vehicle with some corpus-based *stereotypes*, and then with the stereotype's properties. The properties that saliently occur with both, the topic and the vehicle, are returned as results. For Metaphor Magnet, a stereotype is a word that describes the topic/vehicle. The stereotypes and properties are discovered using Google n-grams, as it contains n-grams of the form "X is a Y" that help one understand how X is typically being described.

There are a few works that treat the text components as vectors of a higher dimension in a semantic space. This opens the possibility of using mathematical tools to calculate the similarity of two components, through measuring the distance between their corresponding vectors. Kintsch [6] uses latent semantic analysis (LSA) [2] for modeling the vector space. They generate term vectors that highly correlate with both, the topic and the vehicle; correlation is measured by cosine similarity over the LSA vectors. Metaphor interpretation is represented by the centroid vector of the most similar terms, and it does not necessarily represent a real word.

Terai and Nakagawa [23] use the same algorithm, over a slightly different semantic model.

They use probabilistic latent semantic indexing (PLSI) [5], for finding potential properties, limiting to adjectives and verbs. We go down the same path, in the sense that we use a semantic model for calculating a score for the candidate properties. Similarly, we focus on adjectives and verbs as the only possible interpretations. Terai and Nakagawa also extended their process with a recurrent neural network trained over the properties and scores for finding the dynamic interaction between the properties.

The most relevant work for us is *Meta4meaning* [26], an interpretation system for nominal metaphors. This work uses an LSA along with two dimensionality reduction techniques, Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF). It only considers abstract words as candidate properties. The properties are ranked according to their association strength with both, topic and vehicle. It uses different aggregation methods for combining the association scores of the topic and the vehicle. The system shows a strong performance advantage over the human-annotated dataset provided by [21] compared with other systems.

Following *Meta4meaning*, we build a word-embedding model instead of LSA. Specifically, we use a 300-dimensional GloVe model [19]. Word embeddings, specifically of the type that we are using, outperform SVD for analogy tasks [11]. Since our task is more similar to analogy than to word similarity, we were led to believe that word embeddings may improve performance of metaphor interpretation.

3 Metaphor Processing

Given a metaphor, we begin by generating a list of interpretation candidates. We do this by finding collocations of the topic and vehicle individually, and consider each one of them as a potential candidate. For each candidate c, we calculate a topic semantic score, which is the cosine similarity between c and the k most significant collocations of the topic (k is a parameter) and aggregate it into a single score by averaging all scores. Similarly, we calculate a vehicle semantic score.

In the next step, we calculate two pointwise mutual information (PMI) values, between c and the topic and vehicle respectively. We add the frequency of c as another score and combine all the five score functions in a log-linear structure, with weights assigned to each. The weights are adjusted automatically, as we describe in the following section.

To remove semantically related interpretations from the list, we cluster the results and keep only the highest ranked candidates in each cluster. The remaining candidates are ranked according to their final score and the best n candidates (n, too, is a parameter) are returned as interpretations.

We now describe each step in greater detail.

3.1 Potential Interpretations

In our work, similar to other relevant works, e.g., [21, 26], a metaphor *interpretation* is composed of a single word that conveys the main concept of the metaphor. For example, among the interpretations of the metaphor *city is a jungle* one can find *crazy* and *crowded*. It is natural to assume that an interpretation should be of a class of *describing* words, that is, words that are used for describing objects. Therefore, similar to other related works [25, 26], we consider all adjectives as potential interpretations.

In addition, we add verbs with an *ing* ending as candidates. In [26], they only consider abstract words as potential interpretations. The level of abstractness of a word was measured by Turney et al. [24] automatically for about 11,000 words. To avoid the limitation in using such a list, we did not go that route; we believe that most of the potential interpretations are adjectives.

3.2 Dependency-Based Collocations

Our interpretation process begins with extracting collocations of the vehicle and the topic individually using a relatively large corpus. Specifically, we use DepCC,[1] a dependency-parsed "web-scale corpus" based on Common Crawl.[2] There are 365 million documents in the corpus, comprising about 252B tokens. Among other preprocessing steps,

[1] https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/depcc.html

[2] http://commoncrawl.org

every sentence was given with word dependencies discovered by MaltParser [17]. We only use a fraction of the corpus containing some 1.7B tokens.

Here, we consider as collocation words that are found to be dependent in either the topic or the vehicle, and assigned with a relevant part-of-speech tag: adjective or verb+*ing*. The main assumption is that many potential modifiers of a given noun will appear somewhere in the corpus as a dependent in the dependency graph.

For example, the dependency-based collocations for *school* are: *high*, *elementary*, *old*, *grad*, *middle*, *med*, *private*, *attending*, *graduating*, *secondary*, *leaving*, and *primary*.

To eliminate noisy results that might transpire given that the corpus was generated from the open web, we preserve only candidates that have an entry in WordNet [3].

3.3 Word Association

In parallel with our objective data-driven collocation extraction process, we experimented with word associations as an alternative, more subjective, process for generating interpretation candidates. Word-association norms are repositories of pairs of words and their association frequency in a given population. The first word is a cue or trigger given to participants, and the second is the reported associated word that first came to a subject's mind. For example, *bank* is paired with *money*, because the cue *bank* often elicits the response *money*. Those pairs form various semantic-relation types; some might not be deemed symmetric. Word association norms have been used in psychological and medical research, as well as a device for measuring creativity.

We use the University of South Florida (USF) free association norms [15][3] for generating alternative candidates. This repository contains 5,019 cue words that were given to 6,000 participants beginning in 1973. We utilize this repository by adding all the associated words of the topic and vehicle individually. In this case, we allow words of all parts of speech to be considered as candidates. For example, the associations for cue *school* are: *work*, *college*, *book*, *bus*, *learn*, *study*, *student*, *homework*, *teacher*, *class*, *education*, *USF* (!), *hard*, *boring*, *child*, *house*, *day*, *elementary*, *friend*, *grade*, *time*, *yard*. We evaluate our system's performance with and without the associations; results are reported below.

[3] http://w3.usf.edu/FreeAssociation

3.4 Calculating Semantic Scores

For each candidate we calculate a couple of semantic scores, one for the topic and one for the vehicle. We use word embeddings to transform every word into a continuous vector that captures the meaning of the word, as evidenced in the underlying corpus. We used pre-trained GloVe [19] vectors; specifically, we use the ones that were trained over a 6B token corpus, comprising 400K vectors, each of 300 dimensions. In what follows, we denote the vector of a word v by w_v.

We believe that the most significant collocations of the topic/vehicle tend to reliably represent the way the topic/vehicle, respectively, can be described in different contexts. Therefore, the semantic score $sem(c,t)$ of a candidate c and the topic t is the average cosine similarity between w_c and the vectors of the k most significant collocations of t. Similarly, $sem(c,v)$ is the semantic score of a candidate c and the vehicle v. We experimented with different values for k. Results are reported in the next section.

3.5 Final Scores

For each candidate c, we calculate $npmi(c,t)$ and $npmi(c,v)$, the normalized pointwise mutual information (PMI) values for the topic and vehicle, respectively. Normalized PMI is similar to PMI, except that it is normalized between -1 and 1. The PMI between a candidate c and a noun n is calculated over the dependency graph; that is, we calculate the chances of seeing c as a dependent of n in a dependency graph. We add $freq(c)$, the frequency of c, as another score, calculated over the entire corpus.

To summarize, given a candidate c, the full list of scores is

$$\langle \textbf{\textit{sem}}(c,t), \textbf{\textit{sem}}(c,v), \textbf{\textit{npmi}}(c,t), npmi(c,v), \textbf{\textit{freq}}(c)\rangle.$$

combined using a log-linear structure, with each score amplified by a weight:

$$\mathit{FinalScore}(c) = \sum_{k=1}^{5} \lambda_k \log \mathit{score}_k.$$

We automatically adjust these weights over a development set of metaphors and interpretations to optimize for recall, as explained below. As a result, each candidate is ranked according to its final score.

3.6 Clustering

Lastly, we cluster the list of candidates as a way to deduplicate it. We run clustering using word vectors for finding groups of words that have a strong semantic association of any kind, keeping only the best candidates in each cluster.

We use density-based spatial clustering of applications with noise (DBSCAN) for clustering. This method groups together vectors that are bundled in the space by forcing a minimum number of neighbors. Vectors that do not have the requisite number of neighbors, or in other words occur in low-density areas, are reported as noise and are not placed under any cluster. For us it means that they were not connected with other vectors, so they might have a unique meaning among the listed candidates. We treat such vectors as if they form singletons.

For example, among the interpretation candidates for the metaphor *anger is fire* we find *red* and *black*. After clustering, *black* is removed. As another example, the following candidates for the metaphor *a desert is an oven* may be grouped together: *eating*, *healthy*, *delicious*, *fried*, *spicy*, *leftover*, *veggie*, *steamed*, *lentil*, *roasted*, *homemade*, *yummy*, *creamy*, *glazed*, *seasoning*, *crunchy*, *baking*. (These likely result from the frequent misspelling of "dessert" in the corpus used.)

There are two parameters that need to be configured for DBSCAN: (1) ε – the radius of the consideration area around every vector; and (2) μ – the minimum number of neighbors required in the consideration area. The distance measure should also be configured. We use the common Euclidean distance, which usually shows good performance in a relatively low-dimension space like ours. Below we describe our experimental results, using different values for both parameters. Table 1 shows a few outputs for three different metaphors.

Table 1. Results for several metaphors

Friendship is a rainbow	God is a fire	Typewriter is a dinosaur
beautiful	burning	prehistoric
wonderful	fighting	fossilised
colorful	holy	extinct
forming	sacred	resembling
pink	good	feathered
great	absolute	robotic
bright	powerful	stuffed
magical	cannon	primitive
deep	dangerous	preserved
double	killing	gigantic
happy	almighty	antique
featuring	calling	lumbering
good	great	basal
vibrant	heavy	ancient
glorious	alive	oversized

4 Experimental Results

4.1 Evaluation Set

We evaluate our system with the dataset published by [21], containing 84 unique topic/vehicle pairs that were associated with interpretations by twenty different study participants. Each participant was asked to assign interpretation for different aspects of the pairs, such as treating a pair as a metaphor (e.g. *knowledge*/*power*, from the phrase *knowledge is power*) or as a simile (e.g. *knowledge*/*power*, from *knowledge is like power*). We focus on the interpretation of metaphors, both lexicalized and non-lexicalized.

As a preprocessing step, we lemmatize the interpretations, so as to allow our method's results and the true interpretations to match more smoothly. Additionally, we allow interpretations to match if they are considered as synonyms

Table 2. Topic/vehicle pairs and associated properties

Topic/Vehicle Pair	Associated Properties
Skating/ Flying	Free; Fast; Relaxing
Store/ Zoo	Crowded
Wisdom/ Ocean	Vast; Huge
Job/ Jail	Boring

in WordNet. In this work we focus on nominal metaphors, and since our collocation as well as word-embedding models were trained to handle unigrams, we had to modify some of the metaphors that have multiword vehicles; such multiwords are modified into a single words by eliminating the space characters, knowing it may cause performance reduction; For example, *sermon is a sleeping pill* is modified to *sermon is a sleepingpill*.

Each metaphor might be associated with more than one interpretation. As do other related works [21, 26], we only consider interpretations that were assigned by at least five participants; we call them *qualified interpretations*. This leaves us with only 76 qualified metaphors (i.e. metaphors with at least one qualified interpretation), with two qualified interpretations per metaphor on average. Table 2 shows a few examples of interpretations as assigned by 20 human annotators for the dataset of [21].

4.2 Evaluation Method

To stay in line with related works [26], we report *Recall* @K, which is the average percentage of human-associated interpretations that are found in the top K results. For example, the following results were generated for the pair *skating/flying* from Table 2: *incredible*, *high*, *free*, *great*, *fast*. Therefore, Recall@3 is 33%, while Recall@5 is 66%. We compare our results with [26], which was evaluated on the same dataset following a similar preprocessing step. Therefore, we report on Recall at their reported K's: 5, 10, 15, 25, and 50.

To measure the false positives reported by the system, we evaluate the results with two additional standard metrics: mean reciprocal rank (MRR) and mean average precision (MAP).

4.3 Tuning System Weights

Our log-linear structure is composed of a set of weighted score functions. We adjust the scores using a tuning process over a development set, composed of about 50% of the metaphors. For each weight, we explore a range of possible scores, while we test all possible score combinations taking the brute force approach. For all scores except *freq*, we consider the range $0.1 \; .. \; 1$; because of scale differences, for *freq* we consider the range $1 \; .. \; 10$.

As mentioned, we use DBSCAN to cluster the list of candidates so as to remove some of the semantically related ones. We take a similar brute force approach for tuning the DBSCAN parameters, ε and μ. We also tune n, the number of top results taken from each cluster. For tuning, we use the same development set, evaluated over MRR, MAP and Recall @K values. Table 3 shows the ranges and best values of all the parameters we tune.

We see that both semantic scores get higher weights than the npmi scores, suggesting that the semantic distance as measured by cosine similarity between the vectors of the candidates and the collocations of the topic/vehicle, is effective. The DBSCAN parameters are less stable across different metric optimizations. One thing we learn is that when optimizing for larger values of $@K$, DBSCAN requires dense areas around clustered vectors, resulting in a lower number of clusters. Additionally, the system does not benefit from high values of the DBSCAN n parameter. It turns out that it is better to consider only one interpretation from each cluster.

4.4 Evaluation Results

We evaluate our system against the 76 "qualified" metaphors in the dataset. For each metaphor, our system generates the top 100 interpretation results, which are then compared with the metaphor's human-associated qualified interpretations. For the clustering parameters and scoring weights, we use the tuned values reported in the previous subsections. Since we tune for different evaluation metrics, here we individually use each set of values for generating

Table 3. System parameters tuned to maximize MRR, MAP and Recall@K. The second column shows the range of values considered

Parameter	Range	MRR	MAP	@5	@10	@15	@25	@50
DBSCAN ε	1 .. 6	4	5	4	5	5	4	4
DBSCAN μ	1 .. 5	4	1	6	1	5	5	5
DBSCAN n	1 .. 12	1	1	1	1	1	1	1
sem(c,t)	0.1 .. 1	0.6	0.6	0.6	0.1	0.1	0.6	0.6
sem(c,v)	0.1 .. 1	1.1	1.1	1.1	0.6	0.6	1.1	1.1
npmi(c,t)	0.1 .. 1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
npmi(c,v)	0.1 .. 1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
freq(c)	1 .. 10	3	3	5	7	7	5	3

Table 4. Each row shows evaluation results when using optimal parameter values for the metric mentioned in the first column

Optimization	MRR	MAP	@5	@10	@15	@25	@50
MRR	**0.312**	**0.170**	0.198	0.254	0.278	0.405	0.562
MAP	**0.312**	**0.170**	0.198	0.254	0.278	0.405	0.562
@5	0.302	0.166	**0.207**	0.258	0.270	0.430	0.548
@10	0.233	0.151	0.180	**0.273**	0.322	0.374	0.521
@15	0.245	0.160	0.151	0.262	**0.331**	0.392	0.513
@25	0.302	0.166	**0.207**	0.258	0.270	**0.430**	0.548
@50	**0.312**	**0.170**	0.198	0.254	0.278	0.405	**0.562**

the top 100 results and calculating MRR, MAP and recall at all the relevant K values. Table 4 summarizes the evaluation results at MRR, MAP and Recall@5, @10, @15, @25, and @50, for each set of parameter values. We observe that when optimizing the system for Recall@50 we at least get close to the best result for all other evaluation metrics. Therefore, in what follows we use the parameter values optimized for @50.

We compare our results with the ones reported by Meta4meaning [26], evaluating over the same set of metaphors and following similar preprocessing steps. Table 5 compares the results reported by both systems. While our system somewhat underperforms for the lower values of Recall $@k$, it is doing slightly better on @25 and @50. These results show that, while our system has a better overall coverage, correct interpretations are concentrated more in the lower part of the ranked list that we produce. With more work, we expect to be able to filter out many of the non-associated interpretations, thereby ranking the correct ones higher in the list.

To measure the effect of clustering on the results, we evaluate our system running with and without clustering. When running with clustering, we use the optimized set of parameters, as reported in Table 3. Table 6 compares our system's results, with and without clustering. We learn that when using clustering, our system was able to eliminate

Table 5. Comparison with Meta4meaning

System	MRR	MAP	@5	@10	@15	@25	@50
Meta4meaning	N/A	N/A	**0.221**	**0.303**	**0.339**	0.397	0.454
Ours	0.312	0.170	0.198	0.254	0.278	**0.405**	**0.562**

Table 6. Evaluation results, with and without clustering

Method	@5	@10	@15	@25	@50
w/o clustering	0.198	0.254	0.278	0.351	0.534
w/ clustering	0.198	0.254	0.278	**0.405**	**0.562**

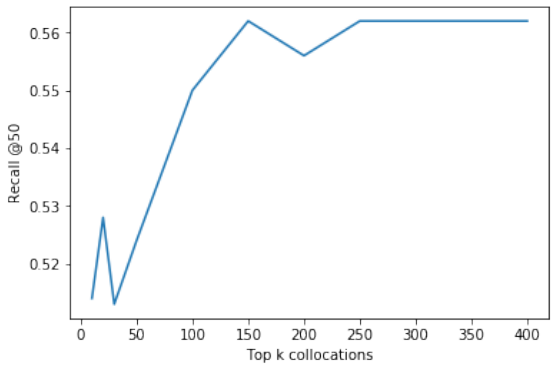


Fig. 1. Evaluation results as Recall@50, measured over different k values for the maximum number of collocations we take from the topic/vehicle for calculating the semantic score

noise in lower parts of the ranked list of candidates, thereby making room for alternative and correct interpretations that ranked lower without clustering.

Recall that our topic/vehicle semantic scores are defined as the cosine similarity between the candidate vector and the top k collocations of the topic/vehicle. We tested our system with different k values; Figure 1 shows evaluation results as Recall@50 when running the system with different k values. Observe that it gets maximized at higher values of k, suggesting that the meaning of the topic/vehicle is usually more complex, and that it takes multiple properties to describe when comparing it vis-à-vis candidate interpretations.

Finally, we check how our system's performance is affected by adding word associations as an additional source for generating interpretation candidates. When we run our system using only dependency-based collocations as candidates, we obtain Recall@50 score of 0.551. This was improved to 0.562 when we add word associations as candidates.

4.5 Improving the Dataset

Overall, the system could not generate even one correct interpretation (among its 50 best results) for 20 out of the 76 evaluated metaphors. Some of those metaphors did not have a correct interpretation anywhere in the list, even beyond the best 50; for example, *music is a medicine*. Taking a closer look at the dataset, we found that some metaphors did not come with any correct interpretation in its interpretation list, even when taking into account all the provided interpretations, not just qualified ones. For example, take the metaphor *education is a stairway*. The suggested interpretations are *higher*, *steps*, *upward*, *long*, *passage*, *ascension*, *climbing* – none of which qualified. Most of these interpretations do not reflect the true meaning of this metaphor (*steps*, *passage* and *climbing* are themselves metaphors; *long* is surely not intended; *higher* and *upward* make little sense); we would rather suggest *enabling* as a more suitable interpretation. For *job is a jail*, the only qualified interpretation is *boring*, while the more accurate interpretation, *confining*, was proposed by fewer then 5 annotators, and therefore did not pass the bar. These are only a few of the examples that encouraged us to perform our own annotation process over the entire dataset. This was done by a native English speaker. We override the original interpretations with the newer ones, resulting in a slightly larger dataset, because with the new annotations some unqualified metaphors now qualify.

In addition to these new annotations, we extended the dataset with 14 new metaphors extracted from [9], among them *words are weapons* and *logic is gravity*. We followed the same annotation process to assign interpretations for the new metaphors. The extended (and improved) dataset contains 98 metaphors with refined interpretations. The full list of modifications

Table 7. Evaluation results when running on different datasets

Dataset	MRR	MAP	@5	@10	@15	@25	@50
Original	0.312	0.170	0.198	0.254	0.278	0.405	0.562
Improved	0.151	0.073	0.051	0.070	0.114	0.171	0.311

can be found in the dataset (published at *to be supplied in the final version*). We intend to extend it even further in the future.

Table 7 compares evaluation results for the original and improved datasets. The degraded results we get for the latter is explained by the fact that, for most metaphors in the dataset, our improvement process removed the majority of suggested interpretations. Fewer human-annotated interpretations means fewer successful matches, making our improved dataset harder to interpret to begin with.

5 Conclusions

We have described a system that interprets nominal metaphors, provided without a context. Given a metaphor, we generate a set of interpretation candidates and rank them according to how strongly they are associated with the topic, as well as with the vehicle. Candidates are generated using two techniques. First, we find collocations of the topic and vehicle, focusing on adjectives as well as gerunds, which were found to be dependent of the topic/vehicle in at least one sentence in a large corpus. We add to that list word associations of both. This addition has proven effective.

Our ranking procedure combines a number of scores assigned for each candidate, which are based on normalized PMI as well as cosine similarity between the representing GloVe vectors of the candidates and the topic/vehicle collocations. The scores are aggregated using a weighted log-linear structure. We tune the weights automatically, optimizing for various evaluation metrics: MRR, MAP and Recall@K for different K values. We found that with small K, the similarity between candidate and topic becomes more important than other score functions. overl In a post-processing step, we cluster the results using DBSCAN and keep only the best candidates out of each cluster. Our system benefits thereby.

Our system was evaluated against a set of metaphors that were assigned with properties by 20 human evaluators. We compare our results with Meta4meaning and obtained competitive results.

Additional work is needed to handle the cases mentioned in the analysis section, especially, cleaning the results from candidates that have an opposite meaning from the ones we are looking for.

Potential future directions include working on additional types of metaphors, as well as additional languages. We plan to improve the current evaluation technique; one option, which we're considering, is to measure the effect of metaphor interpretation on common NLP tasks, such as machine translation. We will also be looking at the analysis of metaphors in context.

Acknowledgment

This work was supported in part by the Blavatnik Family Foundation.

References

1. **Birke, J., Sarkar, A. (2006).** A clustering approach for nearly unsupervised recognition of nonliteral language. 11th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Trento, Italy, pp. 329–336.
2. **Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R. (1990).** Indexing by latent semantic analysis. Journal of the American Society for Information Science, Vol. 41, No. 6, pp. 391–407.
3. **Fellbaum, C.**, editor **(1998).** WordNet: an electronic lexical database. MIT Press.
4. **Gentner, D., Bowdle, B. F., Wolff, P., Boronat, C. (2001).** Metaphor is like analogy. In The Analogical Mind: Perspectives from cognitive science, chapter 6. MIT Press, Cambridge, MA, pp. 199–253.

5. **Hofmann, T. (1999).** Probabilistic latent semantic indexing. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 50–57.
6. **Kintsch, W. (2000).** Metaphor comprehension: A computational theory. Psychonomic Bulletin & Review, Vol. 7, No. 2, pp. 257–266.
7. **Krennmayr, T. (2015).** What corpus linguistics can tell us about metaphor use in newspaper texts. Journalism Studies, Vol. 16, No. 4, pp. 530–546.
8. **Lakoff, G. (1987).** Image metaphors. Metaphor and Symbolic Activity, Vol. 2, No. 3, pp. 219–222.
9. **Lakoff, G., Espenson, J., Schwartz, A. (1991).** The master metaphor list. Technical report, University of California at Berkeley.
10. **Lakoff, G., Johnson, M. (1980).** The metaphorical structure of the human conceptual system. Cognitive Science, Vol. 4, No. 2, pp. 195–208.
11. **Levy, O., Goldberg, Y. (2014).** Neural word embedding as implicit matrix factorization. Advances in Neural Information Processing Systems (NIPS), pp. 2177–2185.
12. **Light, M., Greiff, W. (2002).** Statistical models for the induction and use of selectional preferences. Cognitive Science, Vol. 26, No. 3, pp. 269–281.
13. **Mar, E. (2008).** A Grammar of the English Language. BiblioBazaar.
14. **Mohammad, S. M., Shutova, E., Turney, P. D. (2016).** Metaphor as a medium for emotion: An empirical study. Proceedings of the Joint Conference on Lexical and Computational Semantics (*Sem), Association for Computational Linguistics, Berlin, Germany, pp. 23–33.
15. **Nelson, D. L., McEvoy, C. L., Schreiber, T. A. (2004).** The University of South Florida free association, rhyme, and word fragment norms. Behavior Research Methods, Instruments, & Computers, Vol. 36, No. 3, pp. 402–407.
16. **Neuman, Y., Assaf, D., Cohen, Y., Last, M., Argamon, S., Howard, N., Frieder, O. (2013).** Metaphor identification in large texts corpora. PloS One, Vol. 8, No. 4, pp. e62343.
17. **Nivre, J., Hall, J. (2005).** Maltparser: A language-independent system for data-driven dependency parsing. Proc. of the Fourth Workshop on Treebanks and Linguistic Theories, pp. 13–95.
18. **Ortony, A., Schallert, D. L., Reynolds, R. E., Antos, S. J. (1978).** Interpreting metaphors and idioms: Some effects of context on comprehension. Journal of Verbal Learning and Verbal Behavior, Vol. 17, No. 4, pp. 465–477.
19. **Pennington, J., Socher, R., Manning, C. D. (2014).** GloVe: Global vectors for word representation. Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.
20. **Richards, I. A. (1936; reprinted 1965).** The Philosophy of Rhetoric. Bryn Mawr College. Mary Flexner lectures. Oxford University Press.
21. **Roncero, C., de Almeida, R. G. (2015).** Semantic properties, aptness, familiarity, conventionality, and interpretive diversity scores for 84 metaphors and similes. Behavior Research Methods, Vol. 47, No. 3, pp. 800–812.
22. **Shutova, E. (2015).** Design and evaluation of metaphor processing systems. Computational Linguistics, Vol. 41, No. 4, pp. 579–623.
23. **Terai, A., Nakagawa, M. (2012).** A corpus-based computational model of metaphor understanding consisting of two processes. Cognitive Systems Research, Vol. 19, pp. 30–38.
24. **Turney, P. D., Neuman, Y., Assaf, D., Cohen, Y. (2011).** Literal and metaphorical sense identification through concrete and abstract context. Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, pp. 680–690.
25. **Veale, T., Li, G. (2012).** Specifying viewpoint and information need with affective metaphors: A system demonstration of the metaphor magnet web app/service. Proceedings of the ACL 2012 System Demonstrations, ACL '12, Association for Computational Linguistics, Stroudsburg, PA, pp. 7–12.
26. **Xiao, P., Alnajjar, K., Granroth-Wilding, M., Agres, K., Toivonen, H. (2016).** Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. Proceedings of the Seventh International Conference on Computational Creativity (ICCC), Paris, France, pp. 230–237.

Article received on 21/02/2018; accepted on 18/01/2020.
Corresponding author is Kfir Bar.

ISSN 2007-9737

Constructing Vietnamese WordNet: A Case Study

Khang Nhut Lam[1], Jugal Kalita[2]

[1] Can Tho University,
Vietnam

[2] University of Colorado,
USA

lnkhang@ctu.edu.vn, jkalita@uccs.edu

Abstract. WordNets are commonly used in tasks such as summarizing documents, extracting information, translating and creating other lexical resources. This paper presents experiments in constructing a Vietnamese WordNet (VWN) from a variety of freely published resources in several languages. The VWN has the same structure as the Princeton WordNet. Our algorithm translates several existing WordNets to Vietnamese using a freely available machine translator, removes translation ambiguities by applying ranking methods based on occurrence counts and Google distances on translation candidates. We also establish connections between synsets and extract glosses for synsets. Finally, we carefully look at the VWN created and identify problematic issues in the VWN due to differences in culture and agglutinative morphology of Vietnamese and other languages used.

Keywords. WordNet, Vietnamese, ontology construction.

1 Introduction

A WordNet is a large lexical database where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, the so-called synsets [17]. Each synset represents a distinct concept and consists of a unique synsetID, synset members, and a gloss consisting of a brief definition and one or more examples showing the use of members in the synsets. Synsets are connected to others by means of semantic relations such as hypernymy or generalization, hyponymy or particularization, and meronymy or part-whole relation. Currently, the biggest WordNet is the Princeton WordNet[1] (PWN) constructed manually since 1990. The PWN version 3.0 has 117,659 synsets including 82,415 noun synsets, 13,767 verb synsets, 18,156 adjective synsets and 3,621 adverb synsets.

In this paper, we discuss the feasibility of creating a Vietnamese WordNet (VWN) having the same structure as the PWN by bootstrapping from freely available resources. The remainder of this paper is organized as follows. In Section 2, we discuss related work. Section 3 describes the proposed approaches to build the VWN from existing resources. Results of our experiments and discussion are presented in Section 4. Section 5 concludes the paper.

2 Related Work

The research presented in this paper discusses an efficient method to generate a VWN with the same structure as the PWN. Therefore, this section highlights prior work on constructing WordNets based on the PWN. According to Vossen [25], the two common approaches to build a new WordNet in a target language *T* are the *expand* approach and the *merge* approach. Using the expand method, a new WordNet is created by simply translating the PWN to *T*, whereas using the merge method, an independent WordNet in *T* is firstly built and then aligned to the PWN. There have been a large number of efforts in various languages with the

[1] https://wordnet.princeton.edu/

goal of constructing WordNets. We present a few prominent ones in this section.

2.1 WordNets Created Using the Merge Approach

A French WordNet was constructed from multilingual resources by Sagot and Fiser [20]. The authors performed word alignment and extracted bilingual lexicons from a multilingual corpus; then, every lexical entry was assigned a synsetID obtained from the Balkan WordNet [23]. They also translated the English WordNet to French using dictionaries and thesauri. The French WordNet was finally generated by merging synsets collected from the two methods. Their WordNet contains 32,351 non-empty synsets, and its accuracy based on manual evaluation is 80%.

Gunawan and Saputra [7] generated a prototype version of synsets for an Indonesian WordNet from a monolingual dictionary of Bahasa Indonesia and an Indonesian thesaurus. They first extracted synonym concepts from the thesaurus, combined them with entries in the monolingual dictionary and removed duplicate entries. Finally, a hierarchical clustering technique was applied to merge synsets. Their Bahasa WordNet consists of 60,673 synsets. No evaluation was performed.

A Hindi WordNet[2] has been constructed manually by 'looking up the various list meanings of words in different dictionaries' [4]. The current version has 105,352 unique words and 40,457 synsets. The Hindi WordNet is the first WordNet for Indian languages and has been used to construct WordNets for other Indian languages (e.g., Marathi, Sanskirt and Gujarati) in the IndoWordNet project.

2.2 WordNets Created Using the Expand Approach

Oliver and Climent [18] compared the accuracies of WordNets created by several methods. The first WordNet was created using the Google translation machine to translate a sense-tagged corpus in English to Spanish. The generated WordNet had about 8,000 synsets with accuracy of 80%. In the second method, given a parallel corpus, an analyzer was used to tag senses of words with the English WordNet. Then, constructing a WordNet for Spanish became a word alignment problem. The accuracy of the second approach was lower than that of the first approach, and it depended on the size of the corpus. A bigger corpus increased the accuracy of the created WordNet. They also concluded that sense tagging introduced more errors than statistical machine translation.

Kaji and Watanable [9] constructed a Japanese WordNet by translating the PWN synsets to Japanese, by using a correlation matrix to deal with translation ambiguity. Later, Bond et al. [3] and Isahara et al. [8] constructed another Japanese WordNet by extracting synsets from the PWN and translating them to Japanese using bilingual dictionaries. They enriched the Japanese WordNet using the most common words obtained from different resources. This Japanese WordNet contained 57,238 synsets with 93,834 words.

Sathapornrungkij and Pluempitiwiriyawej [21] proposed a semi-automatic method to construct a Thai WordNet from machine readable dictionaries. They designed a WordNet Builder system which extracted lexical, semantic, and translation relations from the English WordNet and a dictionary. The extracted data was then evaluated according to 13 criteria (e.g., monosemic one-to-one, polysemic one-to-one and polysemic many-to-one). The created Thai WordNet contained 19,582 synsets with a coverage of 80% at 76% accuracy. Later, Akaraputthiporn et al. [1] and Leenoi et al. [14, 15] constructed Thai WordNets from several bilingual dictionaries using a bi-directional translation method. They noted that using different input dictionaries created by different methods such as corpora-based methods or author's expertise produced WordNets with different accuracies. In addition, cultural issues such as categorization, gender, and collective perception needed to be taken into account to maintain the structure of Thai data.

Saveski and Trajkovski [22] constructed a Macedonian WordNet using the expand approach. To remove irrelevant translations, the English synset gloss was translated into Macedonian, and then the Google similarity metric [5] was

[2] http://www.cfilt.iitb.ac.in/wordnet/webhwn/index.php

applied to compute the similarity scores showing the semantic relatedness between the translated gloss and the candidate words. The selected words were words with Google similarity distance with the translated gloss greater than a threshold. The Macedonian WordNet they created had 33,276 synsets.

Lam et al. [13] proposed several methods to create WordNets in many languages with limited resources. The authors generated WordNet synsets for a target language *T* by translating PWN synsets to *T* using the Microsoft Translator. The approach using direct translation (DR), the approach using intermediate WordNets (IW) and the approach using intermediate WordNets and a dictionary (IWND) were introduced to remove translation ambiguities. In the DR approach, synsets in the *T* WordNet were built by simply translating PWN synsets to *T*. The IW approach handled translation ambiguities by using different WordNets with the same structure as the PWN. For each synsetID in PWN, they extracted all synsets of intermediate WordNets and translated to *T*. The objects of their study included resource poor and endangered languages, which do not have many existing lexical resources. Hence, the IWND approach translated synsets having the same synsetID to English, and then translated them to *T*. The correct members of synsets were selected based on the occurrence counts of translation candidates. The authors claimed that the IW approach with 4 intermediate WordNets helped construct better WordNet synsets. They did not establish connections between synsets created.

WordNets created using the expand approach have the same structure as the PWN; however, their quality considering complex agglutinative morphology, presence of culture specific meanings and usages of words is not good compared to those of WordNets built using the merge approach. Generally, the expand approach is more widely used than the other.

3 Proposed Approaches

Generating a new WordNet for a language using the merge approach needs linguistic experts in the language. In addition, the VWN we want to create will have the same structure as the PWN. Therefore, the expand approach is the best choice to construct a VWN. Our work is based on the study of Lam et al. [13], and is divided into 3 parts: creating synsets, establishing connections among synsets and extracting glosses of synsets.

Table 1. Information about WordNets used

WordNet	Synsets	% coverage
FinnWordNet (FWN) [16]	116,763	100%
Japanese WordNet (JWN) [8]	57,184	95%
PWN	117,659	100%
Thai WordNet (TWN) [24]	73,350	81%
WOLF WordNet (WWN) [20]	59,091	92%

3.1 Creating Synsets

To create synsets for the VWN, we use the intermediate WordNets (IW) approach. Lam et al. [13] experimented using the IW approach with different numbers of intermediate WordNets, but they did not know how many intermediate WordNets are good enough to create a new WordNet of high quality. In addition to the WordNets used in their studies, we experiment with one more WordNet, the Thai WordNet. Table 1 presents information about WordNets used. All WordNets used are linked to the PWN version 3.0 and are obtained from the Open Multilingual WordNet [2].

First, we query synsetIDs of all synsets in the PWN. For each synsetID, we extract all members belonging to that particular synset in the PWN and other intermediate WordNets. Then, we translate all synset members in different languages to Vietnamese using a machine translator. As a result of this step, for every synsetID we have a list of translation candidates in Vietnamese. One drawback of the IW approach is that the coverage percentage of synsets created using the IW approach is lower than using the DR and IWND approaches.

To increase the coverage percentage of synsets in the VWN, we improve the method to select translation candidates. The ranking method based on occurrence count is still applied to calculate the

ranking value of translation candidates. The rank of a candidate w is calculated as below:

$$rank_w = \frac{occur_w}{numCandidates} \times \frac{numDstWordNets}{numWordNets}. \quad (1)$$

where:

— $numCandidates$ is the total number of translation candidates of members belonging to a synsetID.

— $occur_w$ is the occurrence count of the word w in the $numCandidates$.

— $numWordNets$ is the number of intermediate WordNets used.

— $numDstWordNets$ is the number of distinct intermediate WordNets that have members translated to the candidate w.

The rank value of each translation candidate is in the range from 0.000 to 1.000. The greater the rank value of the candidate, the higher the possibility that it will become a synset member. Lam et al. [13] select translation candidates based on 3 scenarios: (i) All candidates with the rank values of 1.000 are accepted as correct translations. (ii) If there is no candidate with rank values of 1.000, the candidates with the highest rank value are selected as correct translations. (iii) For each synsetID, if all candidates have the same rank value, they skip all these candidates.

Their approaches to select candidates for each synsetID significantly reduce translation ambiguities; however, an issue is that they discard many correct translations. For instance, members of the synsetID 110399491, with a gloss 'a father or mother; one who begets or one who gives birth to or nurtures and raises a child; a relative who plays the role of guardian', obtained from PWN and JWN are {parent} and {ペアレント}.

Translations of these members are {cha mẹ} and {phụ huynh}, respectively. The criteria for selecting candidates by Lam et al. discard these two candidates which are both correct translations. So, we change the selection method: if all translation candidates of a synset have the same rank value, we compute the Google distance between each translation candidate pair to find the semantic relation among candidates using the NGD formula [6]:

$$NGD(w_1, w_2) = \frac{\frac{max\{logf(w_1), logf(w_2)\} - logf(w_1, w_2)}{logM - min\{logf(w_1), logf(w_2)\}}}{0.7}. \quad (2)$$

where:

— M is the total number of pages indexed by Google[3], nearly 50,500,000,000 at the time we experiment.

— $f(w_1)$ and $f(w_2)$ are the numbers of pages containing w_1 and w_2, respectively.

— $f(w_1, w_2)$ denotes the number of pages containing both w_1 and w_2.

A pair of candidates is accepted as correct translations if the Google distance is smaller than a threshold α, which is 0.450 and is set by experiment. For example, the numbers of pages containing the words (cha mẹ), (phụ huynh) and (cha mẹ, phụ huynh) are respectively 655,000, 515,000 and 20,700. Applying the NGD formula, the NGD value of the pair (cha mẹ, phụ huynh) is 0.420. Therefore, we accept 'cha mẹ' and 'phụ huynh' as correct translations of synset members of synsetID 110399491 in the VWN.

3.2 Establishing Connections Among Synsets

Synsets in PWN are linked to others by semantic relations, which are of 28 types in the PWN version 3.0. There are 285,348 relations among synsets. Lam et al. [13] did not establish connections among the synsets created. We establish connection among synsets in the VWN based on relations among synsets in the PWN using Algorithm 1. First, each Vietnamese synset created $synset_{Vi}$ is mapped to a corresponding $synset_{Pj}$ in the PWN through a synsetID (lines 1-2). Then, for every $synset_{Pj}$ in the PWN, we extract all connections $semRelation_r$ between it and other synsets $synset_{Pk}$ (lines 3-4). Next, we check for the existence of $synset_{Vu}$, which corresponds to $synset_{Pk}$, in the VWN (lines 5-6). If there exists $synset_{Vu}$ in the VWN, we accept and establish the $semRelation_r$ between $synset_{Vi}$ and $synset_{Vu}$ in the VWN (lines 7-8).

[3] http://www.worldwidewebsize.com/

Algorithm 1 Establish connection among synsets in the VWN

Input: synsets in the VWN, synsets in the PWN and their sematic relations

Output: semantic relations among synsets in the VWN

for all $synset_{Vi}$ in the VWN created **do**
  $synset_{Pj} \leftarrow$ map ($synset_{Vi}$,PWN)
  for all $synset_{Pj}$ in the PWN **do**
    Extract all $semRelation_r$ ($synset_{Pj}$, $synset_{Pk}$)
    for all $semRelation_r$($synset_{Pj}$, $synset_{Pk}$) **do**
      $synset_{Vu} \leftarrow$ map ($synset_{Pk}$,VWN)
      if exist $synset_{Vu}$ **then**
        add $semRelation_r$($synset_{Vi}$, $synset_{Vu}$)
      end if
    end for
  end for
end for

Table 2 shows an example of establishing connections between synsetID 110399491 in the VWN with 2 synset members {cha mẹ, phụ huynh}. We note that we do not translate semantic relations to Vietnamese. Currently, the VWN constructed is managed based on the WNSQL project[4].

3.3 Extracting Glosses of Synsets From the Viet WNMS

The project called Viet WNMS[5] has constructed a Vietnamese WordNet for nouns, verbs and adjectives. This Viet WNMS project has been developed using the WNMS tool of the Asian WordNet project (AWN) [19] which provides a platform for building and sharing WordNets in Asian languages based on the PWN. The target of the Viet WNMS project is to build a Vietnamese WordNet consisting of 30,000 synsets and 50,000 words, including the 30,000 most common words in Vietnamese. The Viet WNMS project is divided into 2 parts[6]:

[4]http://wnsql.sourceforge.net/

[5]http://viet.wordnet.vn/wnms/

[6]http://wordnet.vn/vi/chi-tiet/tong-quan-ve-xay-dung-mang-tu-tieng-viet-18-1.html

Table 2. Example of synsets having connections to the synsetID 110399491 in the VWN

Synset ID	Synset member		Gloss	Semantic relation
	PWN	VWN		
107970406	family, family unit	gia đình, hộ gia đình	primary social group; parents and children	member meronym
109772448	adopter, adoptive parent	cha mẹ nuôi	a person who adopts a child of other parents as his or her own child	hyponym
110332385	female parent, mother	mẹ	a woman who has given birth to a child (also used as a term of address to your mother	hyponym
110126708	genitor	cha mẹ ruột	a natural father or mother	hypernym
110654932	stepparent	cha dượng	the spouse of your parent by a subsequent marriage	hyponym
109918248	kid, child	đứa trẻ	a human offspring (son or daughter) of any age	antonym

— Translating the core of the PWN to Vietnamese. According to authors, the core of the PWN are words with high occurrence counts obtained from the BNC corpus[7].

— Manually adding concepts that exist only in Vietnamese. Currently, the Viet WNMS has 40,788 synsets and 67,344 words.

The approach to create the VWN, discussed in this paper based on the IW approach in [13], takes advantages of lexicons in several WordNets having the same structure as the PWN. As a result, our VWN has a better synset coverage percentage and includes common words not only in English but also in several other languages such as French, Finnish, Japanese and Thai. Moreover, our VWN has 4 POSes, including adverbs, whereas the Viet WNMS has 3 POSes. To the best of our knowledge, there is no published paper on this Viet WNMS project. We do not know anything about the structure of this WordNet. However, by manually checking several synsetIDs, we understand that these synsetIDs or synsetOffsets in the Viet WNMS are not the same as in the PWN. Hence,

[7]http://www.natcorp.ox.ac.uk/

Algorithm 2 Extract glosses to synsets in the VWN

Input: the VWN and the Viet WNMS
Output: glosses of synsets in the VWN

1: **for all** words w in the VWN **do**
2: Extract all $synsets_{Ei}$ having w as a synset member from the Viet WNMS
3: $glossViet_i \leftarrow$ getGloss($synsets_{Ei}$)
4: Extract all $synsets_{Vj}$ having w as a synset member from the VWN
5: $glossTrans_j \leftarrow$ getGloss($synsets_{Vj}$)
6: Compute $CosineSim$ of each pair $glossViet_i$ and $glossTrans_j$
7: **if** ($CosineSim$ > β) AND ($CosineSim$ is the greatest) **then**
8: Accept $glossViet_i$ as a gloss of $synset_{Vj}$ in the VWN
9: **end if**
10: **end for**

the Viet WNMS is likely to have a different structure compared to the PWN and our VWN.

