Diseño de un modelo de aprendizaje automático para el análisis de sentimientos en noticias financieras.

Universidad Internacional de Valencia

Titulación:

Máster en Big Data y Ciencia de Datos

Curso académico

2022 – 2023

Alumno:

Beret Grande, Sergio Pablo

D.N.I: 52662708E

Director de TFM:

Francisco Pascual Romero

Convocatoria:

Tercera

De:

Planeta Formación y Universidades

Universidad Internacional de Valencia

Índice

Índice de ilustraciones

Índice de tablas

Resumen

En la actualidad, los datos financieros procedentes de las noticias y de las redes sociales desempeñan un papel importante para predecir el mercado bursátil. El crecimiento exponencial de la información financiera y las diferentes polaridades de sentimiento que diferentes sectores o partes interesadas pueden tener sobre una misma información ha llevado a la necesidad de nuevas tecnologías que recopilen y clasifiquen automáticamente grandes volúmenes de información de forma rápida y sencilla para cada parte interesada.

En este escenario, la plataforma de competiciones de ciencia de datos *CodaLab* lanzó el desafío "*IBERLEF 2023 Task - FinancES. Financial Targeted Sentiment Analysis in Spanish*". Una competición en la que, a partir de un *dataset* compuesto por titulares de medios digitales especializados en economía en castellano, propone la creación de un modelo de aprendizaje automático basado en técnicas NLP (*Natural Language Processing*) para el análisis del sentimiento financiero específico del principal objeto económico de cada titular y de otros agentes económicos relacionados.

El objetivo de este proyecto fin de máster es diseñar y probar una solución al problema planteado en dos etapas diferentes y consecutivas. En primer lugar, se propone identificar el objeto económico principal del titular de la noticia financiera y, en segundo lugar, determinar la polaridad del sentimiento (positivo, neutral o negativo) hacia dicho objetivo en el texto procesado y hacia otros dos agentes económicos como los son las empresas y los consumidores.

En la fase de identificación del objeto económico principal del titular, el mejor resultado obtenido ha sido del 0.5786 de *accuracy* con un modelo *openAI*. Para el análisis de sentimiento del titular sobre el del objeto económico principal, se ha obtenido un f1-score de 0.7922 también con un modelo *openAI*. En los análisis de sentimiento del titular con respecto a las Empresas y a los Consumidores se han obtenido unos f1-score de 0.6163 y 0.6668 respectivamente usando *spaCy* para la extracción de la palabra clave y el preprocesado, y un modelo *Random Forest* como clasificador final.

Palabras clave: Análisis de sentimientos, procesamiento del lenguaje natural, *machine learning*, noticias financieras, análisis de sentimientos dirigido.

Abstract

Currently, financial data derived from news and social media play a significant role in predicting the stock market. The exponential growth of financial information and the diverse sentiment polarities that different sectors or stakeholders may hold regarding the same information have led to the necessity of new technologies that rapidly and easily collect and classify large volumes of information for each interested party.

In this scenario, the data science competition platform CodaLab launched the challenge "IBERLEF 2023 Task - FinancES: Financial Targeted Sentiment Analysis in Spanish". This competition, based on a dataset comprising headlines from specialized Spanish-language digital media in the field of economics, proposes the creation of a machine learning model based on Natural Language Processing (NLP) techniques for analyzing the specific financial sentiment regarding the main economic subject of each headline and other related economic entities.

The aim of this master's thesis project is to design and test a solution to the proposed problem in two different consecutive stages. Firstly, the goal is to identify the main economic subject of the financial news headline, and secondly, to determine the polarity of sentiment (positive, neutral, or negative) towards this subject in the processed text, as well as towards two other economic entities, companies and consumers.

In the phase of identifying the main economic object of the headline, the best result obtained has been a 0.5786 accuracy with an openAI model. For the sentiment analysis of the headline regarding the main economic object, an f1-score of 0.7922 has been obtained, also with an openAI model. In the sentiment analyses of the headline in relation to Companies and Consumers, f1-scores of 0.6163 and 0.6668 respectively have been obtained using spaCy for keyword extraction and preprocessing, and a Random Forest model as the final classifier.

Keywords: Sentiment analysis, natural language processing, machine learning, financial news, targeted sentiment analysis.

Acrónimos

LISTA DE ACRÓNIMOS En este apartado se listan los acrónimos utilizados a lo largo de la memoria de este trabajo, añadiendo el significado de cada una de sus siglas.

AS: Análisis de sentimientos

LLMs: Grandes Modelos de Lenguaje.

NLP: Procesamiento de Lenguaje Natural.

POE: Principal Objetivo Económico.

T-PTLMs: Modelos de lenguaje preentrenados basados en transformadores.

1. Introducción

En este apartado se presenta una visión preliminar de este trabajo de fin de máster, proporcionando una breve introducción a los grandes modelos de lenguaje, al análisis de sentimientos en el ámbito financiero y los desafíos que actualmente presentan, así como la definición del problema que se pretende resolver dentro de la competición de ciencia de datos elegida como objetivo en este proyecto.

