Clasificación de documentos con word2vec

Lucas Antoñanzas del Villar
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
lucantdel@alum.us.es

Jaime García García
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
jaigargar1@alum.us.es

El objetivo principal de este trabajo es desarrollar un sistema de clasificación de fragmentos de texto basado en técnicas de aprendizaje supervisado y no supervisado. Para conseguirlo, utilizaremos fragmentos de artículos periodísticos, etiquetados según sus temáticas. Aplicaremos técnicas de tokenización y eliminación de palabras irrelevantes, así como el uso de stemming y la creación de un vocabulario final.

También implementaremos la técnica de Word2Vec para representar las palabras como vectores numéricos (embeddings), y entrenaremos un modelo sobre el corpus obtenido. Así, conseguiremos explorar diferentes enfoques de clasificación utilizando los resultados obtenidos con Word2Vec y evaluándolos mediante modelos de clasificación, como Naive Bayes multinomial.

Palabras Clave: Corpus, Clasificación, Stopwords, Tokenización, Stemming, Embedding

I. INTRODUCCIÓN

En este proyecto, abordaremos el desafío de clasificar fragmentos de texto en función de su temática utilizando técnicas de aprendizaje supervisado y no supervisado. Nuestro objetivo es desarrollar un sistema capaz de asignar etiquetas o categorías a documentos basándose en la información contenida en ellos.

Para lograr este objetivo, construiremos un corpus que consistirá en fragmentos de artículos periodísticos u otros tipos de documentos, que estarán etiquetados según sus temáticas.

Sobre este corpus, implementaremos varias técnicas aplicando varias librerías, para convertir los textos en vectores de palabras que contengan únicamente información relevante para el problema de clasificación. Crearemos un vocabulario final que representaremos como vectores, y nos adentraremos en el mundo de Word2Vec para explorar diferentes enfoques de clasificación utilizando los resultados obtenidos.

Entrenaremos algunos modelos, como Naive Bayes multinomial, sobre la bolsa de palabras creada antes de aplicar Word2Vec y compararemos los resultados.

A lo largo de este documento, presentaremos los procedimientos llevados a cabo y las metodologías utilizadas para ello. También discutiremos los resultados obtenidos, obteniendo las conclusiones pertinentes. Además, indicaremos las referencias bibliográficas que hemos consultado para comprender su funcionamiento.

II. PRELIMINARES

En esta sección haremos una breve introducción de las técnicas empleadas a lo largo del trabajo.

A. Métodos empleados

- **Tokenización:** es una etapa esencial en el procesamiento de lenguaje natural, que consiste en dividir un texto en unidades más pequeñas llamadas tokens. Estos tokens suelen ser palabras individuales, y el objetivo principal de la tokenización es facilitar el análisis y procesamiento posterior del texto. Al dividir el texto en tokens, se pueden realizar diversas tareas, como el conteo de palabras, la construcción de vocabularios y la clasificación de textos. Implica eliminar los espacios en blanco y utilizar delimitadores, como espacios y signos de puntuación, para separar las palabras.
- **Eliminación de stopwords:** es un paso común en el procesamiento de lenguaje natural que implica filtrar y eliminar palabras que no aportan un significado importante en el análisis de texto. Las stopwords son palabras muy frecuentes en un idioma y que no llevan consigo una carga informativa relevante para el contexto del análisis. Ayuda a reducir el ruido y a mejorar la eficiencia y calidad del análisis de texto. Al eliminar estas palabras vacías, se pueden enfocar los esfuerzos en las palabras clave que aportan un mayor significado y discriminación en la clasificación de documentos. Esto puede mejorar la precisión de los modelos de aprendizaje automático y reducir la dimensionalidad del espacio de características utilizado para representar los textos. Algunos ejemplos comunes de stopwords en inglés son "the", "and", "is", "in".
- **Stemming:** es un proceso en el procesamiento de lenguaje natural que busca reducir una palabra a su forma base o raíz, conocida como "stem". El objetivo del stemming es eliminar las terminaciones y sufijos de las palabras, manteniendo solo la parte principal que lleva consigo el significado básico. Esto ayuda a consolidar palabras similares y reducir la dimensionalidad del espacio de características utilizado para el análisis de texto. Se basa en reglas heurísticas y algoritmos para realizar esta reducción de palabras. Estas reglas buscan eliminar sufijos comunes en un

idioma y así convertir palabras flexionadas o conjugadas en su forma base. Por ejemplo, las palabras "corriendo", "correría" y "correré" se reducirían todas a la forma base "corr". Puede ayudar a mejorar la coherencia y la eficiencia en el análisis de texto, ya que palabras con raíces similares se consideran equivalentes en términos de su significado básico. Esto es especialmente útil cuando se desea realizar tareas como conteo de palabras, análisis de frecuencia o agrupamiento de textos.