We notice that synsets in the Viet WNMS have glosses in Vietnamese, which we believe are constructed manually by experts. Therefore, we extract these glosses and add them to synsets in our VWN using Algorithm 2. We could not use synsetIDs or synsetOffsets to retrieve data from the Viet WNMS. Hence, for each word w in the VWN we created (line 1):

(i) We query all synsets, including their glosses (each of which is called $glossViet$), having w as a synset member in the Viet WNMS (lines 2-3).

(ii) We trace back to all synsets having w as a synset member and translate the corresponding glosses to Vietnamese using a machine translator, the so-called $glossTrans$ (lines 4-5).

Then, we compute a cosine similarity score between each pair of $glossTrans$ and $glossViet$ (line 6). If this score is greater than a threshold β, we accept the $glossViet$ as a correct gloss of that corresponding synset and add them to our VWN. For each $glossTrans$, if there are several $glossViet$s with cosine similarity scores greater than the threshold, we keep the one with the greatest cosine similarity score (lines 7-8).

4 Experiments and Discussion

4.1 Experiments

The synsets and the semantic relations among them in the VWN are evaluated by 8 volunteers who use Vietnamese as mother tongue. We use the same set of 300 synsetIDs, randomly chosen from the synsets we create, and connections among them. Each volunteer is requested to evaluate using a 5-point scale: 5: excellent, 4: good, 3: average, 2: fair and 1: bad.

The VWN is built by translating the PWN and several intermediate WordNets to Vietnamese. The quality of translations and quantity of synsets are highly dependent on machine translators used. Lam et al. [13] used the Microsoft Translator API for translation. When we performed experiments in 2017 for this paper, the Microsoft Translator API was not available for free, and therefore we use the Yandex Translate API[8].

We experimented by constructing VWNs using both our approaches, denoted by IW-NGD, and the IW approach [13] with 4 intermediate WordNets (PWN, FWN, WWN and JWN) and 5 intermediate WordNets (PWN, FWN, WWN, JWN and TWN) using the Yandex Translate API. Table 3 presents the number of synsets, their coverage percentages and average scores of the VWNs built. The VWNs generated using 5 intermediate WordNets have greater numbers of synsets and average scores.

Moreover, the IW-NGD approach creates VWNs of better quality in terms of the numbers of synsets and coverage percentages than the IW approach. The IW-NGD approach with 5 intermediate WordNets creates the best VWN in our experiment. So, we establish links among synsets in the best VWN created. There exist 80,413 semantic relations among 78,285 synsets created in the VWN. The average evaluation score of relations is 3.60.

The Viet WNMS has been published on a website but has limited web service capability. In addition, words in our VWN are not the same as words in the Viet WNMS. In particular, our VWN has many words which do not exist in the Viet WNMS; and contrarily, the Viet WNMS consists

[8] https://tech.yandex.com/translate/

Table 3. VWNs created using different approaches

Approach	Number of intermediate WordNets	Synsets	Average score	% coverage
IW	4	55,048	3.21	46.79%
IW	5	61,808	3.61	52.53%
IW-NGD	4	61,348	3.23	52.14%
IW-NGD	5	78,285	3.73	66.54%

of many words that do not exist in our VWN. Currently, we have queried 2,094 words from the Viet WNMS, and then extracted synsets' glosses for these words.

We carefully evaluate the glosses extracted and find that a value of 0.30 or higher for threshold β finds very good mapped glosses, with an average evaluation score of 4.60. Hence, such synset glosses (the ones extracted from the Viet WNMS) are accepted as the correct glosses and are aligned to the corresponding synsets in our VWN. We have extracted 4,555 glosses for synsets in our VWN. We believe that cooperation between the two Vietnamese WordNets is likely to produce a more extensive WordNet.

Table 4 presents some glosses extracted and aligned to the corresponding synsets in our VWN. In this table, *Member* means the synset member of the *SynsetID* in our VWN, *Gloss in the PWN*: the gloss of the *SynsetID* extracted from the PWN, *GlossTrans*: the translation of the *Gloss in the PWN* generated by a machine translator, *CosineSim*: the cosine similarity score between the *GlossTrans* and the *Gloss extracted* from the Viet WNMS.

4.2 Discussion

Lam et al. [13] and we create VWNs using the IW approach and the same 4 intermediate WordNets. The only different resource used in the prior published experiments and experiments reported in this paper is the machine translator. The previously reported VWN had 72,010 synsets (61.20% coverage percentage) with an average score of 4.26, which is higher than the VWN reported in this paper. The VWN created by Lam et al. [13] was evaluated by native Vietnamese speakers in the US whereas the VWN created in this paper has been evaluated by native Vietnamese speakers in Vietnam. We claim that the translation quality significantly affects the VWN created. Then, an initial important step to build a good WordNet is to use a very good machine translator or dictionaries for translation.

The VWN we created for this paper is managed using WNSQL with 18 tables. The main tables in our project are: linktypes, lexlinks, semlinks, senses, synsets and words. In addition, as mentioned earlier, the PWN has 28 types of semantic relations. We have established only 15 relation types among the synsets we created. One reason for limited connectivity is that many synsets do not exist in the VWN.

Constructing a VWN using the expand approach may lead to problematic issues regarding language gap as discussed below.

— The PWN has concepts which cannot be translated to Vietnamese. For instance, synsetID 107573347 with a gloss 'a canned meat made largely from pork' has one member {Spam} which does not translate well to Vietnamese, although it could possibly be translated to 'một dạng thịt heo đóng hộp' or 'đồ hộp Mỹ[9]'.

— Many concepts in Vietnamese do not exist in English. For example, synsetID 107804323 with a gloss 'grains used as food either unpolished or more often polished' has one member {rice}, which should be translated to 'gạo' in Vietnamese. To the best of our knowledge, in English, 'rice' can be also used for 'cooked rice' or 'boiled rice' which are both translated to 'cơm'. The PWN does not contain synsets pertaining to 'cooked rice' or 'boiled rice'. In Vietnamese, 'gạo' is different from 'cơm'. A similar issue is identified by Sathapornrungkij and Pluempitiwiriyawej [24] when building a Thai WordNet.

— Parts-of-speech (POS) of words in English and their translations in Vietnamese may not be similar. For instance, the word 'sad' in the PWN has only one POS of adjective. This

[9] https://vi.wiktionary.org/wiki/spam#Ti%E1%BA%BFng_Anh

Table 4. Examples of glosses extracted

SynsetId	Member	Gloss extracted	GlossTrans	Gloss in the PWN	Cosine Sim
100887081	sư phạm	nghề của một giáo viên	nghề của một giáo viên	the profession of a teacher	1.00
104161981	ghế	đồ đặc, được thiết kế để ngồi	đồ nội thất, được thiết kế để ngồi	furniture that is designed for sitting on	0.76
300230843	điều chỉnh	sửa đổi để chức năng tốt hơn	sửa đổi cho tốt hơn	modified for the better	0.68
113548105	lọc	loại bỏ các tạp chất	quá trình loại bỏ các tạp chất (như dầu hoặc kim loại hoặc đường)	the process of removing impurities (as from oil or metals or sugar etc.)	0.62
300128572	chưa từng có	không có ví dụ, tiền lệ hoặc sự tương tự trước đây	không có tiền lệ	having no precedent; novel	0.58
301711614	đau đớn	vô cùng đau khổ	thể hiện đau đớn hoặc đau đớn	expressing pain or agony	0.30

word is translated to 'buồn' in Vietnamese. In addition to the POS of adjective, the word 'buồn' has a POS of verb, meaning 'having strong need to do something'[10] and the PWN does not have this concept. Some examples showing the uses of the word 'buồn' are 'buồn ngủ' (sleepy or need to sleep) and 'buồn cười' (to feel like a laugh coming because of something funny (to need to laugh at something)).

5 Conclusion

The purpose of our work presented in this paper has been to study the feasibility of constructing a Vietnamese WordNet with as many synsets as possible by bootstrapping from free lexical resources. We have created synsets and established connections among them.

We intend to improve translation by changing the Yandex Translate API to another better freely available machine translator (if we can find one), and freely available dictionaries [11, 12].

We are contemplating several potential approaches to translate glosses of synsets in the PWN to Vietnamese or to extract glosses of synsets from a Vietnamese corpus. To improve translation quality between English and Vietnamese of glosses, we will use the approach proposed in [10].

In addition, finding a good method to mine or combine information from the Viet WNMS as we have done will definitely improve the quality of our VWN.

[10] https://en.wiktionary.org/wiki/bu%E1%BB%93n

References

1. **Akaraputthiporn, P., Kosawat, K., Aroonmanakun, W. (2009).** A bi-directional translation approach for building Thai WordNet. Asian Language Processing, 2009. IALP'09. International Conference on, IEEE, pp. 97–101.
2. **Bond, F., Foster, R. (2013).** Linking and extending an open multilingual WordNet. Proceedings of the 51st Annual Meeting, volume 1, pp. 1352–1362.
3. **Bond, F., Isahara, H., Kanzaki, K., Uchimoto, K. (2008).** Boot-strapping a WordNet using multiple existing WordNets. Proceedings of the 6th International conference on Language Resources and Evaluation, pp. 1–6.
4. **Chakrabarti, D., Sarma, V., Bhattacharyya, P. (2007).** Complex predicates in Indian language

WordNets. Lexical Resources and Evaluation Journal, Vol. 40, No. 3–4.

5. **Cilibrasi, R. L., Vitanyi, P. M. (2007).** The Google similarity distance. IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3.
6. **Evangelista, A., Kjos-Hanssen, B. (2009).** Google distance between words. Frontiers in Undergraduate Research.
7. **Gunawan, Saputra, A. (2010).** Building synsets for Indonesian WordNet with monolingual lexical resources. Asian Language Processing IALP 2010 International Conference on, IEEE, pp. 297–300.
8. **Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., Kanzaki, K. (2008).** Development of the Japanese WordNet. Proceedings of the 6th International Conference on Language Resources and Evaluation, pp. 2420–2423.
9. **Kaji, H., Watanabe, M. (2006).** Automatic construction of Japanese WordNet. Proceedings of the 5th International Conference on Language Resources and Evaluation.
10. **Lam, K. N., Al Tarouti, F., Kalita, J. (2015).** Phrase translation using a bilingual dictionary and n-gram data: A case study from Vietnamese to English. Proceedings of the 11th Workshop on Multiword Expressions, pp. 65–69.
11. **Lam, K. N., Al Tarouti, F., Kalita, J. K. (2015).** Automatically creating a large number of new bilingual dictionaries. Proceedings of the AAAI Conference on Artificial Intelligence, pp. 2174–2180.
12. **Lam, K. N., Kalita, J. (2013).** Creating reverse bilingual dictionaries. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 524–528.
13. **Lam, K. N., Tarouti, F. A., Kalita, J. (2014aaro).** Automatically constructing WordNet synsets. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, volume 2, pp. 106–111.
14. **Leenoi, D., Supnithi, T., Aroonmanakun, W. (2008).** Building a gold standard for Thai WordNet. Proceeding of The International Conference on Asian Language Processing 2008 (IALP2008), COLIPS, pp. 78–82.
15. **Leenoi, D., Supnithi, T., Aroonmanakun, W. (2009).** Building Thai WordNet with a bi-directional translation method. Asian Language Processing. IALP'09. International Conference on, IEEE, pp. 48–52.
16. **Linden, K., Carlson, L. (2010).** Finnwordnet: Finnish WordNet by translation. LexicoNordica - Nordic Journal of Lexicography, Vol. 17, pp. 119–140.
17. **Miller, G. A. (1995).** WordNet: a lexical database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39–41.
18. **Oliver, A., Climent, S. (2012).** Parallel corpora for WordNet construction: machine translation vs. automatic sense tagging. International Conference on Intelligent Text Processing and Computational Linguistics, Springer, pp. 110–121.
19. **Robkop, K., Thoongsup, S., Charoenporn, T., Sornlertlamvanich, V., Isahara, H. (2010).** Wnms: Connecting the distributed WordNnet in the case of Asian WordNet. Proceedings of the 5th Global WordNet Conference, Narosa Publishing.
20. **Sagot, B., Fiser, D. (2008).** Building a free French WordNet from multilingual resources. Proceedings of OntoLex.
21. **Sathapornrungkij, P., Pluempitiwiriyawej, C. (2005).** Construction of Thai WordNet lexical database from machine readable dictionaries. Proceedings of the 10th Machine Translation Summit, Phuket, Thailand, pp. 78–82.
22. **Saveski, M., Trajkovski, I. (2010).** Automatic construction of WordNets by using machine translation and language modeling. Proceedings of the 13th Multiconference Information Society, Ljubljana, Slovenia.
23. **Stamou, S., Oflazer, K., Pala, K., Christoudoulakis, D., Cristea, D., Tufis, D., Koeva, S., Totkov, G., Dutoit, D., Grigoriadou, M. (2002).** Balkanet: A multilingual semantic network for the Balkan languages. Proceedings of the International WordNet Conference, Mysore, India, pp. 21–25.
24. **Thoongsup, S., Robkop, K., Mokarat, C., Sinthurahat, T., Charoenporn, T., Sornlertlamvanich, V., Isahara, H. (2009).** Thai WordNet construction. Proceedings of the 7th workshop on Asian language resources,

Association for Computational Linguistics, pp. 139–144.

25. **Vossen, P. (2005).** Building WordNets. Irion Technologies. Diaporama électronique.

Article received on 15/02/2018; accepted on 16/01/2020. Corresponding author is Khang Nhut Lam.

ISSN 2007-9737

A Feature-Rich Vietnamese Named Entity Recognition Model

Pham Quang Nhat Minh

Alt Vietnam Co.,
Vietnam

pham.minh@alt.ai

Abstract. In this paper, we present a feature-based named entity recognition (NER) model that achieves the start-of-the-art accuracy for Vietnamese language. We combine word, word-shape features, PoS, chunk, Brown-cluster-based features, and word-embedding-based features in the Conditional Random Fields (CRF) model. We also explore the effects of word segmentation, PoS tagging, and chunking results of many popular Vietnamese NLP toolkits on the accuracy of the proposed feature-based NER model. Up to now, our work is the first work that systematically performs an extrinsic evaluation of basic Vietnamese NLP toolkits on the downstream NER task. Experimental results show that while automatically-generated word segmentation is useful, PoS and chunking information generated by Vietnamese NLP tools does not show their benefits for the proposed feature-based NER model.

Keywords. Feature selection, Vietnamese, named entity recognition.

1 Introduction

Named entity recognition (NER) is an important task in information extraction. The task is to identify in a text, spans that are entities and classify them into pre-defined categories. There have been some conferences and shared tasks for evaluating NER systems in English and other languages, such as MUC-6 [20], CoNLL 2002 [18] and CoNLL 2003 [19].

In Vietnamese language, VLSP 2016 [4] is the first evaluation campaign that aims to systematically compare NER systems for Vietnamese language. Similar to CoNLL 2003 shared-task, in VLSP 2016, four named entity types were considered: person(PER), organization (ORG), location (LOC), and miscellaneous entities (MISC). NER systems in VLSP 2016 adopted either conventional feature-based sequence labeling models such as Conditional Random Fields (CRFs), Maximum-Entropy-Markov Models (MEMMs) or recurrent neural network (RNN) with LSTM units. The first rank NER system in VLSP 2016 applied MEMMs with specific features for Vietnamese NER data [7].

In this paper, we formalize NER task as a sequence-labeling problem and propose a feature-rich NER model for Vietnamese NER, which use word, word-shape features, PoS tags, chunking tags, and features based on two types of word representations: Brown word clusters and word embedding. We adopt CRF [6], a popular sequence-labeling method for our NER model. On the first data set of VLSP NER evaluation with provided word segmentation, PoS, and chunking tags, our system obtained the state-of-the-art F_1 score. Our proposed system significantly outperforms previous work on Vietnamese NER, including a more complicated NER model, which combines bidirectional Long Short-Term Memory (Bi-LSTM), Convolutional Neural Network (CNN), and Conditional Random Fields [16].

There are two NER data sets provided in VLSP 2016 campaign. While the first data set contains word segmentation, PoS, chunking, named entity (NE) information, the second dataset contains only NE information. In the first data set, word segmentation is gold-standard word segmentation. Although PoS tags and chunking tags were generated automatically by public tools, they were partly corrected by annotators during the

annotation process[1]. In the overview paper [4], there is no mention about tools which the VLSP 2016 organizer used to determine PoS and chunking tags.

To date, many published work on Vietnamese NER has reported evaluation results on the first data set. They have used default word segmentation, PoS, and chunking tags provided by organizers of VLSP 2016. However, we could not obtain word segmentation, PoS and chunking tags that way for NER in real scenarios. There is no work that explored the effects of automatically generated word segmentation, PoS, and chunking tags on the accuracy of Vietnamese NER models. Our work will fill that gap by comparing the usage of automatically generated word segmentation, PoS, and chunking tags generated by popular off-the-self Vietnamese NLP toolkits in NER task. Experimental results show that while automatically-generated word-segmentation is useful for a feature-based NER model, PoS and chunking information generated by Vietnamese NLP tools did not give their benefits.

The remainder of the paper is organized as follows. Section 2 presents some related work to our research. In Section 3, we describe our NER system. Next, in Section 4, we present the design of experiments in the paper. In Section 5, we present experimental results achieved on the VLSP 2016 NER data set. Finally, in Section 6, we give conclusions and some remarks.

2 Related Work

Basically, we can categorize machine-learning approaches to NER into conventional machine-learning models and deep-learning models. Conventional machine-learning methods often adopted models such as Conditional Random Fields [6], Hidden Markov Models, Support Vector Machines, or Maximum-Entropy Markov Models. Those methods require to design hand-crafted features for NER [3]. In contrast, deep-learning NER models do not require hand-crafted features but the computational cost in training is very high compared with conventional machine-learning models [2].

For Vietnamese, VLSP community has organized the first evaluation campaign for NER in 2016. Vietnamese NER systems that evaluated on the VLSP 2016 data applied either conventional machine-learning or deep-learning methods. The first rank system in the campaign used MEMM and obtained 89.66% F_1 score on the test data [7].

Recently, Pham and Le-Hong, 2017 [17] incorporated word embedding and syntactic features including PoS, chunk, and regular expressions in Bi-LSTM model and acquired *92.05%* F_1 score. They claimed that automatic syntactic features improve F_1 score about *18%*. Pham et al., 2017 [16] combined Bi-LSTM, CNN, CRF and obtained *92.91%* F_1 score. We argue that syntactic features they used are not really automatic syntactic features because PoS and chunking tags provided in the NER dataset were partly corrected by annotators during the annotation process.

In the best of our understanding, all published Vietnamese NER papers that used the VLSP 2016 NER dataset reported result on the data with default word-segmentation, PoS, chunking tags provided by the VLSP 2016 organizers. There is no work that investigate the effects of automatically generated word-segmentation, PoS tags, and chunking tags by published Vietnamese NLP toolkits to the downstream NER task. Our paper is the first work that addresses that issue.

3 Proposed Feature-Based Vietnamese NER Model

We formalize NER task as a sequence labeling problem by using the B-I-O tagging scheme and we apply a popular sequence labeling model, Conditional Random Fields to the problem. In this section, we briefly describe CRF, and then present features that we used in our model.

[1] We obtained that information thanks to an online discussion with a member in VLSP 2016 organizers

3.1 Conditional Random Fields

Conditional Random Fields [6] is a discriminative probabilistic framework, which directly model conditional probabilities of a tag sequence given a word sequence. Formally, in CRF, the conditional probability of a tag sequence $y = (y_1, y_2, \ldots, y_m)$, given a word sequence $x = (x_1, x_2, \ldots, x_m)$ is defined as follows:

$$P(y|x) = \frac{\exp(w \cdot F(y, x))}{\sum_{y' \in Y} \exp(w \cdot F(y', x))}. \quad (1)$$

where w is the parameter vector to be estimated from training data; $F(y, x) \in \mathbb{R}^d$ is a global feature function that is defined on an entire input sequence and an entire tag sequence; Y is the space of all possible tag sequences. The feature function $F(y, x)$ is calculated by summing local feature functions:

$$F_j(y, x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i). \quad (2)$$

The parameters in CRF can be estimated by maximizing log-likelihood objective function. Parameter estimation in CRF can be done by using iterative scaling algorithms or gradient-based methods [6].

3.2 Features

Basically, features in the proposed NER model are categorized into word, word-shape features, PoS and chunking tag features, features based on word representations including word clusters and word embedding. Note that, we extract unigram and bigram features within the context surrounding the current token with the window size of 5. More specifically, for a feature F of the current word, unigram and bigram features are as follows:

— **unigrams**: F[-2], F[-1], F[0], F[1], F[2].

— **bigrams**: F[-2]F[-1], F[-1]F[0], F[0]F[1], F[1]F[2].

3.2.1 Word Features

We extract word-identity unigrams and bigrams within the window of size 5. We use both word surfaces and their lower-case forms. Beside words, we also extract prefixes and suffixes of surfaces of words within the context of the current word. In our model, we use prefixes and suffixes of lengths from 1 to 4 characters.

3.2.2 Word Shapes

In addition to word identities, we use word shapes to improve prediction ability, especially for unknown or rare words and reduce data spareness problem.

Word shape features are summarized in the Table 1. Among word shape features, we extract both unigram and bigram features for “shaped”, “type”, and “fregex”. For other word shapes, only unigrams are extracted. Features from “fregex” to “wei” were proposed in [7].

3.2.3 PoS and Chunking Tags

Similar to word features, we extract unigrams and bigrams of PoS tags and chunking tags of words within the window of size 5.

3.2.4 Brown Cluster-Based Features

Brown clustering algorithm is a hierarchical clustering algorithm for assigning words to clusters [1]. Each cluster contains words which are semantically similar. Output clusters are represented as bit-strings. In natural language processing, word clusters can be used to tackle the problem of data sparseness by providing lower-dimensional representations of words. The usage of brown-cluster-based features have been explored for named entity recognition in the work of Miller [10], and then widely used in discriminative learning NLP models [5, 21].

Brown-cluster-based features in our NER model include whole bit-string representations of words and their prefixes of lengths 4, 6, 8, and 10. Note that, we only extract unigrams for Brown-cluster-based features.

In experiments, we used the Brown clustering implementation of Liang [9] and applied the tool on the raw text data collected through a Vietnamese

Table 1. Word shape features

Feature	Description	Example
shape	orthographic shapes of the token	"*Đồng*" → "*ULLL*"
shaped	shorten version of shape	"*Đồng*" → "*UL*"
type	category of the token such as "AllUpper", "AllDigit", etc	"1234" → "AllDigit"
fregex	features based on token regular expression [7]	
mix	is mixed case letters	"*iPhone*"
acr	is capitalized letter with period	"*H.*", "*Th.*", "*U.S.*'
ed	token starts with alphabet chars and ends with digits	"*A9*", "*B52*"
hyp	contains hyphen	"*New-York*"
da	is date	"*03-11-1984*", "*03/10*"
na	is name	"*Buôn_Mê_Thuột*"
co	is code	"*21B*"
wei	is weight	"*2kg*"
2d	is two-digit number	"*12*"
4d	is four-digit number	"*1234*"
d&a	contains digits and alphabet	"*12B*"
d&-	contains digits and hyphens	"*9-2*"
d&/	contains digits and backslash	"*9/2*"
d&,	contains digits and comma	"*10,000*"
d&.	contains digits and period	"*10.000*"
up	contains an upper-case character followed by a period	"*M.*"
iu	first character is upper-case	"*Việt_Nam*"
au	all character of the token are upper-case	"*IBM*"
al	all characters are lower-case	"*học_sinh*"
ad	all digits	"*1234*"
ao	all characters are neither alphabet characters nor digits	"*;*"
cu	contains at least one upper-case character	"*iPhone*"
cl	contains at least one lower-case character	"*iPhone*"
ca	contains at least one alphabet character	"*s12456*"
cd	contains at least one digit	"*1A*"
cs	contains at least 1 character that is not alphabet or digit	"*10.000*"

news portal. We performed word clustering on the same preprocessed text data which were used to generate word embeddings in [8]. The number of word clusters used in our experiments is 1000.

3.2.5 Word Embeddings

Word-embedding-based features have been used for a CRF-based Vietnamese NER model in [8]. The basic idea is adding unigram features corresponding to dimensions of word representation vectors.

In the paper, we apply the same word-embedding features as in [8]. We generated pre-trained word vectors by applying Glove [15] on the same text data used to run Brown clustering. The dimension of word vectors in 25.

4 Experimental Design

4.1 Dataset

In experiments, we used the NER data set from VLSP 2016 evaluation campaign with default

Table 2. Statistics of named entities in the VLSP corpus

Entity Types	Training Set	Test Set
Location	6,245	1,379
Organization	1,213	274
Person	7,480	1,294
Miscellaneous names	282	49
All	15,222	2,996

train/test split. There are 16,858 sentences in training data and 2,381 sentences in test data. The data set contains nested entities, yet we only consider first level entities in this paper. The statistics of the data set is shown in Table 2.

The data set provided by VLSP 2016 organizers contains word-segmentation, PoS, and chunking tags along with NER tags. While word-segmentation is manually annotated by human, PoS and chunking tags were automatically determined by tools and then partly corrected by annotators during annotation process.

4.2 CRF Tool and Parameters

In experiments, we adopted CRFsuite [14], an implementation of linear-chain (first-order Markov) CRF. That toolkit allows us to easily incorporate both binary and numeric features such as word embedding features. In training, we use Stochastic Gradient Descent algorithm with L2 regularization and the coefficient for L2 regularization is *3.2*.

4.3 Default and Generated PoS, Chunking Tags

In the VLSP 2016 NER data, PoS and chunking tags were not determined in a fully automatic manner. In our understanding, all published Vietnamese NER work that evaluated on VLSP 2016 data use default word-segmentation, PoS and chunking tags. In real scenarios, we could not obtain PoS and chunking tags that way. In this work, we compare the performance of our NER system in two settings: using default PoS and chunking tags and using PoS and chunking tags generated by off-the-self Vietnamese toolkits. We investigate the effect of PoS, and chunking tags to only our NER model. We plan to do same experiments using other Vietnamese NER models in the future work.

Because of the space limitation, we could not investigate all Vietnamese NLP toolkits in the paper. We choose two Vietnamese toolkits to perform chunking: Underthesea[2] and NNVLP [16]. To perform PoS tagging, we use Underthesea, NNVLP, Pyvi[3], Vitk[4], and VnMarMoT [12]. Those tools are all popular Vietnamese NLP toolkits. We keep the original word-segmentation when we run Vietnamese PoS and chunking tools on the training and test portions of the NER data to reduce the error propagation from word-segmentation tools.

4.4 Default and Generated Word Segmentation

Each word in Vietnamese language may consist of one or more syllables with spaces in between. For instance a location name "*Hà Nội*" consists of two syllables "*Hà*" and "*Nội*". The VLSP 2016 dataset is word segmented, in which spaces between syllables in multi-syllable words were replaced by underscores "_". Because there is no mention about how word segmentation was generated in [4] and organizer reused the dataset for PoS tagged of VLSP project[5], we believe that word segmentation in the VLSP 2016 NER dataset was manually annotated. In this paper, we compare our NER model when we train and test on data with default and generated word segmentation. We also perform an extrinsic evaluation for popular word-segmentation tools in the NER task.

In order to re-generate word segmentation on the training and test data, we remove all word segmentation info in the data, and then run Vietnamese word segmentation tools on the obtained data. We keep the syllables tokenized in the data to avoid boundary-conflict problem in evaluation on the test data segmented by tools. Some tool, such as pyvi tokenizes syllables in the original data into smaller units. For instance "Mr." is tokenized to "*Mr*" and ".". Thus, we choose word segmentation tools that allow us to perform word segmentation on the data with syllables tokenized in advanced. We choose two word segmentation tools, UETSegmenter[6]

[2] https://github.com/magizbox/underthesea
[3] https://pypi.python.org/pypi/pyvi
[4] https://github.com/phuonglh/vn.vitk
[5] http://vlsp.hpda.vn:8080/demo/?\&lang=en
[6] https://github.com/phongnt570/UETsegmenter

and RDRsegmenter[7], which are perfectly fit our need. The two tools obtained good word segmentation results. UETSegmenter obtained 98.82% F_1 score [13], and RDRsegmenter obtained 97.90% F_1 score on the benchmark Vietnamese treebank [11].

4.5 Syllable-Based Model and Word-Based Model

In this paper, we further investigate the effect word segmentation to the proposed Vietnamese NER model by a comparing syllable-based CRF model with a word-based CRF model. In the syllable-based model, BIO tags are tagged on syllable units. In order to generate training and test data for the syllable-based model, we convert BIO tags of words in the original data to BIO tags for syllables. For instance, in word-based model the location "*Hà_Nội*" is tagged with "*B-LOC*" tag, and in syllable-based model, the word will be converted into two syllables with tags: "*Hà/B-LOC*"and "*Nội/I-LOC*". We hypothesize that word-segmentation is useful for NER task and using automatically generated word segmentation improves the accuracy of feature-based NER models against the syllable-based model.

Word embeddings and Brown clusters which we learned for word-based model contained segmented words, so many syllables are not included in vocabularies of them. Therefore, in experiments, we learned word embeddings and Brown clusters for syllable-based model on the unsegmented version of raw text corpora which were used to generate Brown clusters for the word-based model.

5 Main Results

Table 3 shows the accuracy of our NER model and previous NER models using the dataset with default word-segmentation, PoS, chunking tags. In experiments, we use micro-averaged F_1 score, the official evaluation metric in CoNLL 2003 [19] as the evaluation measure. We compare our NER model with following Vietnamese NER models.

[7] https://github.com/datquocnguyen/RDRsegmenter

Table 3. Accuracy of our NER system with full features set and default PoS and chunking tags

System	Precision	Recall	F_1
Vitk [7]	89.56	89.75	89.66
vie-ner-lstm [17]	91.09	93.03	92.05
NNVLP [16]	92.76	93.0	92.91
Our System	**93.87**	**93.99**	**93.93**

— Vitk [7] is the system that obtained the first rank in the VLSP 2016 evaluation campaign. In that work, authors combines regular expressions over tokens and a bidirectional inference method in a sequence labelling model.

— vie-ner-lstm [17] incorporates syntactic features including PoS, Chunk and regular-expression-based features into a bidirectional Long Short-Term Memory (Bi-LSTM) model. They claimed that incorporating automatic syntactic features improves F_1 score about *18%*.

— NNVLP [16] applied Bi-LSTM-CNN-CRF with pre-train word embeddings for Vietnamese language. That model also used default word-segmentation, PoS, chunking tags of VLSP NER dataset.

Results in Table 3 indicated that, our feature-based NER model outperforms the previous work with a large margin. We obtain *93.93%* of F_1 score on the test set, which is *1%* higher than NNVLP system.

5.1 The Effect of PoS and Chunking Tags

In Table 4, we show experimental results of our system when we apply automatic Vietnamese PoS tagging and chunking tools to generate PoS and chunking tags. We can see that with automatically-generated PoS and chunking tags, F_1 score of the system dropped significantly, which is *4.63%*. Incorporating automatically generated PoS and chunk by tools NNVLP or Underthesea did not improve the accuracy of the NER model.

Underthesea tool showed the better result than NNVLP when they were used in our NER model.

A plausible explanation for the result is that chunking tags encode information about boundary of entity mentions. Entities often occur within a noun phrase. Therefore, correct chunking tags will help to improve accuracy of a NER model.

We observe original chunking tags in VLSP NER data and chunking tags generated by NNVLP and by Underthesea, and see that there is a big gap between the original ones and generated ones. The following example shows original chunking tags and generated ones of a sentence in the training data.

— **Original chunking tags**: "Một/B-NP chuyến/B-NP hải_trình/B-NP xuyên/B-VP ba/B-NP nước/B-NP Malaysia/B-NP ,/O Singapore/B-NP ,/O Indonesia/B-NP vừa/O được/B-VP phóng_viên/B-NP Tuổi_Trẻ/B-NP thực_hiện/B-VP ,/O".

— **By NNVLP**: "Một/B-NP chuyến/I-NP hải_trình/I-NP xuyên/B-VP ba/B-NP nước/I-NP Malaysia/I-NP ,/O Singapore/B-NP ,/O Indonesia/B-NP vừa/O được/B-VP phóng_viên/B-NP Tuổi_Trẻ/I-NP thực_hiện/B-VP ,/O".

— **By Underthesea**: Một/B-NP chuyến/B-NP hải_trình/B-NP xuyên/B-VP ba/B-NP nước/I-NP Malaysia/I-NP ,/I-NP Singapore/I-NP ,/I-NP Indonesia/I-NP vừa/B-VP được/I-VP phóng_viên/B-NP Tuổi_Trẻ/I-NP thực_hiện/I-NP ,/O.

In original chunking tags, "Malaysia", "Singapore", "Indonesia" make three noun phrases. NNVLP tool tagged "ba nước Malaysia" ("three countries Malaysia") as one noun phrase, and Underthesea tagged "ba nước Malaysia, Singapore, Indonesia" ("three countries Malaysia, Singapore, Indonesia") as one noun phrase. Underthesea incorrectly tagged "phóng_viên Tuổi_Trẻ thực_hiện" ("done by reporter of Tuoi Tre News") as a noun phrase.

The feature-based NER model learns useful patterns from correct chunking tags. Patterns learned from incorrect generated chunking tags even become noises to the machine-learning model. In the next experiment, we remove chunk features in the model, and compare the accuracy of the model when we use PoS tags generated by different tools. Table 5 indicated that with default

Table 4. Accuracy of our NER system with default and generated PoS, chunking tags; and without PoS and chunking tags

Setting	Precision	Recall	F_1
Default PoS and chunking tags	93.87	93.99	93.93
PoS and chunking tags generated by NNVLP [16]	90.21	86.72	88.43
PoS and chunking tags generated by Underthesea	90.28	88.35	89.3
Without PoS, chunking tags	89.91	90.15	**90.03**

Table 5. Proposed NER systems without chunking tag-based features. We compare default PoS with PoS generated by other tools

Setting	Precision	Recall	F_1
Default PoS tags	90.13	90.55	90.34
PoS by NNVLP [16]	90.05	85.65	88.31
PoS by Underthesea	90.27	88.58	89.42
PoS by Pyvi	90.16	88.72	89.43
PoS by Vtik	89.62	86.42	87.99
PoS by VnMarMoT [12]	90.51	89.15	89.83
Without PoS, chunking tags	89.91	90.15	**90.03**

Table 6. Accuracy of NER system with default and generated word segmentation. We did not use features based on PoS, chunking tags here

Setting	Precision	Recall	F_1
Default Word segmentation	89.91	90.15	90.03
Word segmentation generated by UETSegmenter	87.67	84.95	86.29
Word segmentation generated by RDRsegmenter	89.05	84.98	**86.97**

Table 7. Accuracy of NER system with syllable-based and word-based model. We do not use features based on PoS and chunking tags. "ws" stands for word segmentation

Setting	Precision	Recall	F_1
Syllable-based model	88.78	82.94	85.76
Word-based model with gold ws	89.91	90.15	90.03
Word-based model with ws generated by RDRsegmenter	89.05	84.98	**86.97**

Table 8. Impact of word representation-based features. w2v denotes features based on word embeddings. "cluster" denotes cluster-based features

Setting	Precision	Recall	F_1
(1) = all features with default PoS, Chunk	93.87	93.99	93.93
(2) = (1) - cluster - w2v	91.66	92.02	91.84
(4) = word + word shapes + default PoS	88.01	87.95	87.98
(5) = word + word shapes + cluster + w2v	89.91	90.15	90.03
(6) = word + word-shapes	88.17	88.08	88.13
(7) = word + word-shapes + w2v	88.69	88.72	88.70
(8) = word + word-shapes + cluster	88.96	89.99	89.97

PoS tags, our NER model obtained highest F_1 score. However incorporating PoS tags generated by other PoS tagging tools did not help to improve against the model without PoS and chunking tags.

Since in VLSP 2016 dataset, PoS tags were not automatically generated, we can safely say that automatically generated PoS tags did not give benefits to our feature-based NER model.

5.2 The Effect of Word Segmentation

Similarly, Table 6 shows comparison of the model accuracy with default word-segmentation and word segmentation generated by the two state-of-the-art Vietnamese segmentation tools: UETSegmenter and RDRsegmenter. Word segmentation result of RDRsegmenter leads to better NER accuracy compared with UETSegmenter.

In comparison with using default word-segmentation (which is manually annotated word-segmentation), the F_1 of score of our model with automatic word segmentation decreased about 3%. That suggests that there is still room for improvement of Vietnamese word segmentation, especially in downstream NLP tasks.

Table 7 show results of syllable-based models and word-based models. Experimental results confirm our hypothesis that in Vietnamese, word segmentation is useful for a feature-based NER model. Word-based models outperform syllable-based models even with automatically generated word-segmentation.

5.3 The Effect of Word-Representation-Based Features

In order to evaluate the impact of word representation-based features, we conducted experiments with different feature sets. We start with a feature set, then remove features related to word clusters and word vectors. Results in Table 8 indicated the importance of word representation-based features. Incorporating those features improves F_1 score more than 2%. Resuls also showed that Brown cluster-based features contribute more to the system improvement than word-embedding features. An advantage of word-representations is that they can be learned in the unsupervised fashion from raw-text corpora.

6 Conclusion

In the paper, we presented a feature-based named entity recognition model for Vietnamese language, which obtains the state-of-the-art accuracy on the standard VLSP 2016 NER data set. Using default word-segmentation, PoS, chunking tags provided by VLSP 2016 organizers, our system achieved 93.93% F_1 score. We showed that, in our CRF-based NER model, PoS and chunking features are useful if PoS and chunking tags are precise. However automatically generated PoS and chunking tags did not give their benefit to the accuracy improvement. We pointed out that word-segmentation in Vietnamese language is useful to the downstream NER task, and word-segmentation generated by state-of-the-art Vietnamese word segmentation tools is helpful, but that there is still a big gap between the usage of manually annotated word segmentation and that of automatically generated word segmentation in a feature-based NER model.

References

1. **Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., Lai, J. C. (1992).** Class-based n-gram models of natural language. Comput. Linguist., Vol. 18, No. 4, pp. 467–479.

2. **Chiu, J., Nichols, E. (2016).** Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, Vol. 4, pp. 357–370.
3. **Florian, R., Ittycheriah, A., Jing, H., Zhang, T. (2003).** Named entity recognition through classifier combination. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, volume 4, Association for Computational Linguistics, pp. 168–171.
4. **Huyen, N. T. M., Luong, V. X. (2016).** VLSP 2016 shared task: Named entity recognition. Proceedings of Vietnamese Speech and Language Processing (VLSP).
5. **Koo, T., Carreras, X., Collins, M. (2008).** Simple semi-supervised dependency parsing. Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, pp. 595–603.
6. **Lafferty, J., McCallum, A., Pereira, F. (2001).** Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML, pp. 282–289.
7. **Le, H. P. (2016).** Vietnamese named entity recognition using token regular expressions and bidirectional inference. CoRR, Vol. abs/1610.05652.
8. **Le-Hong, P., Pham, Q. N. M., Pham, T. H., Tran, T. A., Nguyen, D. M. (2017).** An empirical study of discriminative sequence labeling models for vietnamese text processing. Proceedings of the 9th International Conference on Knowledge and Systems Engineering, KSE, pp. 88–93.
9. **Liang, P. (2005).** Semi-supervised learning for natural language. Ph.D. thesis, Massachusetts Institute of Technology.
10. **Miller, S., Guinness, J., Zamanian, A. (2004).** Name tagging with word clusters and discriminative training. **Susan Dumais, D. M., Roukos, S.**, editors, HLT-NAACL 2004: Main Proceedings, Association for Computational Linguistics, Boston, Massachusetts, USA, pp. 337–342.
11. **Nguyen, D. Q., Nguyen, D. Q., Vu, T., Dras, M., Johnson, M. (2018).** A Fast and Accurate Vietnamese Word Segmenter. Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018).
12. **Nguyen, D. Q., Vu, T., Nguyen, D. Q., Dras, M., Johnson, M. (2017).** From Word Segmentation to POS Tagging for Vietnamese. Proceedings of the Australasian Language Technology Association Workshop 2017.
13. **Nguyen, T. P., Le, A. C. (2016).** A hybrid approach to Vietnamese word segmentation. Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016 IEEE RIVF International Conference on, IEEE, pp. 114–119.
14. **Okazaki, N. (2007).** CRFsuite: A fast implementation of conditional random fields (CRFs).
15. **Pennington, J., Socher, R., Manning, C. D. (2014).** Glove: Global vectors for word representation. Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.
16. **Pham, H., Khoai, P. X., Nguyen, T. A., Le-Hong, P. (2017).** NNVLP: A neural network-based Vietnamese language processing toolkit. Proceedings of the IJCNLP 2017, System Demonstrations, pp. 37–40.
17. **Pham, T. H., Le, H. P. (2017).** The importance of automatic syntactic features in Vietnamese named entity recognition. CoRR, Vol. abs/1705.10610.
18. **Sang, E. F. T. K. (2002).** Introduction to the conll-2002 shared task: Language-independent named entity recognition. CoRR, Vol. cs.CL/0209010.
19. **Sang, E. F. T. K., Meulder, F. D. (2003).** Introduction to the conll-2003 shared task: Language-independent named entity recognition. CoNLL.
20. **Sundheim, B. (1995).** Overview of results of the muc-6 evaluation. MUC.
21. **Turian, J., Ratinov, L.-A., Bengio, Y. (2010).** Word representations: A simple and general method for semi-supervised learning. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, pp. 384–394.

Article received on 20/02/2018; accepted on 07/01/2020.
Corresponding author is Pham Quang Nhat Minh.

ISSN 2007-9737

Towards an Automatic Mark-up of Rhetorical Structure in Student Essays

Eckhard Bick

Institute of Language and Communication, University of Southern Denmark, Denmark

eckhard.bick@mail.dk

Abstract. This paper presents and discusses a discourse relation annotation scheme for the MUCH corpus of academic writing, based on Rhetorical Structure Theory (RST). The set of proposed relational tags takes into regard both distinctiveness, pedagogical needs and implementability with automatic rules. We show how a pilot grammar with 180 rules can map discourse relations between existing syntactic nodes, exploiting lower-level grammatical/treebank markup and surface clues such as connectives (e.g., conjunctions and prepositions). In an evaluation of a live run on student essays from teacher training courses, the average false positive rate across the most frequent 21 categories was 26.7% for tags and 17.1% for relation links. Performance was best for categories with a high percentage of rules using surface connectives and, for in-sentence relations, their corresponding dependency links.

Keywords. Rhetorical structure theory, discourse annotation, student essays, MUCH corpus, constraint grammar.

1 Introduction

Over the last decade, corpus linguistics has taken an interest in the quality and pedagogical aspects of academic writing. However, most studies and corpora, e.g., the American MICUSP[1] and the British BAWE[2] corpora have focused on native (L1) speakers, single text versions and lexico-grammatical aspects only (Flowerdew 2010). The Malmö-Chalmers (MUCH) Corpus of Academic Writing as a Process (Eriksson et al. 2012, Wärnsby et al. 2016) breaks new ground by targeting Swedish students' (L2) English essays and aligning drafts, teacher/peer comments and final versions. In addition, MUCH intends to widen annotation scope beyond lexico-grammatical errors to rhetorical structure theory (RST, Mann & Thompson 1988), which not only will add linguistic value to the corpus, but also represents an important step towards a consistent semi-automatic evaluation of student essays for tasks such as grading, proofing and data-driven learning.

Finally, in a process-oriented research perspective, mark-up of rhetorical structure allows a more global interpretation of editing changes made to the texts as a result of teacher or peer intervention. One of the more ambitious goals of the MUCH project, on the corpus annotation side, is therefore the introduction of discourse relations such as reason, purpose, concession, elaboration, evaluation, contrast etc. The presence of such wide-scope mark-up will present a challenge to standard corpus interfaces[3], but it should ultimately allow RST-based searches and statistics and provide an overview of how coherently students structure their essays.