1.1. Contexto

La gestión de datos financieros en el pasado se limitaba a aplicaciones específicas para bancos y compañías financieras. Pero en la actualidad, con la aparición de la Web 2.0, las redes sociales, dispositivos de Internet de las Cosas (IoT) y la popularidad de dispositivos portátiles conectados a internet, hay enormes cantidades de datos económicos y financieros disponibles de forma pública en sitios web y redes sociales.

La disponibilidad inmediata de estos datos puede usarse para analizarlos automáticamente y así monitorear la opinión pública y profesional en tiempo real, recibir alertas tempranas y llevar a cabo un análisis del posible impacto positivo o negativo. Además, el surgimiento de Grandes Modelos de Lenguaje (*Large Language Model* - LLMs) basados en Transformadores (Kalyan *et al*., 2021) ha permitido un aumento del rendimiento de estas tareas en el campo financiero.

El análisis de sentimientos (AS) es una tarea popular dentro del procesamiento del lenguaje natural (NLP) y tiene una aplicación directa y útil en al ámbito financiero (Goodell *et al*., 2023). De forma resumida, el AS consiste en determinar automáticamente si un fragmento de texto tiene un sentido positivo, neutral o negativo. Sin embargo, la información financiera es compleja y suele presentar ambigüedad semántica, por lo que la aplicación del AS en estos textos suele ser un desafío. Los principales problemas del AS en el dominio financiero se pueden resumir en cinco puntos (Pan *et al*., 2023):

1. El lenguaje financiero es inherentemente complejo, ya que los términos financieros se refieren a un contexto social, económico y legal subyacente.
2. El sentimiento financiero de texto depende en gran medida del contexto, ya que una expresión puede tener connotaciones positivas o negativas según el contexto en el que se utilice. Por ejemplo, las expresiones "Crece el salario medio en España" y "Crece el desempleo en 2023" contienen ambas la palabra "crece", pero la polaridad subjetiva de ambas es diferente.
3. Otro aspecto que considerar es que la identificación del principal objetivo económico (POE) al que se refiere el sentimiento es algo complejo de determinar: podría haber varios candidatos posibles o incluso es posible que no se indique explícitamente en el texto.
4. Es difícil establecer el sentimiento general dentro del ámbito financiero, porque un evento puede considerarse positivo o negativo si se tienen en cuenta otros

objetivos aparte del POE. Por ejemplo, hay noticias financieras que son positivas para los bancos, pero negativas para los ciudadanos o para cualquier otro sector empresarial.

5. Por último, la gran cantidad de nuevos LLMs basados en Transformadores disponibles dificulta saber de antemano qué modelo es el más adecuado para el ámbito financiero.

1.2. Motivación

Como se ha comentado anteriormente, los modelos de lenguaje preentrenados basados en transformadores (T-PTLMs) están logrando un gran éxito en casi todas las tareas de procesamiento del lenguaje natural. Estos modelos se basan en la arquitectura de transformadores, aprendizaje auto-supervisado y aprendizaje por transferencia. Los T-PTLMs aprenden representaciones de lenguaje universales a partir de grandes volúmenes de datos de texto utilizando aprendizaje auto-supervisado y transfieren este conocimiento a tareas específicas, evitándose tener que entrenar los modelos específicos desde cero (Kalyan *et al.*, 2021).

En los últimos años, se han organizado varias competiciones sobre análisis semántico dirigido en diferentes eventos, como SemEval (*International Workshop on Semantic Evaluation*)[1], CLEF (*Conference and Labs of the Evaluation Forum*)[2] o IberLEF (*Iberian Languages Evaluation Forum*)[3]. Sin embargo, ninguno de estos eventos anteriores se ha centrado en el ámbito financiero, posiblemente porque la aplicación de los modelos T-PTLMs en textos financieros sigue siendo un desafío por la complejidad intrínseca que presenta este tipo de modelado.

En esta línea, la plataforma de competiciones de ciencia de datos *CodaLab* lanzó el desafío *IBERLEF 2023 Task - FinancES. Financial Targeted Sentiment Analysis in Spanish*[4]. Esta competición tiene como objetivo explorar el análisis de sentimiento dirigido en el ámbito financiero. Específicamente, el enfoque adoptado se basa en que los principales agentes microeconómicos en el mercado de capitales son los consumidores (hogares/individuos), las empresas, los gobiernos y los bancos centrales.

Para desarrollar un método de análisis de sentimientos en el que se consideren los diferentes puntos de vista, se analizará desde tres perspectivas diferentes:

1. Principal objetivo económico (POE) de la noticia.
2. Agente económico individual: empresas.
3. Agente económico individual: consumidores.

El POE es la empresa o activo específico donde se aplica el hecho económico, las empresas son entidades que producen los bienes y servicios que otros consumen y los

[1] https://semeval.github.io/
[2] https://clef-initiative.eu/
[3] http://sepln2023.sepln.org/iberlef/
[4] https://codalab.lisn.upsaclay.fr/competitions/10052

consumidores son los hogares e individuos. Desde estos tres puntos de vista, la noticia tendrá un impacto sobre el POE y sobre los dos agentes económicos que puede ser positivo, negativo o neutro.