- **Embedding:** se refiere a una representación numérica de las palabras o frases en un espacio vectorial. Es una técnica que permite transformar el texto en información numérica, que es más fácil de procesar por algoritmos de aprendizaje automático. Los embeddings capturan características semánticas y contextuales de las palabras o frases, lo que significa que palabras similares o relacionadas tienen representaciones vectoriales cercanas en el espacio. Esto permite que los algoritmos de aprendizaje automático comprendan mejor la similitud y relación entre las palabras durante la clasificación, análisis de sentimiento, traducción automática y otras tareas de procesamiento de lenguaje natural. Una de las técnicas más populares para generar embeddings es Word2Vec.
- **Word2Vec:** es un modelo de aprendizaje automático que se basa en la idea de que las palabras que aparecen en contextos similares tienen significados similares. Para entrenar el modelo, se utilizan técnicas de redes neuronales, específicamente modelos de lenguaje. Estos modelos buscan predecir la probabilidad de una palabra dado su contexto circundante (modelo de lenguaje de predicción) o predecir el contexto circundante dado una palabra (modelo de lenguaje skip-gram). Al entrenar Word2Vec, se genera un espacio vectorial donde cada palabra está representada por un vector denso de valores numéricos. Estos vectores capturan las relaciones semánticas y de similitud entre las palabras. Por ejemplo, las palabras que tienen significados similares o que aparecen en contextos similares tendrán vectores cercanos en el espacio. Una vez que el modelo Word2Vec ha sido entrenado, los vectores resultantes pueden ser utilizados para diversas tareas en el procesamiento de lenguaje natural. Estas representaciones vectoriales de palabras pueden ser utilizadas como características de entrada en algoritmos de aprendizaje automático, como clasificación de textos. También es posible realizar operaciones algebraicas en los vectores para descubrir relaciones semánticas, como encontrar palabras que sean análogas a otras (por ejemplo, "rey" - "hombre" + "mujer" = "reina").
- **Naive Bayes Multinomial:** es un algoritmo de clasificación que se basa en el teorema de Bayes y asume una distribución multinomial para los datos. Este algoritmo es comúnmente utilizado para la clasificación de textos, donde cada documento se representa mediante la frecuencia de ocurrencia de las palabras en un vocabulario dado. La idea básica es calcular la probabilidad de que un documento pertenezca a una determinada categoría, dado el conjunto de palabras que aparecen en el documento. Para ello, se utiliza el teorema de Bayes, que establece que la probabilidad condicional de una clase dado un conjunto de características se puede calcular a partir de la probabilidad a priori de la clase y las probabilidades condicionales de las características dada la categoría. Luego, para clasificar un nuevo documento, se utiliza el teorema de Bayes para calcular la probabilidad de pertenecer a cada clase y se selecciona la clase con la probabilidad más alta como la etiqueta de clasificación del documento. Naive Bayes Multinomial es un algoritmo rápido y eficiente, especialmente adecuado para conjuntos de datos grandes y dispersos, como los datos de texto.
- **RandomForest:** es un algoritmo de aprendizaje automático que se utiliza comúnmente en el procesamiento de lenguaje natural para la clasificación de textos. Pertenece a la categoría de algoritmos de conjunto, que combinan múltiples modelos más simples para tomar decisiones más precisas y robustas. Se basa en la construcción de un conjunto (ensemble) de árboles de decisión. Cada uno se entrena con una muestra aleatoria del conjunto de datos original y utiliza una selección aleatoria de características. Durante la clasificación, cada árbol genera una predicción y la clase final se determina por votación o promedio de las predicciones individuales. El algoritmo puede manejar tanto variables numéricas como categóricas, y es menos propenso al sobreajuste en comparación con un solo árbol de decisión. Esto se debe a la aleatoriedad introducida tanto en las muestras de entrenamiento como en las características seleccionadas para cada árbol, lo que proporciona diversidad al conjunto de árboles y reduce la correlación entre ellos.