The pilot version of the corpus, collected over a 3-year period, contains about 400 essays (500,000 words), but continuous additions and a planned large-scale project, where others are invited to contribute their own texts to the MUCH infrastructure, will eventually lead to a much larger

1 Michigan Corpus of Upper-Level Student Papers [http://micusp.elicorpora.info/]

2 British Academic Written English Corpus [http://www.coventry.ac.uk/research-bank/research-archive/art-design/british-academic-written-english-corpus-bawe/]

3 For visualization of search results, we envision a relational extension to the ELAN linguistic annotator [www.mpi.nl/corpus/html/elan/]

data set. Obviously, the bigger the corpus, the more difficult it becomes to perform annotation by hand, and with ongoing additions to the corpus, any infrastructure based on manual work will eventually run out of funding. As a solution we envision an automatization of the RST mark-up process, with possible post-editing of part or all of the corpus by human annotators during the project period proper.

In principle, the same annotation tool could then also be used independently to assist teachers in their evaluation work, or permit a certain degree of self-evaluation by students. However, automatic annotation of discourse has a notoriously low accuracy with standard machine learning (ML) techniques.

Thus, Forbes-Riley et al. (2016), also working on student essays, report an F-score of 31% even when distinguishing only the 4 level-one categories of the PDTB (plus relation types). As possible issues for their target data the authors cite data noise (spelling and grammatical errors) and the importance of in-domain training data. In order to circumvent these issues, we decided to use a rule-based approach rather than ML, because the former allows transparent domain adaptation with specific rules as well as context-based recognition of grammatical errors (Bick 2015).

The underlying morphosyntactic markup of the MUCH corpus is being carried out using an adapted version of the (rule-based) EngGram parser[4], using the Constraint Grammar (CG) formalism (Karlsson et al. 1995). The EngGram core is a modular system and has been shown to support extensions with higher-level grammars, e.g., for semantic roles and verb frames (Bick 2012).

We therefore decided to maintain methodological and annotational compatibility and extend the EngGram infrastructure to handle RST/discourse relations as well, linking the new annotation to lower level morphosyntactic and dependency mark-up, with named relations holding between clausal arguments. Such use of named relations has recently been introduced to the cg3 compiler (Bick & Didriksen, 2015), and our first experiments in 2014 indicated that the feature is up to the task and indeed can be used to map discourse relations.

However, given the task's semantic and wide-scope nature, automatic annotation at this level is extremely difficult, ambiguity across categories is likely to be high and accuracy bound to be considerably lower than in a low-level task such as part-of-speech tagging.

It is therefore paramount to identify a set of descriptive categories for the task that is large enough to allow meaningful distinctions, yet at the same time small enough to avoid excessive ambiguity which would make it impossible to formulate automatic rules and reduce inter-annotator agreement in a possible human post-editing phase.

2 Frameworks and Annotation Schemes

Common to all discourse analysis approaches is the need for segmentation in order to establish possible arguments for rhetorical/discourse relations. Though segmentation could be based on punctuation and trigger words alone, linguistic segmentation based on syntactic structures provides, if available, a more robust point of departure, because it allows a distinction between clausal and non-clausal on the form side, and verb arguments and free adjuncts in terms of function. This distinction is important, because discourse relations hold between entire predications, rather than between clause constituents. If it can be made to work with sufficient accuracy on student data, the MUCH Constraint Grammar morphosyntactic annotation will provide exactly those distinctions.

A second issue is the treatment of connectives. Though discourse does draw upon a certain number of explicit connectives (therefore, by contrast, according to .., first of all), relations may be implicit and lack surface connectives (about 50%, according to Pitler et al. 2008). A theory that limits itself to relations with a surface connective, though it may be easy to implement (English connectives are fairly predictive, Pitler et al. 2008), will therefore have limited coverage.

[4] The parser can be accessed on-line at http://visl.sdu.dk/visl/en/parsing/automatic/. Our add-on discourse module will be made available at the same site.

To avoid this problem, most theories allow abstract arguments consisting of bracketed token chains without a connective. Discourse arguments may be discontinuous, lists or coordinations, but are usually held together by syntactic coherence. In our Constraint Grammar approach, we exploit this syntactic coherence by tagging relation names onto argument heads. This way, there will always be a surface token to carry the tag, even without explicit connectives.

The third problem is how to establish a reasonable set of relational categories for discourse. In the absence of explicit and unambiguous connectives, too large a category set may lead to inter-annotator disagreement in human annotation, and to low precision in automatic annotation. Conversely, too small a category set may fail to capture important distinctions, restricting the theory's usefulness for pedagogical or linguistic research.

Furthermore, category usefulness is domain-dependent. Thus, spoken discourse exhibits mechanisms (such as repairs) that are absent from written discourse (and which we will therefore ignored for the time being), and scientific papers follow certain topic-organization rules not found in, e.g., news casts.

Two general types of discourse categories can be distinguished: On the one hand, logic-semantic categories such as CAUSE, CONDITION, ALTERNATIVE, on the other hand meta-discourse categories structuring the flow of discourse rather than relating its content: REPAIR, RESTATEMENT, ATTRIBUTION. Though a few categories in the second group are more typical (or even exclusive) of spoken discourse, and the first group is much more important for information extraction and QA, both category classes are relevant for essay evaluation, which is the target domain of the MUCH project. In the following subsections we will discuss three existing mark-up strategies and their choice of categories.

2.1 PENN Discourse Treebank

The PENN Discourse treebank (PDTB Research Group, 2008) adds discourse relations on top of the syntactic annotation, as discourse-level predicates with typically 2 arguments (clauses, vp's, np's, anaphora), just like our own discourse annotation in CG. The scheme distinguishes between explicit and implicit connections, alternative lexicalisations (AltLex) and simple entity-based coherence (EntRel). The first three are associated with discourse senses, comprising four groups of categories:

1 Temporal (asynchronous, precedence, succession),
2 Contingency (cause, condition),
3 Comparison (contrast, concession).
4 Expansion (conjunction, instantiation, restatement, alternative, exception, list),

For the categories in group 2 and 3, a distinction is made between non-pragmatic and pragmatic (e.g., pragmatic cause = justification).

2.2 Ädel's Metadiscourse Categories

Ädel's scheme uses 23 functional metadiscourse categories (Metatext categories, Ädel 2006) and is related to the MICUSP corpus of academic papers and the MICASE corpus of university lectures.

1 Metalinguistic comments (repairing, reformulating, exemplifying a.o.),
2 Discourse organization (topic handling, enumeration, asides, pre-/reviewing),
3 Speech act labels (arguing, exemplifying),
4 References to the audience (managing channel/discipline, message, response).

2.3 RST Treebanks

The Wall Street Journal-based RST Discourse treebank connects elementary discourse units (EDU), mostly clauses, including clausal adverbials (-ing, infinitive or participle clauses) and some phrases, especially PPs, but never clausal subjects or objects (with the exception of arguments of attribution verbs, i.e. cognitive predicates).

The mark-up scheme (Carlson & Marcu 2001) contains 78 relations (53 mononuclear and 25 multinuclear), belonging to 16 classes (attribution, background, cause, comparison, condition, contrast, elaboration, enablement, evaluation, explanation, joint, manner-means, topic-comment, summary, temporal, topic change). In addition, 3

structural relations are used: textual-organization, span and same-unit. For ambiguous cases, a preference order was used to decide on only one relation. Leaner versions of this scheme have been adopted for the Portuguese DiZer annotator (Pardo et al. 2004) and the Spanish (da Cunha et al. 2011) and Basque (Iruskieta et al. 2013) RST treebanks, as well as the multi-source Discourse Relations Reference Corpus (Taboada & Renkema 2008). A related scheme is used by the Potsdam Commentary Corpus (Stede 2004) for German.

2.4 Adopting a Scheme

There is a certain overlap between the RST and PDTB schemes. Both are relational, but with its focus on connectives, PDTB is more surface-oriented and "binary", while RST intends to build a tree structure for so-called EDU's (elementary discourse units). The third scheme (Ädel) is difficult to align to the other two, first because it is non-relational, and second, because it addresses meta-discourse rather than the logic of the discourse proper.

Therefore, even though some of Ädel's categories are equivalent to RST and PDTB categories, they mean something different. Rather than on the comment itself, for instance, focus is on the speech act of saying that this is a comment.

Both types of annotation, discourse and meta-discourse, appear relevant to the text types and intended uses of the MUCH corpus, but while Ädel's metadiscourse categories could be assigned fairly ambiguity-free and "mechanically" with just a large set of paraphrases for the individual category markers, it is linguistically and computationally more challenging to assign potentially ambiguous and underspecified relations between discourse elements.

Also, because of the meta-discourse surface markers, meta-discourse annotation should be more accessible to straight-forward machine learning (ML) techniques. What triggers a discourse relation, on the other hand, is less obvious.

Surface markers are often missing or ambiguous, and it is therefore likely that long distance context and deeper linguistic information will be necessary for the automatic treatment of discourse relations than for the treatment of meta-discourse. Furthermore, a structural annotation, be it binary or tree-based, should profit from structural annotation at lower levels (syntax), and could itself prepare the ground for other high level tasks, e.g., inference and summarization.

We therefore decided to address the more challenging discourse relation mark-up in the MUCH corpus with a Constraint Grammar approach, leaving meta-discourse annotation to a possible later ML stage. Because Constraint Grammar is a token-based approach, we suggest to link the necessary relational tags to the heads of existing syntactic constituents (first of all, clauses). Such a head with all its dependents ("descendents") will then constitute what RST calls elementary discourse units (EDU's), which makes RST a more natural framework than PDTB with its need for implicit (i.e. token-less) connectives[5].

3 Choosing a Category Set

We implemented a pilot discourse grammar in the CG framework, using example sentences from the RST corpus annotation manual for development and formulating relational CG rules for individual RST categories. Based on these experiments, we selected those categories that could be operationalized in terms of text-based linguistic clues (lemma, syntax, semantic roles, verb frames etc.)[6], ending up with a reduced CG set of 33-37[7] RST categories, for each of with we introduced a (mostly 4-letter) abbreviation tag. 11 of these are directly equivalent to adverbial semantic roles, making it possible to directly "translate" the corresponding EngGram tags (e.g., cause, condition, consequence/effect, blue in table 1). Our tag set has a substantial overlap with those cited in (Pardo et al. 2004) and (Da Cunha et al. 2011) who also use a streamlined tag set smaller than the

5 Of course, even with a PDTB category set, connectives could simply be used as names of relations, while still attaching tags to clause heads rather than the connectives themselves, avoiding the problem of missing surface tokens.

6 The presence of an overt surface connector was not a condition, all linguistic hints were considered

7 A few difficult categories are included in the grammar, but filtered back into a hypernym category in actual corpus annotation (* in the table 1).

Table 1. Category tag set

Relational tag	Category name	Relational tag	Category name
BACK*	background	**MEANS**	means
CAUS	cause	**OTHR***	otherwise
CIRC	circumstance	**PREF***	preference
COCL	conclusion	**PSOL**	problem solution
COMP	comparison	**PURP**	purpose
COMT	comment	**QA**	question answering
CONC	concession	**QUOTE**	quote/attribution
COND	condition	**REAS**	reason
CONS	consequence	**RESU**	result
COTR	contrast	**RETQ**	rhetor. question
ELAB	elaboration	**RSTA**	restatement
ENAB*	enablement	**SEQU**	sequence
EVAL	evaluation	**STAR**	statem.-response
EVID	evidence	**SUMA**	summary
EXAM*	example	**TEMP-AFT**	temporal:after
EXPL	explanation	**TEMP-BEF**	temporal:before
ITPR	interpretation	**TEMP-SAM**	temporal:same
LIST	list	**TXTO**	text organisation
MANR	manner		

English original, and differs from the former mainly by including a more fine-grained set of "adverbial" and "illocutionary" RST categories (e.g., temporal, manner and comment, statement-response). Because of this superset-subset correspondence (rather than a many-to-many correspondence), it is possible to automatically convert our CG annotation into the categories used for Spanish and Portuguese.

Another reason for not adopting all categories from the RST scheme was that many are not sufficiently disjunct for our purposes, and difficult to reliably distinguish for both human annotators and CG context rules:

- **Background** is very close to **Circumstance.** Though the latter should contain a temporal element, this needn't be visible, and background information may include time markers, too (tense, adverbs), so it would be easiest to fuse these categories (CIRC).
- **Analogy** should be subsumed under **Comparison** (COMP) because its defining criterion (correspondence in more than one respect) is difficult to operationalize.
- **Antithesis** should be fused with **Contrast** (COTR). The RST manual itself suggests to use nuclearity for the distinction (mononuclear

for Antithesis, multinuclear for Contrast), but for automatic annotation we deem nuclearity too soft a distinction.

- The RST scheme lists some **-[A-Z]** subcategories, for instance negated attribution, **Attribution-N** (e.g., *yesterday's statememt didn't say whether ...*), but negation is a semantic operator not specific to discourse relations, and might better be kept separate. Another case is **Consequence-N** and **Consequence-S,** indicating whether it is the nucleus or the satellite that is the consequence, in analogy to the Cause-Result distinction. Since our own scheme does not distinguish between nucleus and satellite, we will simply use uppercase 'CONS' for the consequence and lower case 'cons' for the underlying situation statement. Similarly, we do not distinguish between **-N** and **-S** forms for RST's categories of **Evaluation**, **Interpretation,** **Problem-Solution-N** and **Summary.**
- A category **Comment-Topic** or **Topic-Comment** is stipulated in the RST scheme, and difficult to distinguish from ordinary (subjective) **Comment** as non-subjective, but examples are close to **Explanation** or **Elaboration** (incl. Definition), so it might be an idea to drop this category.
- A distinction between **Sequence** and **Inverted-Sequence** according to chronological order is not strictly necessary for discourse annotation, and could be left to a TIME-relation parsing stage.
- **Definition** is a separate category in the RST manual, but unless there's actually a verb like "define", definitions read like elaborations, and will be treated as such (ELAB) in our CG scheme. **Example** works a bit like Definition, and could be classified as ELAB, but has so far been kept as an independent category.
- Similarly, the six RST subcategories of Elaboration, **Elaboration-Additional, Elaboration-General-Specific, Elaboration-Object-Attribute, Elaboration-Part-Whole, Elaboration-Process-Step and Elaboration-Set-Member** are just tagged as ELAB. Making these distinctions in an automatic fashion would be challenging, and is left to future research. Elaboration-Process-Step has the added problem of ARG2 being a multi-part list of satellites. In CG, this will either be seen as a coordination (and tagged as a whole), or as multiple parallel arguments.
- The **Hypothetical** seems problematic as a relation and independent category, and is logically subsumed as the parent end of COND (condition) or RESU (result).
- The RST scheme introduces a "symmetric" category for cause/result, **Cause-Result,** which we avoid as superfluous, if true ambiguity/symmetry should occur, double tagging with CAUS and RESU could be used as a fail-safe.
- In the RST scheme, the **Condition** category has a competitor, **Contingency,** for habitual/recurrent conditions or time/place contingencies *(whenever, wherever).* In practice, however, ordinary *where* or *when* can fulfill these functions, too, and the distinction is even more difficult without a connective. We therefore use **Condition** or **Temporal:same** in these cases.
- Finally, the category of **Same-Unit** is not necessary in our scheme, because CG dependency trees do not share the discontinuity problem a constituent grammar would suffer from.

It might be useful to add PDTB categories without a direct match in the RST scheme, in particular **Exception,** which often has clear surface connectives. Furthermore, PDTB categories could be used where a subdivision of RST **Textual-Organization** is desired (e.g., introducing topic, previewing, endophoric marking).

Furthermore, there is the issue of PDTB *pragmatic* versions of certain categories: Pragmatic concession, Pragmatic contrast, Justification (pragmatic reason), Relevance, Implied Assertion. Both RST and PDTB mark topic change, with Topic-Drift / Topic-Shift and Adding-Topic, respectively. However, it seems near impossible to identify surface-oriented or structural clues for these categories in automatic annotation, and bag-of-word comparisons, that would work between texts, are of less use on small chunks such as sentences or paragraphs.

4 Writing a Discourse Grammar

In our CG annotation RST tags appear in upper case for the ARG2 discourse unit, and in lower case for the ARG1 discourse unit, linked by token IDs, e.g., <REF:CONC:+10> and <REF:conc:-10>. For an RST nucleus-satellite relation, ARG2 is the satellite and ARG1 the nucleus. However, our CG annotation does not make the distinction between mono- and multi-nuclear relations. Rather, it will follow the syntactic annotation and call @ADVL constituents for ARG2 satellites[8]. With 2 main clauses, the second will be ARG2, the first ARG1.

The following CG rule, for instance, will tag a concession relation (CONC) between two main verbs (@MV) and their EDU clauses.

- ADDRELATIONS (CONC) (conc) TARGET @MV
- (*-1 ("although" KS) OR ("even=if")
- OR ("though") BARRIER @MV)
- TO (1 (*) LINK *-1 @FS-ADVL
- BARRIER NON-ICL/ADV LINK p @MV) ;

The rule's conditions are basically that the first (TARGET) main verb (@MV) should have a concessive conjunction to the left (*-1) without other verbs in between (BARRIER), that its clause function should be that of adverbial subclause (@FS-ADVL), and that the other (TO) main verb should be the dependency parent (p) of this subclause. An example annotation (word tokens with annotation tags) is shown below for the following sentence:

"Although Scotland has chosen to stick with the union, Cameron will still face political fallout over the vote."

Although [although] <clb> KS @SUB #1->4
Scotland [Scotland] <Proper> <Lcountry> N S @SUBJ> #2->4
has [have] <aux> V PR 3S @FS-ADVL> #3->14
chosen [choose] **<REF:CONC:+10>** <mv> V PCP2 AKT
@ICL-AUX< #4->3 **ID:4**
to [to] INFM @INFM #5->6
stick [stick] <mv> V INF @ICL-<ACC #6->4
with [with] PRP @<PIV #7->6
the [the] <def> ART S/P @>N #8->9
union [union] <HHorg> <def> N S @P< #9->7
, [,] PU @PU #10->0
Cameron [Cameron] <*> <Proper> <hum> N S @SUBJ>
#11->14
will [will] <aux> V PR @FS-STA #12->0
still [still] <atemp> ADV @<ADVL #13->14
face [face] **<REF:conc:-10>** <mv> V INF @ICL-AUX<
#14->12 **ID:14**
political [political] ADJ POS @>N #15->16
fallout [fallout] <event><idf> N S @<ACC #16->14
over [over] PRP @<ADVL #17->14
the [the] <def> ART S/P @>N #18->19
vote [vote] <act-s> <occ> <def> N S @P< #19->17
. [.] PU @PU #20->0

Note that the discourse-level annotation (in red) is fully integrated into the rest of the corpus mark-up. For each token ("word") there are well-defined tag fields, e.g., lemma [....], part-of-speech and morphology (upper case letters), syntactic function (@tags), dependency links (#n->m) and secondary tags such as semantic class (<...>).

Most discourse relations hold between clauses and are therefore tagged on clause heads, i.e. main verbs, but sometimes a discourse function will hold between a prepositional phrase and a main verb. In these cases, we map the relational tag on the semantic head of the pp, i.e. the argument of the prepositions, as in the QUOTE-relation below:

*[No fossils had been **found**], [according_to a NASA **representative**].*

5 Evaluation

Though the focus of this paper is on annotation design decisions such as category set and rule formalism, we have done a small pilot evaluation of the current performance of the parser, using a section of the MUCH corpus containing essays from teacher training courses (85,000 tokens). For the time being, we are interested in methodologically important performance

8 Quotes are an exception to this, with the quoting main clause constituting an ARG2.

Table 2. Category frequency and surface trigger percentage

Relation	n	surf %	Relation	n	surf %
ELAB	3178	0.7	**CIRC**	159	83.0
BACK	832	0	**QA**	115	0
COMT	646	0	**CONC**	113	52.2
QUOTE	561	82.5	**RETQ**	89	0
COORD	376	100	**MEANS**	70	100
COTR	369	96.7	**EVID**	62	100
PURP	361	(infm)	**CONS**	43	72.1
COND	221	100	**RESU**	30	46.7
REAS	214	100	**COMP**	30	100
LIST	206	(adv)	**TEMP-AFT**	21	100
TEMP-SAM	201	100			

differences across categories, rather than in absolute performance as such.

A very important methodological distinction holds between cases, where a discourse relation can be built upon overt surface markers, and where it cannot, assuming the further to be more reliable than the latter. Thus, of 7496 binary relations added in all, 40.3 % were based on rules involving conjunctions and prepositions, or 54.7 % when sentence-internal ELAB (such as relative clauses) was ignored, and even higher when including rules with adverbial lexeme triggers, e.g., LIST.

Since some categories are much more reliant on surface triggers than others (and hence safer), it is possible to use these counts to assign automatic confidence measures or to support informed decisions about selective annotation.

Table 2, containing all categories with n>10, shows, that of the larger categories, QUOTE (quote), COTR (contrast), COND (condition), REAS (reason), MEANS, EVID (evidence) and the temporal categories are the most surface-anchored. PURP (purpose) and LIST could be added, since both have fairly safe constructions, with infinitive markers and certain adverbs as surface markers, respectively.

With a rule-based approach, where part of the research goal is identifying the most operationalizable categories, it is not easy to find or create a manual gold corpus, but we still wanted to know how the individual categories perform in a live parse.

The easiest accessible measure for inspection in this setting is precision, i.e. the percentage of false positive tags and relations (tag % and rel % in table 3). For our experiment, we ran a live parse from raw text, including pos, syntax, frames, roles and, finally, discourse relations, then selecting the first 10 tagged instances of each discourse relation category.

As expected, the "surface-heavy" categories (# in table 3) had a good relation attachment (7.3 % errors compared to 28 % for other categories), because the parser could simply follow the syntactic dependency link based on conjunctions or prepositions, and some of the errors were in fact caused by syntactic parse errors. For the category tags (average 19 % vs. 35 %), the effect was less pronounced, mainly due to ambiguity issues with words such as "as" and "since".

6 Conclusion

We have presented an RST-based discourse annotation scheme for the MUCH corpus, arguing that the category set.

– should have sufficient distinctive power to be useful for linguistic and pedagogical purposes.

Table 3: Precision errors (false positives)

Relation	% cat error	% rel error	Relation	% cat error	% rel error
ELAB	10	40	**CIRC #**	40	10
BACK	40	30	**QA**	40	30
COMT	60	60	**CONC**	10	10
QUOTE #	40	0	**RETQ**	50	40
COORD #	30	0	**MEANS #**	0	30
COTR #	10	0	**EVID #**	0	0
PURP	50	40	**CONS**	40	30
COND #	20	10	**RESU**	0	0
REAS #	10	10	**COMP #**	20	20
LIST	50	0	**TEMP-AFT#**	40	0
TEMP-SAM#	0 (ambi.)	0	**average**	26.7	17.1

- should be implementable as an automatic system, without too fuzzy/many categories.
- should be compatible with, and integratable to, the Constraint Grammar approach used for lower level annotation of the corpus.
- We suggest to largely ignore meta-discourse annotation at the present stage and to focus on discourse relations between existing syntactic nodes. Relation classes should be independent of nucleus-satellite distinctions.

We have implemented and tested a first set of discourse annotation rules to run on top of the EngGram CG parser, prioritizing rules based on surface clues (connector particles such as conjunctions) and confirming our expectation that such rules have a higher precision, for both categories and relation target links, than rules trying to link predications without such clues.

Acknowledgments

The work reported here has been supported by Craafordska Stiftelsen through a pilot grant for the MUCH project. Thanks are also due to the Swedish MUCH team: Anna Wärnsby, Asko Kauppinen, Maria Wiktorsson (Malmö University) and Andreas Eriksson (Chalmers University of Technology), for their work on corpus compilation and valuable discussion feedback.

References

1 **Ädel, A. (2006).** Metadiscourse in L1 and L2. Studies in Corpus Linguistics, John Benjamins Publishing, Vol. 24.

2 **Bick, E. (2012).** Towards a semantic annotation of English television news - building and evaluating a constraint grammar FrameNet. 26th Pacific Asia Conference on Language, Information and Computation, pp. 60–69.

3 **Bick, E., Didriksen, T. (2015).** CG-3 - Beyond classical constraint grammar. 20th Nordic Conference of Computational Linguistics (NODALIA), pp. 31–39.

4 **Bick, E. (2015).** DanProof: Pedagogical spell and grammar checking for Danish. International Conference Recent Advances in Natural Language Processing (RANLP), pp. 55–62.

5 **Carlson, L., Marcu, D. (2001).** Discourse tagging reference manual. ISI Technical Report ISI-TR-545, Information Science Institute, Vol. 54.

6 **Da Cunha, I., Torres-Moreno, J. M., Sierra, G. (2011).** On the development of the RST Spanish treebank. 5th Linguistic Annotation Workshop, pp. 1–10.

7 **Eriksson, A., Finnegan, D., Kauppinen, A., Wiktorsson, M., Wärnsby, A., Withers, P. (2012).** MUCH: The Malmö University-Chalmers Corpus of academic writing as a process. 10th Teaching and Language Corpora Conference (TALC10).

8 **Flowerdew, L. (2010).** Using corpora for writing instruction. In: **O'Keeffe, A., McCarthy, M., eds.,** The Routledge Handbook of Corpus Linguistics, pp. 444–457.

9 **Forbes-Riley, K., Zhang, F., Litman, D. (2016).** Extracting PDTB discourse relations from student essays. 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pp. 117–127. DOI: 10.18653/v1/W16-3615.

10 **Iruskieta, M. Aranzabe, M. J., de Ilarraza, A. D., Gonzalez-Dios, I., Lersundi, M., de Lacalle, O. L. (2013).** The RST Basque treebank: an online search interface to check rhetorical relations. 4th Workshop RST and Discourse Studies, pp 40–49.

11 **Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A. (1995).** Constraint grammar: A language-independent system for parsing unrestricted text. Mouton de Gruyer, pp. 1–88.

12 **Mann, W. C., Thompson, S. A. (1988).** Rhetorical structure theory: Toward a functional theory of text organization. TEXT – Interdisciplinary Journal for the Study of Discourse, Vol. 8, No. 3, pp. 243–281.

13 **Marcu, D., Amorrortu, E., Romera, M. (1999).** Experiments in constructing a corpus of discourse trees. ACL Workshop on Standards and Tools for Discourse Tagging, pp. 48–57.

14 **Pardo, T. A. S., Nunes, G., V., Rino, L. H. M. (2004).** DiZer: An automatic discourse analyzer for Brazilian Portuguese. Advances in artificial Intelligence–SBIA, Lecture Notes in Computer Science, Vol. 3171, pp. 224–234. DOI: 10.1007/978-3-540-28645-5_23.

15 **The PDTB Research Group (2008).** The Penn discourse treebank 2.0 annotation manual. Technical Report IRCS-08-01. Institute for Research in Cognitive Science, University of Pennsylvania.

16 **Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., Joshi, A. K. (2008).** Easily identifiable discourse relations. Technical Reports (CIS), Report 884, Institute for Research in Cognitive Science, University of Pennsylvania.

17 **Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., Webber, B. (2008).** The penn discourse treebank 2.0. 6th International Conference on Language Resources and Evaluation (LREC), pp. 2961–2968.

18 **Stede, M. (2004).** The Potsdam commentary corpus. Workshop on Discourse Annotation, Association for Computational Linguistics, pp. 96–102.

19 **Maite, T., Renkema, M. (2008).** Discourse relations reference Corpus [Corpus]. Simon Fraser University and Tilburg University. Available from: http://www.sfu.ca/rst/06tools/discourse_relations_corpus.html.

20 **Wärnsby, A., Kauppinen, A., Eriksson, A., Wiktorsson, M., Bick, E., Olsson, L. -J. (2016**). Building interdisciplinary bridges - MUCH: The Malmö University-Chalmers Corpus of academic writing as a process. In: **Olga, T., Gardner, A. C., Honkapohja, A., Chevalier, S., eds**., New Approaches to English Linguistics: Building Bridges, John Benjamins Publishing, pp. 197–211.

Article received on 18/02/2018; accepted on 11/01/2021. Corresponding author is Eckhard Bick.

ISSN 2007-9737

Distributional Word Vectors as Semantic Maps Framework

Amir Bakarov

National Research University Higher School of Economics, Moscow, Russia

amirbakarov@gmail.com

Abstract. Distributional Semantics Models are one of the most ubiquitous tools in Natural Language Processing. However, it is still unclear how to optimize such models for specific tasks and how to evaluate them in a general setting (having ability to be successfully applied to any language task in mind). We argue that benefits of intrinsic distributional semantic models evaluation could be questioned since the notion of their "general quality" possibly does not exist; distributional semantic models, however, can be considered as a part of Semantic Maps framework which formalizes the notion of linguistic representativeness on the lexical level.

Keywords. Word embeddings, distributional word vectors, semantic maps.

1 Introduction

The Semantic Maps framework in linguistics aims to describe patterns of multifunctionality of grammatical units without grounding to monosemic and polysemic analyses [38]. The core concept of this framework is a semantic map, geometrical representation of grammatical functions (such as uses, meanings, and contexts of grammatical morphemes) as interlinked constituents a so-called "semantic space", a structure that implies graph theory mechanisms and claims to generalize the configuration of functions shown by the map across linguistic phenomena and different languages.

This structure could be viewed as a representation of conceptual similarity between different semantic functions [39], certain scholarly studies, though, do not impose such attribution, and interpret Semantic Maps as a compact description of attested variation, imposing a question of whether this framework may reflect extra-cognitive factors (diachronic or communicative) [37].

Semantic Maps are constructed with a core principle of "contiguity / connectivity requirement" in mind, functions that are often associated with one and the same linguistic expression are represented as nodes adjacent to each other, or as a contiguous region in a semantic map [73], but it does not imply that one and the same linguistic expression represented through an association with several nodes should be analyzed as polysemic.

Therefore, Semantic Maps claim that multifunctionality of a gram occurs only when the various functions of the gram are similar, for example, as one of the possible application of Semantic Maps is separation of polysemy from accidental homonymy, where formally identical elements have unrelated meanings [28].

Ideally, a complete theory of grammatical meaning would allow us to deductively leverage Semantic Maps for deriving language-independent functions as well as their relative positions at the map structure, as functions in Semantic Maps are distributed in a way that allows each gram from each language to occupy a contiguous area on a map.

However, given the data of only one language, we can not be sure which functions to represent on the map in the first place [88]. All in all, Semantic Maps have become a popular method in grammatical typology, being used for capturing both synchronic facts and patterns of development [41].

One of the types of Semantic Maps on which we focus in this article is called Probabilistic Semantic Maps, a way of constructing Semantic Maps through statistical methods based on correspondence analysis of relative occurrences of particular linguistic expressions in different contexts across one or multiple languages [88]. “Semantic space” operated by this type of Semantic Maps is expressed topologically by closeness of nodes in the Semantic Map graph. WordNet [48] is one of the most well-known examples of Probabistic Semantic Maps.

It is manually constructed using heuristic judgments on the similarity concepts as a medium of multifunctionality. Recent scholarly studies propose an alternative to the manual construction of such maps with a Distributional Semantics theory, a context-based, non-compositional approach to meaning [40], following the claim that the meaning of a word can be determined based on patterns of co-occurrence in a corpus [53].

The fundamental assumption in Distributional Semantics is that the word meaning is distributed across contexts of its use, and lexical representations are quantitative functions of their global distributions, which can be viewed as so called Word Vectors.

Given metric as a measure of similarity of words corresponding to given vectors, one can use it as a proxy for semantic relations between corresponding words. This metric can be formally represented with an any kind of similarity measure between vectors, like cosine similarity:

$$\cos(x,\ y) = \frac{x \cdot y}{||x|| \cdot ||y||}. \tag{1}$$

Here x and y are compared vectors. Distributional Semantics has a number of features that make it different from other semantic theories and made it able to become the most ubiquitous semantic concept nowadays:

1. **Distributional word representations** are context-sensitive: linguistic contexts in which words occur construct their semantic constitution.
2. **Distributional word representations** are inherently distributed, i.e., captured word meaning lies in overall distributional history of this word rather than in certain set of explicitly observable features.
3. **Distributional word representations** are gradual, so the captured meaning differs not merely for the contexts they appear, but also for how saliently these contexts describe the combinatorial behavior of these words.

Far from being a “theory-neutral” approach to semantics, Distributional Semantics has been used to test linguistic hypotheses, and there is some evidence supporting the view that semantic associations and textual co-occurrences are related [96]. Distributional Semantics is particularly well suited to describing those aspects of meaning that interact with syntax, such as argument structure.

Usually scholars distinguish two taxonomic classes of distributional semantic models [13]. The first one is based on explicit counts of word co-occurrences in a corpus. Such counts can be done by finding all word-per-word occurrences and measuring the degree of mutual information inheriting from this connection.

Ubiquitous example of a distributional semantic model based on such counts is a model of *Latent Semantic Analysis* [92] which actually builds sparse word-word matrix for pointwise mutual information of word co-occurrences and applies dimensionality reduction on this matrix. This class of distributional semantic models is therefore called count-based distributional semantics.

Models from the second taxonomic class are based on sampling the training corpus with a sliding window, so each word is initialized with a feature vector which values are optimized to accurately predict sequence of words in the corpus given an input sequence of words (language modeling). Since usually this task is resolved with the help of artificial neural networks, such distributional semantic models are usually called neural-based (or prediction-based) distributional semantic models.

Meaning representations captured as input weights of these networks are called word

embeddings [35]. Despite such taxonomy is nominal, several recent works have proven the effectiveness of predictive models against count-based models [13].

Nowadays the neural-based architectures gained the most popularity in the community along with such algorithms as *Continuous Skip-Gram* or *Continuous Bag-of-Words* [106], *Global Vectors* [118], *FastText* [24], and others. Despite distributional hypothesis gained most attention after Harris's work [71], the pre-requisites and early versions of this hypothesis were formulated by other post-Bloomfieldian American structuralists: Martin Koos, Charles Hockett and George Trager [125].

To this end, distributional hypothesis has much connection with structuralist hypothesis, or formal approach to linguistic in general [50]. As in structural models, words in distributional hypothesis are defined according to their features in the lexicon, and the meanings are defined by contrasts in these sets of features: in distributional hypothesis words' contexts of use play the role of these features, while structuralist hypothesis relies on manually handcrafted properties.

This leads to lower linguistic motivation of distributional hypothesis comparing to the structuralist one: unsupervised feature construction is more convenient for downstream tasks, but they can lack an intrinsic meaning [25]. This issue of intrinsic vagueness grows from the application purpose of distributional hypothesis: as [125] puts it, the structuralist distributional procedure was originally introduced for phonemic analysis, and only after few time turned into a general methodology able to be applied to every linguistic level.

This procedure was a way for linguists to ground their analyses on firm methodological base, avoiding any argument based on meaning as an identity criterion for linguistic elements [64]. But mapping distributional hypothesis to word studies can go against semantic theories relying on precise identity criteria for semantic content of words since distributional semantics build this criteria on a generalization of paradigmatic relations built upon on linguistic distributions.

For example, we can formulate distributional hypothesis as the ability of the degree of semantic similarity between two linguistic expressions A and B to be considered as a function of the similarity of linguistic contexts in which A and B can appear [96]. One can observe that this actually inherits Bloomfield's refusal of meaning as an linguistic explanans [23], defining meaning as a similarity of words' distribution (in other words, it supports the idea that an exploration of a number of contexts of a word can evidence some of its semantic properties). But while Bloomfield assumes that research of meaning goes beyond linguistic research, the distributional hypothesis implicitly puts a solid empirical (statistical) foundation for meaning analysis.

However, we assume that meaning becomes a part of empirical studies only at those aspects that can be defined through distributional analysis procedures. From the position of cognitive linguistics, many more aspects of meaning are not ever fixed in written traditions [63]. Through the view of conceptual hypothesis and prototypical hypothesis (or functional approach in general), the distributional theory also could be motivated through the same methods that appear as ingredients of human conceptualization (particularly, the linguistic contexts).

According to [93], contexts intrinsically embody conceptual representation of aspects of the world. Such representations commonly propose a functional explanation in terms of the principles governing the process of conceptualizing the word. Therefore linguistic distributions and meaning proposed by them are explained in many cases by embodied conceptualisation.

Certain psycholinguists, e.g. [109], assume that repeated encounters of a word in various linguistic contexts eventually determine the formation of a contextual representation. This view on meaning highly relates with that Wittgenstein puts as "meaning of word is its use in the language" [143].

However, formal (Harris') and functional (Miller's) views on distributional theory have a few in common. While Harris puts distributional analysis as a purely empirical method of linguistic research, Miller assumes a cognitive background of meaning

and claims that it goes beyond purely statistical investigation. To this end, linguistic motivation of distributional semantic models relies on dichotomy of these theories, and can be explained from both these sides (either as be criticized).

If we want to support the claim of Distributional Semantics as a legit linguistic framework that can be viewed from the perspective of Semantic Maps theory, we should accept assumptions of one of these views. From the structural views, we must posit that language is a system of inter-related units and structures and that every unit of language is related to the others within the same system. From the conceptual views, we should assume that mental representations which encode the human understanding of the world contain the primitive conceptual elements of which meanings are built.

However, despite all this recent progress, structural and conceptual theories are still much more heuristics than practice and yet lack a strong experimental basis. Therefore, the possibility to support the legitimacy of Distributional Semantics as a linguistic framework can be doubted.

To this end, we suggest to turn the investigations of linguistic legitimacy of distributional semantics to the computational side, which proposes more objective framework. As probabilistic semantic maps can be viewed as modeling the semantics of linguistic diversity (and they do so to the extent that the sample, which is an underlying typological database, is representative of the populatio, which is the entire linguistic diversity), we can pose a general question is whether semantic maps based on linguistic data can model universal semantic space, claimed to be the ultimate aim of Semantic Maps framework in the beginning of this article.

If semantic space is both mental and universal, it must be both comprehensive and robust. Robust means that different samples of languages and of semantic functions are assumed to yield highly similar maps representing the full range of semantic diversity encountered in natural languages. Comprehensive means that all semantic categories encountered in the database must be well-represented [141].

While being more focused on “the semantics of individual lexical items, their configurations in lexical field or individual processes of word formation” rather than on “typologically relevant features in the grammatical structure of the lexicon” [95], we assume that the primary benchmark for distributional semantic models as semantic maps can be proposed from the perspective of evaluation of their features as models of lexicon. Existing approaches to evaluation of distributional semantic models divide on perspectives of extrinsic evaluation (evaluation on downstream tasks) and intrinsic evaluation (evaluation of inner properties of the models) [128].

Former methods are based on the ability of distributional word vectors to be used as the feature vectors of supervised machine learning algorithms used in one of various downstream natural language processing tasks. On the opposite, methods of intrinsic evaluation are experiments in which word vectors are compared with human judgements on words relations. Manually created sets of words are often used to get human assessments, and then these assessments are compared with word vectors.

Intrinsic evaluation relies on in vivo experiments to obtain human judgments from assessors. Such an estimate could be used as an absolute measure of the quality of word vectors since it reports the similarity of lexical semantics inferred by a distributional semantic model to the lexical semantics determined by humans. To this end, I consider distributional semantic model representative of language only in case they demonstrate decent evaluation performance.

Intrinsic evaluation approaches like the “word semantic similarity task” (will be covered in more detail in the next section) do not use such formalizable a strict notion of the model’s performance. From an intrinsic evaluation perspective, word embeddings are usually assessed using our (humans’) understandings of relationships between words (and other lexical units), for example, by collecting human annotations of a so-called “word semantic similarity”.

Usually, intrinsic evaluation relies on psycholinguistic tasks which collect human judgments on the “gold standard” of different properties of the lexicon. These tasks (experiments) are conducted either in the laboratory

with a limited set of examinees (**judgments collected in-house**) or on crowd-sourcing Web platforms like Mechanical Turk, attracting an unlimited number of participants (**judgments collected through crowd-sourcing**) [98].

Sometimes the assessors are asked to evaluate the quality of word embeddings directly, for instance, when different models produce different judgments on word relations, and the task of an assessor is to tell which model works better. This type of intrinsic evaluation is called **comparative intrinsic evaluation** [128], in opposite to the regular or "absolute" intrinsic evaluation, it allows not to estimate the absolute quality of the word vectors, but to find the most adequate vectors in a given set.

As it was mentioned, unlike "extrinsic" evaluation, the "intrinsic" approach tries to assess a more general notion of word embeddings performance (particularly, can DSMs be used as a proper formalization of a lexicon?) but at the same time, it relies on less formalizable and more vague concepts. Particularly, is unclear which type of relationships the word embeddings should reflect (synonymy? co-hyponymy? something else), and if the model takes into account not the type of relationships that we know (like synonymy), but other types of relationships, how should we assess it [47].

Scholar works on this topic tend to face various methodological problems, such as lack of proper test sets (resulting in adjusting the models to the data trying to increase their quality) or absence of the statistical significance tests. One of the main issues with most of the scholarly research on this topic is that there is no strict definition of an evaluation method in the field of distributional semantics (after all, if the notion of word meaning could not be even defined properly, how the notion of its modeling evaluation could be defined?).

Therefore, we consider by the method of word embeddings evaluation any way or attempt of finding a link (correlation) between a DSM and **any** data that hypothetically could carry information about lexical semantics. The evaluation representativeness obviously depends on the degree of plausibility of the hypothetical amount of lexicon information in the data one tries to use for evaluation, but the general intuition is that we are not able to strictly evaluate this amount.

2 Empirical Benchmarks

2.1 Semantic Similarity

The most well-known benchmark of **word semantic similarity** directly assesses the ability of DSMs to report representative distance between the word vectors in terms of the ability of this distance to be grounded to human assumptions on that distance between corresponding words.

For example, if the so-called "distance" between *cup* and *mug* (defined in a continuous interval $0, 1$) predicted by the model is 0.8, then we assume that the distance is reported correctly by the model of the human assessor asked to estimate the "distance" between these words (whatever it means depending on the annotation guidelines) outputs a similar value.

These distances, both DSM's and human's one, are collected on a range of pairs of words, and we expect to find a meaningful correlation between these two sets (usually, more than one assessor is used for the sake of reliability of the provided scores). Having two different models, we consider the better model the more correlated are the predictions [13].

Word similarity benchmark is also one of the oldest ones, its roots go back to 1965 when the first experiments on human judgments on word semantic similarity were conducted to test the distributional hypothesis from the psychology perspective [124] (in 1978 a similar work was carried out in [112]).

Despite the strong psycholinguistic background of this method, it is one of the most frequently criticized in the community, obviously for subjectivity and vagueness [54, 47, 18], there are a lot of potential linguistic, psychological [85] and social factors [117], which could introduce bias in the assessments [61].

The task is also very much dependent on possible connotations in the word lists [98], and the ambiguity of the overall assessment, different works propose different definitions of semantic similarity, while some scholars define

it as co-hyponymy (like the relation between the words *machine* and *bicycle*) [137], others define it as synonymy (like in a word pair *mug* and *cup*) [77].

It was also argued that the notion of semantic similarity inherits not only semantic connections of words but also some morphological and graphemic features of word representations [87].

Among other criticized features of word semantic similarity, there is also the lack of correlation between these human assessments and the performance of word embeddings on extrinsic methods [33, 132], the low inter-rater agreement between annotators [77], the factor of assessors getting tired when annotating a large number of pairs [27], poor ability of numerical labels to fully describe all types of relations between words (it is suggested that it will be better to describe the degree of word similarity in a natural language [108]), and the misconduct of thematic roles relations [44].