Definido esto, la competición propone dos tareas. Por un lado, una tarea que consiste en identificar y extraer el POE en el texto y realizar un análisis de sentimientos para determinar la polaridad emocional hacia dicho objetivo del titular. Por otro lado, una segunda tarea debe evaluar el impacto del titular en el resto de los agentes económicos, es decir, empresas y consumidores.

Los retos que esta tarea implica son principalmente dos:

1. Detección del objetivo. La identificación del POE del titular se ve obstaculizada por la longitud reducida del texto y las características lingüísticas de los titulares.
2. Clasificación de sentimiento multidimensional. A diferencia de las tareas tradicionales en las que se identifican múltiples objetivos dentro del alcance de cada texto procesado individualmente, aquí cada titular de noticias se refiere a una única entidad objetivo, pero también se consideran las posturas de los otros agentes económicos (empresas y consumidores).

Para ello, se dispone de un conjunto de datos compuesto por titulares de noticias en español, recopilados de periódicos digitales especializados en noticias económicas, financieras y políticas.

Cada titular se etiquetó manualmente por la organización con el POE y la polaridad del sentimiento en las tres dimensiones indicas anteriormente: POE, empresas y consumidores. Es decir, dado un titular, se clasificó manualmente como positivo, neutral o negativo para las tres entidades específicas. El conjunto de datos final está compuesto por 6300 titulares de noticias.

1.3. Estructura del documento

En esta sección se revisa la estructura que tiene el presente documento, acompañada de una breve descripción que permita al lector conocer de antemano los aspectos principales de cada capítulo.

· **Capítulo 2**: **Objetivos.** En este capítulo se definen los objetivos generales y específicos marcados para el proyecto.

· **Capítulo 3: Estado del Arte y Marco Teórico**. Se proporciona una visión general y se revisan los últimos estudios sobre el NLP en los textos financieros. Se realiza también una breve introducción teórica de algunos conceptos teóricos claves en el desarrollo de este proyecto.

· **Capítulo 4: Desarrollo del proyecto y resultados**. En este capítulo se presenta la metodología CRISP-DM, seguida en el desarrollo del proyecto; se define el problema

a resolver y se describen todos los modelos usados. Por último, se evalúan todos los resultados obtenidos.

· **Capítulo 5: Conclusión y trabajos futuros**. En este capítulo se resumen las líneas seguidas y se realiza una conclusión de los resultados obtenidos. Por último, se describen una serie de líneas futuras como propuestas.

· **Referencias**: Es un listado de todas las referencias usadas en el proyecto.

· **Apéndice – repositorio GitHub**: Por último, se enlaza el repositorio de GitHub que contiene los *scripts* usados para cada modelo, así como el enlace desde dónde descargar el *dataset* de entrenamiento.

2. Objetivos

En este apartado se exponen los objetivos principales marcados para este proyecto, definiendo un objetivo principal general que se desgranará en varios objetivos parciales.

2.1. Objetivo general

El principal objetivo establecido para este proyecto es:

Explorar diferentes modelos y herramientas para al desafío planteado en *IBERLEF 2023* y presentar la mejor solución encontrada para resolver el problema.

2.2. Objetivos específicos

Como ya se ha mencionado en puntos anteriores, el problema del análisis de sentimientos en el ámbito financiero presenta ciertos desafíos para los que no existe una solución óptima reconocida. Por ello, es conveniente explorar diversas técnicas de preprocesamiento y modelos de NLP con el fin de proponer una solución relativamente óptima. Se exponen los siguientes objetivos específicos:

1. Analizar otras líneas de trabajo que hayan afrontado este tipo de problemas o similares.
2. Pruebas de resolución del problema con bibliotecas estándar de NLP como NLTK o spaCy.
3. Pruebas de resolución del problema con modelos T-PTLMs preentrenados para tareas de NLP como BERT o GPT-2 de openAI.
4. Pruebas de resolución del problema utilizando soluciones mixtas de las anteriores.
5. Explorar la interpretabilidad y la explicabilidad de las predicciones realizadas por el modelo seleccionado usando librerías como LIME (*Local Interpretable Model-Agnostic Explanations*) o SHAP (*SHapley Additive exPlanations*).

3. Estado del Arte y Marco teórico

En este apartado se presenta un resumen del estado del arte y una revisión de la literatura de estudios centrados en el uso de técnicas de procesamiento de lenguaje natural (NLP) para el análisis de sentimientos (AS) y su aplicación en el ámbito financiero.

3.1. Procesamiento del Lenguaje Natural

El NLP es una rama de la inteligencia artificial que se enfoca en la interacción entre las computadoras y el lenguaje humano. Su objetivo principal es permitir que las máquinas comprendan, interpreten y generen texto de manera similar a como lo hacen los seres humanos.

Tradicionalmente, la interacción entre humanos y las computadoras se realiza a través de un lenguaje de programación. Cuando se trata de la interacción del lenguaje humano con la máquina, lograr esta comunicación es bastante desafiante, ya que el lenguaje humano es altamente ambiguo, contiene jergas con significados inusuales y contextos sociales. Para ello, el NLP realiza dos tareas principales: el análisis sintáctico y el análisis semántico (Johri *et al*., 2021). El análisis sintáctico se utiliza para organizar las palabras en la oración de tal manera que comience a tener sentido gramatical. Ayuda a la NLP a evaluar el significado de la oración en función de las reglas gramaticales. El análisis semántico se realiza para descubrir el significado detrás de las palabras y su uso en una oración; se aplica en la NLP para comprender la estructura y el significado de una oración.

