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:

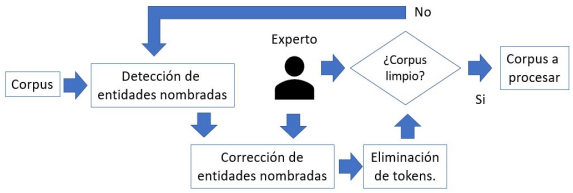

Fig. 1. Proceso de limpieza del corpus

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

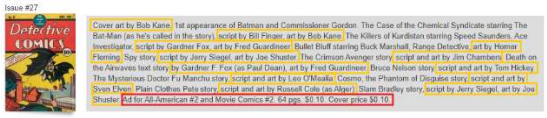

Fig. 3. Descripción del cómic Detective Comics #27 (1939)

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

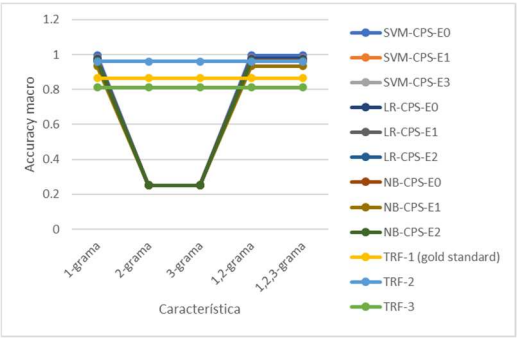

Fig. 4. *Accuracy* macro de entrenamiento

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

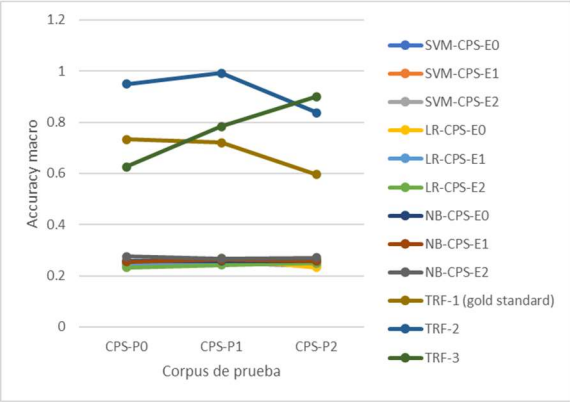

Fig. 5. Accuracy macro de pruebas

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.