It is also unclear whether such embeddings reflect enduring properties of language or if they are sensitive to inconsequential variations in the source documents [6, 8]. Datasets:

1. **SimVerb-3500**, 3 500 pairs of verbs assessed by semantic similarity (that means that pairs that are related but not similar have a fairly low rating) with a scale from 0 to 4 [59].
2. **MEN** (acronym for Marco, Elia and Nam), 3 000 pairs assessed by semantic relatedness with a discrete scale from 0 to 50 [27].
3. **RW** (acronym for Rare Word), 2 034 pairs of words with low occurrences (rare words) assessed by semantic similarity with a scale from 0 to 10 [100].
4. **SimLex-999**, 999 pairs assessed with a strong respect to semantic similarity with a scale from 0 to 10 [77].
5. **SemEval-2017**, 500 pairs assessed by semantic similarity with a scale from 0 to 4 prepared for the *SemEval-2017 Task 2* (*Multilingual and Cross-lingual Semantic Word Similarity*) [30]. Notably, dataset contains not only words, but also collocations (*e.g. climate change*).
6. **MTurk-771** (acronym for Mechanical Turk), 771 pairs assessed by semantic relatedness with a scale from 0 to 5 [69].
7. **WordSim-353**, 353 pairs assessed by semantic similarity (however, some researchers find the instructions for assessors ambiguous with respect to similarity and association) with a scale from 0 to 10 [51].
8. **MTurk-287**, 287 pairs assessed by semantic relatedness with a scale from 0 to 5 [122].
9. **WordSim-353-REL**, 252 pairs, a subset of WordSim-353 containing no pairs of similar concepts [3].
10. **WordSim-353-SIM**, 203 pairs, a subset of WordSim-353 containing similar or unassociated (to mark all pairs that receive a low rating as unassociated) pairs [3].
11. **Verb-143**, 143 pairs of verbs assessed by semantic similarity with a scale from 0 to 4 [12].
12. **YP-130** (acronym for Yang and Powers), 130 pairs of verbs assessed by semantic similarity with a scale from 0 to 4 [144].
13. **RG-65** (acronym for Rubenstein and Goodenough), 65 pairs assessed by semantic similarity with a scale from 0 to 4 [124].
14. **MC-30** (acronym for Miller and Charles), 30 pairs, a subset of RG-65 which contains 10 pairs with high similarity, 10 with middle similarity and 10 with low similarity [109]. Also, there is a subset of MC-30 called **MC-28** which excludes 2 pairs not represented in WordNet [123].

2.1.1 Synonym Detection

The so-called Synonym Detection task is very close to the previously described task of Semantic Similarity, but while it also assesses the ability of DSMs to provide reliable distances between words, it does not rely on an absolute degree of similarity in terms of a scalar value. Instead, we assume that

we can do the thing by finding the most similar word relative to a set of other words.

So, given a word a and a set $K = b_1, b_2, b_3$, the task is to find b_i which is the most synonymous (semantically similar in terms of the word semantic similarity task) to a [13].

For example, for the target *levied* one must choose between *imposed* (correct), *believed*, *requested* and *correlated*. The task of a DSM is to find the word vector with the smallest distance to the vector of the specified word.

Taking into account all the criticism of the word semantic similarity method, moving from the absolute measure to the relative measure could probably exclude a lot of problems of this task (score bias, lack of assessments interpretability, etc.).

On the other hand, the creation of a dataset for evaluation in this task is more complicated and raises certain new questions (for example, how to properly choose the words to form the set K). Datasets that could be used for evaluation on this task presented in a form of 5-word tuples in which one word is a target word, and 4 words are potential synonyms where the only one is a correct answer:

1. **RDWP** (acronym for Reader's Digest Word Power Game; also mentioned as RD-300), 300 synonym questions (5-word tuples) [82].
2. **TOEFL** (acronym for Test of English as a Foreign Language), 80 questions [92].
3. **ESL** (acronym for English as a Second Language), 50 questions [134].

2.2 Word Analogy

The task of Word Analogy (in some works being also called *analogical reasoning*, *linguistic regularities* and *word semantic coherence*) implies the intuition that the arithmetic operations in a word vector space should have a common sense reasoning.

Given a set of three words, a, $a*$ and b, the task is to identify such word $b*$ that the relation b:$b*$ is the same as the relation a:$a*$ [133, 119, 13]. For instance, having $a = Paris$, $b = France$, $c = Moscow$, the target word would be $Russia$ since the relation $a : b$ is $capital : country$, so one needs need to find the capital of which country is *Moscow*. There are different metrics that can be used in this benchmark, though:

— *3CosAdd* (and a similar metric *3CosMul*) proposed in the original *Word2Vec* paper is based on arithmetic operations in vector space (addition and multiplication of cosine distances) [107].

— *PairDir* modifies *3CosAdd*, taking into account the direction of the resulting vectors in these operations [97].

— *Analogy Space Evaluation* metric compares the distances between words directly without finding the nearest neighbors [32].

This task was also criticized and investigated at [110]. A theoretical investigation of analogy phenomena of word vectors was presented in [60]. There was a concept of temporal word analogies also introduced [131].

[52] also gives much attention to the problem of analogy solving. We also provide a list of datasets which could be used for evaluation on this task. As [62] notes, datasets designed for *semantic relation extraction task* could also be used to compile a word analogy set.

In this case, it also worth looking at the *Lexical Relation* set which is a compilation of different semantic relation datasets including *BLESS* [16] (12 458 word pairs with a relation comprising 15 relation types) [140] and the *Semantic Neighbors* set (14 682 word pairs with a relation comprising 2 relation types, meaningful and random) [115].

1. **WordRep**, 118 292 623 analogy questions (4-word tuples) divided into 26 semantic classes, a superset of *Google Analogy* with additional data from WordNet [57].
2. **BATS** (acronym for Bigger Analogy Test Set), 99 200 questions divided into 4 classes (*inflectional morphology*, *derivational morphology*, *lexicographic semantics* and *encyclopedic semantics*) and 10 smaller subclasses. [62].

3. **Google Analogy** (also called Semantic-Syntactic Word Relationship Dataset), 19 544 questions divided into 2 classes (*morphological relations* and *semantic relations*) and 10 smaller subclasses (8 869 semantic questions and 10 675 morphological questions) [105].
4. **SemEval-2012**, 10 014 questions divided into 10 semantic classes and 79 subclasses prepared for the *SemEval-2017 Task 2* (*Measuring Degrees of Relational Similarity*) [86].
5. **MSR** (acronym for Microsoft Research Syntactic Analogies), 8 000 questions divided into 16 morphological classes [107].
6. **SAT** (acronym for Scholastic Aptitude Test), 5 610 questions divided into 374 semantic classes [136].
7. **JAIR** (acronym for Journal of Artificial Intelligence Research), 430 questions divided into 20 semantic classes. Notably, dataset contains not only words but collocations (like *solar system*) [135].
8. New analogical reasoning dataset [72].

2.3 Thematic Fit

The method of Thematic Fit (also called *selectional preference* in [13]) is to separate different thematic roles of arguments of a predicate and to find how well the word embeddings could find most semantically similar noun for a verb that is used in a specific role.

For humans, a certain verb could cause a person to expect that a certain role must be filled with a certain noun (e.g., for the argument *to cut* the most expected argument in the *object* role is *pie*) [127].

Experiments propose assessments of adequacy score of the tuple $\{$verb, noun, thematic role$\}$ (for example, *people eat* is more common phrase than *eat people*, so the pair *people* and *eat* would have the higher score) [139].

Some researchers consider another variation of this method, proposing the task of assessing a pair of words n (noun) and v (verb) by the most frequent role in which n used with v (e.g., pair $people, eat$ would be classified as the *subject* since it is more common to use $people$ as a subject with that verb) [15].

It is unclear, though, which method of obtaining an embedding for a thematic role to distinguish different roles of the argument is the most adequate, some researchers propose a method of vectorization of "slots" for certain thematic roles, which are obtained by averaging several most frequent nouns encountered in a given role [15].

1. **MSTNN** (abbreviation mentioned in [127]), 1 444 *verb-object-subject* pairs [103].
2. **GDS** (acronym for Greenberg, Sayeed and Danberg), 720 *verb-object* pairs. The dataset is additionally divided into a subsample containing only polysemous verbs (*GDS-poly*) and a subsample containing monosemous verbs (*GDS-mono*) [65].
3. **F-Inst & F-Loc** (acronym for Ferretti-Instrument and Ferretti-Location), 522 verbs pairs which are split to a subset of 248 verbs with associated *instruments* (*F-Inst*) and a subset of 274 verbs with associated *locations* (*F-Loc*) [49].
4. **P07** (acronym for Pado), 414 *verb-object-subject* pairs [114].
5. **UP** (acronym for Ulrike and Pado), 211 *verb-noun* pairs, the set of roles is unlimited [113].
6. **MT98** (acronym for McRae and Tanenhaus), a subset of 200 verbs from *MSTNN* where each verb has two nouns, one is a good subject, but a bad object, and one which is a good object, but a bad subject [104].

2.4 Concept Categorization

The method of Concept Categorization assesses a word vector space's ability to be split into distinguishable categories, i.e., clusters. Given a set of words, we want to map each word into a meaningful category which can have common sense reasoning (for example, for words *dog*,

elephant, *robin*, *crow*, the first two make one cluster which is *mammals* and the last two form another second cluster which is *birds*; the cluster name is not necessary to be formulated) [13].

Lexical-typological research typically asks questions such as how languages categorize particular domains (human body, kinship relations, color, motion, perception, etc.) by means of lexical items, what parameters underlie categorization, whether languages are completely free to "carve up" the domains at an infinite and arbitrary number of places or whether there are limits on this, and whether any categories are universal (e.g., say 'relative', 'body', or 'red').

The critique of such a method addresses the question of either choosing the most appropriate clustering algorithm or choosing the most adequate metric for evaluating clustering quality.

A different way to approach this evaluation method was introduced in the works related to categorical modularity, which is a graph modularity metric based on the k-nearest neighbor graph constructed with embedding vectors of words from a fixed set of semantic categories, in which the goal is to measure the proportion of words that have nearest neighbors within the same categories [31].

The underlying principle is that in good embeddings, words in the same semantic category should be closer to each other than to words in different categories.

The authors quantify this by building the k-nearest neighbor graph with a fixed set of words' semantic categories and computing the graph's modularity for a given embedding space. Modularity measures the strength of division of a graph with densely connected groups of vertices, with sparser connections between groups [55]. The datasets for the Word Categorization task are presented with sets of words classified into a number of certain categories.

1. **BM** (acronym for Battig and Montague), 5 321 words divided into 56 categories [17].
2. **AP** (acronym for Almuhareb and Poesio), 402 words divided into 21 categories [4].
3. **BLESS** (acronym for Baroni and Lenci Evaluation of Semantic Spaces), 200 words divided into 27 semantics classes [16]. Despite the fact that BLESS was designed for another type for evaluation, it is also possible to use this dataset in a word categorization task, as in [83].
4. **ESSLLI-2008** (acronym for the European Summer School in Logic, Language and Information), 45 words divided into 9 semantic classes (or 5 in less detailed categorization); the dataset was used in a shared task on a *Lexical Semantics Workshop on ESSLI-2008* [14].

2.5 Outlier Word Detection

This method of Outlier Word Detection evaluates the same feature of word embeddings as the word categorization method (it also proposes clustering), but the task is not to divide a set of words into a certain amount of clusters, but to identify a semantically anomalous word in an already formed cluster (for example, for a set $\{$orange, banana, lemon, book, orange$\}$ which are mostly fruits, the word *book* is the outlier since it is not a fruit) [29].

Some researchers propose a very similar method called *evaluation of coherence in semantic space*. The idea of this method is, given a set of three words – word a, the two words a_1 and a_2 which are the closest to a in an embedding space are found, – a word b is chosen randomly from the model's dictionary (this word probably would not be so semantically similar to a), and the task of a human assessor is to correctly identify b (the outlier) in the set a, a_1, a_2, b [128]. The more words are identified correctly, the better is the model.

1. **8-8-8 Dataset**, 8 clusters, each is represented by a set of 8 words with 8 outliers [29].
2. **WordSim-500**, 500 clusters, each is represented by a set of 8 words with 5 to 7 outliers [20].

3 "Subconcsious" Experimental Evaluation Tasks

As we mentioned in the previous section, because the notion of DSMs quality is not bounded only by standard benchmark performance, extending to the territory of a more global question of building a lexicon model, we also attempt to overview experimental evaluation tasks that might not be industry applicable (yet), but which can provide important insights from linguistic and scientific points of view on distributional semantics. Moreover, with the recent trends in the community, these methods start to get out of the "experimental" zone and started to get more attention from different researchers both from cognitive sciences and from language technology [7].

Later in this section, we describe different ways of collecting cognitive data and their application to DSMs evaluation in more detail.

3.1 Semantic Priming

A semantic priming behavioral experiment is based on a hypothesis that a human should read a word faster if it is preceded by a semantically related word (which can introduce an association in a brain). Within the experiment, the time of reading a specified word a (called the *target word*) is compared with the time required to read it when it occurs after a word b_1 and with the time required to read it in a case it occurs after a word b_2.

If the reading time of the word b_1 is lower than the reading time of the word b_2, than the word b_1 is claimed to be semantically related to a (b_1 is called *prime*, or *prime word*, or *stimulus word*) [46, 9]. Reading time is tracked using eye-tracking or safe-paced reading [101, 94], [84, 76, 102, 126].

The most notable dataset used for semantic priming experiments is the *Semantic Priming Project*, containing 6 337 pairs of words. The data is collected from 768 subjects for 1 661 target words. Every word pair is presented in four versions: first, depending on the time interval on the demonstration of the target and non-target words which is 70 and 200ms (this interval is called *stimulus onset asynchronies, SOA*), and, second, depending on the task for the priming, naming task or lexical decision task [79].

3.2 Functional Magnetic Resonance Imaging

One of the most ubiquitous ways to analyze neural activity in a human brain is functional magnetic resonance imaging (functional MRI, fMRI), which records changes in blood level on the brain cortex (bloodoxygen-level-dependent (BOLD) responses), while the brain is presented with certain stimuli. BOLD responses are commonly represented as dense 4-D arrays of the measured data, where time series of the blood flow-related activity measured in tens of thousands of voxels (which are small areas of size equal to approximately 2x2x2 mm^3) are measured across the brain.

These excitations are hypothesized to be elicited primarily by the presented stimulus (with minor background contamination due to respiration, heartbeat, or movement). Hypothetically, the stimuli find their representations in these voxel patterns, and the works aiming to map DSMs data and fMRI data usually rely on matching these two types of data in different ways, particularly, to use word embeddings to try to predict voxel's activation [80, 78].

The evaluation method is based on using as a gold standard the data of fMRI experiments which measures changes associated with blood flow in certain parts of the brain by fixating regions of the blood flow at certain time intervals (once a second, for instance).

The idea is that the blood flow and the neuronal activation patterns correlate, so one could identify parts of the brain that are activated. In the field of neurolinguistics, reading or listening to the text is usually considered to be a stimulus for this activity. The obtained data is presented as a set of voxels reporting the level of neuronal activity in different small parts of the brain.

It is not clear how to obtain data on reading single words, since the minimum time interval on fixating blood flow is about 1 second; some researchers try to train a regression model to compute the average brain activation vectors for each word or to use aggregate statistics to obtain

vector representations of fMRI data using it is as a gold standard [130, 1].

One could try to use *StudyForrest* [70] dataset which offers data on listening to the audio track of the "Forrest Gump" movie in German, or the *Word Processing* dataset which contains readings for various natural language words on English [43]. Most of the studies, though, do not try to compare different word embeddings models to each other, but just try to figure out whether they are capable to encode abstract information [138].

3.3 Electroencephalography

Electroencephalography (EEG) records the electrical activity of the brain, and the idea is that the amplitude of the impulses in the brain that occur on words (such response is called N400, it is an early response elicited by every word of a sentence) stores information about lexical semantics since the interpretation of the response is usually generalized by the hypothesis that the worse the word fits the context (which could be both sentence context and word context), the higher is the amplitude of the signal.

The amplitude differences of a tuple of words are able to be simulated through the average cosine distances of word embeddings, so it is hypothetically could be used as a gold standard data for evaluation [116, 45, 129].

3.4 Eye Movement Data

This evaluation method is based on using as a gold standard the data of human eye movement obtained. Such data could be obtained through an instrument called *eye-tracker* which tracks the movement of a pupil and a time of fixation on certain words while a person reads text from the computer screen, and such data hypothetically could carry some information about lexical semantics.

The eye-tracker assigns to each word a set of features reporting characteristics of its reading: how many milliseconds the gaze was fixated on this word, how many times the gaze returned to it, etc. Such feature sets can be compared with word embedding vectors, considering word vectors as another "feature set", the correlation between such vectors and word embeddings (for instance, on predicting k nearest neighbors to a certain word) can report the quality of a DSM [130, 10].

We are aware only of two publicly available English eye movement datasets that one could use in their experiments. The first is the **Provo Corpus** [99] which consists of data of reading 55 paragraphs from 84 native speakers. This dataset could be converted into a list of 1 185 words each of which is associated with a set of 26 eye movement features.

The second dataset is the **Ghent Eye-Tracking Corpus (GECO)** [36] containing data of reading 5 000 sentences from monolingual and bilingual English speakers (33 participants overall). After converting one could obtain a dataset of 987 words, each associated with 48 features.

4 Experimental Data-Driven Evaluation Methods

4.1 QVEC

Building the inverted index of a collection of documents in which each is responsible for a certain category of human knowledge like super-senses in WordNet (e.g. *food*, *animal*, etc.), we can construct the so-called "thesaurus vectors" and use them a proxy for evaluating word embeddings.

The dimensionality of these "thesaurus vectors" is the size of the document collection, and each component in these vectors reports the number of occurrences of the word in a certain document.

For the sake of computational efficiency (to process the large collections), we can also map one component of an embeddings vector to multiple components of thesaurus vectors (or vice versa if the collection is too small) [132].

In the original paper presenting this method, the authors used a so-called "conceptual thesaurus" based on WordNet, but we believe that a set of documents that claims to contain a comprehensive set of the knowledge categories can be used to obtain the "thesaurus vectors".

For instance, *Wikipedia*, which was already similarly used for document vectorization, referring to a method of *Explicit Semantic Analysis*

[56] which was considered for the task of cross-language information retrieval.

4.2 Dictionary Definition Graph

Co-occurrences of words in dictionary definitions could carry information about their relationships [2], we construct a digraph from the set of dictionaries where the nodes are represented by the words, and the values of the edges connecting the word a to the word b are represented by the number of all occurrences of the word b in all the definitions of the word a.

Transforming this graph to a matrix, we obtain a "dictionary vector" for each word, and use these vectors as a proxy of evaluation. Alternatively, one can represent the edges not with simply frequencies of the co-occurrences but the amounts of time when b was encountered as a head in the dependency syntax tree (such an idea can help to identify similarities based on phrases like *a cat is an animal*).

4.3 Cross-Match Test

A Cross-match test is a technique of finding similarity between two high-dimensional sets used to compare blood samples in medicine, and we can use this method for evaluating word embeddings as well.

Determining whether the two sets of values are sampled from the same distribution, we measure the statistical significance of a model, if the correlation of a sample of vectors of two different word vector models is low, then the two compared models probably use different features of the corpus, so it is probably a good result [67].

4.4 Semantic Difference

Characterizing words of the distinctive features (*attributes*), we consider each word in a pair associated with a certain set of attributes. The distance between words is calculated as the difference between the Cartesian product multiplied by the attributes of the word vectors, we can select a pair of attributes of the same category for each pair of non-abstract words (e.g. the category could be *size*, and the distinctive attributed could be *big* and *small*) [90].

There is a certain amount of databases where words are associated with sets of different attributes. One of the examples of such bases is a previously mentioned *BLESS* dataset, which contains 200 pairs of words (for example, for the [*motorcycle*, *moped*] word pair these are the two sets of attributes: [*large*, *small*] and [*fast*, *slow*]) [16].

Another example is *Feature Norms Dataset* containing 24 963 pairs of words, for which a least one pair of distinctive features is selected (for example, for the pair [*airplane*, *helicopter*] the *existence of wings* is selected) [90].

4.5 Semantic Networks

In manually constructed knowledge graphs like WordNet [74], *semantic networks*, the words are organized following their semantic distinctive features based on judgments of the linguists. These graphs provide a measure of similarity for word pairs based on the shortest path in a graph, so such similarity measure can be used as a proxy for the similarity measure of the same pair calculated by word embeddings to evaluate its quality [3].

4.6 Phonosemantic Analysis

The general (and very heuristic) intuition is that the form of a linguistic sign is not arbitrary and has a relationship to its meaning. If that is true, we can use phonosemantic patterns of the word (its phonemes or characters) as a proxy for its meaning. To calculate the phonosemantic difference between two words, one could measure using Levenshtein distance measure, and such metric could be used as a gold standard for evaluation [68]. Notably, this observation was confirmed not only for the Latin alphabet but also for Cyrillic [91].

4.7 Bi-Gram Co-Occurrence Frequency

The distance between the words vectors representing words of a phrase group (e.g. *noun + adjective*) should correlate with the frequency of this group in a corpus (bi-gram co-occurrence frequency). In other words, bi-gram co-occurrence frequency could be used as a gold standard [89].

4.8 Image-Based Evaluation

The method relies on image vectors for word embeddings evaluation and explores whether the similarity spaces generated by two disparate algorithms give rise to similar similarities among high-frequency items [5].

4.9 Closed Domain Evaluation

The method aims at the evaluation of word (and sentence) embeddings from specialized corpora in concept-focused domains [111]; the authors suggest using so-called “ground truths” as a proxy for evaluation [19].

Based on a QUINE corpus, the evaluation consists of a semi-formal definition of the relations of some key terms to other terms, and by defining these interrelations between terms in the corpus, the expert knowledge of the meaning of a term within the corpus is reflected by how the term relates to other terms.

In the case of our Neo-Latin corpus, the domain expert identified that definition (definition) and axioma (axiom) are functional synonyms of principium (principle). Similar to the task discussed above, to successfully complete this task, the cosine distance of the vector of a given target term has to be nearer to the vectors of their functional synonyms than alternative terms.

In the case of principium, definitio and axioma, the cosine distance of the vectors of these terms is expected to be nearer to each other than to other terms. Such a conceptual evaluation grounded in expert knowledge provides a method to evaluate word embeddings intrinsically and, thereby, the quality of their consistency [22].

4.10 Consistency Evaluation

The model is considered consistent if its output does not vary when its input should not trigger variation (i.e., because it is sampled from the same text). Thus, a model can only be as consistent as the input data it is trained on and it requires the experimenter to compute data consistency in addition to vector space consistency.

To evaluate data consistency, we create vectors for target terms in a domain corpus under two conditions: a) random sampling; b) equal split. The “equal split” condition simply corresponds to splitting the data in the middle, thus obtaining two subcorpora of equal size and in diachronic order. Given a pre-trained background space kept frozen across experiments, the vector representation of a target is generated by simple vector addition over its context words.

Therefore, the obtained vector directly represents the context the target term occurs in, and consequently, similar representations (in terms of cosine similarity) mean that the target term is used similarly in different parts of a book/corpus, and is thus consistently learnable. Crucially, though, this measure may interact with data size [21]. A similar metric considered as “reliability“ was checked in [75].

5 Conclusion

The core concern of lexical typology, i.e., how languages express meanings by words, can be approached from slightly different perspectives. We can start from the meanings, or concepts, and ask how these are expressed in different languages, among other things, how semantic domains are distributed among the lexical items across languages.

Lexico-typological research can also start from the expressions (lexemes) and ask what different meanings can be expressed by them or by lexemes that are related to them synchronically and/or diachronically. In this survey we systematized the existing attempts to answer a question of what is a good distributional semantic model and we highlighted that this question always implicitly supposes a question of what is a good model

of the lexicon, and therefore another question of what lexicon is.

We tried to make this paper useful for engineers from the industry as well as linguistics from academia, so we extensively described both well-known "empirical" evaluation methods (such as word similarity task and word analogy task) and experimental methods based on the use of thesauri, semantic networks, or even neuroimaging data.

To not extend this paper to a monstrous size, we tried to focus on overviewing and discussing only those works which were dedicated to both a) "traditional" distributional semantic models aiming to produce representations of lexical units (e.g. Word2Vec), b) the problem of evaluation of the quality of such models or the representations produced by such models, nevertheless what was meant by "quality" by the authors of such works.

What we believe is that the notion of the quality of distributional word representations is heavily tied to the notion of their linguistic representativeness, i.e., the degree of being a proper model of a lexicon. Despite this notion of representativeness being grounded to theoretical linguistics and the legacy of formal analysis of semantics, we have shown in Section 3 dedicated to the so-called subconscious evaluation methods that the exploration of cognitive dynamics could be a promising direction towards understanding mechanisms of distributional semantics.

Another view on the quality of the DSMs that we consider important for further studies (despite it has not been included in this survey) considers the reliability of distributional semantic models from the position of fairness and prejudices [11] a representative model should not contain prejudices against certain groups of people (by their gender, ethnicity, sexual orientation, etc.).

If we accept this assumption, we should understand how underlying mechanisms of such fairness bias look like, and how to automatically remove bias from vector spaces. We consider that this fairness property is also an important feature of distributional semantics, and experiments grounding either to DSM's quality, performance, reliability, linguistic motivation, or whatnot, also should have such issue in mind.

As we mentioned in the first section of the paper, the trends in NLP are now heavily shifted towards sentence embeddings, and traditional word vector representations became a "niche" topic. However, we assume that algorithms like ELMo [120], BERT [42] and GPT-2 [121] do not provide so much of scientific interest from the position of the lexicon as traditional word embeddings, as they are not relying on single words, but adopt a more "syntactically savvy" notion of linguistic contexts, in which word semantics are reconstructed by specific syntactic configurations.

There is a hypothesis that such evaluation of representation in context is more reliable, and experiments on such context-based models would be more representative. But context-based setting just grounds to one of the semantic views which are not assumed to be absolutely correct [58].

The theory that lexical semantics is not grounded to the context [81] gains motivation from cognitive studies. It basically gives the main attention to how words obtain meaning in human cognition and interact with other linguistic units, while the context-sensitive approach is not compatible with the idea that syntax a priori acts as the scaffolding that guides distributional analysis. This survey could be considered as a small step forward to bigger planned research of computational formalisms for lexical semantics.

We plan to give more attention to other semantic theories and their theoretical background to propose a more detailed exploration of distributional semantics, for example, the referential one. One of the possible directions of further work on the topic of this survey goes to the intersection of referential and distributional theory. Taking roots from structural theory, certain theories try to ground distributional hypothesis to an interpretable framework [26].

Probably one of the most notorious works in this field goes to **Compositional Distributional Semantic** framework [34], DisCoCat, which suggests construction representations of sentences or documents not through arithmetic operations on word vectors, but by categorical logical operators.

Recent studies propose its experimental support [66] and we assume that such approach could be

more efficient than common distributional theory on certain downstream tasks like anaphora and ellipsis resolution [142].

We hope that the work done by writing this survey at least could be helpful to scholars to look at distributional semantics from a new scope and start to treat word vectors not only as black-box tools for resolving downstream tasks but as linguistic formalisms that have their benefits and limitations from the language perspective.

Despite most of the experimental studies on a similar topic doubting the generalizing ability of distributional semantics, we do not suggest refusing it! In opposite, we argue that we should give more attention to its detailed investigation. But as a matter of fact, we should not narrow the computational semantics research to this theory.

As we know the limitations of this theory, we can draw ideas from other semantics theories to overcome them. Maybe it is time to shake off the dust from abandoned semantics theories and revise their ideas since feasibly the forgotten evening/morning star is the one that leads us to clarity.

References

1. **Abnar, S., Ahmed, R., Mijnheer, M., Zuidema, W. (2018).** Experiential, distributional and dependency-based word embeddings have complementary roles in decoding brain activity. Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018), pp. 57–66.
2. **Acs, J., Kornai, A. (2016).** Evaluating embeddings on dictionary-based similarity. Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, pp. 78–82.
3. **Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., Soroa, A. (2009).** A study on similarity and relatedness using distributional and wordnet-based approaches. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 19–27.
4. **Almuhareb, A. (2006).** Attributes in lexical acquisition. Ph.D. thesis, University of Essex.
5. **Amatuni, A., He, E., Bergelson, E. (2018).** Preserved structure across vector space representations. arXiv preprint arXiv:1802.00840.
6. **Antoniak, M., Mimno, D. (2018).** Evaluating the stability of embedding-based word similarities. Transactions of the Association for Computational Linguistics, Vol. 6, pp. 107–119.
7. **Artemova, E., Bakarov, A., Artemov, A., Burnaev, E., Sharaev, M. (2020).** Data-driven models and computational tools for neurolinguistics: a language technology perspective. Journal of Cognitive Science, Vol. 21, No. 1, pp. 15–52.
8. **Asr, F. T., Zinkov, R., Jones, M. (2018).** Querying word embeddings for similarity and relatedness. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1, pp. 675–684.
9. **Auguste, J., Rey, A., Favre, B. (2017).** Evaluation of word embeddings against cognitive processes: primed reaction times in lexical decision and naming tasks. Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, pp. 21–26.
10. **Bakarov, A. (2018).** Can eye movement data be used as ground truth for word embeddings evaluation? arXiv preprint arXiv:1804.08749.
11. **Bakarov, A. (2020).** Did you just assume my vector? detecting gender stereotypes in word embeddings. International Conference on Analysis of Images, Social Networks and Texts, Springer, pp. 3–10.
12. **Baker, S., Reichart, R., Korhonen, A. (2014).** An unsupervised model for instance level subcategorization acquisition. EMNLP, pp. 278–289.
13. **Baroni, M., Dinu, G., Kruszewski, G. (2014).** Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 238–247.
14. **Baroni, M., Evert, S., Lenci, A. (2008).** ESSLLI 2008 workshop on distributional lexical semantics.

Hamburg, Germany: Association for Logic, Language and Information.

15. **Baroni, M., Lenci, A. (2010).** Distributional memory: A general framework for corpus-based semantics. Computational Linguistics, Vol. 36, No. 4, pp. 673–721.
16. **Baroni, M., Lenci, A. (2011).** How we blessed distributional semantic evaluation. Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, Association for Computational Linguistics, pp. 1–10.
17. **Baroni, M., Murphy, B., Barbu, E., Poesio, M. (2010).** Strudel: A corpus-based semantic model based on properties and types. Cognitive science, Vol. 34, No. 2, pp. 222–254.
18. **Batchkarov, M., Kober, T., Reffin, J., Weeds, J., Weir, D. (2016).** A critique of word similarity as a method for evaluating distributional semantic models. Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, Association for Computational Linguistics, pp. 7–12.
19. **Betti, A., Reynaert, M., Ossenkoppele, T., Oortwijn, Y., Salway, A., Bloem, J. (2020).** Expert concept-modeling ground truth construction for word embeddings evaluation in concept-focused domains. Proceedings of the 28th International Conference on Computational Linguistics, pp. 6690–6702.
20. **Blair, P., Merhav, Y., Barry, J. (2016).** Automated generation of multilingual clusters for the evaluation of distributed representations. arXiv preprint arXiv:1611.01547.
21. **Bloem, J., Fokkens, A., Herbelot, A. (2019).** Evaluating the consistency of word embeddings from small data. Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pp. 132–141.
22. **Bloem, J., Parisi, M. C., Reynaert, M., Oortwijn, Y., Betti, A. (2020).** Distributional semantics for neo-latin. Proceedings of LT4HALA 2020-1st Workshop on Language Technologies for Historical and Ancient Languages, pp. 84–93.
23. **Bloomfield, L. (1914).** An introduction to the study of language. H. Holt.
24. **Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2016).** Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.
25. **Boleda, G., Erk, K. (2015).** Distributional semantic features as semantic primitives - or not. 2015 AAAI Spring Symposium Series.
26. **Boleda, G., Herbelot, A. (2016).** Formal distributional semantics: Introduction to the special issue. Computational Linguistics, Vol. 42, No. 4, pp. 619–635.
27. **Bruni, E., Tran, N. K., Baroni, M. (2014).** Multimodal distributional semantics. J. Artif. Intell. Res.(JAIR), Vol. 49, No. 2014, pp. 1–47.
28. Bybee, J. L., Perkins, R. D., Pagliuca, W., others **(1994).** The evolution of grammar: Tense, aspect, and modality in the languages of the world, volume 196. University of Chicago Press Chicago.
29. **Camacho-Collados, J., Navigli, R. (2016).** Find the word that does not belong: A framework for an intrinsic evaluation of word vector representations. ACL Workshop on Evaluating Vector Space Representations for NLP, pp. 43–50.
30. **Camacho-Collados, J., Pilehvar, M. T., Collier, N., Navigli, R. (2017).** Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval 2017). Vancouver, Canada.
31. **Casacuberta, S., Halevy, K., Blasi, D. E. (2021).** Evaluating word embeddings with categorical modularity. arXiv preprint arXiv:2106.00877.
32. **Che, X., Ring, N., Raschkowski, W., Yang, H., Meinel, C. (2017).** Traversal-free word vector evaluation in analogy space. Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, pp. 11–15.
33. **Chiu, B., Korhonen, A., Pyysalo, S. (2016).** Intrinsic evaluation of word vectors fails to predict extrinsic performance. Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 1–6.
34. **Coecke, B., Sadrzadeh, M., Clark, S. (2010).** Mathematical foundations for a compositional distributed model of meaning. Lambek Festschirft, Linguistic Analysis, Vol. 36.
35. **Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P. (2011).** Natural

language processing (almost) from scratch. Journal of Machine Learning Research, Vol. 12, No. Aug, pp. 2493–2537.

36. **Cop, U., Dirix, N., Drieghe, D., Duyck, W. (2017).** Presenting geco: An eyetracking corpus of monolingual and bilingual sentence reading. Behavior research methods, Vol. 49, No. 2, pp. 602–615.
37. **Cristofaro, S. (2010).** Semantic maps and mental representation. Linguistic Discovery, Vol. 8, No. 1.
38. **Croft, W. (2002).** Typology and universals. Cambridge University Press.
39. **Croft, W., Poole, K. T. (2008).** Inferring universals from grammatical variation: Multidimensional scaling for typological analysis.
40. **Cruse, A. (2010).** Meaning in language: An introduction to semantics and pragmatics.
41. **Cysouw, M., Haspelmath, M., Malchukov, A. (2010).** Introduction to the special issue "semantic maps: Methods and applications". Linguistic Discovery, Vol. 8, No. 1, pp. 1–3.
42. **Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018).** Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
43. **Duncan, K. J., Pattamadilok, C., Knierim, I., Devlin, J. T. (2009).** Consistency and variability in functional localisers. Neuroimage, Vol. 46, No. 4, pp. 1018–1026.
44. **Erk, K. (2016).** What do you know about an alligator when you know the company it keeps? Semantics and Pragmatics, Vol. 9, pp. 17–1.
45. **Ettinger, A., Feldman, N. H., Resnik, P., Phillips, C. (2016).** Modeling N400 amplitude using vector space models of word representation. Proceedings of the 38th annual conference of the Cognitive Science Society, pp. 1445–1450.
46. **Ettinger, A., Linzen, T. (2016).** Evaluating vector space models using human semantic priming results. Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 72–77.
47. **Faruqui, M., Tsvetkov, Y., Rastogi, P., Dyer, C. (2016).** Problems with evaluation of word embeddings using word similarity tasks. arXiv preprint arXiv:1605.02276.
48. **Fellbaum, C. (2010).** Wordnet. In Theory and applications of ontology: computer applications. Springer, pp. 231–243.
49. **Ferretti, T. R., McRae, K., Hatherell, A. (2001).** Integrating verbs, situation schemas, and thematic role concepts. Journal of Memory and Language, Vol. 44, No. 4, pp. 516–547.
50. **Fillmore, C. J. (1976).** Frame semantics and the nature of language. Annals of the New York Academy of Sciences, Vol. 280, No. 1, pp. 20–32.
51. **Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E. (2001).** Placing search in context: The concept revisited. Proceedings of the 10th international conference on World Wide Web, ACM, pp. 406–414.
52. **Finley, G., Farmer, S., Pakhomov, S. (2017).** What analogies reveal about word vectors and their compositionality. Proceedings of the 6th joint conference on lexical and computational semantics (* SEM 2017), pp. 1–11.
53. **Firth, J. R. (1957).** A synopsis of linguistic theory, 1930-1955. Studies in linguistic analysis.
54. **Friedman, H. H., Amoo, T. (1999).** Rating the rating scales. Friedman, Hershey H. and Amoo, Taiwo (1999)." Rating the Rating Scales." Journal of Marketing Management, Winter, pp. 114–123.
55. **Fujinuma, Y., Boyd-Graber, J., Paul, M. (2019).** A resource-free evaluation metric for cross-lingual word embeddings based on graph modularity. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4952–4962.
56. **Gabrilovich, E., Markovitch, S. (2007).** Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJcAI, volume 7, pp. 1606–1611.
57. **Gao, B., Bian, J., Liu, T.-Y. (2014).** Wordrep: A benchmark for research on learning word representations. arXiv preprint arXiv:1407.1640.
58. **Geeraerts, D. (2010).** Theories of lexical semantics. Oxford University Press.
59. **Gerz, D., Vulić, I., Hill, F., Reichart, R., Korhonen, A. (2016).** Simverb-3500: A large-scale evaluation set of verb similarity. arXiv preprint arXiv:1608.00869.

60. **Gittens, A., Achlioptas, D., Mahoney, M. W. (2017).** Skip-gram- zipf+ uniform= vector additivity. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 69–76.

61. **Gladkova, A., Drozd, A. (2016).** Intrinsic evaluations of word embeddings: What can we do better? Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, pp. 36–42.

62. **Gladkova, A., Drozd, A., Matsuoka, S. (2016).** Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. Proceedings of the NAACL Student Research Workshop, pp. 8–15.

63. **Glynn, D., Fischer, K. (2010).** Quantitative methods in cognitive semantics: Corpus-driven approaches, volume 46. Walter de Gruyter.

64. **Goldsmith, J. A. (2005).** The legacy of zellig harris: Language and information into the 21st century, vol. 1: Philosophy of science, syntax and semantics. Language, Vol. 81, No. 3, pp. 719–736.

65. **Greenberg, C., Demberg, V., Sayeed, A. (2015).** Verb polysemy and frequency effects in thematic fit modeling. Proceedings of the 6th Workshop on Cognitive Modeling and Computational Linguistics. Association for Computational Linguistics, Denver, Colorado, pp. 48–57.

66. **Grefenstette, E., Sadrzadeh, M. (2011).** Experimental support for a categorical compositional distributional model of meaning. Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 1394–1404.

67. **Gurnani, N. (2017).** Hypothesis testing based intrinsic evaluation of word embeddings. arXiv preprint arXiv:1709.00831.

68. **Gutiérrez, E. D., Levy, R., Bergen, B. (2016).** Finding non-arbitrary form-meaning systematicity using string-metric learning for kernel regression. ACL (1).

69. **Halawi, G., Dror, G., Gabrilovich, E., Koren, Y. (2012).** Large-scale learning of word relatedness with constraints. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 1406–1414.

70. **Hanke, M., Baumgartner, F. J., Ibe, P., Kaule, F. R., Pollmann, S., Speck, O., Zinke, W., Stadler, J. (2014).** A high-resolution 7-tesla fmri dataset from complex natural stimulation with an audio movie. Scientific data, Vol. 1, pp. 140003.

71. **Harris, Z. S. (1954).** Distributional structure. Word, Vol. 10, No. 2-3, pp. 146–162.

72. **Hashimoto, T. B., Alvarez-Melis, D., Jaakkola, T. S. (2016).** Word embeddings as metric recovery in semantic spaces. Transactions of the Association for Computational Linguistics, Vol. 4, pp. 273–286.

73. **Haspelmath, M. (2003).** The geometry of grammatical meaning: Semantic maps and cross-linguistic comparison. In The new psychology of language. Psychology Press, pp. 217–248.

74. **Hearst, M. (1998).** Wordnet: An electronic lexical database and some of its applications.

75. **Hellrich, J., Hahn, U. (2016).** Bad company - neighborhoods in neural embedding spaces considered harmful. Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers, pp. 2785–2796.

76. **Herdağdelen, A., Erk, K., Baroni, M. (2009).** Measuring semantic relatedness with vector space models and random walks. Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, Association for Computational Linguistics, pp. 50–53.

77. **Hill, F., Reichart, R., Korhonen, A. (2016).** Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics.

78. **Hollenstein, N., Van der Lek, A., Zhang, C. (2020).** Cognival in action: An interface for customizable cognitive word embedding evaluation. Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations, pp. 34–40.

79. **Hutchison, K. A., Balota, D. A., Neely, J. H., Cortese, M. J., Cohen-Shikora, E. R., Tse, C.-S., Yap, M. J., Bengson, J. J., Niemeyer, D., Buchanan, E. (2013).** The semantic priming project. Behavior Research Methods, Vol. 45, No. 4, pp. 1099–1114.

80. **Huth, A. G., de Heer, W. A., Griffiths, T. L., Theunissen, F. E., Gallant, J. L. (2016).** Natural speech reveals the semantic maps that tile human cerebral cortex. Nature, Vol. 532, No. 7600, pp. 453–458.

81. **Jackendoff, R. (1976).** Toward an explanatory semantic representation. Linguistic inquiry, Vol. 7, No. 1, pp. 89–150.

82. **Jarmasz, M., Szpakowicz, S. (2004).** Roget's thesaurus and semantic similarity. Recent Advances in Natural Language Processing III: Selected Papers from RANLP, Vol. 2003, pp. 111.

83. **Jastrzebski, S., Leśniak, D., Czarnecki, W. M. (2017).** How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. arXiv preprint arXiv:1702.02170.

84. **Jones, M. N., Kintsch, W., Mewhort, D. J. (2006).** High-dimensional semantic space accounts of priming. Journal of memory and language, Vol. 55, No. 4, pp. 534–552.

85. **Joseph, K., Carley, K. M. (2016).** Relating semantic similarity and semantic association to how humans label other people. Proceedings of the First Workshop on NLP and Computational Social Science, pp. 1–10.

86. **Jurgens, D. A., Turney, P. D., Mohammad, S. M., Holyoak, K. J. (2012).** Semeval-2012 task 2: Measuring degrees of relational similarity. Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, Association for Computational Linguistics, pp. 356–364.

87. **Kiela, D., Hill, F., Clark, S. (2015).** Specializing word embeddings for similarity or relatedness. EMNLP, pp. 2044–2048.

88. **Koptjevskaja-Tamm, M., Rakhilina, E., Vanhove, M. (2015).** The semantics of lexical typology. The Routledge Handbook of Semantics, pp. 434.

89. **Kornai, A., Kracht, M. (2015).** Lexical semantics and model theory: Together at last? Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015), pp. 51–61.

90. **Krebs, A., Paperno, D. (2016).** Capturing discriminative attributes in a distributional space: Task proposal. ACL 2016, pp. 51.

91. **Kutuzov, A. (2017).** ArbitrAriness of linguistic sign questioned: correlation between word form and meaning in russian. Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii, Vol. 1, No. 16 (23), pp. 109–120.

92. **Landauer, T. K., Dumais, S. T. (1997).** A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, Vol. 104, No. 2, pp. 211.

93. **Langacker, R. W. (1987).** Foundations of cognitive grammar: Theoretical prerequisites, volume 1. Stanford university press.

94. **Lapesa, G., Evert, S. (2013).** Evaluating neighbor rank and distance measures as predictors of semantic priming. Proceedings of the ACL Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2013), pp. 66–74.

95. **Lehmann, C. (1990).** Towards lexical typology. Studies in typology and diachrony: Papers presented to Joseph H. Greenberg on his 75th birthday, volume 161, pp. 185.

96. **Lenci, A. (2008).** Distributional semantics in linguistic and cognitive research. Italian journal of linguistics, Vol. 20, No. 1, pp. 1–31.

97. **Levy, O., Goldberg, Y. (2014).** Linguistic regularities in sparse and explicit word representations. Proceedings of the eighteenth conference on computational natural language learning, pp. 171–180.