Recientemente, el campo del Procesamiento del Lenguaje Natural (NLP) ha experimentado un auge sin precedentes gracias a dos factores fundamentales que han revolucionado por completo la forma en que abordamos el procesamiento del lenguaje: el aprendizaje profundo y la disponibilidad de grandes volúmenes de datos textuales.

El aprendizaje profundo o *deep learning*, una rama de la inteligencia artificial basada en redes neuronales artificiales ha sido un catalizador clave en la evolución del NLP. A medida que se desarrollaron arquitecturas de redes neuronales más profundas y complejas, se logró un aumento sustancial en la capacidad de las máquinas para comprender el lenguaje humano de manera más precisa y sofisticada. Modelos como BERT (*Bidirectional Encoder Representations from Transformers*) (Jacob Devlin, 2019) y GPT (*Generative Pre-trained Transformer*) (Ashish Vaswani, 2017) han impulsado avances significativos en tareas como el análisis de sentimientos, la traducción automática y la generación de texto coherente y contextual.

Un componente esencial para el éxito del aprendizaje profundo en NLP es la disponibilidad de grandes cantidades de datos textuales. Estos datos se utilizan para entrenar modelos, permitiendo que las máquinas adquieran una comprensión más profunda del lenguaje humano. La web, las redes sociales y las enormes colecciones

de texto digitalizado han proporcionado un flujo constante de datos que alimentan estos modelos y les permiten capturar la riqueza de la expresión humana.

3.1.1. Herramientas estándar en NLP

Dependiendo de la complejidad del problema a resolver, así como del tiempo disponible, muchos desarrolladores trabajan con bibliotecas y herramientas de NLP públicas "listas para usar" que pueden proporcionar buenos resultados en muchas tareas de NLP.

Dos de las bibliotecas más conocidas y utilizadas (Gorod, 2021) (Gupta, 2021) en el ámbito del NLP son NLTK (*Natural Language Toolkit*) (Loper & Bird, 2002) y spaCy (Honnibal & Montani, 2016), ambas escritas en Python y diseñadas para abordar diversas tareas de procesamiento del lenguaje natural.

NLTK es una biblioteca de código abierto que se ha convertido en una opción predilecta para muchos investigadores en NLP debido a su amplia gama de recursos y algoritmos lingüísticos. NLTK ofrece una amplia gama de herramientas como funciones la tokenización y análisis sintáctico, funciones para la lematización y el *stemming*, y clasificadores de texto.

spaCy es otra biblioteca de procesamiento del lenguaje natural en Python, conocida por su eficiencia y velocidad en el procesamiento de texto a gran escala. A nivel técnico, spaCy proporciona modelos pre-entrenados para varios idiomas que incluyen etiquetas POS (partes de la oración) y reconocimiento de entidades nombradas (NER), lo que facilita el análisis de texto sin necesidad de entrenar modelos desde cero, realiza análisis sintácticos y semánticos y utiliza un algoritmo de tokenización inteligente (tiene en cuenta reglas lingüísticas específicas del idioma).

3.2. NLP para análisis de sentimientos

En la actualidad, con la aparición de la Web 2.0, las redes sociales, los dispositivos de Internet de las Cosas (IoT) y el uso de dispositivos portátiles conectados a internet ha impulsado el crecimiento de los datos generados por los medios digitales y por los propios usuarios. Los medios publican información en las redes prácticamente en tiempo real mientras que los usuarios tienen libertad para expresar sus opiniones o discutir cualquier tema en blogs, redes sociales, sitios web de comercio electrónico, foros, etc.

Sin embargo, es difícil procesar manualmente una cantidad masiva de datos textuales debido a las limitaciones humanas obvias de tiempo y capacidad. El análisis de sentimientos (AS) puede ser un método para revelar información sobre contenido no estructurado al analizar automáticamente la información, las opiniones, las emociones y actitudes de las noticias de los medios digitales o bien de las personas hacia un evento, empresa, individuo o tema específico basado en basado en datos generados por los usuarios o en los titulares de los medios (Yi-Le Chan *et al.*, 2022).

Tanto las opiniones generadas por los usuarios como los titulares de noticias suelen consistir en "un objetivo" y un "sentimiento sobre el objetivo" (Liu, 2020). Por ejemplo, "El precio de la acción A (objetivo) está aumentando (sentimiento sobre el objetivo)": el AS identifica el objetivo y expone el sentimiento que lo acompaña. El AS es un proceso que necesita de una serie de subtareas de NLP, como la extracción de aspectos, el reconocimiento de nombres de entidades, la detección de subjetividad y la detección de sarcasmo (Xing *et al*., 2018).