III. METODOLOGÍA

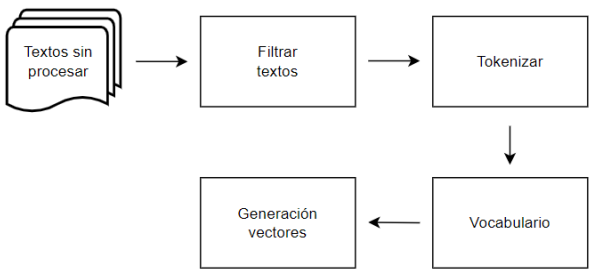


Fig. 1. Diagrama representativo del proceso realizado

1) Construcción del corpus: construiremos un corpus que consistirá en fragmentos de artículos periodísticos u otros tipos de documentos, para esto hemos utilizado la generación de un .csv a partir de muchos textos periodísticos clasificados por temáticas. Es importante tener en cuenta que el tamaño del corpus debe ser adecuado para las tareas posteriores, por lo

que es recomendable comenzar con un corpus de tamaño moderado y aumentarlo en función de los resultados obtenidos.

2) Tokenización y eliminación de stopwords: a partir de una función que transformará los textos en secuencias de palabras significativas, pasando todas las palabras a minúsculas y eliminando palabras irrelevantes (stopwords) y símbolos de puntuación.

3) Stemming: a partir de una función que además de tokenizar los textos (haciendo uso de la función creada en el apartado anterior), reduce las palabras a su forma base o raíz, usando la librería NLTK. Esto nos permitirá transformar los textos en vectores de palabras que contengan información relevante para el problema de clasificación.

4) Word2Vec: implementaremos un modelo Word2Vec para la vectorización de los textos preprocesados. El primer paso es crear el modelo, donde hay varios parámetros muy influyentes en el desempeño de este. Luego, debemos construir el vocabulario con el corpus, teniendo encuenta la frecuencia de las palabras en este. Por último, entrenaremos este modelo.

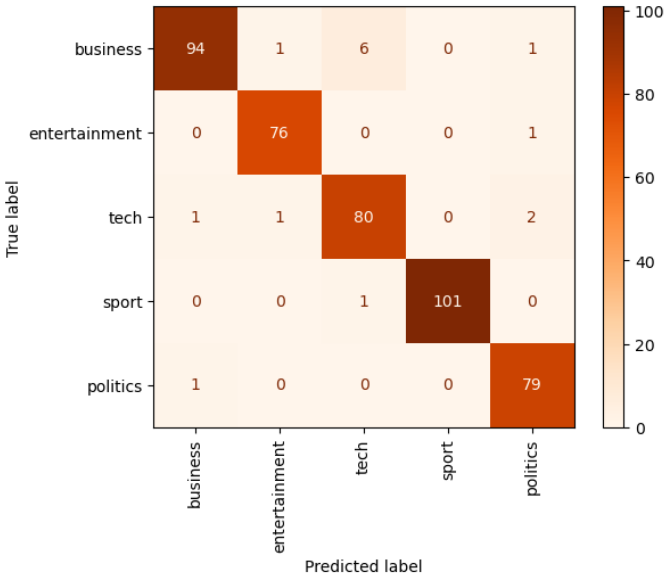
5) Scikit-Learn: siguiendo el flujo típico de trabajo de la librería scikit-learn, dividimos los textos en dos conjuntos, uno de entrenamiento, para entrenar los modelos, y otro de prueba, para predecir las categorías. Para entrenar el modelo Naive Bayes multinomial, los conjuntos serán texto, es decir, una bolsa de palabras mientras que para el modelo RandomForest los conjuntos estarán formados por vectores generados por el modelo Word2Vec del apartado anterior. Por último, mostraremos el resultado de cada uno de los modelos con un informe de la clasificación y una matriz de confusión.