98. **Liza, F. F., Grzes, M. (2016).** An improved crowdsourcing based evaluation technique for word embedding methods. ACL 2016, pp. 55.

99. **Luke, S. G., Christianson, K. (2017).** The provo corpus: A large eye-tracking corpus with predictability norms. Behavior Research Methods, pp. 1–8.

100. **Luong, T., Socher, R., Manning, C. D. (2013).** Better word representations with recursive neural networks for morphology. CoNLL, pp. 104–113.

101. **Mandera, P., Keuleers, E., Brysbaert, M. (2017).** Explaining human performance in psycholinguistic tasks with models of semantic similarity based on

prediction and counting: A review and empirical validation. Journal of Memory and Language, Vol. 92, pp. 57–78.

102. **McDonald, S., Brew, C. (2004).** A distributional model of semantic context effects in lexical processing. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 17.

103. **McRae, K., Ferretti, Liane Amyote, T. R. (1997).** Thematic roles as verb-specific concepts. Language and cognitive processes, Vol. 12, No. 2-3, pp. 137–176.

104. **McRae, K., Spivey-Knowlton, M. J., Tanenhaus, M. K. (1998).** Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. Journal of Memory and Language, Vol. 38, No. 3, pp. 283–312.

105. **Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013).** Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

106. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, pp. 3111–3119.

107. **Mikolov, T., Yih, W.-t., Zweig, G. (2013).** Linguistic regularities in continuous space word representations. hlt-Naacl, volume 13, pp. 746–751.

108. **Milajevs, D., Griffiths, S. (2016).** A proposal for linguistic similarity datasets based on commonality lists. arXiv preprint arXiv:1605.04553.

109. **Miller, G. A., Charles, W. G. (1991).** Contextual correlates of semantic similarity. Language and cognitive processes, Vol. 6, No. 1, pp. 1–28.

110. **Nissim, M., van Noord, R., van der Goot, R. (2019).** Fair is better than sensational: Man is to doctor as woman is to doctor. arXiv preprint arXiv:1905.09866.

111. **Nooralahzadeh, F., Øvrelid, L., Lønning, J. T. (2018).** Evaluation of domain-specific word embeddings using knowledge resources. Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).

112. **Osgood, C. E., Suci, G. J., Tannenbaum, P. H. (1978).** The measurement of meaning. 1957. Urbana: University of Illinois Press.

113. **Padó, S., Lapata, M. (2007).** Dependency-based construction of semantic space models. Computational Linguistics, Vol. 33, No. 2, pp. 161–199.

114. **Padó, U. (2007).** The integration of syntax and semantic plausibility in a wide-coverage model of human sentence processing.

115. Panchenko, A., others **(2013).** Similarity measures for semantic relation extraction. Ph.D. thesis, PhD thesis, Université catholique de Louvain & Bauman Moscow State Technical University.

116. **Parviz, M., Johnson, M., Johnson, B., Brock, J. (2011).** Using language models and latent semantic analysis to characterise the N400m neural response. Proceedings of the Australasian Language Technology Association Workshop 2011, pp. 38–46.

117. **Peinelt, N., Liakata, M., Nguyen, D. (2019).** Aiming beyond the obvious: Identifying non-obvious cases in semantic similarity datasets. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2792–2798.

118. **Pennington, J., Socher, R., Manning, C. (2014).** Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

119. **Pereira, F., Gershman, S., Ritter, S., Botvinick, M. (2016).** A comparative evaluation of off-the-shelf distributed semantic representations for modelling behavioural data. Cognitive neuropsychology, Vol. 33, No. 3-4, pp. 175–190.

120. **Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018).** Deep contextualized word representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237.

121. **Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019).** Language models are unsupervised multitask learners. OpenAI Blog, Vol. 1, pp. 8.

122. Radinsky, K., Agichtein, E., Gabrilovich, E., Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. Proceedings of the 20th international conference on World wide web, ACM, pp. 337–346.

123. Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.

124. Rubenstein, H., Goodenough, J. B. (1965). Contextual correlates of synonymy. Communications of the ACM, Vol. 8, No. 10, pp. 627–633.

125. Sahlgren, M. (2008). The distributional hypothesis. Italian Journal of Disability Studies, Vol. 20, pp. 33–53.

126. Salicchi, L., Lenci, A., Chersoni, E. (2021). Looking for a role for word embeddings in eye-tracking features prediction: Does semantic similarity help? Proceedings of the 14th International Conference on Computational Semantics (IWCS), pp. 87–92.

127. Sayeed, A., Greenberg, C., Demberg, V. (2016). Thematic fit evaluation: an aspect of selectional preferences. ACL 2016, pp. 99.

128. Schnabel, T., Labutov, I., Mimno, D. M., Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. EMNLP, pp. 298–307.

129. Schwartz, D., Mitchell, T. (2019). Understanding language-elicited EEG data by predicting it from a fine-tuned language model. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 43–57.

130. Søgaard, A. (2016). Evaluating word embeddings with fMRI and eye-tracking. ACL 2016, pp. 116.

131. Szymanski, T. (2017). Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: short papers), pp. 448–453.

132. Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G., Dyer, C. (2015). Evaluation of word vector representations by subspace alignment. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2049–2054.

133. Turian, J., Ratinov, L., Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics, pp. 384–394.

134. Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. Machine Learning: ECML 2001, pp. 491–502.

135. Turney, P. D. (2008). The latent relation mapping engine: Algorithm and experiments. Journal of Artificial Intelligence Research, Vol. 33, pp. 615–655.

136. Turney, P. D., Littman, M. L., Bigham, J., Shnayder, V. (2003). Combining independent modules to solve multiple-choice synonym and analogy problems. arXiv preprint cs/0309035.

137. Turney, P. D., Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research, Vol. 37, pp. 141–188.

138. Utsumi, A. (2020). Exploring what is encoded in distributional word vectors: A neurobiologically motivated analysis. Cognitive Science, Vol. 44, No. 6, pp. e12844.

139. Vandekerckhove, B., Sandra, D., Daelemans, W. (2009). A robust and extensible exemplar-based model of thematic fit. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 826–834.

140. Vylomova, E., Rimell, L., Cohn, T., Baldwin, T. (2015). Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. arXiv preprint arXiv:1509.01692.

141. Wälchli, B. (2010). Similarity semantics and building probabilistic semantic maps from parallel texts. Linguistic Discovery, Vol. 8, No. 1, pp. 331–371.

142. Wijnholds, G., Sadrzadeh, M. (2019). A typedriven vector semantics for ellipsis with anaphora using lambek calculus with limited contraction. arXiv preprint arXiv:1905.01647.

143. Wittgenstein, L. (2010). Philosophical investigations. John Wiley & Sons.

144. Yang, D., Powers, D. M. (2006). Verb similarity on the taxonomy of wordnet. The Third International WordNet Conference: GWC 2006, Masaryk University.

Article received on 07/04/2022; accepted on 20/07/2022.
Corresponding author is Amir Bakarov.

ISSN 2007-9737

Semi-Automatic Alignment of Multilingual Parts of Speech Tagsets

S. Yashothara, R. T. Uthayasanker, G. V. Dias, S. Jayasena

University of Moratuwa,
Department of Computer Science and Engineering,
Sri Lanka

{yashoshan,rtuthaya,gihan,sanath}@cse.mrt.ac.lk

Abstract. We cast the problem of mapping a pair of Parts of Speech (POS) tagsets as a labelled tree mapping problem and present a general-purpose semi-automatic POS tree alignment algorithm to solve the alignment. This algorithm can be used to align two POS tagsets of different languages or the same language. We evaluate its usefulness using POS tagsets of two languages: Tamil and Sinhala. The proposed approach shows that manual effort in prior approaches is drastically reduced due to the proposed algorithm and eliminates the need to create new POS tagsets.

Keywords. Parts of speech, POS tagset mapping, POS tagset alignment, semi-automatic approach, BIS tagset, UOM tagset, Tamil NLP, Sinhala NLP.

1 Introduction

Parts of Speech (POS) is a category in which a word is assigned conforming to its morpho-syntactic functions [1]. The process of assigning the POS label to words in a given text is an important aspect of natural language processing. The initial task of any POS tagging process is choosing various POS tags that are word classes such as nouns, verbs, adjectives, etc., in a language.

The importance of POS tagging has led various researchers to work independently in developing POS tags for a language. It limited the ability to reuse tagged corpus among NLP researchers in the same language. Subsequently, there have been efforts to standardize POS tagset for a language [3]. While standardizing POS tagset for a given language, researchers also found the importance of standardizing POS tagsets for similar languages [4]. A multilingual POS agreement facilitates cross-language compatibility between different languages and ensures that common parts of different languages are tagged alike [5]. Yet, most of the tagsets capture features of a particular language, and it is not easy to tag data in other languages. The imbalance in tagsets obstructs the interoperability and reusability of tagged corpora. Furthermore, it limited the ability to reuse tagged corpus among NLP researchers in low resource languages with data shortages, especially tagged data.

POS agreement between multiple languages is useful because: (1) reusability of annotated corpora, (2) interoperability across different languages, (3) capture more detailed morphological and syntactic features of these languages, (4) achieve cross-linguistic compatibility among different languages corpora, (5) make sure the common category in different languages is tagged the same way, (6) useful for building and evaluating unsupervised cross-lingual taggers, and (7) development of multilingual corpora [4]. The POS Agreement for Multilingualism can be used for machine translation, parsing, named entity recognition, coreference resolution, sentimental analysis, question answering, and code-mixing [4]. However, alignment is still challenging due to the cost of multi-language experts, time-consuming and manual effort.

Prior efforts at the POS Agreement focused on developing a framework for standardizing POS tagsets for a given language family and mapping from different tagsets to universal sets. Despite the standardization of POS tagsets, researchers developed new and evolving tagsets by in-depth

consideration of morpho-phrasal features [6]. Therefore, aligning the already generated POS tagsets is necessary. There are some approaches to map existing tagsets to a universal tagset [1]. However, no attempt has been made to align within a language or between language tags. This paper focuses on a novel approach called 'POS tagset alignment of different languages'.

Further, it is the ever semi-automatic alignment of POS tagsets. POS alignment is the process of determining correspondences between tagsets between two languages P_1 and P_2, without creating a new tagset. POS alignment can be done in three ways: (1) equal alignment, (2) subset alignment, and (3) complex alignment. It can be useful to integrate multiple POS tagsets. POS alignment is better than POS standardization as it covers better granularity and no new tagset.

In this study, we chose Tamil and Sinhala languages, which gain importance since both languages are acknowledged as official languages in Sri Lanka. Furthermore, these efforts are gaining more reputation as these two languages are considered low resource languages. Sinhala language belongs to the Indo Aryan language family, and Tamil language belongs to the Dravidian family.

As two languages that have been associated for a long time, they share striking similarities in morphology and syntax. It makes sense for the alignment of tagsets that can utilize this similarity to facilitate mapping tagsets to each other.

Therefore, in this research, BIS tagset was selected for the Tamil language as it is the standard tagset for the Indian language. University of Moratuwa (UOM) tagset was chosen for the Sinhala language as it covers the most morpho-syntactic features. We derived a POS alignment between those tagsets using a semi-automatic approach. Semi-automated alignment was a better approach that simplified the challenges of alignment.

2 Related Work

Previous efforts at the POS Agreement focused primarily on developing a framework for POS tagset standardization of a language group and using the POS standardization guidelines to create a new standardized tagset or map from various tree-bank tagsets to a universal set.

2.1 Existing Approaches on POS Standardization

NLP researchers around the world focus on several POS standardization efforts. EAGLES guidelines [5] resulted from such an initial blow to create common standards across languages. EAGLES Guidelines provide analytical information about text language, especially for identifying morpho-phrases and syntax related to computer linguistics. In this approach, they did not create a new standardized tagset using their guidelines. It became the foundation for several other kinds of research [4, 7, 8, 9] in leveraging morpho-syntactic and syntactic features to develop common standards across multiple languages.

The LE-PAROLE project [7] established a multilingual corpus for fourteen European languages, an Extended morpho-syntactically annotated, language-specific set of features according to a common basic PAROLE tagset. MULTEXT [8] focused on multilingual tools, integration, and linguistic features, with extensions in other languages.

Still, this project also mostly focuses on European languages to make the standardization among them. However, a spin-off MULTEXT-EAST [9] gradually added morpho-syntactic descriptions of sixteen languages, including Persian or Uralic languages. The MULTEXT-EAST dataset embodies the EAGLES-based morpho-syntactic specifications, morpho-syntactic lexicons, and annotated multilingual corpora.

One of the earliest works on Indian language standardization was by Baskerville et al. in designing a common POS tagset for eight languages. Hierarchical and decomposable tagsets were used in the framework as it is a recognized method for creating a common tagset framework for multiple languages [4].

The BIS has released the Unified Parts of Speech (POS) Standard in Indian languages considering the morphologic, syntactic features of Indian languages. The top-level is subdivided into the next two levels [3]. Nitish Chandra et al. claimed that the tagset for which taggers perform best should be the standard tagset [10]. Unlike

prior efforts, designing a new common framework was not the focus of Nitish Chandra et al. [10].

POS standardization focuses on designing a common tagset framework that can exploit similarity. Mapping from the existing tagset to the standardized tagset is not considered in the above approaches. Nevertheless, there are some on mapping from different tree-bank tagsets to the universal tagset.

2.2 Existing Approaches to Mapping from Different Tree-Bank Tagsets to Universal Set

Instead of standardizing morpho-syntactic tagging, there are some efforts of mapping existing tagsets to universal tagset, which they created. A Universal Parts-of-Speech Tagset was proposed by McDonald et al. The tagset consists of twelve universal parts-of-speech categories. In addition to the tagset, they evolved a mapping from 25 different tree-bank tagsets to this universal set. As a result, this universal tagset and mapping generate a dataset consisting of common parts-of-speech for 22 different languages. When corpora with a common tagset are inaccessible, they manually define a mapping from the language or the tree bank-specific fine-grained tagset to the universal tagset [1].

Zeman and Resnik worked on Interset Project, which was used in cross-language parser adaptation [11]. In this approach, a tagset of a language is converted into the universal tagset using an encoding algorithm implemented in the support library. The above project serves as an intermediate step on the way from tagset A to tagset B. They covered twenty tagsets in ten languages.

Zeman and Resnik claim that their approach is different from McDonald et al. McDonald et al. did not need to be studied in-depth, as they removed much of the language-specific information, except for the basic parts of speech that are universally found. On the contrary, Interset eliminates as little as possible because they keep what they find anywhere. Direct conversion from one language to another language did not focus on this approach.

An international collaborative project called the "Universal Dependencies project" proposes a scheme for the treebank annotation, suitable for a wide variety of languages and assists cross-linguistic study [12]. The universal annotation guidelines were built on Google Universal Part of Speech tagset. Forty languages are covered in the current version 1.3. But in this approach also, they didn't focus on the direct conversion from one language to another language.

The majority of researchers focused on mapping several tagsets to a universal tagset using the guidelines developed. Despite the standards, researchers kept introducing tagsets, which posed key challenges for standardization using universal tagset. As POS tagsets become widely used, there is a growing need to align tagset between multiple languages and the need to align multiple tagsets to one tagset [15].

3 Background

We briefly introduce the Parts of speech tagset alignment problem in this section by adapting the knowledge from ontology and schema alignment. In the ontology alignment also, researchers matched entities to determine an alignment between different ontologies.

Nevertheless, since the direct mapping of the same labelled tagsets is impossible in all POS tagset alignment cases, this is a more challenging problem than ontology alignment. Most ontology alignment approaches are semiautomatic as they couldn't receive the best output using an automatic process. So in this paper also, the focus is based on a semi-automatic process.

The POS tagset alignment problem is to find a set of correspondences between two languages' tagsets P_1 and P_2. Because tagsets can be modeled as trees, the problem is often cast as a matching problem between such trees. A tagset tree, P, is defined as P= (V, E), where V is the set of labelled vertices representing the tags and E is the set of edges representing the relations, which is a set of ordered 2-subsets of V.

Definition 1 (Alignment, correspondence Maα). Given two tagsets P_1 and P_2, an alignment between P_1 and P_2 is a set of correspondences: $(x_a, , y_\alpha, , r)$ with $x_a \in P_1$ and $y_\alpha \in P_2$ being the two matched entities, r being a relationship holding between x_a and y_α, in this correspondence:

$M_{a\alpha}$: { $x_{a,}$, $y_{\alpha,}$, r},
x_a : { x^1_a , x^2_a , ….., x^s_a },
y_α : { y^1_α , y^2_α , ……., y^t_α },
r {=, ⊆ , ⊇, … }.

Each assignment variable $M_{a\alpha}$, M is the confidence between the alignment of two languages, and x_a is the tag from one language, and y_α is the tag from another language. Here P_1 language has 's' no of tags and P_2 language has 't' no of tags. Many possible relationships are held between x_a and $y_{\alpha,}$ but they mostly fall into equal and subsumption relationships.

An equal relationship means one language tagset can equally align with another language tagset. Sometimes a POS tag in one language may not be mapped directly to another language POS tag. This mostly occurs when a number of aspects used in the specialization of a POS tag differ between languages.

For example, the Sinhala language does not have animate/ inanimate verb categories, but Tamil does. It is also possible that a POS tag in one language does not occur in another language. In this case, we will not be able to map the POS tag at all. Every language has some specific features. But we need to map these kinds of tags as well. If we cannot find an exact match for a tag, abstract level tagsets can be aligned through the adaptation knowledge of EAGLES guidelines.

4 Approach

To agree on multiple language POS tagset, researchers adopted various strategies as discussed above. Some derived a new tagset capturing the morpho-syntactic features of some specific languages (Bureau of Indian Standard), and some mapped existing POS tagsets to a universal POS tagset. However, both approaches introduce a new POS tagset.

Unlike these prior approaches, we took an entirely new angle. We cast the problem of heterogeneity in POS tagsets as an alignment of two labelled trees and proposed a novel semi-automatic approach algorithm to solve. We evaluated our algorithm using a representative POS tagset chosen from Sinhala and Tamil languages. We chose these language pairs because (1) accessibility of the data and expertise, (2) they are low resourced languages, and (3) they are official languages in Sri Lanka, where we conduct the research.

Below, the rationales behind choosing the representative tagset from each language are described. Then, a semi-automatic POS alignment algorithm is presented.

4.1 Tagset Selection

As there are several tagsets available in each language, selections of a proper POS tagset is essential for this study. While choosing a tagset of a language, usability and standardization are considered. The following subsections describe the identified POS tagsets of Sinhala and Tamil and how the proper tagset is selected to align.

Sinhala Tagsets. There are two tagsets available for the Sinhala language, such as the University of Colombo School of Computing (UCSC) tagset, developed by University of Colombo [16] and the UOM tagset by University of Moratuwa [17]. UCSC tagset contains 29 tags. There are three versions in the UCSC tagset. UOM tagset is an extended version of the UCSC tagset by overcoming the following issues: (1) uncovered word classes in the UCSC tagset (2) multiple words the 'any' category (out of the 100,000 words in the manually POS tagged corpus, 3989 words do not fall into any category), (3) inconsistent tagging, and (4) unfocus on inflection based grammatical variations [17].

There are three levels in this tagset, following a hierarchical structure. Altogether, they came up with 148 tags. Level I contains the primary top-level part of speech. Level II tagset is generated by adding inflected forms to Level I. Level II tagset is consisted of thirty tags [17]. UOM tagset is selected for this study because of the above mentioned significant limitations in the UCSC tagset. Table 1 shows the selected UOM tagset at the second level.

Tamil Tagsets. For the Tamil language, there are plenty of tagsets. We considered nine tagsets [3, 6, 10, 18, 19, 20, 21, 22, 23, 24] before choosing an appropriate one for this study. Bureau of Indian Standards (BIS) is recommended as a common tagset for POS annotation of Indian languages. Many tags in BIS are same as LDC-IL tagset. It

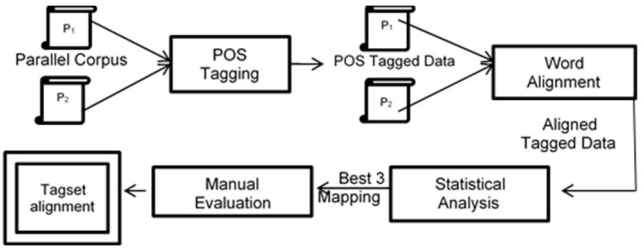


Fig. 1. Workflow of the semi-automatic POS tagsets alignment of P_1 and P_2 languages

groups unknown, Punctuation and residual into one tag. It has 11 tags in level I and 32 tags in Level II tags.

Level II is made by further subdividing the level I tags [3].

We chose BIS Tamil Tagset since it is the officially accepted standard tagset for Tamil language.

In our approach, the third level of both language tagsets is not considered. The third level captures inflection based grammatical variations of the language. We choose to omit Level III for the following reasons: (1) no apparent impact in most applications, (2) the deeper levels are inflectional forms than being POS classes, (3) more time for tagging, and (4) a large number of tags will lead to more complexity, which reduces the tagging accuracy [19].

4.2 Semi-Automatic Algorithm for POS Tagset Alignment

We proposed a semi-automatic approach for the tagsets alignments. Figure 1 describes the workflow of the semi-automatic POS tagsets alignment. The proposed semi-automatic approach requires parallel corpus. Therefore, the parallel corpus of languages P_1 & P_2 were annotated using respective automatic POS taggers.

Then the tagged parallel corpora were word-aligned using a word alignment tool. Then, the top three maps for each POS tag were selected based on the word order and presented to the human evaluators. The experts pruned the provided mappings and arrived at a final quality and complete alignment. Below we present every workflow step and tool used for this approach in a descriptive manner.

We have access to the Sinhala-Tamil parallel corpus of government official documents, which contains approximately 40,000 words. The parallel corpus was manually cleaned and aligned by three professional translators. Then, the parallel corpus was annotated using the automatic POS tagger of both languages. We used an automatic POS tagger developed by Dhanalakshmi et al. for the Tamil language as it gave higher accuracy among all available taggers. Likewise, we used an automatic POS tagger [17] based on SVM from the University of Moratuwa to annotate the Sinhala corpus.

Once the annotation was done for both the sides of the parallel corpus, parallel text was word-aligned using a word alignment tool. This study used GIZA++ [25] as a word alignment tool, giving our dataset higher accuracy. GIZA++ can perform word alignments in two directions for each pair of languages by considering one language as the source and another as the target. The intersection of both directions is taken as the resulting alignment [25].

Based on the word alignment, we retrieved the best-aligned words for the given words. It resulted in any tag of one language can be mapped to any tag of the other. There are 35 tags from the BIS tagset and 30 tags from the UOM tagset in our study. Therefore, there can be 30*35 (1050) possible alignments of tags. Further, to refine this alignment, statistical values of this mapping was considered. The highest three mappings were considered as the possible aligned tags.

The highest three mappings were derived using an automatic program by counting words belonging to each mapping. The general idea is to consider all the tag alignments of both languages generated from the GIZA++ algorithm and choose the most frequent of them as the correct alignment.

Table 1. Alignment of BIS tagset and UOM tagset

UOM Tags	BIS Tags	Example		
Common Noun	Common Noun/Echo words	மரம்	ගස	Tree
Adjectival Noun		பாடசாலை,	පාසල්	School
Case marker	Common/proper	க்கு ,உடைய	ට, ගේ	to, 's
Proper noun	Proper noun	ஜான்	ජොන්	John
Pronoun/Deterministic Pronoun	Personal Pronoun	நான், நீ	මම, ඔබ	I, you
Pronoun	Reflexive Pronoun	தான்	-	Myself
	Reciprocal Pronoun	ஒருவருக்கொருவர,அவனவன்	එක එක්කෙනාට,ඔවුනොවුන්	each other
Questioning Pronouns	Question words	என்ன, எப்படி	කුමක්ද, කෙසේද	what, how
Question-Based Pronouns	Relative Pronoun	எங்கே, எது	කොහේ, කවර	where, which
Determiners	Deictic	இவன், இவள்	මේ, සියලු	this, all
	Relative	அவ்வீடு, இவ்வீடு	ඒ ගෙදර, මේ ගෙදර	That home, this home
Verbal Participle	Verbal participle	பார்த்து	බලා	Looked
Verb finite	Verb finite	செய்தான்	කළේය	Did (he)
Preposition in compound verb		-	ඉටු, සිදු	-
Nouns in Compound Verb		படிக்கின்றான்	පාඩම් කරනවා	Study
Adjective in Compound Verbs		கூட்டபடுகின்றது	වැඩි කරනවා	Increasing
Nipathana		போதும், காணாது	ඇති, මදි	Enough/ not enough
Modal auxiliary	Verb auxiliary	முடியும், வேண்டும்	හැකි, යුතු	Can, should
Verb Non-Finite	Infinite Verb	விழ	වැටීමට වගේ	like to fall
	Conditional Verb	நடந்தால்	ඇවිද්දොත්	If walk
Verbal Noun	Verbal Gerund	படித்தல்	ඉගෙනීම	Studying
	Verbal noun	படிப்பு	-	Study
Adverb	Adverb	விரைவாக	වේගයෙන්	Fast
Adjective	Adjective	மிருதுவாக	සුමුදු	Smooth
	Relative Participle	நடந்த	ඇවිද	Walked (kid)
Conjunction	Coordinator	உம், மற்றும்	හෝ, සහ	Or, and
	Subordinator	என்று, என	යනු, යැයි	That
Particle	Default Particles	மட்டும், கூட	ය, ද, ම	Only, also
	Classifier	அட்டும்	-	-
	Intensifier	அதி, வேக, மிக	ඉතා	Most, speed
	Negation	இல்லை	නෑ, නැත	No
Interjection	Interjection	ஐயோ	අයියෝ	Oh
Postposition	Postposition	பற்றி,குறித்து	ගැන	Related
Number	Cardinal	ஒன்று, 1	එක, 1	One, 1
	Ordinal	முதல், இரண்டாம்	පළමුවන, දෙවන	First, second
Punctuation/Full stop	Punctuation	/?,:"	/?,:"	/?,:"
	Symbol	$, &,*,(	$, &,*,(	$, &,*,(
Foreign word	Foreign Residuals	கார்	කාරය	Car
Abbreviation	Unknown	மு.ப	පෙ.ව	a.m

Nevertheless, in our approach, we chose the top three frequent aligned tags and cross-checked them with bilingual experts to finalize the alignments. For example, "Nipathana" in UOM tags aligned with "Verb Finite" and "Common noun" mostly in BIS tagset. From the linguistic point of view, it does have to align with "Verb finite".

5 Results and Discussion

Through the experiment, some possible relationships are held between the BIS tagset and the UOM tagset. We reported identified four types of relationships with examples. After the manual inspection, table 1 shows the POS tagset alignment between the BIS tagset and the UOM tagset.

Two linguistics did a manual review to avoid bias. There are eight equal relationships, 22 subsumption relationships, one complex relationship and no non mapped relationships.

5.1 Equal Relationship

Some POS alignments hold an equal relationship. An equal relationship implies one language tagset can equally align with the tagset in another language. As mentioned in Table 1, some POS alignments fall under the equal relationship. The adverb in the Tamil language is directly mapped to the Sinhala language adverb node.

Modal auxiliary in UOM tagset and Verbal auxiliary in BIS tagset are equally aligned. Verbal participle, Common noun, Postpositions, Foreign words and Punctuation in both languages are fallen in an equal relationship as it has the same features. Questioning pronouns words are used to ask a question. Therefore, that is equivalently aligned with question words in the BIS tagset.

5.2 Subsumption Relationship

In most cases, a POS tag in the Sinhala language is not mapped directly to the Tamil language POS tag. Most of those tags fall under the subsumption relationship. Nipathana is a category in the Sinhala language but does not have a direct mapping tag in the Tamil language. Therefore, Nipathana has to map with the finite verb category in the Tamil language (subsumption ⊆ ⊇). A conjunction is specialized into subordinator and coordinator in the Tamil language. So these two subcategories are aligned to parent node conjunction in Sinhala language (subsumption ⊆ Relationship). It often happens when some of the features used to specialise a POS tag vary between languages.

BIS tagset does have five categories of pronouns, while there are only four categories in the UOM tagset. As a result, we are not able to equally align those tags.

The Personal, Reflexive and Reciprocal pronouns from the BIS tagset are subsumption aligned with the Pronoun tag in the UOM tagset.

Deterministic pronouns in the UOM tagset are aligned to personal pronouns in the BIS tagset. Furthermore, the category of personal pronouns can contain other words except for deterministic pronouns.

Question-based pronouns are used to show the uncertainty of a noun/noun phrase of interest. So it aligns with the Relative pronoun in the BIS tagset. But Relative pronouns can contain other words than question-based pronouns.

E.g.: *I don't know who did this.*

இதை யார் செய்தது என்று எனக்கு தெரியாது.

මෙය කළේ කවුදැයි මම නොදනිමි.

There are two types of demonstrative in the BIS tagset, while the UOM tagset has only one category. The subcategories Deictic and Relative are aligned to the Determiners tag. Particles are further divided into five subcategories in the BIS tagset, while only a parent node Particles are in the UOM tagset.

Hence, the subcategories are mapped to Particles in the UOM tagset using a subsumption relationship. General, ordinal and cardinal are the three categories of Quantifiers in the BIS tagset. Yet, the UOM tagset only have a Number category. Thus, three subcategories are aligned with the Number category.

Full stop in the UOM tagset does have a subsumption relationship with Punctuation in the BIS tagset. Like that, Symbol in the BIS tagset is aligned with the Punctuation category of the UOM tagset. As BIS tagset do not have a proper tag for Abbreviation in UOM tagset, it takes the subsumption relationship with Unknown tag. Echo

words in the BIS tagset are aligned to the Common noun in the UOM tagset.

A noun in Compound Verb is another category of noun in the Sinhala language. It is a combination of nouns and verbs. The noun, which makes a compound verb, is called as a noun in the compound verb. There is no matching translation in English and Tamil since all compound verbs in the Sinhala language is normal verb in English and Tamil. In this example, the first part of the verb is identified as 'Noun in the compound verb'. Therefore, this 'Noun in Compound verb' tag is subsumption mapped with the Finite verb tag of the BIS tagset.

E.g. එයා පාඩම් කරනවා.

He is studying.

அவன் படிக்கிறான்.

The adjectival noun is a common noun that acts as an adjective to describe another noun. When a common noun is used as an adjectival noun, it always takes the base, plural form of the common noun. For example, in a noun phrase like 'පාසල් වත්ත (school garden)', 'පාසල් (school)' is an adjectival noun that describes the main common noun 'වත්ත (garden)'. However, according to the Tamil grammar rule, if a noun expresses another noun, it cannot be categorized under the adjective category. So that 'Adjectival noun' is mapped with the common noun in the BIS tagset.

Further, adjectives are categorized into three subcategories Adjective, Adjectival Noun, and Adjective in Compound Verbs. As we saw above, the Adjectival Noun tag is aligned to the Common noun tag.

The adjective in Compound Verb is a combination of Adjective + Verb. The first word in such compound verbs will be tagged as an adjective in compound verbs. In the example' වැඩි කරනවා (increase)', වැඩි is an adjective and කරනවා is a verb. However, Tamil, we can write this as 'கூட்டபடுகின்றது'.

Hence, Tamil has no matching translation for the adjective in the compound verb since all compound verbs in Sinhala are normal in Tamil. Thus, 'Adjective in the Compound verb' is mapped with the Finite verb tag of the BIS tagset. The remaining subcategory, 'Adjective,' is aligned to the adjective in the BIS tagset.

Non-finite and finite verb forms often constitute mixed categories from the syntactic point of view. The syntactic properties of participles overlap with adjectives. Relative participle from verb category in BIS tagset also maps with the adjective in UOM tagset. Similarly, gerunds and verbal nouns BIS tagset are aligned to Verbal nouns in the UOM tagset. However, they retain their verbal arguments. Usually, these words are tagged as forms of verbs. Likewise, infinite verb and conditional verb in the BIS tagset align to the non-finite verb category in the UOM tagset.

Some other categories in UOM tagset also fall under the Verb category of the BIS tagset. Similar to 'Adjective in Compound Verb', 'Preposition in the compound verb' is one of the categories in the UOM tagset, which does not have a meaning by them but, when combined with another verb, make up a compound verb. In the example' ඉටු කරයි (does)', ඉටු is a preposition and කරයි is a verb. However, Tamil, we can write this as 'செய்கிறார்'. Hence, Tamil has no matching translation for the preposition in the compound verb, since all compound verbs in Sinhala are normal in Tamil. Thus, 'Preposition in the Compound verb' is mapped with the Finite verb tag of the BIS tagset.

Nipathana is a tag in the UOM tagset, which is used alone in some contexts and as a postposition. However, Tamil language does not have an exact match for this category. This category is mapped with the Finite verb tag by considering the usability of this category:

E.g., ඇති (*Enough*) - போதும்,

නැති (*not having*) – கிடையாது.

5.3 Complex Relationship

Some features in the POS tagset are unique to the particular language. Those features may map to another category or categories when it comes to alignment. There are some complex alignments when we try to map POS tagsets of the Sinhala and Tamil languages. Hence, we went deep into the grammar of both languages to find out the relationship for those categories.

Sinhala and Tamil nouns are morphologically inflected based on the case. The suffix is attached to the nouns to indicate the case. According to Sinhala language rules, detaching these case marking suffixes from the main noun is incorrect.

However, some Sinhala writers tend to separate this case marking suffix from the main noun. Therefore, unlike the Tamil language, the Sinhala language has space between the noun and its case marker. Subsequently, a new POS tag was added, "Case marker" in Sinhala but not Tamil. The case marker does not have an English meaning on its own. According to the previous tagset alignment in the Sinhala language, this tagset must align with a common noun or proper noun. Therefore, this alignment falls into the composite relationship.

For example, nominative form of ගස - gasa "the tree" can be inflected as ගසට - gasata "to the tree". ගසට - gasata can be written as ගසට - gasata or ගස ට - gasa ta. In the second case ට - ta has to be tagged as a case marker.

However, in the Tamil language, it will be "மரத்துக்கு" and tagged under the common noun category. This correspondence is fallen into the composite relationship.

POS alignment depicts the grammar of the language to a certain level. In addition, it is a good starting point for the study of language divergence.

6 Conclusion and Future Works

We showed that heterogeneity in POS tagsets can be cast into the labelled tree alignment problem. We presented a generic language-independent semi-automatic algorithm to align POS tagsets to provide high-quality alignment. Manual effort and time are reduced compared to previous approaches by this algorithm.

We have presented a quality alignment between the Sinhala UOM tagset and the Tamil BIS tagset. Even though these two languages have been in contact for an extended period, the grammars are not identical and have a significant difference. We listed numerous examples from real tagsets of Tamil and Sinhala languages to illustrate the most difficult parts of tagsets alignment. Each mapping includes the top three maps obtained using an automated word counting program to present the layout. However, in our approach, even though we choose the top three frequent mappings, all alignments fall within the top two mappings.

The solutions we propose follow the ultimate goal of minimizing information loss and creating a new tagset. This approach is language-independent, and we could apply for the different tagsets, which belong to a language. POS alignment is used to study the similarity and dissimilarity of grammar quantitatively. In addition, it is a good starting point for the study of language divergence.

In the future, we plan to extend this study for different tagsets, which either belong to a different language or the same language.

Acknowledgment

This project was partly supported by a Senate Research Committee (SRC) Grant funds awarded by the University of Moratuwa and funds from the Department of Official Languages of Sri Lanka.

References

1. **McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Tackstrom, O., et al. (2013).** Universal dependency annotation for multilingual parsing. 51st Annual Meeting of the Association for Computational Linguistics (ACL), Vol. 2, pp. 92–97.
2. **Hardie, A. (2004).** The computational analysis of morphosyntactic categories in Urdu. PhD thesis Lancaster University.
3. **Bureau Indian Standard. (n.d.).** Unified parts of speech (POS) standard in Indian languages. Ministry of Communications & Information Technology Govt. of India.
4. **Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Jha, G. N., Rajendran, S., Saravanan, K., Sobha, L. (2008).** Designing a common POS-Tagset framework for Indian languages. 6th Workshop on Asian Language Resources, pp. 89–92.

5. **Leech, G., Wilson, A. (1996).** Recommendations for the morphosyntactic annotation of corpora: EAGLES document EAG-TCWG-MAC/R. Expert Advisor Group on Language Engineering Standard.
6. **Selvam, M., Natarajan, A. M. (2009).** Improvement of rule based morphological analysis and POS tagging in Tamil language via projection and induction techniques. International journal of computers, Vol. 3, No. 4, pp. 357–367.
7. **Suzanne, N., Volz, L. (1996).** Multilingual corpus tagset specifications. MLAP PAROLE 63œ386 WP 4.4.
8. **Ide, N., Véronis, J. (1994).** Multext: Multilingual text tools and corpora. 15th International Conference on Computational Linguistics, Vol. 1, pp. 588–592. DOI: 10.3115/991886.991990.
9. **Erjavec, T. (2004).** Multext-east version 3: Multilingual morphosyntactic specifications, lexicons and corpora. 4th International Conference on Language Resources and Evaluation, pp. 1535–1538.
10. **Chandra, N., Kumawat, S., Srivastava, V. (2014).** Various tagsets for Indian languages and their performance in part of speech tagging. 5th IRF International Conference.
11. **Zeman, D. (2008).** Reusable tagset conversion using tagset drivers. Sixth International Conference on Language Resources and Evaluation (LREC'08), pp. 28–30.
12. **Sulubacak, U., Gokirmak, M., Tyers, F., Coltekin, C., Nivre, J., Eryigit, G. (2016**). Universal dependencies for Turkish. 26th International Conference on Computational Linguistics: Technical Papers, pp. 3444–3454.
13. **De Marneffe, M. C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J. M., Manning, C. D. (2014).** Universal Stanford dependencies: a cross-linguistic typology. 9th International Conference on Language Resources and Evaluation (LREC), pp. 4585–4592.
14. **Tsarfaty, R. (2013).** A unified morpho-syntactic scheme of Stanford dependencies. 51st Annual Meeting of the Association for Computational Linguistics, Vol. 2, pp. 578–584.
15. **Doan, A., Halevy, A. Y. (2005).** Semantic-Integration Research in the Database Community A Brief Survey. AI Magazine, Vol. 26, No. 1. DOI: 10.1609/aimag. v26i1.1801.
16. **Gunasekara, D., Welgama, W. V., Weerasinghe, A. R. (2016**). Hybrid part of speech tagger for Sinhala language. 16th International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 041–048. DOI: 10.1109/ICTER.2016.7829897.
17. **Fernando, S., Ranathunga, S., Jayasena, S., Dias, G. (2016).** Comprehensive part-of-speech tagset and SVM Based POS tagger for Sinhala. 6th Workshop on South and Southeast Asian Natural Language Processing, pp. 173–182.
18. **Central Institute of Indian Languages. (n.d.).** Ministry of Education Government of India. http:///www.ciil.org/.
19. **Dhanalakshmi, V., Padmavathy, P., Anan-Kumar, M., Soman, K. P., Rajendran, S. (2009).** Chunker for Tamil. International Conference on Advances in Recent Technologies in Communication and Computing, pp. 436–438. DOI: 10.1109/ ARTCom.2009.191.
20. **Dandapat, S. (2010).** MSRI Part-of-Speech Annotation Interface.
21. **Lakshmana, P. S., Geetha, T. (2008).** Morpheme based language model for Tamil part-of-speech tagging. Polibits, No. 38, pp. 19–25.
22. **Ramanathan, M., Chidambaram, V., Patro, A. (n.d).** An attempt at multilingual POS tagging for Tamil. Available from http://pages.cs. wisc.edu/~madhurm/CS769_final_report.pdf.
23. **Shiva IIIT (2018).** A Parts-of-Speech tagset for Indian languages. http://shiva.iiit.ac.in/ SPSAL2007/iiit_tagset_guidelines.
24. **Ramasamy, L., Žabokrtský, Z. (2014).** Tamil dependency treebank v0.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL).
25. **Och, F. J. (2003).** Minimum error rate training in statistical machine translation. 41st Annual Meeting on Association for Computational Linguistics, pp. 160–167. DOI: 10.3115/1075096.1075117.

26. Zeman, D. (2010). Hard problems of tagset conversion. 2nd International Conference on Global Interoperability for Language Resources, pp. 181–185.

27. Leech, G. (1997). Grammatical tagging. **Garside, Leech and McEnery, eds.**, Corpus Annotation: Linguistic Information from Computer Text Corpora. Longman.

28. Petrov, S., Das, D., McDonald, R. (2011). A universal part-of-speech tagset. arXiv:1104.2086v1. DOI: 10.48550/arXiv.1104.2086.

29. Zeman D. (2004). Parsing with a statistical dependency model. PhD thesis, Univerzita Karlova v Praze.

Article received on 12/09/2018; accepted on 07/01/2021. Corresponding author is S. Yashothara.

ISSN 2007-9737

Machine Translation for Low-Resource English-Mizo Pair Encountering Tonal Words

Vanlalmuansangi Khenglawt[1], Sahinur Rahman Laskar[2], Partha Pakray[2], Riyanka Manna[3], Ajoy Kumar Khan[1]

[1] Mizoram University,
Department of Computer Engineering,
India

[2] National Institute of Technology,
Department of Computer Science and Engineering,
India

[3] Gandhi Institute of Technology and Management,
Department of Computer Science and Engineering,
India

mzut208@mzu.edu.in, {sahinur_rs, partha}@cse.nits.ac.in,
{riyankamanna16, ajoyiitg}@gmail.com

Abstract. Machine translation is one of the most powerful natural language processing applications for preserving and upgrading low-resource language. Mizo language is considered as low-resource since there is limited availability of resources. Therefore, it is a challenging task for English-Mizo language pair translation. Moreover, Mizo is a tonal language, where a word can express different meanings depending on a variety of tones. There are four variations of tones, namely high, low, rising, and falling. A tone marker is used to represent each of the tones, which is added to the vowels to indicate tone variation. Addressing tonal words in machine translation for such a low-resource pair is another challenging issue. In this paper, the English-Mizo corpus is developed where parallel sentences having tonal words are incorporated. The different machine translation models are explored based on statistical machine translation and neural machine translation for the baseline systems. Furthermore, the proposed approach attempts to augment the train data by expanding parallel data having tonal words and achieves state-of-the-art results for both forward and backward translations encountering tonal words.

Keywords. English-Mizo, machine translation, low-resource, tonal.

1 Introduction

Language is a method of communication for individuals of varied cultures everywhere in the world. The language barrier prevents communication between different cultures. Machine translation (MT) commonly uses to address the problem and serves as a bridge for language barriers among people of divergent linguistic backgrounds.

In MT, one natural or human spoken language translates into another natural or human spoken language. Natural language is of three categories based on the availability of resources. The categories include high, medium, and low-resource. The resources comprise works of native speakers, online data, and computational resources.

The resource-poor languages classify into the low-resource category that has restricted online resources [25, 32]. Moreover, a low-resource language pair is considered based on the minimal amount of data required for training a model [9].

The proper definition of low-resource language pair puts forward a challenging research question itself. However, if the training data is under 1 million parallel sentences, it is considered a low-resource language pair [12]. The native speakers play a vital role in different aspects of the language, including the quality and quantity of the data.

Most of the world languages are recognized under the low-resource category based on the availability of resources. The MT works are limited in India's north-eastern region, and the languages considered as low-resource languages include Assamese, Boro, Manipuri, Khasi, Kokborok, and Mizo.

1.1 Low-Resource Pair: English–Mizo

The Mizo[1] language belongs to the Sino-Tibetan family of languages. It is spoken natively by the Mizo people (also known as Lushai) in the Mizoram state of India and Chin State in Burma. Mizoram is one of the states of India, situated in the northeastern parts of the country.