La polaridad puede expresarse en forma de rango, desde una clasificación binaria ('positivo' o 'negativo') hasta una clasificación más compleja (por ejemplo, 'muy negativo', 'negativo', 'neutral', 'positivo' o 'muy positivo'). Se pueden distinguir tres niveles de análisis según el grado de especificidad: basado en el documento, basado en la oración y basado en aspectos (ABSA) (Ligthart *et al*., 2021). El enfoque basado en el documento asume que cada documento expresa solo un sentimiento principal. Con el enfoque basado en la oración, se calcula un sentimiento para cada oración en el documento. ABSA divide los textos en subtemas y asigna un sentimiento a cada uno, convirtiéndose así en el enfoque más sofisticado para llevar a cabo el análisis de sentimiento (Pan *et al*., 2023).

Los enfoques tradicionales para el AS generalmente se clasifican en tres categorías (Yadav *et al*., 2019): (1) enfoques basados en léxicos, para compilar un diccionario de sentimientos que determine el sentimiento a nivel de palabra, (2) enfoques de aprendizaje automático, para adoptar características hechas a mano en el entrenamiento del clasificador de la red no neuronal para clasificar palabras en sus etiquetas de sentimiento correspondientes, y (3) enfoques de aprendizaje profundo, para entrenar un modelo complejo de red neuronal para capturar mejor características semánticas significativas y abstractas para el análisis de sentimientos.

Sin embargo, los enfoques orientados al aprendizaje supervisado pueden tener problemas al no existir datos de entrenamiento suficientes y por el alto costo que puede suponer el etiquetado de datos (Pan & Yang, 2009). En este sentido, el aprendizaje por transferencia en NLP ha recibido una mayor atención en los últimos tiempos ya que puede aprovechar el conocimiento existente y mejorar el rendimiento de tareas objetivo.

3.2.1. Clasificación automática de textos

El problema del AS se suele modelar como un problema de clasificación (Jain & Dandannavar, 2018), en el que un clasificador recibe un texto previamente procesado y devuelve una categoría, por ejemplo, positiva, neutra o negativa. En un clasificador de aprendizaje automático se entrena un modelo con datos previamente etiquetados (aprendizaje supervisado) tal y como se muestra en la Ilustración 1.

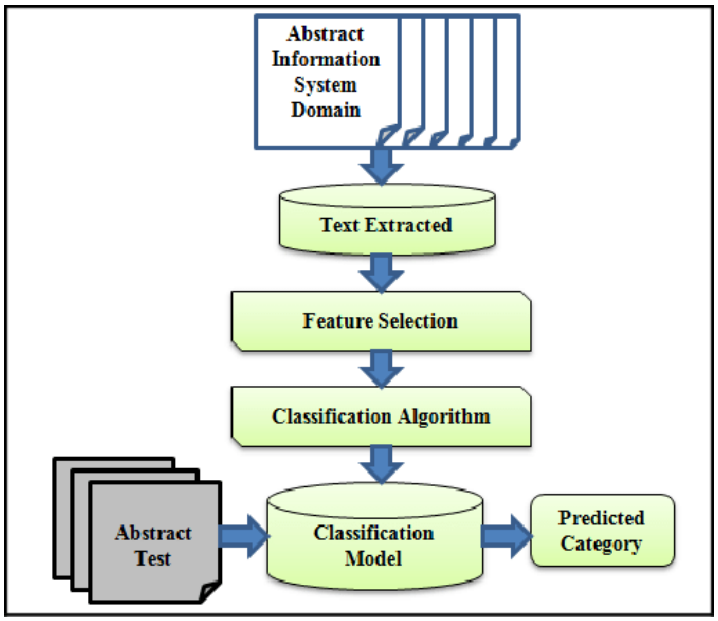

Ilustración 1. Clasificación automática de textos. Fuente: https://www.researchgate.net/figure/Feature-Selection-for-Classification-Model_fig1_267711810

3.2.2. Extracción de características del texto

El primer paso en un clasificador de texto de aprendizaje automático es de transformar los datos textuales en representaciones numéricas, comúnmente conocido como vectorización. Este paso es indispensable ya que los modelos clasificadores, en especial aquellos destinados para el análisis de sentimientos, requieren entradas numéricas para funcionar correctamente. Los textos, compuestos por palabras, frases y oraciones, son datos no estructurados y altamente dimensionales, por lo que convertirlos en vectores permite que los algoritmos de aprendizaje automático los procesen de manera efectiva.

Al vectorizar el texto, convertimos la información lingüística en variables cuantitativas, permitiendo así que los modelos puedan discernir patrones, relaciones y semántica, fundamentales para categorizar y analizar los sentimientos expresados en el texto.

BoW: *bag-of-words*

Un enfoque clásico para esta tarea de vectorización es el de la técnica conocida como *bag-of-words* (BoW o bolsa de palabras).

El método *bag-of-words* (Salton & McGill, 1971) es una forma de representación que convierte texto en vectores de longitud fija, al contar cuántas veces aparece cada palabra ignorando el orden.

La idea detrás de este modelo es simple y nos permite representar texto a través de un vector de características numéricas. Los pasos para su construcción son:

1. Crear un vocabulario de palabras únicas para todo el conjunto total de textos que tengamos.

2. Para cada uno de los textos, se construye un vector de características que contiene el número de veces que cada palabra del vocabulario aparece dentro de él.