IV. Resultados

Tras entrenar los modelos Naive Bayes Multinomial y Random Forest hemos obtenido los siguientes resultados al predecir las categorías del conjunto de prueba:

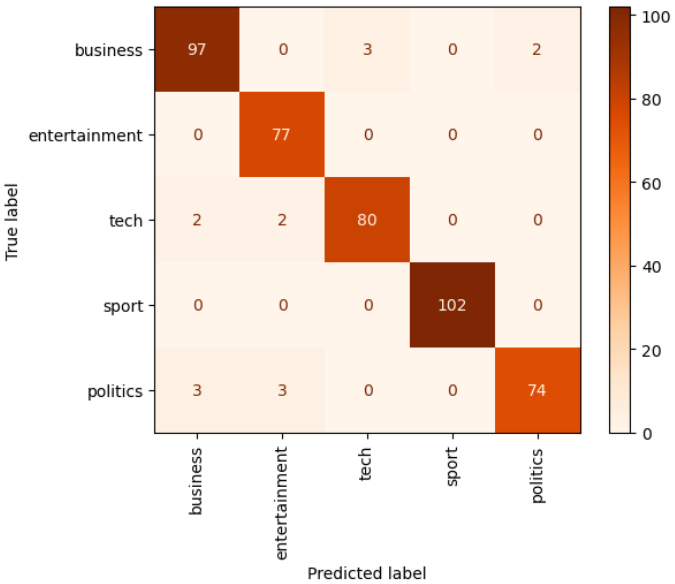
- **MultinomialNB:**

	precision	recall	f1-score	support
business	0.99	0.93	0.96	102
entertainment	0.99	0.96	0.97	77
politics	0.92	0.96	0.94	84
sport	1.00	1.00	1.00	102
tech	0.95	1.00	0.98	80
accuracy			0.97	445
macro avg	0.97	0.97	0.97	445
weighted avg	0.97	0.97	0.97	445



- **RandomForestClassifier:**

	precision	recall	f1-score	support
business	0.94	0.97	0.96	102
entertainment	0.99	0.95	0.97	77
politics	0.95	0.96	0.96	84
sport	1.00	0.98	0.99	102
tech	0.96	0.97	0.97	80
accuracy			0.97	445
macro avg	0.97	0.97	0.97	445
weighted avg	0.97	0.97	0.97	445



Como podemos observar, los resultados son muy muy similares, incluso hemos llegado a pensar que sea un error que los resultados de las predicciones sean tan buenos y parecidos, pero tras revisar varias veces el código por completo hemos

llegado a la conclusión de que hemos entrenado muy bien el modelo, y por ello hemos obtenido resultados con tan buena precisión.

V. Conclusiones

Tras la realización del trabajo, queda claro que la construcción de un corpus adecuado, la tokenización de los textos y la eliminación de stopwords han sido pasos cruciales en la preparación de los datos. A su vez, el uso de las técnicas de aprendizaje supervisado y la vectorización de textos ofrece una solución bastante efectiva para clasificar fragmentos de texto basados en su temática.

Además, la aplicación de técnicas como stemming y el uso de Word2Vec para generar embeddings, permiten capturar las relaciones semánticas y significados implícitos en el texto, mejorando así la precisión de los resultados.

La clasificación de documentos puede lograrse mediante la implementación de modelos, en este caso hemos aplicado Naive Bayes multinomial sobre la bolsa de palabras previa a la aplicación de Word2Vec.

En conjunto, estos enfoques han aportado una solución versátil y automatizada para la clasificación de textos en diversos contextos.

Referencias

[1] *Página web del curso IA de Ingeniería del Software.* https://www.cs.us.es/cursos/iais

[2] *Problem-solving with ML: automatic document classification.* https://cloud.google.com/blog/products/ai-machine-learning/problem-solving-with-ml-automatic-document-classification

[3] *Gensim Word2Vec Tutorial by Pierre Megret (Kaggle).* *https://www.kaggle.com/code/pierremegret/gensim-word2vec-tutorial#Getting-Started*

[4] *Scikit-Lean Documentation.*

https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[5] *Gensim Documentation.* https://radimrehurek.com/gensim/models/word2vec.html#

[6] *Tutorial TF-IDF vs Word2Vec For Text Classification [How To In Python With And Without CNN].* https://spotintelligence.com/2023/02/15/word2vec-for-text-classification/

[7] *ChatGPT.* https://chat.openai.com

[8] *INTRO al Natural Language Processing (NLP) #1 - ¡De PALABRAS a VECTORES!* https://www.youtube.com/watch?v=Tg1MjMIVArc&list=PL-Ogd76BhmcAQXovVph6ZBjObrdxxxeCu&index=2&ab_channel=DotCSV

[9] *INTRO al Natural Language Processing (NLP) #2 - ¿Qué es un EMBEDDING?* https://www.youtube.com/watch?v=RkYuH_K7Fx4&list=PL-Ogd76BhmcAQXovVph6ZBjObrdxxxeCu&index=10&ab_channel=DotCSV