It shares borders with three states in northeast India: Tripura, Assam, and Manipur. Additionally, the state also shares a border with two of the neighbouring countries: Myanmar and Bangladesh. The name Mizoram comes from the words *"Mi"*, which means people, *"Zo"*, which means hill, and *"Ram"*, which means land.

Thus, the word Mizoram implies a 'hilly people's land' [27]. It holds the second least populated state with a population of 830,846 according to the 2011 Census of India[2]. The Mizo language [24, 27] is mainly based on the *Lusei* dialect and many words are also derived from its surrounding Mizo sub-tribes and sub-clan.

The writing system of the Mizo language is based on the Roman script. The Mizo alphabet has 25 letters including 3 letters with a combination of two letters represented as one letter: *AW, CH, NG*. Among the alphabets there are six vowels which are *A, AW, E, I, O, U*. A circumflex ∧ was subsequently added to the vowels to demonstrate long vowels, which were inadequate to completely express Mizo tone. A vowel is a syllabic language unit pronounced with no stricture within the vocal tract. Each of the vowels has its meaning by itself, and they represent the tone of a word. All the other alphabets are a consonant which has no meaning by itself but can be merged to form a syllable with a vowel.

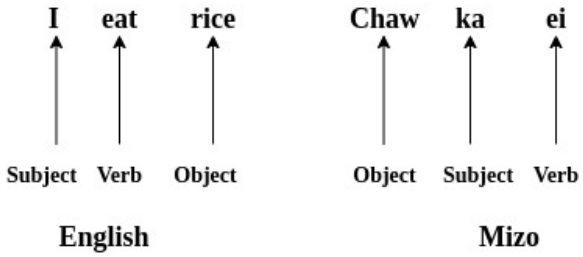


Fig. 1. Example of English–Mizo word order

A consonant is a speech tone that is articulated with a complete or partial closure of the vocal tract in articulatory phonetics. Unlike English, Mizo is a tonal language, where the lexical meaning of words is influenced by the pitch of a syllable. The structure and the word order of the Mizo Language are also different from the English language, the declarative word order of Mizo is OSV (object-subject-verb).

Fig. 1 presents an example of a Mizo sentence with its translation in English. In contrast to English, all proper names have a gender suffix in Mizo, a letter *'i'* is added at the end of every female proper name and a letter *'a'* is added at the end of every male proper name to distinguish the gender.

In terms of pronoun, there is no distinction between gender in Mizo while there is a clear distinction of gender when using the pronoun in English. However, Mizo uses the same number system like English. The English–Mizo can be considered as a low-resource pair based on the limited availability of resources, namely, parallel corpus and monolingual data of Mizo.

1.2 Motivation

Beneath every language, there is a culture involved. A language is defined by the people living in the area, their origin, traditions, custom, cuisine, and many more.

[1] https://en.wikipedia.org/wiki/Mizo_language

[2] https://www.censusindia.gov.in/2011Census/Language_MTs.html

Table 1. Variation of tone (a) in Mizo

Type	Tone (a)
High tone	á
Low tone	à
Rising tone	ă
Falling tone	â

Therefore, a language not only means for communication, but defines the people using them. There are many languages across the world which are extinct. It can be due to rapid change in the advancement of different technologies where there is a requirement for a dominant language like English language.

The reason can also be negligence by the native people, where their language is given less priority. It can get easily extinct when the language is not properly passed on to the younger generations. Spoken languages without written form are more likely to get extinct.

However, it is also endangered as a low-resource language where a minority of the population uses the language. As the language becomes extinct, the culture dies along with it. Therefore, the preservation of language from extinction is highly necessary, especially for low-resource languages.

With Mizo language being a low-resource language, it is imperative for preservation. Machine translation is capable of preservation of low resource language as it breaks language barriers. Since English language is considered to be the most dominant language, English-Mizo machine translation can enhance the limitations of the Mizo language in today's digital world.

Therefore, a low-resource language like the Mizo language has a chance for survival and is capable of encountering technological advances with Machine Translation. There are very limited machine translation works on English–Mizo pair [30, 15], that lags in encountering tonal words of Mizo. Apart from this, automatic translations like Google and Bing cover 109 and 70 languages across the globe, but the Mizo language lags in.

This is due to the lack of standard corpus. In this paper, we have considered machine translation work of English–Mizo pair by encountering challenges of Mizo tonal words. From the best of our knowledge, no prior work available that encounters Mizo tonal words in such low-resource English–Mizo pair translation. The contributions of this work are as follows:

— Detailed survey of linguistic challenges in English–Mizo machine translation.

— Created EnMzCorp1.0:English–Mizo corpus.

— Evaluated baseline systems for low-resource English–Mizo pair, encountering tonal words through different machine translation models.

— Proposed approach investigates with data augmentation technique and achieved state-of-the-art results for English–Mizo pair translation.

— Analysis is reported for inspecting errors on predicted translation.

2 Challenges of English–Mizo Machine Translation

Translation of a language is not a simple task. There are several challenges to be dealt with when translating one language to another. Like many other languages, the Mizo language deals with several challenges. This section has surveyed linguistic challenges.

2.1 Tonal Words

A language is treated as a tonal language when its tone influences the meaning of the word. Mizo language is undoubtedly a tonal language, which can lead to certain challenges for machine translation. Variation in tones and contour tones can alter the meaning of particular words.

The type of pitch used is capable of automatically determining the grammatical forms of that specific word. Many linguists have concluded the Mizo language to be of four tones, while some conclude it to be more than four tones by considering two ways of vowel sound: long vowel and short vowel.

Table 2. Example of different meaning of the word *Buk* in Mizo

Mizo Word	Tone	English Meaning	Mizo	English
buk	High tone *(búk)*	Hut	Kan ramah **búk** sak ka duh.	I want to build hut in our land.
	Low tone *(bùk)*	Bushy	He Ui hian mei a nei **bùk** hle mai.	This dog has a bushy tail.
	Rising tone *(bŭk)*	Unstable	He dawhkan hi a **bŭk** ania.	This table is unstable.
	Falling tone *(bûk)*	Weight	Khawngaihin heng hi min lo **bûk** sak teh.	Please weight this for me.
lei	High tone *(léi)*	Tongue	Doctor in ka **léi** chhuah turin min ti.	The doctor asked me to stick out my tongue.
	Low tone *(lèi)*	Soil	Thlai chí tuh nan **lèi** an chŏ.	They dig up the ground to plant seeds.
	Rising tone *(lěi)*	Buy	Thil ka **lěi**.	I am buying something.
awm	High tone *(áwm)*	To be present	Vawiin seminar ah a **áwm** m?	Is he present today at the seminar?
	Low tone *(àwm)*	To look after/stay	Ka naute chu kan nauàwmtu in a **àwm**	My baby is look after by our nanny.
	Rising tone *(ăwm)*	Chest	A **ăwm** nat avangin doctor hnenah a inentir.	She went to the doctor complaining of chest pains.
	Falling tone *(âwm)*	Probably/likely	Inneihna ah a kal a **âwm** viau ani.	It is very likely that he will go to the wedding.

However, the Mizo tone framework accepts four tones: High (H), Low (L), Rising (R), and Falling (F) [6]. The tones are also named in Mizo as *'Ri sang'*, *'Ri hniam'*, *'Ri lawn'* and *'Ri kuai'* respectively. Linguist had created a tone-marker for each of the tone to indicate the tone variation in the Mizo Language, which are listed in the following Table 1.

The four different tones used in Mizo words can indicate different meanings in the English word, as shown in Table 2. For example, the Mizo word 'buk' can indicate different meanings in English words like 'bushy', 'weight', 'hut/camp', 'unstable', which is to determine based on the tone used. The Mizo is undeniably a tonal language where a change in tone will completely alter a word's meaning.

However, in the writing system, the indication of tonal words is neglected and not correctly considered. Most of the writings in Mizo use only circumflex $\wedge$ for indication of tone. Furthermore, it is also an understudied language with a limited resource in terms of tones. Based on the four tones applicable to the five vowels (a, e, i, o, u), we have identified $4 \times 5 = 20$ possible types that exist in Mizo.

2.2 Tonal-Polysemy Words

Mizo language is also rich in polysemy words where the intonation is the same, yet its meaning is different. Polysemy is a side of linguistics ambiguity that considerations the multiplicity of word meanings. Table 3 presents examples of tonal-polysemy words in Mizo. It is a simple fact of common parlance, and people gleefully interpret correct results without conscious effort.

However, polysemy is largely impervious to any generalized natural language processing task. As tonal languages go, the Mizo language is one of the most complicated languages. It is a tonal language where not only a particular word has several tones, but also it is a language in which the pitch of the word defines the meaning. However, polysemy is the association of a word with at least two distinct purposes. Since polysemy words have the same tone, the pitch of the word alone cannot define the word. Therefore, a complete understanding of the nearby word or understanding the whole sentence's context is necessary. A few polysemy words in the Mizo language can also act as both noun and verb. For example:

— Engzat nge **mikhual** in thlen ? **(Noun)**
I lo zin hunah ka **mikhual** ang che **(Verb)**

— Ruah a sur dawn sia, **púk** ah hian awm mai ang u **(Noun)**
I pawisa ka lo **púk** ang e, I phal em? **(Verb)**.

Moreover, the few extraordinary words can change their tone depending on the phrase used but still have the same meaning. For example:

— **lěi - Buy** (Raising tone)

  – Thil ka **lěi** → I am buying something. (Raising tone)
  – Khawiah nge I **lêi?** → where did you buy? (sound as falling tone)

— **áng – will** (High tone)

Table 3. Example of tonal-polysemy words in Mizo

Tonal-Ploysemy	Tone	English Meaning	Mizo	English
ăng	Rising tone (ă)	To open the mouth	I ka **ăng** rawh le.	Open your mouth.
		Talk angrily	Kha kha ti suh a tia, a **ăng** vak a.	”Don’t do that!” she shouted angrily.
búl	High tone (ú)	Beginning	A **búl** atangin lehkha kha chhiar rawh.	Read the paper from the beginning.
		Stump	Kawtah sawn thing **búl** a awm.	There is a tree stump at the courtyard.
		Near	Helai **búl** velah hian thingpui dawr a awm hnai m?	Is there a restaurant nearby?

- Ka ti vek **áng** ⟶ I will do everything. (High tone)
- Chhang hi ka zai sak **àng** che ⟶ I will cut this cake for you. (sound as low tone)

— **Chhûm – boil** (Falling tone)

- Naute tui I pek dăwn chuan I **chhûm** so phawt dawn nia ⟶ If you give water to a small baby, you have to boil it first. (falling tone)
- I **chhúm** zawhah gas off rawh ⟶ Off the gas after you boil. (sound as high tone).

2.3 Symbolic Words

Apart from the tone, a few symbols are used in the writings of the Mizo sentence. In many places, – (hyphen) is found, used for continuing English (non-Mizo) word with Mizo word to appear as one word. It is used after figures.

Another famous symbol is *’n*, which is used after the noun to show possession with the noun. It works as putting (apostrophe) in the English sentence. Table 4 demonstrates symbolic words in Mizo. Moreover, there are a few words that are significant with having affix words.

For instance, *‘ah’* is an affix word that is a preposition (can be used as: at, on, upon, in, into) depending on the sentence. Combining the same word and the affixed word to produce one syllable of a linguistic unit may lead to a different meaning but an entirely correct Mizo word. For example:

— *Ru-ah* (steal) ⟶ *Ruah* (Rain)

— *Chi-ah* (salt) ⟶ *Chiah* (Dip).

We have tackled the above challenges in two ways. First, we have extracted Mizo tonal and symbolic words from the monolingual corpus of Mizo. Then, manually translated into corresponding English words.

Secondly, Mizo tonal sentences are extracted from monolingual data. Then, the best-trained baseline model (Mizo to English) is applied to generate pseudo-English sentences.

To improve the Mizo tonal word’s translation quality, we have augmented the parallel train data by injecting more tonal word information. The data statistics and proposed approach are described in Sect. 5 and 7.

3 Machine Translation

Machine translation removes human intervention from a translation of one natural language to another using automatic translation, thereby resolving linguistically ambiguous problems. It is divided into two broad categories: rule-based and corpus-based approaches. The knowledge-driven approach is another name for a rule-based approach based on the linguistic information of the language.

The rule-based translation system is built using a set of grammatical rules and linguistic experts. Although the rule based methods have reasonable translation accuracy, it requires a considerable amount of time and effort to pre-design a set of translation rules and the languages’ grammatical

Table 4. Example of symbolic words in Mizo

	Hyphen	'n
Symbolic Words	8,307-in	worker-te'n
	database-ah	Lalruatkima'n
	district-a	20-te'n
	police-te	hnathawktute'n

structures. The corpus-based approach is also known as the data-driven approach.

The corpus-based approach can self-learn using bilingual corpora that require a considerable volume of bilingual content in both the source and target languages.

The corpus-based approach acquires translation information using these parallel data. There has been a significant change in the translation method from rule-based to corpus-based.

Since relying on parallel sentences is more practical than complex grammatical rules with linguistic experts and knowledge in NLP techniques. Example-based Machine translation (EBMT), statistical machine translation (SMT), and neural machine translation (NMT) are the three methods of corpus-based machine translation.

The EBMT requires a parallel corpus, and the central concept is text similarity. It identifies the approximately matching sentences (i.e., examples) using a point-to-point mapping and similarity measures such as word, syntactic, or semantic similarity. The retrieval module and the adaptation module are the two modules that make up the translation method.

For a given input sentence, the retrieval module finds identical parallel sentences from the corpus.

The adaptation module determines the parts of translation to be reused from the retrieval module.

The relevant match concerning the source language is used in case it does not match. The two most common corpus-based MT are SMT and NMT, which are described in the following subsections.

3.1 SMT

In the corpus-based approach, the main drawback of EBMT is that in real-time scenarios, we can not cover various types of sentences by examples only. To encounter this issue, statistical machine translation (SMT) is introduced [14, 13].

In this approach, a statistical model in which the parameters are computed from bilingual corpus analysis. The translation problem is reformulated using a mathematical reasoning problem. In SMT, there are different forms of translation: word based translation, phrase based translation, syntax based translation, and hierarchical phrase-based translation.

Out of which phrase-based translation is the most widely used. Before NMT, phrase-based SMT achieves a state-of-the-art approach. SMT consists of three modules: translation model (TM), language model (LM), and decoder. Consider the translation task of English to Mizo, where the best Mizo translation (m_{best}) for the source English sentence (e) is formulated using Eq. 1:

$$m_{best} = \arg\max_{m} P(m \mid e). \tag{1}$$

To estimate $P(m \mid e)$ for the given source-target sentences, the probability distribution of all possible target sentences is required, which is achieved by understanding what makes a good translation. Any good translation should possess two aspects: adequacy and fluency.

The target sentence should keep the same meaning as the source sentence, which is known as adequacy, and the target sentence should be fluent. Both adequacy and fluency factors must be balanced to yield a good translation. This can be formulated based on Bayes Theorem by the extension of Eq.1 as shown in Eq.2:

$$\begin{aligned} m_{\text{best}} &= \arg\max_{m} \ \text{adequacy}(e \mid m) \times \text{fluency}(m), \\ &= \arg\max_{m} P(e \mid m) \times P(m). \end{aligned} \tag{2}$$

In the SMT, TM and LM are used to compute $P(e \mid m)$ and $P(m)$. The decoder is responsible for $\arg\max_m$ to search for the best translation. The TM model collects phrase pairs from parallel

data and then used to estimate the probable target words/phrases as shown in the Eq.3:

$$P(e \mid m) = \frac{\text{count}(e,\ m)}{\sum_e \text{count}(e,\ m)}. \tag{3}$$

The LM reorders the obtained target words/phrases from TM to predict syntactically correct target sentences for ensuring fluency of translation.

The LM is estimated from monolingual target data, where the target sentence is modelled by the conditional probability of each word given the previous words in the sentence. This modelling is also known as n-gram LM. Lastly, the decoder utilizes a beam search strategy to find out the best possible translation. The abstract pictorial representation of SMT is shown in Fig. 2.

3.2 NMT

In the MT task, the NMT approach attains state-of-the-art for both high and low resource pair translations [1, 30, 29, 18, 21]. NMT can learn the model in an end-to-end manner by mapping the source and target sentence.

The main problem with SMT is that SMT creates a model context by considering a set of phrases of limited size. As the phrase size increases, the data sparsity will reduce the quality.

Likewise, feed-forward based NMT calculates the phrase pairs score by considering the length of the fixed phrases. But in real-time translation, the phrase length of both source and target are not fixed. Therefore, recurrent neural networks (RNN) based NMT [5, 4] is introduced to tackle variable-length phrases.

RNN can process each word in a sentence of arbitrary length via continuous space representations. These representations can assist the long-distance relationship among words in a sentence. Also, RNN updates and maintains a memory known as a state during the processing of each word.

The Eq. 4 represents probability of a sentence S and $S_1, S_2, ..S_n$ denotes a sequence of n words. The RNN based LM [26] can be represented by considering the Equation 5, where next word S_{t+1} is predicted for the given current word S_t and previous words $S_1....S_{t-1}$:

$$\begin{aligned} P(S) &= P(S_1, S_2....Sn), \\ &= \prod_{t=1}^{n} P(S_t \mid S_1....S_{t-1}), \end{aligned} \tag{4}$$

$$\begin{aligned} S_{t+1} &= \arg\max_{t+1}(P(S_{t+1} \mid S_1....S_t)), \qquad (5) \\ &= \arg\max_{t+1}(p_t), \\ p_t &= \text{softmax}(y_t), \qquad (6) \\ y_t &= h_t^1 W_0, \qquad (7) \\ h_t^1 &= \tanh([h_t^0; h_{t-1}^1]W_h), \qquad (8) \\ h_t^0 &= S_t E. \qquad (9) \end{aligned}$$

The RNN based LM processes each word in a sentence at every time step t to predict the next word. Here, consider the vocabulary size and all hidden layers as V and M, respectively. The current word S_t is transformed into continuous space representation via indexing into the embedding matrix E provides h_t^0.

The embedding S_t having vector size V equivalent to the vocabulary size, where indexing is performed through one-hot vector representation. In one-hot vector representation, a "1" indicates the current word's index position, and for all other positions is denoted by a "0".

This helps to create the embedding through the multiplication of the one-hot vector having size $1 \times V$ with the embedding matrix of size $V \times M$. The RNN based LM maintains memory using a hidden state, h_{t-1}^1 called as the previous state. When the first word encounters in the sequence, the previous state is set to all zeros' vector.

The previous state generates the concatenated vector and the embedding h_t^0 and then multiply with the matrix W_h having size $2 \times M \times M$ followed by the $tanh$ non-linear function. As a result, the hidden state h_t^1 is obtained at the current timestamp t, and the current hidden state represents the previous state for the next consecutive word in the sequence.

The obtained h_t^1 is used for mapping to a vector y_t having size N via multiplication with the matrix

W_0. Then, softmax function is used to transform the vector y_t (also knows as logits) into probability values which provides the vector p_t.

The predicted next word is the optimum probability value corresponding to the index position. This process of predicting S_{t+1} given S_t is required to update the neural network parameters by computing the cross-entropy loss between next word predictor p_t and the actual next word S_{t+1}. The cross-entropy loss is calculated for the entire sequence in the forward pass of the neural network.

Then, the obtained total loss is used to calculate the prediction error through the backward pass. Further, RNN considers long short-term memory (LSTM) [10] or gated recurrent unit (GRU) [3] for encoding and decoding to enhance learning long-term features.

There are two main units of NMT: encoder and decoder, where the encoder is used to compact the whole input/source sentence into a context vector and the context vector is decoded to the output/target sentence by the decoder. Such basic encoder-decoder based NMT unable to capture all important information if the sequence is too long.

Therefore, the attention mechanism comes into existence [1, 23] that allows the decoder to focus on different segments of the sequence locally (part of the sequence) as well as globally (associating all the words of the sequence).

Fig. 3 depicts attention-based RNN, where the input Mizo sentence *"Thil ka lei˘"* is translated into the target English sentence *"I am buying something"*. The drawback of RNN is that input processing follows in a strict temporal order, which means it computes context in one direction based on preceding words, not on future words. RNN impotent to look ahead into future words. BRNN (Bidirectional RNN) [1] resolves this issue by utilizing two distinct RNNs, one for the forward direction and another for the backward direction.

In [33], a BRNN based model improves translation accuracy on low-resource pairs like English–Hindi, English–Tamil. Moreover, the convolutional neural network (CNN) based NMT is introduced [11, 8] by taking advantage of parallelizing operation and considering relative positions of the tokens instead of the temporal dependency among the tokens of the sequence.

But it lags behind features of RNN to enhance the encoding of the source sentence. The demerits of CNN-based approaches require many layers to hold long-term dependency, making the network large or complex without ever succeeding, which seems to be impractical. To handle such an issue, a transformer-based NMT comes in [38].

The idea behind the transformer model is to encode each position and apply a self-attention mechanism to connect two different words, which would be parallelized to accelerate learning. Unlike the traditional attention mechanism, the self-attention mechanism calculates attention several times, which is known as multi-head attention.

However, both SMT and NMT require minimal training data to provide a promising result, which is a significant problem for low-resource pairs like English–Mizo. It is a challenging task to prepare the parallel and monolingual corpora for English–Mizo.

4 Related Work

This section focuses on existing MT work done on English–Mizo and other low-resource pairs. In MT, there are limited existing works available for English–Mizo pair [30, 15, 16]. A comparative study [30] in English to Mizo translation was performed between SMT and NMT, where NMT outperforms SMT.

In [15, 16], various attention-based NMT models, RNN and BRNN, have been examined in English to Mizo translation with parallel data only. Monolingual data is not incorporated to improve such low-resource pair translation. Furthermore, no previous work is found that focuses on tonal words of Mizo in MT in both directions of translation, i.e., English to Mizo and vice versa.

Besides, MT related works include recognizing named entity classes [2], Multi-word Expressions (MWE) [24], and resource building and POS tagging for Mizo language [27]. The NMT has been investigated with RNN for low-resource pairs like English to Punjabi, English to Tamil, and English

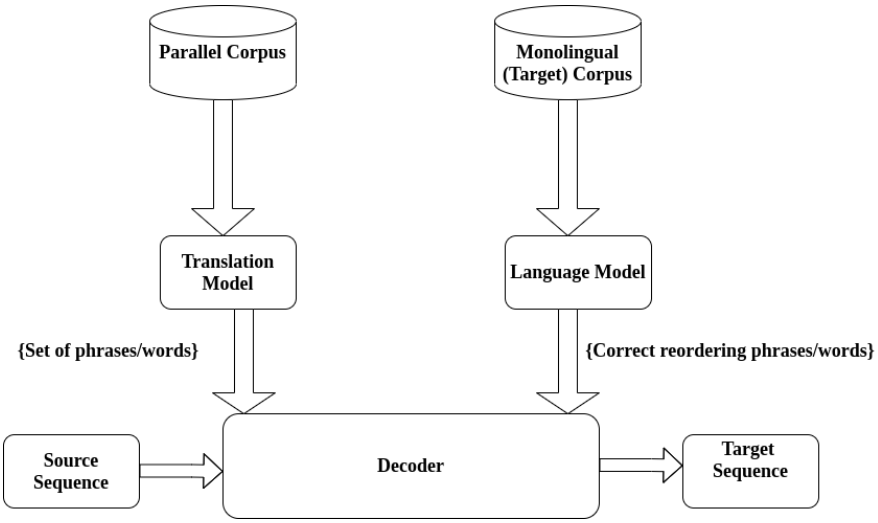


Fig. 2. Abstract diagram of phrase-based SMT

to Hindi and observed that performance increases with an increase in parallel train data [29].

In [34, 18], English to Hindi translation on the benchmark dataset, the NMT shows promising results. For low-resource pair translation like English to Vietnamese and English to Farsi, NMT improved performance through the recurrent units with multiple blocks and a trainable routing network [41].

Moreover, among similar language pair translations in WMT19, NMT systems attained remarkable performance on Hindi-Nepali [20]. With monolingual data to address the low-resource language problem, a filtering approach for the pseudo-parallel corpus is proposed to increase the parallel training corpus.

Despite achieving state-of-the-art performance in various language pairs, the NMT demands parallel corpus, which is a big challenge in low-resource pairs. To address this issue, a monolingual data-based NMT has been introduced without modifying system architecture [35].

By applying BackTranslation (BT) on low resource language monolingual data, the low-resource target sentences can be generated using the NMT trained model. Then the obtained synthetic parallel data can be used as additional parallel training data.

However, the NMT performance degrades by directly augmenting BT data in the original parallel data. Therefore, to improve NMT performance, BT data filtering is necessary before adding with original parallel data [40]. In the context of low resource tonal language like Burmese with English pair, NMT with BT strategy shows remarkable performance [39].

Moreover, unsupervised pre-train based NMT is introduced [37, 17], where monolingual data of both source and target sentences are pre-trained and then fine-tuned the trained model with original parallel data.

5 EnMzCorp1.0: English–Mizo Corpus

The low-resource English–Mizo (En-Mz) pair has limited available options for parallel and monolingual data of Mizo. We have explored different viable resources to prepare the corpus, which discusses in the following subsections.

5.1 Corpus Details

We have prepared an En-Mz parallel corpus that contains a total of 130,441 sentences. Also, monolingual data of Mizo is prepared. The parallel corpus is collected from various online sources

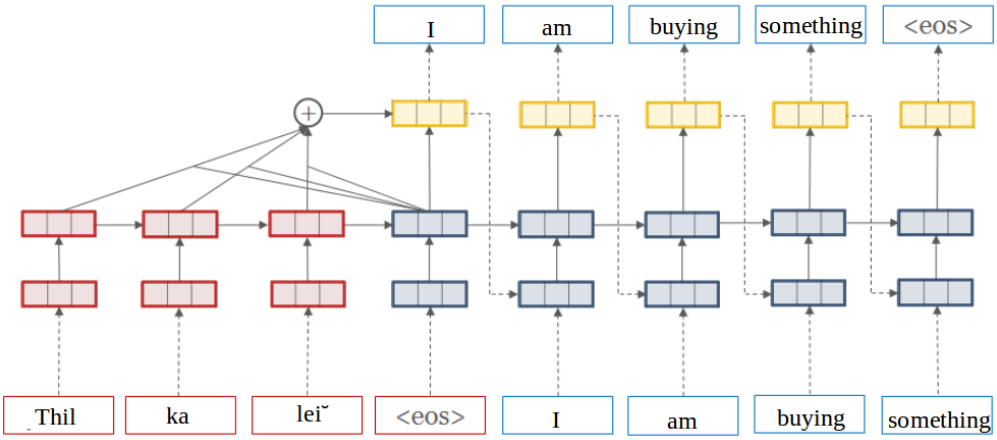


Fig. 3. English–Mizo NMT system (attention-based RNN)

namely, Bible[3], online dictionary (Glosbe)[4], Government websites[5] [6] and different web pages / blogs. Table 6 presents the corpus of sources with statistics, and Table 5 demonstrates example sentences collected from various sources. In Table 5, tonal words in the sentences are marked as bold.

5.2 Corpus Extraction Approaches

We have used a web crawling technique, namely Scrapy, which is an open-source framework. In Scrapy, xpath of each element is coded with a degree of generalization, which helps to crawl numerous web pages by replicating multiple web pages. To extract text from the PDF/image files, Google OCR[7] tool is used.

It is mainly used to extract Mizo data from text book[8] (Government website) and Table 8 presents the extracted data statistics for the same. Fig. 4 depicts the overall data acquisition. Moreover, we have used manual effort to prepare parallel data, mainly government websites extracted data.

[3] https://www.bible.com/
[4] https://glosbe.com/en/lus
[5] https://finance.mizoram.gov.in/
[6] https://dipr.mizoram.gov.in/
[7] https://cloud.google.com/vision/
[8] https://scert.mizoram.gov.in/page/english-medium

From the monolingual data of Mizo, tonal and symbolic words are extracted and translated manually to their corresponding English words. The manual process alignment took a period of 2 to 3 months by the first author. Moreover, the Mizo sentences are cross-verified by hiring a linguistic expert of Mizo, who is a native speaker and possesses linguistic knowledge of Mizo.

5.3 Data Cleaning and Split

The prepared corpus contains noise like too many special characters, web-link (URLs), blank lines, and duplicates. Therefore, we have removed noise and the duplicate sentences, the total number of parallel sentences reduced to 118,449.

During data cleaning, conversion of lower-case and removal of punctuation is not performed as in [19] to maintain the semantic contextual meaning. Table 7 presents the split data for the train, validation, and test data.

During the partition of validation and test data, we have considered those sentences which have tonal words. We have also considered two test sets, namely Test Set-1 for in-domain data from the split data and Test Set-2 for out-domain data that includes different types of tonal words having maximum length of 15 words, which we have prepared manually.

Table 5. Example of parallel and monolingual sentences

Corpus	En	Mz	Source
Parallel	In the beginning God created the heavens and the earth.	A **tîrin** Pathianin lei leh **vân** a siam a.	Bible
	He will guide the humble in justice.	Retheite chu dik takin ro a **rêlsak** ang.	
	What questions do we need to answer?	Eng zawhnate nge kan **chhân** ang?	Glosbe
	What is humility?	**Inngaihtlâwmna** chu eng nge ni?	
	GSDP which is at an approximate level compared to previous year's figure.	GSDP atanga **chhût** erawh hi chu nikum dinhmun nen a intluk tlang a ni.	Government Website
	And the gate was shut as soon as the pursuers had gone out.	A **ûmtute** chu an chhuah veleh kulh kawngka chu an **khâr** ta a.	
	advance	**hmasâwn**	Tonal Word (Manually Prepared)
	punch	**hnék**	
	At Famous	'Famous'-ah	Symbolic word Manually Prepared)
	God for ever	kumkhua-in—Pathian	
Monolingual		Schedule tribe-te chu income tax **àwl** an ni thin tih sawiin Zoramthanga chuan.	Web pages/Blogs/Text Book
		Mi **tlâwmte** chu a kawng a **zirtîr thîn**.	

Table 6. Corpus sources and statistics

Corpus	Source	Sentences	Tokens	
			En	Mz
Parallel	Bible	26,086	684,093	866,317
	Online Dictionary (Glosbe)	70,496	1,438,445	1,674,435
	Government Websites	31,518	402,90	653,65
	Tonal and Symbolic words (Manually Prepared)	2,341	2,341	2,341
	Total	130,441	2,165,169	2,608,458
Monolingual	Web Pages/Blogs/Text Book	1,943,023	-	25,813,315

Following [19], we have considered small test data in comparison to training data because it is used for the baseline system. In the train data, out of 115,249 Mizo sentences, 44,604 sentences have tonal words.

5.4 Domain Coverage

Our corpus EnMzCorp1.0 covers various domains: Bible, daily usage, Government messages/notices, elementary textbook, dictionary, and general-domains.

6 Baseline System

We have considered phrase-based SMT [14] and sequence-to-sequence model-based NMT (recurrent neural network (RNN), bidirectional RNN (BRNN)) for baseline systems to provide benchmark translation accuracy for both the directions of translations in English-Mizo pair. We have utilized our EnMzCorp1.0 dataset and monolingual data of English (3 million sentences) from WMT16[9].

6.1 Experimental Setup

We have followed SMT and NMT setup by employing Moses[10] and OpenNMT-py[11]toolkit respectively. The SMT and NMT setup is used for

[9] http://www.statmt.org/wmt16/translation-task.html

[10] http://www.statmt.org/moses/

[11] https://github.com/OpenNMT/OpenNMT-py

Table 7. Statistics for train, valid and test set

Type	Sentences	Tokens	
		En	Mz
Train	115,249	1,308,563	1,462,070
Validation	3,000	78,083	82,470
Test Set-1	200	5,181	5,523
Test Set-2	200	1,312	1,608

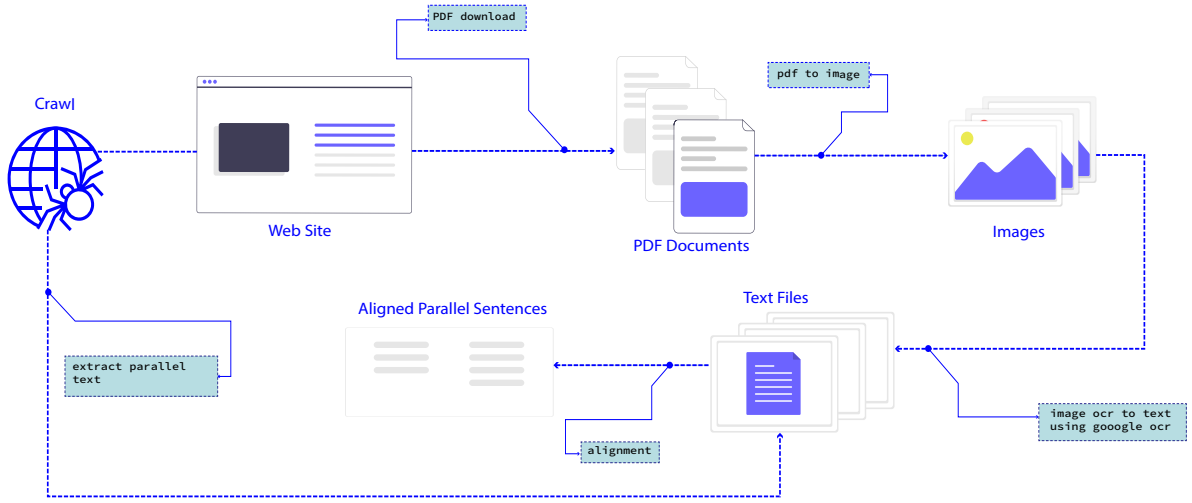


Fig. 4. Data acquisition

Table 8. Extracted Mz data using Google OCR

Monolingual Data	Sentences	Tokens
Mz	14,957	243,657

building phrase-based (PBSMT), RNN and BRNN based NMT systems.

For PBSMT, GIZA++ and IRSTLM [7] are utilized to produce phrase pairs and language models following the default settings of Moses. For RNN and BRNN, a 2-layer long short term memory (LSTM) network of encoder-decoder architecture with attention is used [1].

The LSTM contains 500 units at each layer. The Adam optimizer with a learning rate of 0.001 and drop-outs 0.3 is used in RNN and BRNN models. We have used unsupervised pre-trained word vectors of monolingual data using Glove[12] [31] and pre-trained up to 100 iterations with embedding vector size 200.

6.2 Results

To evaluate predicted sentences, automatic evaluation metrics and human evaluation are considered. The automatic evaluation metrics viz. bilingual evaluation understudy (BLEU) [28], translation edit rate (TER) [36], metric for evaluation of translation with explicit ordering (METEOR) [22] and F-measure.

6.2.1 BLEU

It utilizes the modified precision of n-gram by comparing the n-grams of the candidate

[12] https://github.com/stanfordnlp/GloVe

Table 9. BLEU scores of baseline systems

Translation	Test Data	PBSMT	RNN	BRNN
En to Mz	Test Set-1	16.23	18.27	18.41
	Test Set-2	2.60	3.24	3.44
Mz to En	Test Set-1	17.18	19.20	20.12
	Test Set-2	2.75	3.47	3.49

(predicted) translation with the n-grams of the reference translation. Eq. 10 represents the formula for the computation of the BLEU score. Here, P_l, R_l denote the length of the predicted and reference translation, respectively.

Pd_i represents precision score of i^{th} gram. To deals with too short translation, a brevity penalty is equal to 1.0 is considered when the candidate translation length is the same as the length of any reference translation.

It is recommended to consider the lower value of n when the translation system is not adequate [30]. In this work, $n = 3$ is considered because the BLEU score tends to zero while crossing the tri-gram score. Table 9 presents the BLEU scores for both the directions of translation:

$$\text{BLEU} = \min\left(1,\ \frac{P_l}{R_l}\right)\left(\sum_{i=1}^{n} Pd_i\right)^{\frac{1}{n}}. \qquad (10)$$

6.2.2 TER

It is an automatic metric used to calculate the number of actions required to update a candidate translation to align with the reference translation. It is a technique used in MT for measuring the amount of post-editing effort needed for the output of machine translation. TER is computed in Eq. 11 by dividing the number of edits (N_{ed}) needed to adjust the candidate translation to match the reference translation by the reference translation's length (L_{rw}):

$$\text{TER} = \frac{N_{ed}}{L_{rw}}. \qquad (11)$$

Several possible edits include insertion, deletion, the substitution of single words, and shifts of word sequences. The cost of all the edits is the same. Consider the following scenario of candidate translation and reference translation where the mismatch is highlighted by italics:

— Reference translation: Fruits are healthy *tasty and nutritious* loaded with *fiber* and vitamin,

— Candidate translation: Fruits are *tasty and* healthy loaded with *minerals antioxidant* and vitamin.

From the above scenario, even if the candidate translation is fluent, TER, on the other hand, would not accept it as an exact match. The possible edits are as follows:

— tasty and : shift (1 edit),

— nutritious : insertion (1 edit),

— minerals antioxidant : substitution for fiber (2 edits).

The total number of edits is 4 (one shift, one insertion, and two substitutions). The length of the reference word is 11. Therefore, TER score becomes $\frac{4}{11} = 36\%$. Lower the value of the TER score, accuracy will be good. Table 10 presents TER scores.

6.2.3 METEOR and F-measure

Meteor is calculated by computing a word alignment based on matching the three modules: an explicit word, stem word, and synonym word between the predicted and reference translation.

These three modules work together to ensure the alignment between the two translations. The

uni-gram precision P_{ug} and uni-gram recall R_{ug} are calculated using Eq. 12 and 13:

$$P_{ug} = \frac{T_m}{p_n}, \tag{12}$$

$$R_{ug} = \frac{T_m}{r_n}. \tag{13}$$

Where, p_n and r_n are number of uni-grams in the predicted, reference translation respectively. T_m denotes total number of matched uni-grams word between candidate and reference translation.

Table 10. TER (%) scores of baseline systems

Translation	Test Data	PBSMT	RNN	BRNN
En to Mz	Test Set-1	80.1	78.90	75.0
	Test Set-2	102.6	102.4	101.8
Mz to En	Test Set-1	76.80	74.50	73.60
	Test Set-2	95.30	93.80	93.40

Table 11. METEOR scores of baseline systems

Translation	Test Data	PBSMT	RNN	BRNN
En to Mz	Test Set-1	0.1626	0.1795	0.1812
	Test Set-2	0.0792	0.0794	0.0811
Mz to En	Test Set-1	0.1783	0.1856	0.1904
	Test Set-2	0.0893	0.0920	0.0925

Table 12. F-measure scores of baseline systems

Translation	Test Data	PBSMT	RNN	BRNN
En to Mz	Test Set-1	0.3832	0.4139	0.4179
	Test Set-2	0.1872	0.1961	0.1970
Mz to En	Test Set-1	0.4049	0.4175	0.4316
	Test Set-2	0.2103	0.2114	0.2140

During the computation of METEOR score, F-measure score is calculated, which is the harmonic mean of precision P_{ug} and recall R_{ug} as shown in Eq. 14.

$$\mathrm{F-measure} = \frac{2 \times P_{ug} \times R_{ug}}{P_{ug} + R_{ug}}. \tag{14}$$

Also, F-mean is calculated by the parameterized harmonic mean of the precision P_{ug} and recall R_{ug}. Then, METEOR is computed using Eq. 15:

$$\mathrm{METEOR} = (1 - \mathrm{Pen}) \times F_{\mathrm{mean}}. \tag{15}$$

Here, fragmentation penalty (Pen) is calculated by fragmentation fraction (frag) and γ in Eq. 16 to account for the degree to which the uni-grams in both translations are in the same order. γ is the maximum penalty which is determined by the value ranges from 0-1.

To compute fragmentation fraction (frag), the number of chunks (ch), which is a group of matched uni-grams that are adjacent to each other with having the same word order in both the translations, is divided by the number of matches (m) as given in Eq. 17. METEOR and F-measure are assigned, ranging from 0 to 1 in each segment. Table 11 and 12 present METEOR and F-measure scores:

$$\mathrm{Pen} = \gamma \times \mathrm{Frag}, \tag{16}$$

$$\mathrm{Frag} = \frac{ch}{m}. \tag{17}$$

6.2.4 HE

Human evaluation (HE) is a manual evaluation metric that is used for evaluating the predicted sentence of the machine translation systems [30].

As automated evaluation metrics fail to assess all critical aspects of translation accuracy, the human evaluator with a linguistic expert has evaluated the predicted translation. The linguistic expert engaged in human evaluation is acquainted with both the Mizo and English language.

The expert is well-versed with the complexities and challenges of the Mizo language. Based on adequacy, fluency, and overall rating, a human evaluator evaluates the predicted translations. Adequacy is measured using the contextual meaning of the predicted translation that corresponds to the reference translation.

Fluency is measured by considering the good formation of the predicted sentence in the target language, regardless of whether it corresponds to the reference translation. By computing an average score of both adequacy and fluency, the overall rating is measured. Considering an example of a reference translation as:

"Small businesses have been exempted from the tax increase" and the predicted translation as *"I am putting my hand on my table"*.

Here, the predicted translation is considered inadequate since it contains a different contextual meaning with the corresponding reference translation. The predicted sentence is also fluent; even though the meaning is entirely different from the reference translation, it is a well-formed sentence in the target language. The overall rating[13] considers the average of the adequacy as well as fluency.

Table 13. HE (Overall Rating (%)) scores of baseline systems

Translation	Test Data	PBSMT	RNN	BRNN
En to Mz	Test Set-1	28.56	29.40	31.92
	Test Set-2	17.40	18.80	19.60
Mz to En	Test Set-1	29.24	30.08	32.92
	Test Set-2	18.60	19.20	20.80

Table 14. Augmented train data statistics

Parallel Corpus	Sentences	Tokens	
		En	Mz
Synthetic	33,229	550,238	610,376
Synthetic + Original	148,478	1,858,801	2,072,446

The assessment criteria are measured on a scale of 1-5, with higher values indicating better performance [30]. The rating score is assigned for 50 predicted test sentences (randomly chosen). Table 13 reports human evaluation scores which are calculated using Eq. 18.

Where n_{TAR} is the total average rating scores of adequacy, and fluency. Here, n_{TBR} is calculated by multiplying best rating score with total number of questions, i.e., $5 \times 50 = 250$:

$$\mathrm{HE}(\mathrm{Overall\ Rating}) = \frac{n_{\mathrm{TAR}}}{n_{\mathrm{TBR}}} \times 100\%, \qquad (18)$$

7 Proposed Approach

Our proposed approach is based on BT [35] strategy without modifying the model architecture. It consists of three operations. First, extraction of Mizo sentences having tonal words from monolingual data of Mizo. Secondly, extracted Mizo tonal sentences are used to generate the English synthetic sentences via the best translation model (BRNN) of Mizo to English obtained from the baseline system.

Then, the synthetic parallel corpus is augmented with the original parallel corpus. The main goal of the first two operations is to expand the parallel train data by increasing the Mizo tonal sentences. Lastly, the augmented data is used for training the NMT model (BRNN) independently for each direction of translations. Fig. 5 depicts the pictorial diagram of the proposed approach. Since the original train data contains only 44,604 Mizo tonal sentences, we have extracted 44,000 Mizo tonal sentences by considering a maximum word length of 10.

However, we have removed blank lines, single word sentences from the synthetic English sentences and their corresponding Mizo sentences. Thus, the synthetic parallel corpus contains a total of 33,229 sentences as shown in Table 14. The evaluated results are reported in Table 16, 15.

8 Analysis

Among the automatic evaluation scores in Table 9, 10, 11, 12 and 16 on both test data, the proposed approach attains higher accuracy than baseline systems. The accuracy on Test Set-1 (in-domain) is better than Test Set-2 (out-domain).

It is noticed that Mizo to English translational evaluation scores outperforms English to Mizo. Because train data contains more Mizo tokens in comparison to English tokens, as mentioned in Table 7. Therefore the model encoded more Mizo word frequency, and the decoder can produce a better translation for Mizo-to-English.