Mostramos esto con un ejemplo. Supongamos que quisiéramos vectorizar las siguiente tres frases (nos referiremos a ellas como documentos):

· *the cat sat*

· *the cat sat in the hat*

· *the cat with the hat*

Primero definimos nuestro vocabulario, que es el conjunto de todas las palabras encontradas en nuestro conjunto de documentos. Las únicas palabras que se encuentran en los 3 documentos anteriores son: *the*, *cat*, *sat*, *in*, *the*, *hat* y *with*.

En segundo lugar, para vectorizar nuestros documentos, todo lo que tenemos que hacer es contar cuántas veces aparece cada palabra como se muestra en la Tabla 1:

Documento	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Tabla 1. Ejemplo de bag-of-words

De esta forma, se puede representar cada documento con un vector de tamaño 6:

· *the cat sat*: [1,1,1,0,0,0]

· *the cat sat in the hat*: [2,1,1,1,1,0]

· *the cat with the hat*: [2,1,0,0,1,1]

Este método es muy básico, y se observa fácilmente que perdemos información contextual, como por ejemplo, en qué parte del documento apareció la palabra. Como su propio nombre indica, BoW es como una bolsa literal de palabras: solo te dice qué palabras ocurren en el documento, no dónde ocurrieron.

TF-IDF (Term Frequency-Inverse Document Frequency)

Una técnica más avanzada para vectorizar el texto es la técnica de TF-IDF (siglas de Frecuencia de Término - Frecuencia Inversa de Documento) (Jones, 1972). Es una técnica de ponderación que refleja la importancia de una palabra para un documento en relación con un conjunto de documentos o corpus.

Está dividida en dos partes:

1. TF (*Term Frequency*): Mide la frecuencia de una palabra en un documento. Esto significa que cuanto más a menudo aparece una palabra en un documento, más alto será el valor de TF.

$$TF(t,d) = \frac{número\ de\ veces\ que\ el\ término\ t\ aparece\ en\ el\ documento\ d}{\text{número total de términos en el documento d}}$$

Ecuación 1. TF (Term Frequency)

2. IDF (*Inverse Document Frequency*): Mide lo informativa que es una palabra en todo el corpus de documentos. La idea es que, si una palabra aparece en muchos documentos, no será un buen discriminador.

$$IDF(t,D) = \log \frac{número\ total\ de\ documentos}{\text{número de documentos con el término t}}$$

Ecuación 2. IDF (Inverse Document Frequency)

TF-IDF sería el producto de TF e IDF, y sirve para dar más peso a las palabras que son más específicas para un documento particular.

$$TF - IDF(t,d,D) = TF(t,d)\ x\ IDF(t,D)$$

Ecuación 3. TF-IDF (Term Frequency-Inverse Document Frequency)

La técnica de TF-IDF, como ya hemos mencionado, se utiliza para transformar documentos en vectores numéricos, que pueden ser usados como entrada para clasificadores y otros modelos de aprendizaje automático. En esta representación vectorial, cada dimensión corresponde a un término (palabra) del vocabulario, y el valor en esa dimensión es el peso TF-IDF calculado para ese término en ese documento.

Mostramos a continuación un ejemplo del vectorizador TF-IDF. Supongamos que quisiéramos vectorizar las siguientes dos frases (documentos):

A: *Jupiter is the largest planet*

B: *Mars is the fourth planet from the Sun*

PALABRA	A: TF	B: TF	IDF	A: TF-IDF	B: TF-IDF
Jupiter	1/5	0	0.301	0.06	0
is	1/5	1/8	0	0	0
the	1/5	2/8	0	0	0
largest	1/5	0	0.301	0.06	0
planet	1/5	1/8	0	0	0
Mars	0	1/8	0.301	0	0.04
fourth	0	1/8	0.301	0	0.04
from	0	1/8	0.301	0	0.04
Sun	0	1/8	0.301	0	0.04

Tabla 2. Ejemplo de vectorización TF-IDF

La Tabla 2 muestra el TF-IDF de cada documento, así como los cálculos intermedios. En nuestro ejemplo, dado que tenemos dos documentos en el corpus, N=2. El documento A tiene 5 palabras y el documento B tiene 8. El paso inicial es crear un vocabulario de palabras únicas (columna 'palabra') y calcular el TF para cada documento. El TF será más alto para las palabras que aparecen con frecuencia en un documento y menos para palabras menos usadas.

Luego se calcula el IDF de cada palabra, que representa la medida de la importancia de dicha palabra. La frecuencia de término (TF) no considera la importancia de las palabras, ya que algunas palabras como "de", "y", etc. pueden estar presentes con mucha frecuencia, pero son de poca importancia. IDF proporciona un peso a cada palabra basado en su frecuencia en el corpus D.

Por último, se aplica TF-IDF en los documentos A y B, obteniendo un vector de dimensión igual a las palabras del vocabulario. El valor correspondiente a cada palabra representa la importancia de esa palabra en un documento particular.

Concluyendo, la técnica TF-IDF:

- Va más allá de contar frecuencias y pondera las palabras según su importancia relativa en el documento y en todo el corpus.
- Palabras que son comunes en todo el corpus recibirán pesos más bajos.
- Palabras que son raras, y, por lo tanto, más informativas o significativas, recibirán pesos más altos.
- Sin embargo, al igual que BoW, TF-IDF también ignora el orden de las palabras.

Word Embeddings

Recientemente, se han introducido técnicas innovadoras para la extracción de características centradas en los *embeddings* de palabras o *Word Embeddings*. Estas técnicas se fundamentan en la hipótesis de que una palabra puede ser caracterizada por su contexto de aparición, es decir, por las palabras que la acompañan.

Similar al modelo de BoW, estas técnicas también generan vectores de características numéricas; no obstante, dichos vectores contienen números reales que representan coordenadas en un espacio vectorial específico. Esto permite calcular la proximidad o similitud semántica entre palabras mediante la distancia entre sus vectores correspondientes. Así, palabras cuya representación vectorial sea más cercana tendrán semánticas similares. Existen diversas métricas para calcular estas distancias o similitudes, siendo la distancia euclídea y la similitud coseno entre las más prevalentes (Kumar & Mehrotra, 2022).

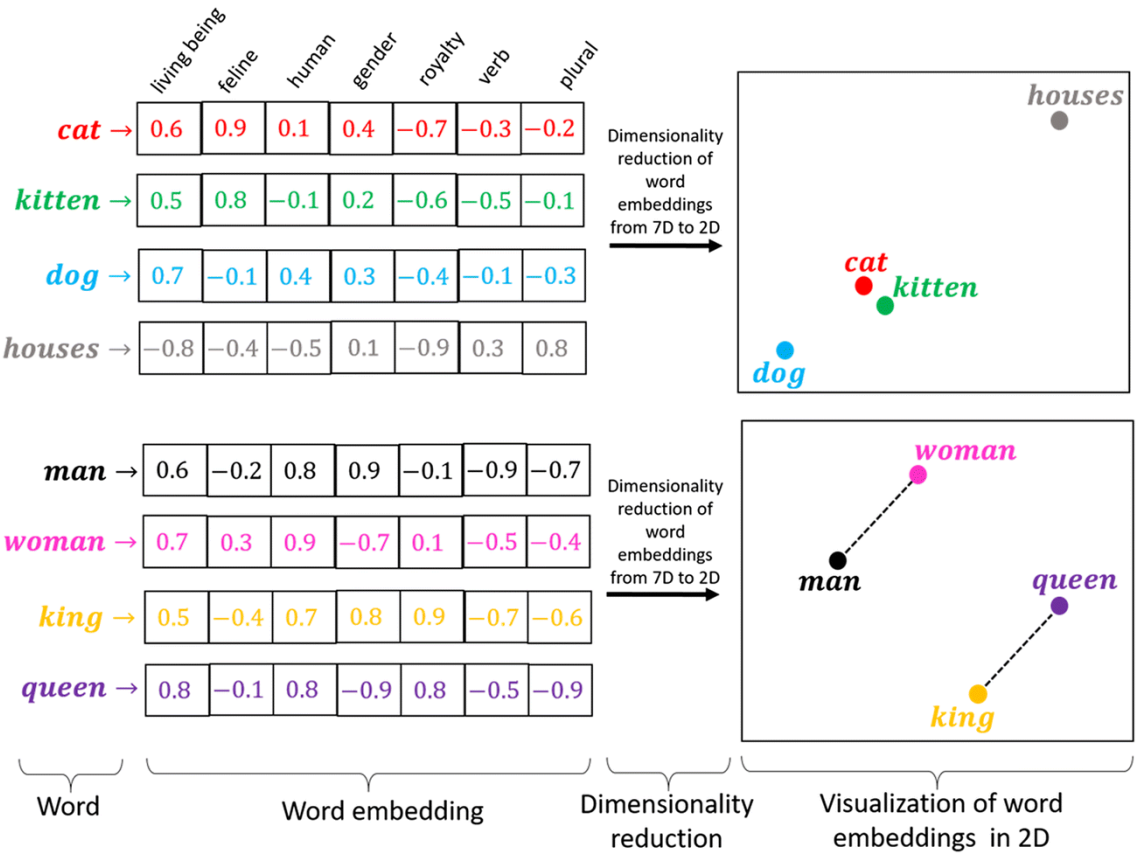

Ilustración 2. Ejemplo de Word Embeddings. Fuente: https://medium.com/@hari4om/word-embedding-d816f643140

En la Ilustración 2 se visualiza de forma clara un ejemplo de la obtención de vectores de características mediante *Word Embeddings*. Se representan cuatro vectores correspondientes a "*cat*" (gato), "*kitten*" (gatito), "*dog*" (perro) y "*houses*" (casas). Cada vector se descompone en siete dimensiones distintas, representando categorías semánticas como "*living being*" (ser vivo), "*feline*" (felino), "*human*" (humano), "*gender*"

(género), "*royalty*" (realeza), "*verb*" (verbo) y "*plural*" (plural). El valor asignado en cada dimensión del vector refleja el nivel de asociación de la palabra a la categoría denotada. De este modo, "*cat*" y "*kitten*" presentan altos valores en "*feline*", contrastando con "*dog*" y "*houses*", que exhiben bajos valores en dicha dimensión. Este tipo de representación es muy útil para comprender las relaciones semánticas y sintácticas entre diferentes palabras, permitiendo así un análisis más profundo y enriquecido del lenguaje.