It is observed that all the system's output encountering tonal words has poor translational quality. Moreover, predicted output suffers under translation, impotent in named-entity prediction and out-of-vocabulary issues. By the following notations, the samples of predicted sentences are considered below to inspect the errors.

[13] https://nlp.amrita.edu/mtil_cen/\#results

— ST: Source Test sentence.

— RT: Reference/Target sentence.

— PD1: Predicted sentence by the proposed approach.

— PD2: Predicted sentence by the BRNN.

— PD3: Predicted sentence by the RNN.

— PD4: Predicted sentence by the PBSMT.

1. Sample predicted sentence for En-to-Mz (partial adequacy but good in fluency)

Table 15. Human evaluation results of the proposed approach

Translation	Test Data	HE Overall Rating (%)
En to Mz	Test Set-1	32.24
	Test Set-2	20.40
Mz to En	Test Set-1	33.48
	Test Set-2	21.80

Table 16. Automatic evaluation results of the proposed approach

Translation	Test Data	BLEU	TER (%)	METEOR	F-measure
En to Mz	Test Set-1	20.21	73.4	0.1851	0.4272
	Test Set-2	4.04	100.5	0.0868	0.1992
Mz to En	Test Set-1	20.31	71.9	0.2022	0.4501
	Test Set-2	4.10	92.4	0.0931	0.2230

— ST: *They dig up the ground to plant seeds.*

— RT: *Thlai chí tuh nan lèi an chŏ.*

— PD1: *Mau hmanga lei laih an rél a.*

— PD2: *Lung chi hrang hrang an han thlen chuan.*

— PD3: *Lung angin lei an khuar a.*

— PD4: *Pialtlêp chu an kân a.*

Discussion: The PD1 has encountered the tonal words *"lei"* meaning *"ground"* and generated the relevant meaning of the tonal words. But it is unable to detect the tone marker *è*. The word *"dig"* in the source sentence is predicted as *"laih"* which is correct and is also having a similar meaning as the tonal word *"chŏ"* in the reference sentence.

The Engliah meaning of the proposed approach is *"They decide to dig the ground with bamboo"*. In the predicted sentence, the word *"Mau"* means *"bamboo"* and a tonal word *"rél"* means *"decide"* are encountered which are not relevant to the source sentence. Both PD2 and PD3 predictions are inadequate and not fluent. However, PD4 translation is also inadequate but fluent.

Thus, in terms of total words, the proposed approach can identify the tonal words, but the other baseline systems do not consider it for translation. As compared to baseline systems translation, the proposed approach has the best-predicted sentence since most of the words are correctly predicted. Therefore, it attains partial adequacy but good in terms of fluency.

2. Sample predicted sentence for Mz-to-En (partial adequacy but good in fluency)

— ST: *Naupang ruàlin pawnah an nghak.*

— RT: *A group of children waited outside the door.*

— PD1: *They are waiting for the child.*

— PD2: *shun*

— PD3: *There*

— PD4: *books*

Discussion: The PD1 has identified the tonal word *"ruàlin"* in the source sentence and predicted it as "they" which can be accepted as similar meaning with *"group"* in the reference translation.

But PD2, PD3 and PD3 do not recognize the tonal word. They have incorrectly predicted the sentence by only one word which is completely inadequate and also not fluent.

However, in the PD1, the contextual meaning of the predicted sentence is partially adequate as compared to the reference translation. In terms of fluency, it is a well-formed Mizo sentence.

3. Sample predicted sentence for En-to-Mz (inadequacy but good in fluency)

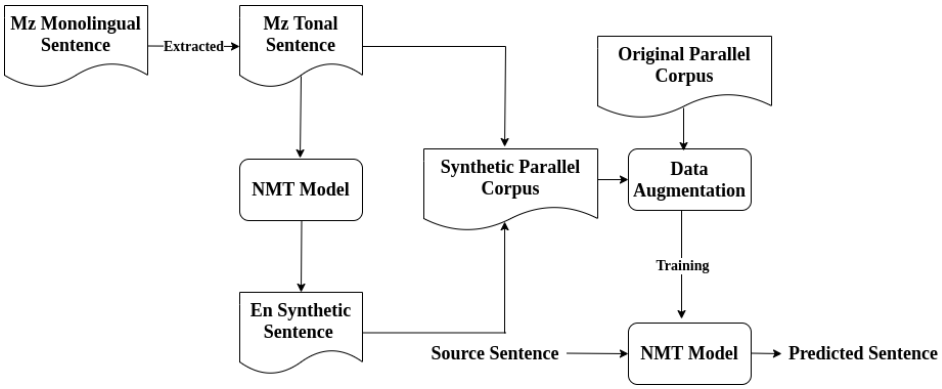


Fig. 5. Proposed Approach

— ST: *There are many ants that crowd around sugar.*

— RT: *Fanghmir tam deuhin chini an bâwm luai luai mai.*

— PD1: *Gas agency tam tak an awm a.*

— PD2: *Hnathawk vêlah chuan huaisen takin a awm a.*

— PD3: *Hnathawk vêlah chuan mipa tam tak an awm a.*

— PD4: *Chutah chuan mipa tam tak an awm a.*

Discussion: Both the PD1 and PD4 have not generated tonal words in their predicted sentence, while a tonal word *"bâwm"* means *"crowd"* appears in the reference text.

However, PD2 and PD3 have both generated a tonal word *"vêlah"* which means *"about"* in their predicted sentence which is not relevant to the reference translation.

For all the systems, the predicted translations are contextually inadequate with a completely different meaning as compared to the reference translation. But in terms of fluency, the predicted sentences of all the systems are good in fluency.

4. Sample predicted sentence for Mz-to-En (inadequacy but good in fluency)

— ST: *A hma a ka lo tilo kha ka ă hle mai.*

— RT: *I was foolish not to have done it before.*

— PD1: *I was very sorry that he had not come before him.*

— PD2: *I did not know how I was good.*

— PD3: *I didn't know how he didn't know it.*

— PD4: *I did not know him until he was saying.*

Discussion: A tonal word *"ă"* which means *"foolish"* appears in the source sentence, but none of the systems are able to detect the source's tonal word. Here, the contextual meaning of all the predicted sentences is completely different from the reference translation.

Therefore, they are termed as inadequate. As the predicted sentence is a well-formed and proper sentence of the target language, it is considered to be good in fluency.

5. Sample predicted sentence of named-entity error (En-to-Mz)

— ST: *They moved the goal posts wider apart.*

— RT: *Goal bàn an sawn zau.*

— PD1: *Ruahpui vânâwn chu nasa takin an chelh a.*

— PD2: *Thalai chu an tum ber tur tlat a ni.*

— PD3: *latitudinal*

— PD4: *Mitin chuan an ramri chu an pan ta a.*

ISSN 2007-9737

Discussion: A tonal word *"vânâwn"* means *"down pour"* is generated in the PD1. However, there is no relevant word in the reference translation. On the other hand, a tonal word *"bàn"* appears in the reference translation, but all the systems are unable to correctly generate the tonal word in their predicted sentence.

There are multiple errors in the named entity as the word *"goal"* appears in both source text as well as reference text, but none to the systems have correctly generated in their predicted sentence. Therefore, due to huge errors in named entities and contextually different predictions, the predicted sentences of all the systems are inadequate.

In terms of fluency, parts of the prediction in PD1 and PD2 are correct so they are partially fluent. However, PD3 predicts non-Mizo words and PD4 predicts a proper Mizo sentence. Therefore, it is good in fluency but inadequate.

6. Sample predicted sentence of named-entity error (Mz-to-En)

— ST: *I ka ăng rawh le.*

— RT: *Open your mouth.*

— PD1: *hushaby*

— PD2: *I make it for you.*

— PD3: *Let me get your grave.*

— PD4: *I have to make it for y.*

Discussion: A tonal word *"ăng"* which means *"open mouth"* appears in the source sentence but none of the systems are able to detect the source's tonal word. All the systems have encountered named-entity errors in their predicted sentences. While the reference translation is *"Open your mouth"*.

None of the systems predicted the word *"open"* and *"mouth"*. PD1 predicts as *"hushaby"* which is completely inadequate but fluent. Likewise, PD2 and PD4 have both predicted a contextually different sentences but perfectly fluent. However, PD3 predicts an improper English sentence which is also inadequate.

7. Sample predicted sentence of over-prediction (En-to-Mz)

— ST: *Two children answered the teacher's question simultaneously.*

— RT: *Naupang pahnih chuan zirtirtu zawhna a ruálin an chhăng.*

— PD1: *Naupangte chuan zawhna an chhâng a, zawhna pahnih an chhâng a.*

— PD2: *Fa pahnih chuan ¡unk¿ zawhna pakhat chu an chhâng a.*

— PD3: *Fapa pahnih chuan zawhna pakhat chu an chhâng a.*

— PD4: *16 Naupang pahnih chuan zawhna pakhat a chhâng a.*

Discussion: Two tonal words *"chhăng"* and *"ruálin"* appear in the reference translation. A word *"answered"* in the source text is correctly predicted by all the systems as *"chhâng"*. But in all the predicted sentences, the tone marker is changed in *"chhâng"* which is a falling tone while in the reference translation it is a rising tone.

However, a tonal word *"ruálin"* from the reference translation which means *"simultaneously"* is unable to be generated by all the systems in their predicted sentence. From all the predicted sentences it can be noticed that all of the systems encountered over-prediction.

As the number of questions is not mentioned in the source test sentence, however, all the systems have predicted a sentence which includes the number of questions. PD1 predicts two questions while PD2, PD3 and PD4 predict one question.

Even though the predicted sentences by all the systems are incomplete and inadequate but all are well-formed and therefore, it is fluent.

8. Sample predicted sentence of over-prediction (Mz-to-En)

— ST: *kha kha ti suh a tia, a ăng vak a.*

— RT: *Don't do that! she shouted angrily.*

— PD1: *And do not do it in judgment and in crook.*

— PD2: *Do not do that which is great in the eyes of him who is ¡unk¿*

— PD3: *not*

— PD4: *And don't do not do that which is right in the eyes of Yahweh .*

Discussion: A tonal word *"ăng"* which means *"shouted"* appears in the source sentence but none of the systems are able to detect the source's tonal word. The PD1 is over-predicted by adding *"judgment and in crook"* which does not appear in the reference translation. Similarly, PD2 and PD4 have also been over-predicted by adding several words apart from the reference sentence.

Although, it is inadequate but good in fluency. Besides, *"¡unk¿"* is detected as part of the predicted sentence in PD2. However, PD3 has predicted only a single word *"not"* which is inadequate.

9. Sample predicted sentence of under prediction (En-to-Mz)

— ST: *There was a bomb blast yesterday.*

— RT: *Niminah bàwm a puak.*

— PD1: *¡unk¿ puak a awm a.*

— PD2 : *Nimin puak puak thei a awm.*

— PD3: *Zanin chu a puak puak.*

— PD4:*Niminah tu a lo awm.*

Discussion: A tonal word *"bàwm"* means *"bomb"* appears in the reference translation. However, none of the systems can correctly generate the tonal word in the predicted sentences.

In the PD1, *"¡unk¿"* is generated as part of the predicted sentence. However, the prediction of all the systems encountered under-prediction as *"bomb"* and *"yesterday"* are not generated in the PD1.

Although, it is inadequate but good in fluency. Similarly, PD2 and PD3 have not mentioned *"bomb"* and PD4 does not mention *"bomb blast"*. The predicted sentences of PD2, PD3 and PD4 are inadequate and it is not a well formed Mizo sentence.

10. Sample predicted sentence of under prediction (Mz-to-En)

— ST: *kan bill kan pek hnuah èngzah nge la bâng áng?*

— RT: *How much will we have left over once we've paid our bill?*

— PD1: *And when we give the bill.*

— PD2: *When our bill of our bill.*

— PD3: *After the bill of our bill.*

— PD4: *And when we get the Memorial, what does it ¡unk¿*

Discussion: Three tonal words *"èngzah"* means *"How much"*, *"bâng"* means *"left"* and *"áng"* means *"will"* is encountered in the source sentence, but none of the systems are able to detect the source's tonal word. All the systems encountered under prediction where the predicted sentence predicts only part of the reference translation.

It is inadequate as the contextual meaning of the reference translation is different from the predicted sentence of the systems. In terms of fluency, it is not a well-formed Mizo sentence.

9 Conclusion and Future Work

In this article, we have developed EnMzCorp1.0 for the English-Mizo corpus, and the same has been used to build baseline systems for English to Mizo and vice-versa translations encountering tonal words.

The dataset will be available here[14]. Moreover, the proposed approach based on the data augmentation technique attains higher translation accuracy than baseline systems.

From the analysis of predicted translations, it is realized that the system needs to be improved to encounter Mizo tonal words. In the future, we will increase the size of the dataset and explore the knowledge-transfer-based NMT approach for improvement.

[14] https://github.com/cnlp-nits/EnMzCorp1.0

Acknowledgments

The authors are thankful to the Department of Computer Science and Engineering and Center for Natural Language Processing (CNLP) at the National Institute of Technology, Silchar for providing the requisite support and infrastructure to execute this work.

References

1. **Bahdanau, D., Cho, K., Bengio, Y. (2015).** Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, pp. 1–15.
2. **Bentham, J., Pakray, P., Majumder, G., Lalbiaknia, S., Gelbukh, A. (2016).** Identification of rules for recognition of named entity classes in Mizo language. 2016 Fifteenth Mexican International Conference on Artificial Intelligence (MICAI), IEEE, pp. 8–13.
3. **Cho, K., van Merriënboer, B., Bahdanau, D., Bengio, Y. (2014).** On the properties of neural machine translation: Encoder–decoder approaches. Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, Doha, Qatar, pp. 103–111.
4. **Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y. (2014).** On the properties of neural machine translation: Encoder-decoder approaches. **Wu, D., Carpuat, M., Carreras, X., Vecchi, E. M.**, editors, Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014, Association for Computational Linguistics, pp. 103–111.
5. **Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014).** Learning phrase representations using RNN encoder–decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734.
6. **Fanai, L. T. (1992).** Some aspects of the lexical phonology of Mizo and English an autosegmental approach.
7. **Federico, M., Bertoldi, N., Cettolo, M. (2008).** IRSTLM: an open source toolkit for handling large scale language models. INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22-26, 2008, ISCA, pp. 1618–1621.
8. **Gehring, J., Auli, M., Grangier, D., Dauphin, Y. (2017).** A convolutional encoder model for neural machine translation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, pp. 123–135.
9. **Gu, J., Hassan, H., Devlin, J., Li, V. O. (2018).** Universal neural machine translation for extremely low resource languages. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, pp. 344–354.
10. **Hochreiter, S., Schmidhuber, J. (1997).** Long short-term memory. Neural Comput., Vol. 9, No. 8, pp. 1735–1780.
11. **Kalchbrenner, N., Blunsom, P. (2013).** Recurrent continuous translation models. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, pp. 1700–1709.
12. **Kocmi, T. (2020).** Exploring benefits of transfer learning in neural machine translation. CoRR, Vol. abs/2001.01622.
13. **Koehn, P. (2010).** Statistical Machine Translation. Cambridge University Press, USA, 1st edition.
14. **Koehn, P., Och, F. J., Marcu, D. (2003).** Statistical phrase-based translation. Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 127–133.
15. **Lalrempuii, C., Soni, B. (2020).** Attention-based english to Mizo neural machine translation. Machine Learning, Image Processing, Network Security and Data Sciences, Springer Singapore, Singapore, pp. 193–203.

16. **Lalrempuii, C., Soni, B., Pakray, P. (2021).** An improved English-to-Mizo neural machine translation. ACM Trans. Asian Low-Resour. Lang. Inf. Process., Vol. 20, No. 4.

17. **Lample, G., Conneau, A. (2019).** Cross-lingual language model pretraining. Advances in Neural Information Processing Systems (NeurIPS).

18. **Laskar, S. R., Dutta, A., Pakray, P., Bandyopadhyay, S. (2019).** Neural machine translation: English to Hindi. 2019 IEEE Conference on Information and Communication Technology, pp. 1–6.

19. **Laskar, S. R., Faiz Ur Rahman Khilji Darsh Kaushik, A., Pakray, P., Bandyopadhyay, S. (2021).** EnKhCorp1.0: An English–Khasi corpus. Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021), Association for Machine Translation in the Americas, Virtual, pp. 89–95.

20. **Laskar, S. R., Khilji, A. F. U. R., Pakray, P., Bandyopadhyay, S. (2020).** Hindi-Marathi cross lingual model. Proceedings of the Fifth Conference on Machine Translation, Association for Computational Linguistics, Online, pp. 396–401.

21. **Laskar, S. R., Pakray, P., Bandyopadhyay, S. (2019).** Neural machine translation: Hindi-Nepali. Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2), Association for Computational Linguistics, Florence, Italy, pp. 202–207.

22. **Lavie, A., Denkowski, M. J. (2009).** The meteor metric for automatic evaluation of machine translation. Machine Translation, Vol. 23, No. 2–3, pp. 105–115.

23. **Luong, T., Pham, H., Manning, C. D. (2015).** Effective approaches to attention-based neural machine translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, pp. 1412–1421.

24. **Majumder, G., Pakray, P., Khiangte, Z., Gelbukh, A. (2018).** Multiword expressions (mwe) for Mizo language: Literature survey. **Gelbukh, A.**, editor, Computational Linguistics and Intelligent Text Processing, Springer International Publishing, Cham, pp. 623–635.

25. **Megerdoomian, K., Parvaz, D. (2008).** Low-density language bootstrapping: the case of Tajiki Persian. Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco, European Language Resources Association, pp. 3293–3298.

26. **Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S. (2010).** Recurrent neural network based language model. volume 2, pp. 1045–1048.

27. **Pakray, P., Pal, A., Majumder, G., Gelbukh, A. (2015).** Resource building and parts-of-speech (pos) tagging for the Mizo language. 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI), pp. 3–7.

28. **Papineni, K., Roukos, S., Ward, T., Zhu, W.-J. (2002).** BLEU: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 311–318.

29. **Pathak, A., Pakray, P. (2018).** Neural machine translation for Indian languages. Journal of Intelligent Systems, pp. 1–13.

30. **Pathak, A., Pakray, P., Bentham, J. (2018).** English–Mizo machine translation using neural and statistical approaches. Neural Computing and Applications, Vol. 30, pp. 1–17.

31. **Pennington, J., Socher, R., Manning, C. D. (2014).** Glove: Global vectors for word representation. **Moschitti, A., Pang, B., Daelemans, W.**, editors, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL, pp. 1532–1543.

32. **Probst, K., Brown, R. D., Carbonell, J. G., Lavie, A., Levin, L., Peterson, E. (2003).** Design and implementation of controlled elicitation for machine translation of low-density languages.

33. **Ramesh, S. H., Sankaranarayanan, K. P. (2018).** Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, New Orleans, Louisiana, USA, pp. 112–119.

34. **Saini, S., Sahula, V. (2018).** Neural machine translation for english to Hindi. 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), pp. 1–6.

35. **Sennrich, R., Haddow, B., Birch, A. (2016).** Improving neural machine translation models with monolingual data. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, pp. 86–96.

36. **Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J. (2006).** A study of translation edit rate with targeted human annotation. In Proceedings of Association for Machine Translation in the Americas, pp. 223–231.

37. **Variš, D., Bojar, O. (2019).** Unsupervised pretraining for neural machine translation using elastic weight consolidation. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, Florence, Italy, pp. 130–135.

38. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., Polosukhin, I. (2017).** Attention is all you need. In **Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R.**, editors, Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 5998–6008.

39. **Wang, R., Sun, H., Chen, K., Ding, C., Utiyama, M., Sumita, E. (2019).** English-Myanmar supervised and unsupervised NMT: NICT's machine translation systems at WAT-2019. Proceedings of the 6th Workshop on Asian Translation, Association for Computational Linguistics, Hong Kong, China, pp. 90–93.

40. **Wu, L., Wang, Y., Xia, Y., Qin, T., Lai, J., Liu, T.-Y. (2019).** Exploiting monolingual data at scale for neural machine translation. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, pp. 4207–4216.

41. **Zaremoodi, P., Buntine, W., Haffari, G. (2018).** Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Melbourne, Australia, pp. 656–661.

Article received on 04/02/2022; accepted on 20/05/2022.
Corresponding author is Partha Pakray.

ISSN 2007-9737

Tropical Cyclone Simulations with WRF Using High Performance Computing

Jorge Clouthier-Lopez[1], Ricardo Barrón-Fernández[1], David Alberto Salas-de-León[2]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico

[2] Universidad Nacional Autónoma de México, Instituto de Ciencias del Mar y Limnología, Mexico

clouthier@gmail.com, barron2131@gmail.com, dsalas@unam.mx

Abstract. Tropical cyclone (TC) Bud occurred over the Eastern Pacific Ocean from 9 June to 16 June 2018 reaching, a category-4 hurricane status (H4) due to its strong sustained winds and gusts. In this study, we simulate this TC to reproduce its track path, direction, and strength to determine the best model physics configurations that weather agencies could use to forecast TC tracks over the Eastern Pacific Ocean, adjacent to the Mexican coastline. To achieve this goal, the sensitivity for the impact of different microphysics and cumulus parameterization schemes is carried out through high-performance computer simulations, with the WRF model using the cluster *CÓDICE B2 at Centro de Investigación en Computación (CIC) of the Instituto Politécnico Nacional (IPN)*. The realism of the TC for the different schemes is assessed by comparing the simulations and the best track data taken from the National Hurricane Center (NHC-NOAA). The NCEP-GFS forecast data is used as initial and boundary conditions. The evolution of wind and minimum pressure at sea level for the different physics combination runs are also compared with the best track data. We found that the track paths and intensities improve when Sea Surface Temperature (SST) is allowed to evolve with the modeled atmosphere via computer simulations.

Keywords. Numerical weather prediction (NWP), weather mesoscale computer modeling, Tropical cyclone.

1 Introduction

Tropical cyclones (TCs) are weather-rotating systems that form associated with thunderstorms. These systems are characterized by a low-pressure center and extreme wind velocities [1, 2]. They go through four stages, from lower to higher intensity: tropical disturbance, tropical depression, tropical storm, and hurricane. TCs originate over tropical or subtropical ocean regions with sea surface temperatures (SSTs) around 27 °C.

These weather systems can cause deaths and considerable damage over coastal areas [3, 4, 5]. When TCs occur near the coast, like TC Bud, they can indirectly impact over distant inland areas. When they occur adjacent to the continent, convective storms develop over distant inland areas. According to [6], in the tropical Americas, a single TC can cause intense precipitation over inland arid regions.

Therefore, it is of fundamental importance to track down TCs well ahead to minimize societal damage and economic impacts. In general, numerical weather prediction (NWP) models, that run over supercomputers, are used to forecast the paths and strengths of TCs. The latter are highly interrelated with other atmospheric phenomena and strongly dependent on the variations of the SST, which depends on ocean feedbacks between the atmosphere and the ocean mixing layer.

NWP models use initial and boundary conditions to solve atmospheric dynamic and thermodynamic systems of partial differential equations with the assumptions of some semi-empirical physical approximations, known as

parameterizations or parameterization schemes. The latter are linked with sub-grid scale processes that cannot be resolved explicitly by the governing equations implemented in the computer models.

The simulation of TCs with NWP models is very sensitive to the parameterizations used during the model runs. Several studies on the impact of parameterization schemes, when simulating mesoscale weather events, can be found in the literature.

It has been established that not only the physics options [7] but also the initial and boundary conditions in the initialization of NWP models [8, 9, 10, 11] play a decisive role in weather simulations. NWP models are currently used for both research and operational forecasting by different research and weather forecasting agencies around the world.

A sensitivity study is the most adequate to quantify the effect of different physics options; and, in turn, to predict the track path of a TC and its strength over a determined geographical region. Previous studies have made an effort to identify the optimum parameterization schemes and customize NWP models for the simulation and forecast of TCs over specific regions of the world.

For example, [12] documented a comprehensive assessment of the performance of parameterization schemes for TC computer simulations.

[13] performed a sensitivity study of the physics parameterizations when simulating the TC Jal over the Indian Ocean.

[14] performed sensitivity studies to evaluate cumulus parameterization schemes.

[15] conducted several sensitivity experiments for five TCs over the Bay of Bengal and found that the combination of the Kain-Fritsch for convection, the Yonsei University planetary boundary layer (PBL), the LIN for microphysics, and the NOAH for land surface schemes had the best performance to reproduce both the tracks and intensities.

Additionally, data assimilation has also been applied to provide data observations during the running of computer simulations to nudge the development of TC track paths, being considered a successful approach for improving TC forecasts [16, 17, 18]. However, this technique does not let the physical processes to evolve naturally. Besides, data assimilation techniques are very expensive, and they are not suitable to be implemented for forecasting purposes due to the high demand of computer resources.

In contrast to previous studies, we allowed the SST to evolve with the modeled atmosphere, for each model configuration, via the evolution of the computer simulations. According to the ocean temperature feedback between the modeled atmosphere and the ocean mixing layer, the purpose is to get reliable performance when the SST changes are taken into account in the development of TC Bud.

In this article, two ensembles of simulations with various parameterization schemes were carried out to define its members, from 00:00 UTC 10 June to 00:00 UTC 16 June 2018. A sensitivity study for a set of cumulus and microphysics parameterizations schemes was conducted to evaluate the track path and intensity of TC Bud. Three microphysics schemes, three cumulus schemes, and a moisture-advection-based trigger scheme were assessed. In one ensemble, the SST was fixed during the simulation period, while in the other one it was allowed to evolve with the modeled atmosphere via the computer simulations. The latter ensemble showed the best results.

The Weather Research and Forecasting (WRF-ARW) model was used with a new hybrid terrain-following sigma-pressure vertical coordinate. A parent domain and two child nested domains were established to define the grids over which the governing equations are solved. In the innermost domain, the assessment was performed with a grid spacing of 2 km.

This article is organized as follows. Section 2 describes the case study. Section 3 describes the model setup and methodology. Section 4 presents the results. Finally, in section 5, the conclusion is presented.

2 Case Study

TC Bud indirectly impacted the Mexican continental territory; when it occurred over the Eastern Pacific Ocean, heavy precipitation and flooding were present over inland Mexico. In Fig. 1a, the satellite spatial distribution of precipitation is observed. Fig. 1a was obtained and produced

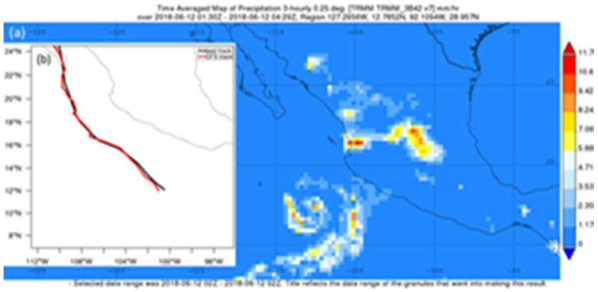


Fig. 1. (a) Spatial distribution of observed averaged precipitation by satellite, at 02:00 UTC, on 12 June, produced with the Giovanni online data system, developed and maintained by the NASA GES DISC. (b) TC Bud's best track (in black) and (in red) the track from the Global Forecasting System (GFS)

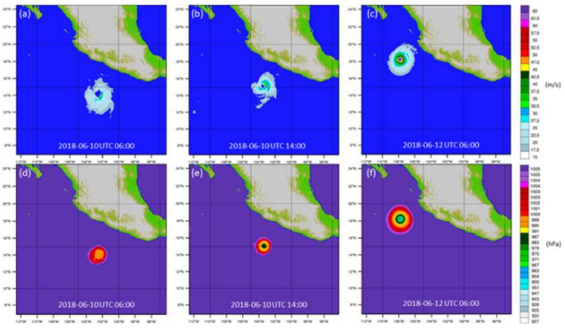


Fig. 2. Evolution of TC Bud from 06:00 UTC on June 10 to 06:00 UTC on June 12. The maximum wind speed (MWS) is at the top, and the sea level pressure (SLP) is at the bottom, according to a WRF-ARW simulation

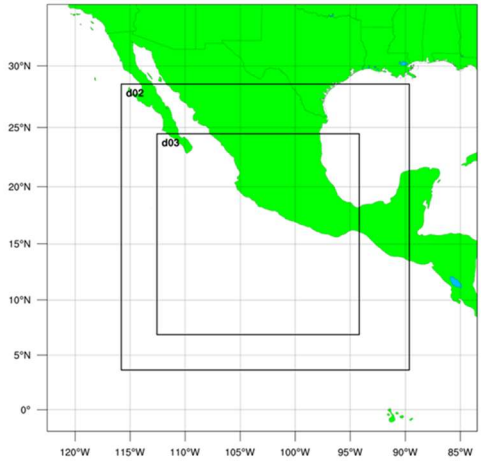


Fig. 3. Domain setup for the study. The assessment was carried over the innermost domain, d03, with a grid spacing of 2 km

with the Giovanni online data system, developed and maintained by the NASA GES DISC.

This TC occurred from 18:00 UTC 9 June to 06:00 UTC 16 June 2018, adjacent to the Mexican coastline, over the Eastern Pacific Ocean. In Fig. 1b, the best track (in black) is shown. Also, in Fig. 1b, the track obtained from the Global Forecast System (GFS) (in red) is also shown. The precursor of the TC was an easterly wave that emerged off the west coast of Africa on 29 May 2018 then entered the eastern North Pacific Ocean on 6 June. Later, a low pressure formed on 9 June, and it started to rotate.

Consequently, a tropical depression, 528 km south of Acapulco Guerrero, formed at 18:00 UTC. It became a tropical storm on 10 June at 00:00 UTC, reaching hurricane status on 10 June at 18:00 UTC. The maximum wind speeds reached 61.7 m/s at 00:00 UTC on 12 June near Manzanillo. Later, Bud became a tropical storm on 13 June at noon UTC, landfalling over Baja California Sur on 15 June at 02:00 UTC, with wind speeds of 20.5 m/s and a central pressure of 999 hPa. Subsequently, it crossed Baja California Sur, weakening due to the interaction with land.

The TC converted into a convection-free post-tropical cyclone on 15 June at noon UTC, and the isobaric contours opened by 00:00 UTC on 16. Then Bud dissipated by 06:00 UTC on 16 June. In Fig. 2, a representation of part of the evolution of the TC Bud is shown. In this figure, the maximum wind speed (MWS) and the sea level pressure (SLP), obtained from one computer simulation, are shown.

3 Modeling Setup and Methodology

The (WRF-ARW) model, version 4.1.3 [19], was used to simulate TC Bud. The model has a fully compressible and non-hydrostatic dynamic core. It allows to simulate and forecast mesoscale convective systems with many parameterization schemes. This mesoscale numerical model was developed for research and operational forecasting. The WRF-ARW model was implemented in the *CÓDICE B2 cluster at Centro de Investigación en Computación (CIC) of the Instituto Politécnico Nacional (IPN)*, using 16 nodes. Each node has 28 CPU cores.

Table 1. Ensemble members of the ensemble with RTG-SST forcing

Simulation	Simulation acronym	Microphysics parameterization	Cumulus parameterization	
1	LIN-KF-RTG	Purdue Lin scheme	Kain-Fritsch scheme	
2	LIN-BMJ-RTG	Purdue Lin scheme	Betts-Miller-Janjic scheme	
3	LIN-SAS-RTG	Purdue Lin scheme	Simplified Arakawa-Schubert Scheme	
4	LIN-KF-TR-RTG	Purdue Lin scheme	Kain-Fritsch scheme	Moisture-advection-based trigger scheme
5	THOM-KF-RTG	Thompson scheme	Kain-Fritsch scheme	
6	THOM-BMJ-RTG	Thompson scheme	Betts-Miller-Janjic scheme	
7	THOM-SAS-RTG	Thompson scheme	Simplified Arakawa-Schubert scheme	
8	THOM-KF-TR-RTG	Thompson scheme	Kain-Fritsch scheme	Moisture-advection-based trigger scheme
9	WSM6-KF-RTG	WRF Single-moment 6-class scheme	Kain-Fritsch scheme	
10	WSM6-BMJ-RTG	WRF Single-moment 6-class scheme	Betts-Miller-Janjic scheme	
11	WSM6-SAS-RTG	WRF Single-moment 6-class scheme	Simplified Arakawa-Schubert scheme	
12	WSM6-KF-TR-RTG	WRF Single-moment 6-class scheme	Kain-Fritsch scheme	Moisture-advection-based trigger scheme

In each of the two ensembles, the outer domain was set with a grid spacing of 18 km, while the innermost domain with a grid spacing of 2 km, using a ratio of 1 to 3 and two-way nesting.

In order to assess the different cumulus and microphysics parameterizations at 2 km grid spacing (over the innermost domain, d03, shown in Fig. 3), the initial and boundary conditions (IBCs) were downscaled. In Fig. 3, the nested domains d01, d02, and d03 with 18, 6, and 2 km grid spacing, respectively, are shown.

The IBCs were obtained from the National Center for Environmental Prediction (NCEP) FNL operational global analysis and forecast dataset, which is available with a grid spacing of 0.25° at 6 h intervals.

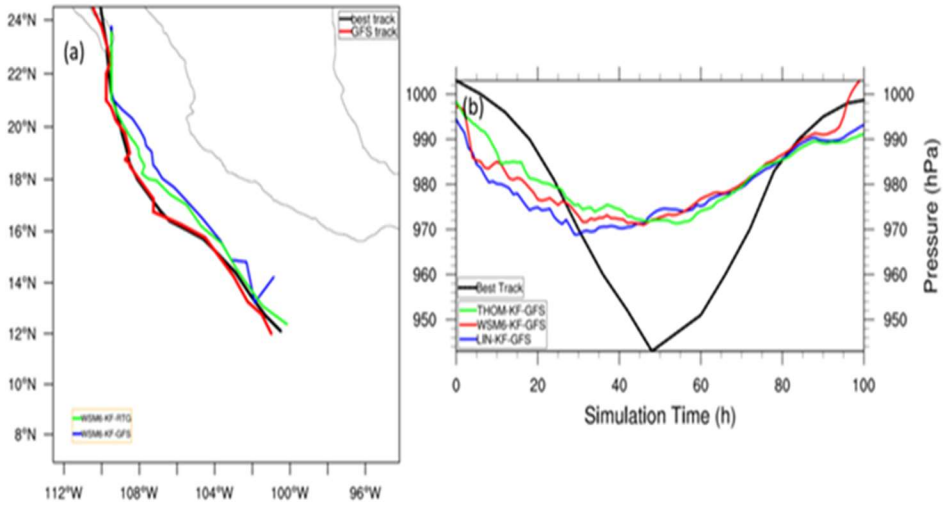


Fig. 4. (a) Track paths with longitude (deg) and latitude (deg) coordinates for WSM6-KF-RTG (with SST forcing) and WSM6-KF-GFS (without SST forcing) ensemble members. (b) Sea Level Pressure (SLP) for the THOM-KF-GFS, WSM6-KF-GFS, and LIN-KF-GFS ensemble members (without SST forcing). Best track observations are also included; as well as the track from the Global Forecasting System (GFS) in (a), shown in red The simulated track paths, the SLP along the tracks, and the MWS for the different cumulus and microphysics schemes, for the ensemble with SST forcing are presented from Fig. 5 to Fig. 7. Although the computer simulations can reproduce the best track path, they are not capable enough to predict the MWS

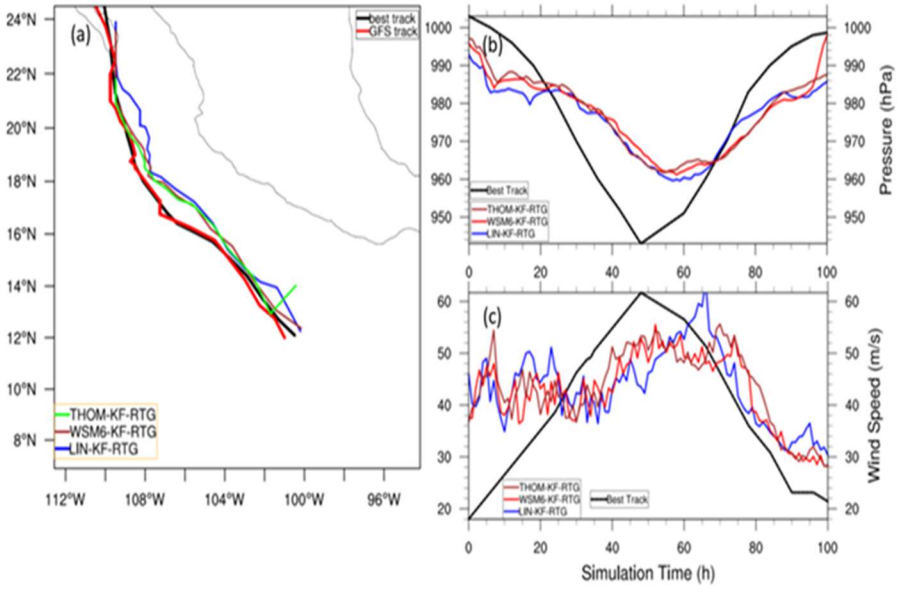


Fig. 5. (a) Track paths with longitude (deg) and latitude (deg) coordinates for THOM-KF-RTG, WSM6-KF-RTG, and LIN-KF-RTG (with SST forcing) ensemble members. (b) Sea Level Pressure (SLP) for each of the ensemble members in (a). (c) Wind Speed, the maximum values of TC Bud (MWS), for each of the ensemble members in (a). Best track observations are also included; as well as the track from the Global Forecasting System (GFS) in (a), shown in red

In this dataset, the files that contain the IBCs are made with the same model used in the Global Forecast System (GFS). To obtain more inputs, the data was interpolated at intervals of 3 h and used to update the outermost domain during each simulation, every 3 h. The innermost domain, covering the MTC region and the adjacent Pacific Ocean, is comprehended by a horizontal grid of 886 × 886 points. The model was configured with 51 vertical levels, using the new hybrid sigma-pressure vertical coordinate that follows the terrain and gradually makes the transition to constant pressure surfaces, reducing the numerical noise at higher levels [20, 21].

A 3 s time-step for the innermost domain was selected via the evolution of the simulations. A time-step ratio of three was chosen for the middle and outermost domains. In comparison to greater time-steps, a short time-step ensures a better simulation performance in terms of satisfying the Courant-Friedrich-Lewy condition [22]. The model needed to be warmed up (spin up) 17 h before analysis since simulation results are sensitive to the spin-up time [23].

Three different microphysics and cumulus schemes, with and without convection triggering, were varied one at a time–while the other schemes were left unchanged in each ensemble. The microphysics schemes used were the Purdue Lin scheme [24], the WRF single-moment 6-class scheme [25], and the Thompson scheme [26]. The cumulus schemes used were the Kain-Fritsch scheme [27], the Betts-Miller-Janjic scheme [28], and the simplified Arakawa-Schubert scheme [29]. The unchanged schemes were the Yonsei University scheme [30] for the planetary boundary layer, the MM5 similarity scheme [31, 32, 33, 34, 35] for the surface layer, the unified Noah land surface model [36] for the land surface, the Dudhia shortwave scheme [37] for the shortwave radiation, and the RRTM Longwave scheme [38] for the longwave radiation. A moisture advection-based trigger scheme was also considered.

In the first ensemble, the SST was provided, from the Real-time Global Sea Surface Temperature Analysis product (RTG-SST), for the period simulated over the ocean surface. In the second ensemble, just the IBCs from NCEP FNL were used, and the names for each ensemble member will end in "-GFS." Each ensemble member of the first ensemble, which performed better than the second, is listed in Table 1. During the numerical simulations, the SST was modified according to the response of the surface winds and changes in radiative fluxes using the scheme proposed by [39]. The change in SST on a daily scale is not considerable; however, subtle changes could indirectly impact the simulated atmosphere. During TCs, surface winds cause ocean mixing over the first few meters, changing the SSTs.

Table 2. Mean track error (in km): ensembles without SST forcing

LIN-KF-GFS	110.062
LIN-BMJ-GFS	83.624
LIN-SAS-GFS	44.6
LIN-KF-TR-GFS	44.551
THOM-KF-GFS	64.651
THOM-BMJ-GFS	95.147
THOM-KF-TR-GFS	35.337
THOM-SAS-GFS	46.612
WSM6-KF-GFS	79.583
WSM6-BMJ-GFS	76.741
WSM6-SAS-GFS	56.172
WSM6-KF-TR-GFS	34.471

Table 3. Mean track error (in km) taking into account SST forcing

THOM-KF-RTG	47.002
WSM6-KF-RTG	47.031
LIN-KF-RTG	70.041

Table 4. Mean track error (km) taking into account SST forcing

THOM-KF-TR-RTG	32.207
WSM6-KF-TR-RTG	26.314
LIN-KF-TR-RTG	39.551

Table 5. Mean track error (km) taking into account SST forcing

THOM-BMJ-RTG	85.028
WSM6-BMJ-RTG	68.875
LIN-BMJ-RTG	78.262

Table 6. Mean track error (km) taking into account SST forcing

THOM-SAS-RTG	57.524
WSM6-SAS-RTG	46.091
LIN-SAS-RTG	62.864

According to [40], negative feedback from wind-driven ocean mixing can be reflected in numerical simulations.

4 Results

Both cumulus and microphysics schemes play an important role when forecasting TCs. Cumulus schemes are associated with redistribution of moisture and heat, in the vertical direction, but they do not depend on the latent heat generated during condensation and deposition of water vapor. Besides, latent heat is important in the vertical development of convection inside TCs. During lifting, latent heat is released due to condensation or deposition. On the other hand, microphysics schemes depend on the moisture distribution in the troposphere, where heat transfer occurs between the ocean surface and the uppermost layers of the atmosphere.

The simulated tracks with and without SST forcing were compared with the best track data. The simulated paths closer to the best track path are those obtained from the ensemble members with SST forcing, which are discussed in this section. However, in Table 2, the mean track error for the ensemble members without SST forcing is shown. The tracking error was calculated as the distance between a point in the simulated track path and a point in the best track path along the corresponding great circle for all the ensemble members. Then, the average track error for each track was obtained.

Fig. 4a provides the simulated track paths for the WSM6-KF-RTG (with SST forcing) and WSM6-KF-GFS (without SST forcing) of TC

Bud. It is observed that the ensemble member with SST forcing is closer to the best track path than the one without SST forcing. Besides, in Fig. 4b, the SLP for each of the THOM-KF-GFS, WSM6-KF-GFS, and LIN-KF-GFS ensemble members, along with the best track SLP (in black), is shown. The SST was not allowed to evolve with the modeled atmosphere with the Kain-Fritsch cumulus parameterization in the latter ensemble members.

It is noticed that the mentioned ensemble members overestimated the SLP of the TC. The same occurred for all other ensemble members of the ensemble without SST forcing. However, the simulated SLP improves when SST is allowed to evolve with the modeled atmosphere for most ensemble members with SST forcing.

This can be appreciated in Fig. 5b when the SLP improves compared to Fig. 4b.

The simulated track paths, the SLP along the tracks, and the MWS for the different cumulus and microphysics schemes, for the ensemble with SST forcing are presented from Fig. 5 to Fig. 7. Although the computer simulations can reproduce the best track path, they are not capable enough to predict the MWS. The mean track errors for the THOM-KF-RTG and the WSM6-KF-RTG ensemble members, shown in Table 3, when the Kain-Fritsch cumulus scheme is used, are lower than the mean track error for the LIN-KF-RTG ensemble member. This is appreciated in the track paths shown in Fig. 5a. However, in the first hours of simulation, the track from THOM-KF-RTG considerably departs from the best track, but it presents the lowest mean track error along with the whole TC event.

In Table 4, the mean track error for the THOM-KF-TR-RTG, WSM6-KF-TR-RTG, and LIN-KF-TR-RTG ensemble members are shown. These ensemble members consider a moisture-advection-based trigger scheme for the Kain-Fritsch cumulus scheme. The latter considerably improved the mean track error for the three microphysics schemes.

In these three ensemble members, with three different microphysics schemes, the THOM-KF-TR-RTG was the one that performed best, appreciated in Fig. 6a. However, the LIN-KF-TR-RTG is the ensemble member that best describes pressure reduction; see Fig. 6b. The THOM-KF-TR-RTG and LIN-KF-TR-RTG ensemble members qualitatively describe wind speed increase, although they do not reach the maximum value of the best track data for this variable.

Microphysics schemes that depend on cumulus schemes at coarse grid spacing domains are important in regional weather models for providing atmospheric heat and moisture tendencies [41].

Vertical flux of cloud, precipitation, and sedimentation processes of hydrometeors are also included in the microphysics schemes. The microphysics schemes used, as aforementioned,

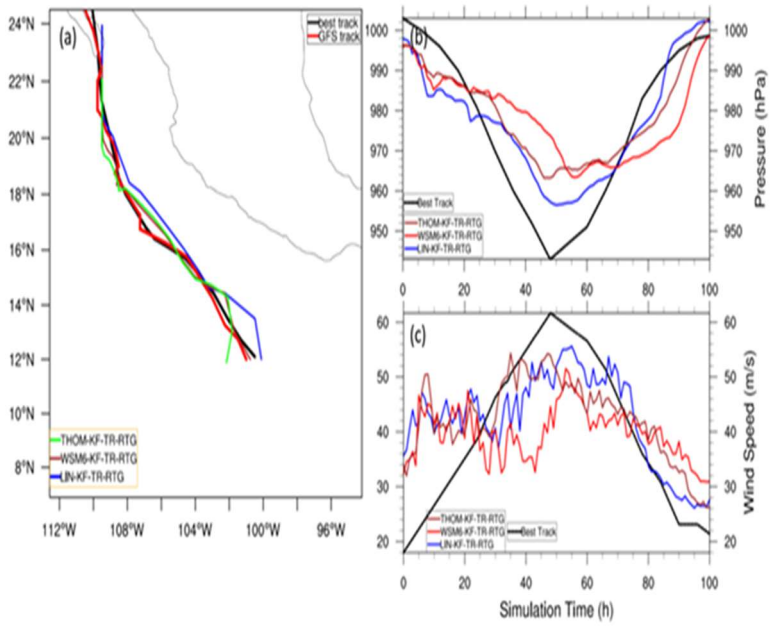


Fig. 6. As in Fig. 5, but for THOM-KF-TR-RTG, WSM6-KF-TR-RTG, and LIN-KF-TR-RTG ensemble members, see Table 1 for acronyms

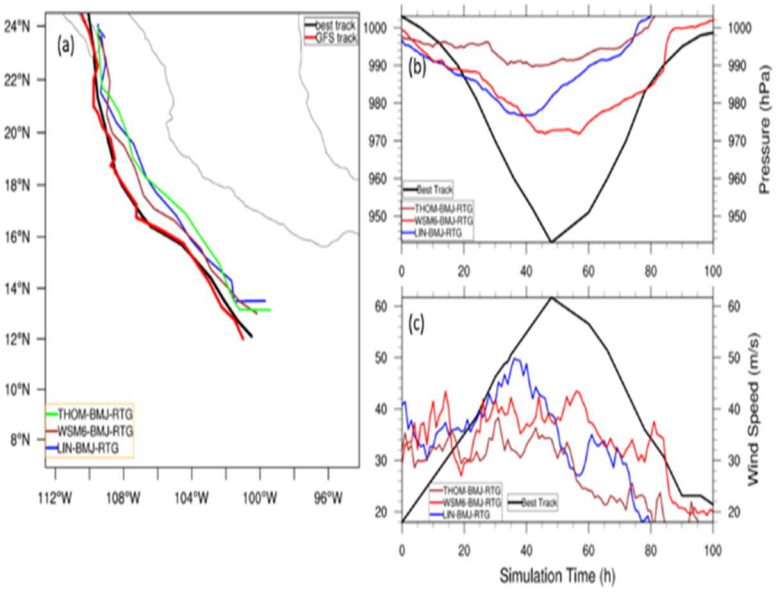


Fig. 7. As in Fig. 5, but for THOM-BMJ-RTG, WSM6-BMJ-RTG, and LIN-BMJ-RTG ensemble members, see Table 1 for acronyms

were the Purdue Lin, the WRF single moment 6-class, and the Thompson scheme.

The first includes cloud water, cloud ice, non-precipitable water, rain snow, and graupel [24]. In the second scheme, the ice crystal number concentration is expressed as a function of the ice amount [25]. Finally, the Thompson scheme is a 6-class microphysics scheme with graupel, ice, and

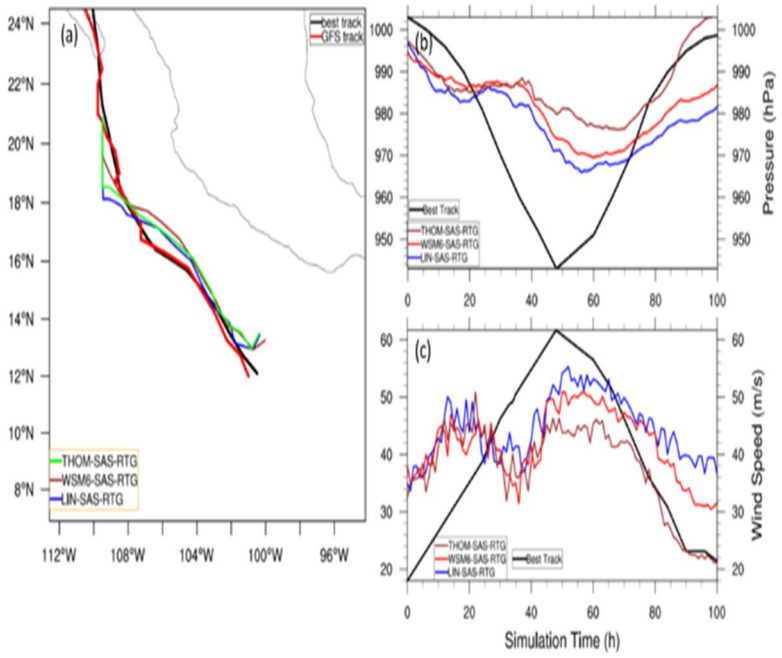


Fig. 8. As in Fig. 5, but for THOM-SAS-RTG, WSM6-SAS-RTG, and LIN-SAS-RTG ensemble members, see Table 1 for acronyms

rain number. It includes a generalized gamma distribution for each hydrometeor species with snow parameterization, depending on the ice water content and the temperature.

When the Betts-Miller-Janjic cumulus scheme was used in the computer simulations, the mean track errors, when compared to the best track data, were worse than the previous results. In Table 5, the mean track error for each ensemble member with this cumulus scheme is shown.

The simulated MWSs was underestimated by all the different combinations that defined each ensemble member. The SLP did not exactly reproduce the pressure evolution from the best track information, but this variable described the observed minimum for each ensemble members with SST forcing. The mean SLP for the ensemble members with the Simplified Arakawa-Schubert cumulus scheme is presented in Fig. 8b.

The evolution of MWS for each ensemble member is shown in Fig. 8c. The mean track errors for the THOM-SAS-RTG and the LIN-SAS-RTG ensemble members did not improve, showing greater values than those without SST forcing, presented in Table 6.

5 Conclusion

The simulation of TC Bud using the WRF-ARW model, using the cluster CÓDICE B2, is documented by studying the sensitivity of different cumulus and microphysics schemes. This event was selected because TC Bud reached category-4 hurricane status, spending its lifespan over the Eastern Pacific Ocean and dissipating over the Gulf of California without traversing Mexico. Based on our experimental results, we summarize the following findings:

i. Overall, the computer simulations have performed well in predicting the track path of the TC.

ii. Nonetheless, in any case, regardless of the cumulus and the microphysics schemes, the simulations underestimated the strength

(expressed by either the SLP or the MWS) of the cyclone.

iii. The best ensemble members were those in which the moisture-advection-based trigger scheme, with the Kain-Fritsch cumulus scheme, was used, regardless of the microphysics schemes. This was reflected in the obtained best mean track error values.

iv. The track paths calculated from the THOM-SAS-RTG and the LIN-SAS-RTG ensemble members did not show an improvement compared to those that did not include SST forcing.

Acknowledgments

This work has been funded in part by *Instituto Politécnico Nacional* under grant SIP-2021083. We thank the Centro de Investigación en Computación of the Instituto Politécnico Nacional, at Mexico City, for providing us with high-performance-computing resources, through the CÓDICE B2 supercomputer. The analysis and visualizations in Fig. 1a were produced with the Giovanni online data system, developed and maintained by the NASA GES DISC.

References

1. **Emanuel, K. (2010).** Tropical cyclone activity downscaled from NOAA-CIRES reanalysis, 1908–1958. Journal of Advances in Modeling Earth Systems (JAMES), Vol. 2, No. 1, pp. 1–12. DOI: 10.3894/james.2010.2.1.
2. **Hamill, T. M., Whitaker, J. S., Fiorino, M., Benjamin, S. G. (2011).** Global ensemble predictions of 2009's tropical cyclones initialized with an ensemble Kalman filter. Monthly Weather Review, Vol. 139, pp. 668–688. DOI: 10.1175/2010mwr3456.1.
3. **Emanuel, K. (2005).** Increasing destructiveness of tropical cyclones over the past 30 years. Nature, Vol. 436, pp. 686–688. DOI: 10.1038/nature03906.
4. **Bouwer, L. M. (2011).** Have disaster losses increased due to anthropogenic climate change? Bulletin of the American Meteorological Society, Vol. 92, No. 1, pp. 39–46. DOI: 10.1175/2010bams3092.1.
5. **Mendelsohn, R., Emanuel, K., Chonabayashi, S., Bakkensen, L. (2012).** The impact of climate change on global tropical cyclone damage. Nature Climate Change, Vol. 2, No. 3, pp. 205–209. DOI: 10.1038/nclimate1357.
6. **Dominguez, C., Magaña, V. (2018).** The role of tropical cyclones in precipitation over the tropical and subtropical North America. Frontiers in Earth Science, Vol. 6. DOI: 10.3389/feart.2018.00019.
7. **Rao, D. V. B., Srinivas, D. (2014).** Multi-Physics ensemble prediction of tropical cyclone movement over Bay of Bengal. Natural Hazards, Vol. 70, No. 1, pp. 883–902. DOI: 10.1007/s11069-013-0852-2.
8. **Bongirwar, V., Rakesh, V., Kishtawal, C. M., Joshi, P. C. (2011).** Impact of satellite observed microwave SST on the simulation of tropical cyclones. Natural Hazards, Vol. 58, No. 3, pp. 929–944. DOI: 10.1007/s11069-010-9699-y.
9. **Liu, J., Yang, S., Ma, L., Bao, X., Wang, D., Xu, D. (2013).** An initialization scheme for tropical cyclone numerical prediction by enhancing humidity in deep-convection region. Journal of Applied Meteorology and Climatology, Vol. 52, No. 10, pp. 2260–2277. DOI: 10.1175/jamc-d-12-0310.1.
10. **Xue, M., Schleif, J., Kong, F., Thomas, K. W., Wang, Y., Zhu, K. (2013).** Track and intensity forecasting of Hurricanes: impact of convection-permitting resolution and global ensemble Kalman filter analysis on 2010 Atlantic season forecasts. Weather and Forecasting, Vol. 28, No. 6, pp. 1366–1384. DOI: 10.1175/waf-d-12-00063.1.
11. **Anisetty, S. K. A. V. P., Huang, C. Y., Chen, S. Y. (2014).** Impact of FORMOSAT-3/COSMIC radio occultation data on the prediction of super cyclone Gonu (2007): a case study. Natural Hazards, Vol. 70, No. 2, pp: 1209–1230. DOI: 10.1007/s11069-013-0870-0.
12. **Osuri, K. K., Mohanty, U. C., Routray, A., Kulkarni, M. A., Mohapatra, M. (2012).** Customization of WRF-ARW model with

physical parameterization schemes for the simulation of tropical cyclones over North Indian Ocean. Natural Hazards, Vol. 63, No. 3, pp. 1337–1359. DOI: 10.1007/s11069-011-9862-0.

13. **Chandrasekar, R., Balaji, C. (2012).** Sensitivity of tropical cyclone Jal simulations to physics parameterizations. Journal of Earth System Science, Vol. 121, No. 4, pp. 923–946. DOI: 10.1007/s12040-012-0212-8.
14. **Fadnavis, S., Deshpande, M., Ghude, S. D., Raj, P. E. (2014).** Simulation of severe thunder storm event: a case study over Pune, India. Nat Hazards, Vol. 72, No. 2, pp. 927–943. DOI: 10.1007/s11069-014-1047-1.
15. **Srinivas, C. V., Rao, D. V. B., Yesubabu, V., Baskaran, R., Venkatraman, B. (2013).** Tropical cyclone predictions over the Bay of Bengal using the high-resolution Advanced Research Weather Research and Forecasting (ARW) model. Quarterly Journal of the Royal Meteorological Society, Vol. 139, No. 676, pp. 1810–1825. DOI: 10.1002/qj.2064.
16. **Munsell, E. B., Zhang, F. (2014).** Prediction and uncertainty of Hurricane Sandy (2012) explored through a real-time cloud-permitting ensemble analysis and forecast system assimilating airborne Doppler radar observations. Journal of Advances in Modeling Earth Systems (JAMES), Vol. 6, No. 1, pp. 38–58. DOI: 10.1002/2013ms000297.
17. **Srivastava, K., Bhardwaj, R. (2014).** Analysis and very short range forecast of cyclone ‘‘AILA’’ with radar data assimilation with rapid intermittent cycle using ARPS 3DVAR and cloud analysis techniques. Meteorology and Atmospheric Physics, Vol. 124, No. 1, pp. 97–111. DOI: 10.1007/s00703-014-0307-7
18. **Subramani, D., Chandrasekar, R., Ramanujam, K. S., Balaji, C. (2014).** A new ensemble based data assimilation algorithm to improve track prediction of tropical cyclones. Natural Hazards, Vol. 71, No. 1, pp. 659–682. DOI: 10.1007/s11069-013-0942-1.
19. **Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Liu, Z., Berner, J., Wang, W., Powers, J. G., Duda, M.G., Barker, D. M., et al. (2019).** A Description of the advanced research WRF version 4. NCAR Technical Note NCAR/TN-556+STR. DOI: 10.5065/1dfh-6p97.
20. **Powers, J. G., Klemp, J. B., Skamarock, W. C., Davis, C. A., Dudhia, J., Gill, D. O., Coen, J. L., Gochis, D. J., Ahmadov, R., Peckham, S. E., et al. (2017).** The weather research and forecasting model: Overview, system efforts, and future directions. Bulletin of the American Meteorological Society, Vol. 98, No. 8, pp. 1717–1737. DOI: 10.1175/BAMS-D-15-00308.1.
21. **Beck, J., Brown, J., Dudhia, J., Gill, D., Hertneky, T., Klemp, J., Wang, W., Williams, C., Hu, M., James, E., et al. (2020).** An Evaluation of a hybrid, terrain-following vertical coordinate in the WRF-Based RAP and HRRR models. Weather and Forecasting, Vol. 35, No. 3, pp. 1081–1096. DOI: 10.1175/WAF-D-19-0146.1.
22. **Gnedin, N. Y., Semenov, V. A., Kravtsov, A. V. (2018).** Enforcing the Courant–Friedrichs–Lewy condition in explicitly conservative local time stepping schemes. Journal of Computational Physics, Vol. 359, pp. 93–105. DOI: 10.1016/j.jcp.2018.01.008.
23. **Bonekamp, P. N. J., Collier, E., Immerzeel, W. W. (2018).** The impact of spatial resolution, land use, and spinup time on resolving spatial precipitation patterns in the Himalayas. Journal of Hydrometeorology, Vol. 19, No. 10, pp. 1565–1581. DOI: 10.1175/JHM-D-17-0212.1.
24. **Chen, S. -H., Sun, W. -Y. (2002).** A one-dimensional time dependent cloud model. Journal of the Meteorological Society of Japan, Vol. 80, No. 1, pp. 99–118. DOI: 10.2151/jmsj.80.99
25. **Hong, S. -Y., Lim, J. -O., J. (2006).** The WRF single-moment 6-class microphysics scheme (WSM6). Journal of the Korean Meteorological Society, Vol. 42, No. 2, pp. 129–151.
26. **Thompson, G., Field, P. R., Rasmussen, R. M., Hall, W. D. (2008).** Explicit forecasts of winter precipitation using an improved bulk microphysics scheme. Part II: Implementation of a New Snow Parameterization. Monthly Weather Review, Vol. 136, No. 12, pp. 5095–5115. DOI: 10.1175/2008MWR2387.1

27. **Kain, J. S., (2004).** The Kain-Fritsch convective parameterization: An update. Journal of Applied Meteorology and Climatology, Vol. 43, No. 1, pp. 170–181. DOI: 10.1175/1520-0450(2004)043<0170:TKCPAU>2.0.CO;2.
28. **Janjic, Z. I. (1994).** The step-mountain eta coordinate model: Further developments of the convection, viscous sublayer, and turbulence closure schemes. Monthly Weather Review, Vol. 122, No. 5, pp. 927–945. DOI: 10.1175/1520-0493(1994)122<0927:TSMECM>2.0.CO;2.
29. **Han, J., Pan, H. -L. (2011).** Revision of convection and vertical diffusion schemes in the NCEP global forecast system. Weather Forecasting, Vol. 26, No. 4, pp. 520–533. DOI: 10.1175/WAF-D-10-05038.1.
30. **Hong, S. -Y., Noh, Y., Dudhia, J. (2006).** A new vertical diffusion package with an explicit treatment of entrainment processes. Monthly Weather Review, Vol. 134, No. 9, pp. 2318–2341. DOI: 10.1175/MWR3199.1.
31. **Paulson, C. A. (1970).** The mathematical representation of wind speed and temperature profiles in the unstable atmospheric surface layer. Journal of Applied Meteorology and Climatology, Vol. 9, No. 6, pp. 857–861. DOI: 10.1175/1520-0450(1970)009<0857:TMROWS>2.0.CO;2.
32. **Dyer, A. J., Hicks, B. B. (1970).** Flux-gradient relationships in the constant flux layer. Quarterly Journal of the Royal Meteorological Society, Vol. 96, No. 410, pp. 715–721. DOI: 10.1002/qj.49709641012.
33. **Webb, E. K. (1970).** Profile relationships: The log-linear range, and extension to strong stability. Quarterly Journal of the Royal Meteorological Society, Vol. 96, No. 407, pp. 67–90. DOI: 10.1002/qj.49709640708.
34. **Beljaars, A. C. M. (1994).** The parameterization of surface fluxes in large-scale models under free convection. Quarterly Journal of the Royal Meteorological Society, Vol. 121, No. 522, pp. 255–270. DOI: 10.1002/qj.49712152203.
35. **Zhang, D., Anthes, R. A. (1982).** A high-resolution model of the planetary boundary layer sensitivity tests and comparisons with SESAME-79 data. Journal of Applied Meteorology (1962–1982), Vol. 21, No. 11, pp. 1594–1609. DOI: 10.1175/1520-0450(1982)021<1594:AHRMOT>2.0.CO;2.
36. **Tewari, M., Chen, F., Wang, W., Dudhia, J., LeMone, M. A., Mitchell, K., Ek, M., Gayno, G., Wegiel, J., Cuenca, R. H. (2004).** Implementation and verification of the unified NOAH land surface model in the WRF model. 20th Conference on weather analysis and forecasting/16th conference on numerical weather prediction, pp. 11–15.
37. **Dudhia, J., (1989).** Numerical study of convection observed during the Winter Monsoon Experiment using a mesoscale two-dimensional model. Journal of the Atmospheric Sciences, Vol. 46, No. 20, pp. 3077–3107. DOI: 10.1175/1520-0469(1989)046<3077:NSOCOD>2.0.CO;2.
38. **Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., Clough, S. A. (1997).** Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. Journal of Geophysical Research, Vol. 102, No. D14, pp. 16663–16682. DOI: 10.1029/97JD00237.
39. **Zeng, X., Beljaars, A. (2005).** A prognostic scheme of sea surface skin temperature for modeling and data assimilation. Geophysical Research Letters, Vol. 32, No. 14. DOI: 10.1029/2005GL023030.
40. **Emanuel, K., DesAutels, C., Holloway, C., Korty, R. (2004).** Environmental control of tropical cyclone intensity. Journal of the Atmospheric Sciences, Vol. 61, No. 7, pp. 843–858. DOI: 10.1175/1520-0469(2004)061<0843:ECOTCI>2.0.CO;2.
41. **Nasrollahi, N., AghaKouchak, A., Li, J., Gao, X., Hsu, K., Sorooshian, S. (2012).** Assessing the impacts of different WRF precipitation physics in hurricane simulations. Weather and Forecast, Vol. 27, No. 4, pp. 1003–1016. DOI: 10.1175/waf-d-10-05000.1.

Article received on 04/11/2021; accepted on 01/03/2022.
Corresponding author is Ricardo Barrón-Fernández.

Fuzzy Distribution Sets

Ildar Z. Batyrshin

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

batyr1@gmail.com

Abstract. We introduce a new type of fuzzy set called Fuzzy Distribution Set (FDS). Fuzzy distribution sets are fuzzy sets defined on a finite domain subject to a sum of membership values equal to 1. Such fuzzy sets can serve as models of subjective probability distributions and subjective weight distributions. Considering these distributions as fuzzy sets gives a possibility to extend on such distributions the operations of fuzzy sets and, more generally, the calculus of fuzzy restrictions developed during the last decades. Recently Yager introduced the concept of negation of probability distributions. In our works, we studied several classes of such negations. Here we consider an involutive negation of probability distributions as a complement of FDS. We introduce the operations of union and intersection of fuzzy distribution sets. These basic operations on FDS can serve as a basis for the application of fuzzy logic methods to subjective probability and weight distributions. These operations can be used for the development of reasoning models with subjective probability distributions and subjective weighting functions. Weight distributions can be used in multi-criteria, multi-person, and multi-attribute decision-making models.

Keywords. Fuzzy set, probability distribution, weight distribution, complement of distributions, union of distributions, intersection of distributions.

1 Introduction

Recently R. Yager introduced an operation of the negation of probability distributions [1]. Such negation operation can be used for formalizing sentences like "NOT High Price," where "High Price" denotes a probability distribution defined on a set of prices. Dozen of papers used Yager's negation for modeling uncertain information related to probability distributions (pd) [4].

Also, some new negations of pd have been proposed [2-4]. Since probability distributions can be considered as some kind of probabilistic predicates, the natural question appears: How to define the operations of disjunction and conjunction of probability distributions? The disjunction operation could give a possibility to formalize sentences like "High OR Very High Price." On the other hand, one can use the conjunction operations for formalizing sentences like "Which of my friends called my father AND promised to come today?". Here, subjective probability distributions "called" and " promised to come today " are defined on the set of my friends.

We construct new logical operations on the set of probability distributions as extensions of operations used in fuzzy set theory [5-12]. A probability distribution is considered a special type of fuzzy set called Fuzzy Distribution Set (FDS). We define operations AND, OR, and NOT on the set of probability distributions as operations of intersection, union, and complement of these fuzzy sets. The involutive negation of probability distributions proposed in [3] is used as a complement of fuzzy distribution sets. The union and intersection operations of FDS are constructed as extensions of similar operations defined on the set of fuzzy sets. Formally, as a fuzzy distribution set, one can consider a weight distribution used in multi-criteria, multi-person, and multi-attribute decision-making. For this reason, fuzzy distribution sets can also serve as models of subjective weight distributions.

In such an interpretation, the fuzzy set-theoretic operations considered in this paper can also be used as operations on the set of subjective weight distributions.

The paper has the following structure. Section 2 gives a definition of a fuzzy distribution set and defines a complement of such fuzzy sets as an involutive negation of probability distributions. In Section 3, we introduce the union operations of FDS. Sections 4 and 5 propose the methods of construction of the intersection of FDS. Section 6 contains an example of the intersection and union of subjective probability distributions. Section 7 contains the conclusion.

2 Complement of Fuzzy Distribution Sets

Let $X = \{x_1, \dots, x_n\}$ be a finite non-empty set, $(n > 1)$.

Definition 1. A *fuzzy distribution set* (FDS) defined on X is a function $d: X \to [0,1]$ subject to:

$$\sum_{i=1}^{n} d(x_i) = 1. \quad (1)$$

Denote $d_i = d(x_i), \quad i = 1, \dots, n.$ From (1), we have:

$$\sum_{i=1}^{n} d_i = 1. \quad (2)$$

The set of *membership values* $D = \{d_1, \dots, d_n\}$ will be called a *distribution*. When several distributions considered in some problem are defined on the same domain $X = \{x_1, \dots, x_n\}$ it is convenient to represent a distribution $D = \{d_1, \dots, d_n\}$ as n-tuple $D = (d_1, \dots, d_n)$. If it is not confusing, we will use both of these representations of distributions defined on a domain X. In such notations, the distribution $D = (0.8, 0.2, 0, \dots, 0)$ will denote the distribution D with: $d_1 = 0.8$, $d_2 = 0.2$, and $d_i = 0$ for $i = 3, \dots, n$.

The distribution $D_{(i)} = (d_1, \dots, d_n)$ satisfying the property: $d_i = 1$ for some $i = 1, \dots, n$, and $d_j = 0$ for all $j \neq i$, will be referred to as a *degenerate* or *point distribution*. For example, for $i = 1$ and $i = n$ we have the following point distributions: $D_{(1)} = (1,0, \dots,0)$, and $D_{(n)} = (0, \dots,0,1)$.

The simplest example of distribution is the *uniform distribution*:

$$D_U = \left(\frac{1}{n}, \dots, \frac{1}{n}\right).$$

Let $\mathcal{D}_n$ be the set of all fuzzy distribution sets defined on the set $X = \{x_1, \dots, x_n\}$.

Definition 2. A *complement* of a fuzzy distribution set is a function $com: \mathcal{D}_n \to \mathcal{D}_n$ such that for any fuzzy distribution set $D = (d_1, \dots, d_n)$ in $\mathcal{D}_n$ the distribution $C = com(D) = (c_1, \dots, c_n)$ satisfies for all $i, j = 1, \dots, n$, the following properties:

$$\text{if } d_i \leq d_j, \text{then } c_i \geq c_j.$$

From the definition of the complement, it follows for all $i, j = 1, \dots, n$:

$$0 \leq c_i \leq 1, \qquad \sum_{i=1}^{n} c_i = 1,$$

$$\text{if } d_i = d_j, \text{ then } c_i = c_j.$$

A *negator* N is a function of distribution values d_i taking values in [0,1] point-by-point transforming distribution $D = (d_1, \dots, d_n)$ into its complement:

$$com(D) = \big(N(d_1), \dots, N(d_n)\big).$$

Hence, for all $i, j = 1, \dots, n$, the following properties are satisfied [2]:

$$0 \leq N(d_i) \leq 1, \quad \sum_{i=1}^{n} N(d_i) = 1,$$

$$\text{if } d_i \leq d_j, \text{ then } N(d_i) \geq N(d_j),$$

$$\text{if } d_i = d_j, \text{ then } N(d_i) = N(d_j).$$

We will say that a negator N *generates* a complement $com(D) = \big(N(d_1), \dots, N(d_n)\big)$ of FDS D and distribution $D = (d_1, \dots, d_n)$.

A negator N is called a *distribution-independent* [2] if for any distribution $D = (d_1, \dots, d_n)$ in $\mathcal{D}_n$ the negator $N(d_i)$ depends only on the value d_i but not on other values d_j from D. A negator that is not distribution-independent will be referred to as *distribution-dependent*.

Consider examples of negators [2-4].

Yager negator is defined for all d in [0,1] as follows [1]:

$$N_Y(d) = \frac{1-d}{n-1}.$$

It is a distribution-independent negator. For any distribution $D = (d_1, \dots, d_n)$ in $\mathcal{D}_n$ it defines a complement of D as follows:

$$com_Y(D) = \left(\frac{1-d_1}{n-1}, \dots, \frac{1-d_n}{n-1}\right).$$

The *uniform negator* is defined for all d in [0,1] as follows [2]:

$$N_U(d) = \frac{1}{n}.$$

It is another example of a distribution-independent negator. For any distribution $D = (d_1, \dots, d_n)$ in $\mathcal{D}_n$ negator N_U defines its complement as follows:

$$com_U(D) = \left(\frac{1}{n}, \dots, \frac{1}{n}\right) = D_U.$$

The following negator N_B, introduced in [3], is defined for any distribution $D = (d_1, \dots, d_n)$ in $\mathcal{D}_n$ and all d_i, $i = 1, \dots, n$ as follows:

$$N_B(d_i) = \frac{\max(D)+\min(D)-d_i}{n(\max(D)+\min(D))-1} = \frac{MD-d_i}{nMD-1}. \quad (3)$$

where $\max(D) = \max\{d_1, \dots, d_n\}$, $\min(D) = \min\{d_1, \dots, d_n\}$ and $MD = \max(D) + \min(D)$. This negator is an example of a distribution-dependent negator. It depends not only on membership value d_i, but also on maximal and minimal values of the distribution D.

Generally, this distribution-dependent negator can be denoted, for example, as $N_B(D, d_i)$, but, for simplicity of notations, we denote it here as $N_B(d_i)$.

The complement com of fuzzy distribution sets is called *involutive*, if for all distributions $D = (d_1, \dots, d_n)$ in $\mathcal{D}_n$ it satisfies the following *involutivity property*:

$$com\big(com(D)\big) = D.$$

This property is fulfilled for the complement $\overline{A}$ of crisp, non-fuzzy sets: $\overline{\overline{A}} = A$, so it is also convenient to have involutive complements for fuzzy distribution sets. The complements of FDS based on Yager negator, uniform negator, and many other negators are non-involutive [2-4]. The complement of FDS based on the negator N_B (3):

$$com_B(D) = \big(N_B(d_1), \dots, N_B(d_n)\big).$$

is involutive, satisfying the property:

$$com_B\big(com_B(D)\big) = D.$$

3 Union of Fuzzy Distribution Sets

As in fuzzy sets theory, we can define the union of fuzzy distribution sets element-by-element using a disjunction operation (disjunctor) applied to elements of FDS. Let us use t-conorms [9] as such disjunctors for constructing the union of FDS.

T-conorm is a function $S: [0,1] \times [0,1] \to [0,1]$, such that for all a, b, c in [0,1] the following properties are satisfied [9]:

$S(a,b) = S(b,a)$ (commutativity),

$S\big(a, S(b,c)\big) = S(S(a,b), c)$ (associativity),

$S(a,b) \le S(a,c)$, whenever $b \le c$ (monotonicity),

$S(a,0) = S(0,a) = a$ (boundary condition).

The following boundary condition follows from the properties of t-conorms:

$S(1,a) = S(a,1) = 1.$

Here are the examples of the basic t-conorms:

$S_M(a,b) = max(a,b)$ (maximum),

$S_P(a,b) = a + b - ab$ (probabilistic sum).

For all t-conorms S and all a, b in [0,1] it is fulfilled:

$$S_M(a,b) \le S(a,b).$$

Definition 3. A *union* of fuzzy distribution sets based on T-conors S is a function $un: \mathcal{D}_n \times \mathcal{D}_n \to \mathcal{D}_n$, defined for all FDS $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ in $\mathcal{D}_n$ as follows:

$$un(A,B) = C = (c_1, \dots, c_n), \quad (4)$$

where

$$c_i = \frac{S(a_i,b_i)}{\sum_{i=1}^{n} S(a_i,b_i)}, i = 1, \dots, n. \quad (5)$$

It is easy to see that (4) is a distribution. From the boundary conditions and (2), it follows that the denominator of (5) is positive. From the definition of t-conorms for all $i = 1, \dots, n$, it follows:

$$0 \le c_i \le 1,$$

and from (5) we have:

$$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \frac{S(a_i,b_i)}{\sum_{i=1}^{n} S(a_i,b_i)} = 1.$$

From the properties of t-conorms, it follows that for all A, B in $\mathcal{D}_n$ the property of commutativity is satisfied:

$$un(A,B) = un(B,A).$$

The function $DIS: [0,1] \times [0,1] \to [0,1]$ defined for any distributions $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ in $\mathcal{D}_n$ by

$$DIS(a_i, b_i) = \frac{S(a_i, b_i)}{\sum_{i=1}^{n} S(a_i, b_i)} \quad (6)$$

will be called a *disjunctor*. The following properties of disjunctor (6) follow from the properties of t-conorms for all $i, j = 1, \dots, n$:

Commutativity:

$$DIS(a_i, b_i) = DIS(b_i, a_i).$$

Monotonicity:

if $a_i \le a_j$ and $b_i \le b_j$, then $DIS(a_i, b_i) \le DIS(a_j, b_j)$.

For t-conorm $S_M(a, b) = \max(a, b)$, we obtain the following disjunctor:

$$DIS_M(a_i, b_i) = \frac{max(a_i, b_i)}{\sum_{i=1}^{n} max(a_i, b_i)}.$$

For t-conorm $S_P(a, b) = a + b - ab$, we obtain the following disjunctor:

$$DIS_P(a_i, b_i) = \frac{a_i + b_i - a_i b_i}{\sum_{i=1}^{n} (a_i + b_i - a_i b_i)} = \frac{a_i + b_i - a_i b_i}{2 - \sum_{i=1}^{n} a_i b_i}.$$

4 Intersection of Fuzzy Distribution Sets Using t-Norms

T-norm is a function $T: [0,1] \times [0,1] \to [0,1]$, such that for all a, b, c in [0,1] the following properties are satisfied [9]:

$T(a, b) = T(b, a)$ (commutativity),

$T\big(a, T(b, c)\big) = T(T(a, b), c)$ (associativity),

$T(a, b) \le T(a, c)$, whenever $b \le c$ (monotonicity),

$T(a, 1) = T(1, a) = a$ (boundary condition).

The following boundary condition follows from the properties of t-norms:

$T(0, a) = T(a, 0) = 0.$

Here are the examples of the basic t-norms:

$T_M(a, b) = mi\,n(a, b)$, (minimum),

$T_P(a, b) = ab$ (product).

For all t-norms T and all a, b in [0,1], it is fulfilled:

$$T(a, b) \le T_M(a, b).$$

Definition 4. An intersection of fuzzy distribution sets based on t-norm T is a function $int: \mathcal{D}_n \times \mathcal{D}_n \to \mathcal{D}_n$, defined for all FDS $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ in $\mathcal{D}_n$, as follows:

$$int(A, B) = C = (c_1, \dots, c_n),$$

where, for all $i = 1, \dots, n$, c_i is defined by:

$$c_i = \frac{1}{n}, \quad \text{if} \quad T(a_i, b_i) = 0 \quad \text{for all } i = 1, \dots, n,$$

otherwise:

$$c_i = \frac{T(a_i, b_i)}{\sum_{i=1}^{n} T(a_i, b_i)}. \quad (7)$$

The function $CON: [0,1] \times [0,1] \to [0,1]$ defined for any distributions $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ in $\mathcal{D}_n$ by

$$CON(a_i, b_i) = \begin{cases} \frac{1}{n}, if\ T(a_i, b_i) = 0\ for\ all\ i = 1, \dots, n, \\ \frac{T(a_i, b_i)}{\sum_{i=1}^{n} T(a_i, b_i)}\ otherwise, \end{cases}$$

will be called a *conjunctor*. The following properties of conjunctor follow from the properties of t-norms for all $i, j = 1, \dots, n$:

Commutativity:

$$CON(a_i, b_i) = CON(b_i, a_i).$$

Monotonicity:

if $a_i \le a_j$ and $b_i \le b_j$, then $ON(a_i, b_i) \le CON(a_j, b_j)$.

5 Intersection of Fuzzy Distribution Sets Using Union and Complement

From De Morgan law of crisp sets:

$$\overline{A \cap B} = \overline{A} \cup \overline{B}.$$

And the involutivity of complement $\overline{\overline{A}} = A$ we have:

$$A \cap B = \overline{\overline{A} \cup \overline{B}}.$$

We will use this formula to define a new intersection of fuzzy distribution sets.

Definition 5. An intersection of fuzzy distribution sets is a function $int: \mathcal{D}_n \times \mathcal{D}_n \to \mathcal{D}_n$, defined for all FDS $A = (a_1, \dots, a_n)$ and $B = (b_1, \dots, b_n)$ in $\mathcal{D}_n$, as follows:

$$int(A, B) = com_B(un(com_B(A), com_B(B))).$$

such that in the distribution:

$$int(A,B) = C = (c_1, \dots, c_n).$$

c_i is defined for all $i = 1, \dots, n$, by:

$$c_i = N_B\left(DIS\big(N_B(a_i), N_B(b_i)\big)\right).$$

Step-by-step, we obtain:

$$N_B(a_i) = \frac{\max(A)+\min(A)-a_i}{n(\max(A)+\min(A))-1}.$$
$$N_B(b_i) = \frac{\max(B)+\min(B)-b_i}{n(\max(B)+\min(B))-1}.$$
$$DIS\big(N_B(a_i), N_B(b_i)\big) = \frac{S(N_B(a_i), N_B(b_i))}{\sum_{i=1}^{n} S(N_B(a_i), N_B(b_i))}.$$

Denoting $e_i = DIS\big(N_B(a_i), N_B(b_i)\big)$ and $E = (e_1, \dots, e_n)$, then finally we obtain:

$$c_i = N_B(e_i) = \frac{\max(E)+\min(E)-e_i}{n(\max(E)+\min(E))-1}.$$

Suppose you with your wife want to evaluate how many neighbors will come to your open party. She supposes that you will have 5-6 guests, and you suppose to have 8-9 guests. Suppose you can represent these evaluations as subjective probability distributions defined on the set:

$$X = \{4, \mathbf{5}, \mathbf{6}, 7, \mathbf{8}, \mathbf{9}, 10\}.$$

For your wife as:

$$W = (0.1, \mathbf{0.3}, \mathbf{0}, \mathbf{3}, 0.2, 0.1, 0, 0)$$

and for you, as:

$$Y = (0, 0, 0, 0.1, \mathbf{0.4}, \mathbf{0}, \mathbf{4}, 0.1).$$

Using the formula (7) with minimum and product conjunctions, you obtain intersections of these distributions as possible solutions to your problem.

Using minimum t-norm T_M you obtain

$$int_M(W,Y) = (0, 0, 0, \mathbf{0.5}, \mathbf{0.5}, 0, 0),$$

i.e., $p(7) = p(8) = 0.5$, and $p(k) = 0$, for $k = 4,5,6,9,10$.

Using product t-norm T_P you obtain:

$$int_P(W,Y) = (0, 0, 0, \mathbf{0.33}, \mathbf{0.67}, 0, 0).$$

i.e., $p(7) = 0.33, p(8) = 0.67$, and $p(k) = 0$, for $k = 4,5,6,9,10$.

Compared with initial subjective evaluations of the number of guests: 5-6, and 8-9, the applied models give 7-8 as the more probable number of guests for both t-norms used in the model, but the product t-norm evaluates the number of 8 guests as the most probable.

Evaluation of the probable number of guests using union operation gives the following results for maximum S_M and probabilistic sum S_P t-conorms:

$$un_M(W,Y) = (0.056, \mathbf{0.167}, \mathbf{0.167}, 0.111, \mathbf{0.222}, \mathbf{0.222}, 0.056)$$

$$un_P(W,Y) = \begin{pmatrix} 0.052, \mathbf{0.155}, \mathbf{0.155}, 0.144, \mathbf{0.237}, \\ \mathbf{0.206}, 0.052 \\ X = \{4, \mathbf{5}, \mathbf{6}, 7, \mathbf{8}, \mathbf{9}, 10\} \end{pmatrix}.$$

These distributions are defined on the domain.

All elements of this domain have non-zero probabilities with local maximums for 5-6 and 8-9 guests corresponding to the initial subjective distributions but considering 8-9 as more probable.

The product t-conorm refines the probability values giving the most probable value for 8 guests.

Similar methods can be used for aggregating subjectively defined weight distributions widely used in multi-criteria, multi-objective, multi-person, and multi-alternative decision-making.

7 Conclusion

The paper proposed a new type of fuzzy set that can be used to model subjective probability distributions and subjective weight distributions. Fuzzy logic allows the building of models of subjective human reasoning and behavior.

It has numerous applications in control, pattern recognition, decision-making, machine learning, and other research areas. The paper paves a new way to apply the power of fuzzy logic for modeling and processing subjective probability distributions and subjective weight distributions.

In future works, we plan to extend the methods of processing fuzzy distribution sets by applying generalized fuzzy conjunction and disjunction operations and other fuzzy logic techniques (in the wide sense) [7-12].

Acknowledgments

This work was partially supported by the project SIP 20220857 of the Mexican National Polytechnic Institute. The author thanks Dr. Grigori Sidorov for his valuable comments.

References

1. **Yager, R. R. (2014).** On the maximum entropy negation of a probability distribution. IEEE Transactions on Fuzzy Systems, Vol. 23, No. 5, pp. 1899–1902. DOI: 10.1109/TFUZZ.2014. 2374211.
2. **Batyrshin, I., Villa-Vargas, L. A., Ramirez-Salinas, M. A., Salinas-Rosales, M., Kubysheva, N. (2021).** Generating negations of probability distributions. Soft Computing, Vol. 25, No. 12, pp. 7929–7935. DOI: 10.1007/s00500-021-05802-5.
3. **Batyrshin, I. Z. (2021).** Contracting and involutive negations of probability distributions. Mathematics, Vol. 9, No. 19, pp. 1–11, DOI: 10.3390/math 9192389.
4. **Batyrshin, I. Z., Kubysheva, N. I., Bayrasheva, V. R., Kosheleva, O., Kreinovich, V. (2021).** Negations of Probability Distributions: A Survey. Computación y Sistemas, Vol. 25, No. 4, pp. 775–781. DOI: 10.13053/cys-25-4-4094.
5. **Zadeh, L. A. (1965).** Fuzzy Sets. Information and Control, Vol. 8, pp. 338–353.
6. **Zadeh, L. A. (1975).** Calculus of fuzzy restrictions. In **Zadeh, L. A., Fu, K. S., Tanaka, K., Shimura, M., editors,** Fuzzy sets and their applications to cognitive and decision processes. Academic Press, pp. 1–39. DOI: 10.1016/B978-0-12-775260-0.50006-2.
7. **Klir, G. J., Yuan, B. (Eds.). (1996).** Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh. Vol. 6. World Scientific, Vol. 6.
8. **Jang, J. S. R., Sun, C. T., Mizutani, E. (1997).** Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. IEEE Transactions On Automatic Control, Vol. 42, No. 10, pp. 1482–1484.
9. **Klement, E. P., Mesiar, R., Pap, E. (2013).** Triangular norms. In Springer Science & Business Media. DOI: 10.1007/978-94-015-9540-7.
10. **Nguyen, H. T., Walker, C., Walker, E. A. (2018).** A first course in fuzzy logic. Chapman and Hall/CRC, DOI: 10.1201/9780429505546.
11. **Batyrshin, I., Kaynak, O. (1999).** Parametric classes of generalized conjunction and disjunction operations for fuzzy modeling. IEEE Transactions on Fuzzy Systems, Vol. 7, No. 5, pp. 586–596. DOI: 10.1109/91.797981.
12. **Batyrshin, I., Kaynak, O., Rudas, I. (2002).** Fuzzy modeling based on generalized conjunction operations. IEEE Transactions on Fuzzy Systems, Vol. 10, No. 5, pp. 678–683. DOI: 10.1109/TFUZZ. 2002.803500.

Article received on 20/04/2022; accepted on 08/07/2022.

Corresponding author is Ildar Z. Batyrshin.