Al representar estos vectores en un espacio bidimensional, es evidente que los vectores de "*cat*" (gato) y "*kitten*" (gatito) se posicionan muy próximos el uno al otro debido a su similitud semántica. El vector de "*dog*" (perro) se sitúa a una distancia mayor de estos, reflejando su menor similitud semántica con "*cat*" y "*kitten*", aunque manteniendo cierta proximidad por su condición de ser vivo animal. En contraste, "*houses*" (casas), al no compartir similitud semántica con las palabras mencionadas, se localiza considerablemente alejada de estas.

En la sección inferior de la Ilustración 2 se presentan cuatro vectores adicionales, con sus respectivas coordenadas en el espacio multidimensional. Al visualizar dichos vectores en dos dimensiones, se percibe que la distancia y dirección entre los vectores de "*man*" (hombre) y "*woman*" (mujer) son análogas a las de "*king*" (rey) y "*queen*" (reina). Este fenómeno sugiere que el modelo reconoce una relación semántica entre "*man*" y "*woman*" comparable a la existente entre "*king*" y "*queen*".

Este análisis visual evidencia la capacidad del modelo de *Word Embeddings* para discernir y representar relaciones semánticas y similitudes entre palabras diversas. La proximidad y dirección de los vectores en el espacio representan relaciones semánticas sutiles y contextualizaciones específicas, permitiendo inferencias detalladas sobre las conexiones inherentes entre palabras distintas.

Word2Vec

Uno de los modelos de *Word Embeddings* más conocidos y utilizados es Word2Vec. Word2vec (Mikolov *et al.*, 2013) es una red neuronal de dos capas que procesa texto vectorizando palabras. Su entrada es un corpus de texto y su salida es un conjunto de vectores: vectores de características que representan palabras en ese corpus. Aunque Word2vec no es una red neuronal profunda, convierte el texto en una forma numérica que las redes neuronales profundas pueden entender.

El modelo Word2Vec tiene como finalidad principal agrupar, dentro del espacio vectorial, los vectores de palabras que son similares entre sí. En términos matemáticos, su propósito es detectar similitudes. Este modelo genera vectores que constituyen representaciones numéricas distribuidas de las características de las palabras, incorporando aspectos como el contexto de palabras individuales, todo ello sin intervención humana.

Con una cantidad suficiente de datos, aplicaciones y contextos, Word2Vec es capaz de realizar inferencias altamente precisas acerca del significado de una palabra, basándose en sus apariciones anteriores. Dichas inferencias pueden emplearse para

establecer asociaciones entre palabras (por ejemplo, “hombre” se relaciona con “niño” de la misma manera que “mujer” se relaciona con “niña”), o para clasificar documentos por temas y agruparlos. Estos agrupamientos pueden ser fundamentales para realizar búsquedas, análisis de sentimientos y generar recomendaciones en áreas tan variadas como la investigación científica, la exploración legal, el comercio electrónico y la gestión de relaciones con clientes.

El producto final de la red neuronal de Word2Vec es un vocabulario en el que a cada término se le adjunta un vector. Este vector puede ser la entrada de una red neuronal de aprendizaje profundo o simplemente se puede consultar para descubrir relaciones entre palabras.

Al medir la similitud mediante el coseno, una ausencia de similitud se representa por un ángulo de 90 grados, mientras que una similitud total, 1, se representa por un ángulo de 0 grados, correspondiente a una superposición completa; por ejemplo, Suecia es idéntica a Suecia, mientras que Noruega tiene una distancia de coseno de 0.760124 desde Suecia, la más alta de cualquier otro país.

A continuación, se presenta una lista de palabras asociadas con “Suecia” (*Sweden*) mediante el uso de Word2Vec, ordenadas según su proximidad:

Word	Cosine distance
norway	0.760124
denmark	0.715460
finland	0.620022
switzerland	0.588132
belgium	0.585835
netherlands	0.574631
iceland	0.562368
estonia	0.547621
slovenia	0.531408

Ilustración 3. Ejemplo de similitud de la palabra “Suecia” (Sweden) mediante el uso de Word2Vec. Fuente: https://wiki.pathmind.com/word2vec

Los países escandinavos y otros países del norte de Europa con situaciones económicas y culturales similares se encuentran entre los nueve primeros en la Ilustración 3.

Word2Vec, como método de w*ord embeddings* que es, vectoriza palabras, transformando el lenguaje natural en un formato procesable por computadora, lo que nos permite comenzar a ejecutar operaciones matemáticas avanzadas en palabras para discernir sus similitudes. De esta forma, puede entrenar con palabras buscando la relación con otras palabras que aparezcan en su contexto dentro del corpus de entrada.

Este proceso se puede realizar de dos maneras (Ilustración 4): utilizando el contexto para predecir una palabra objetivo, metodología conocida como bolsa continua de palabras (*Continuous Bag-Of-Words* o CBOW), o utilizando una palabra para predecir un contexto objetivo, lo que se denomina *Skip-Gram*.

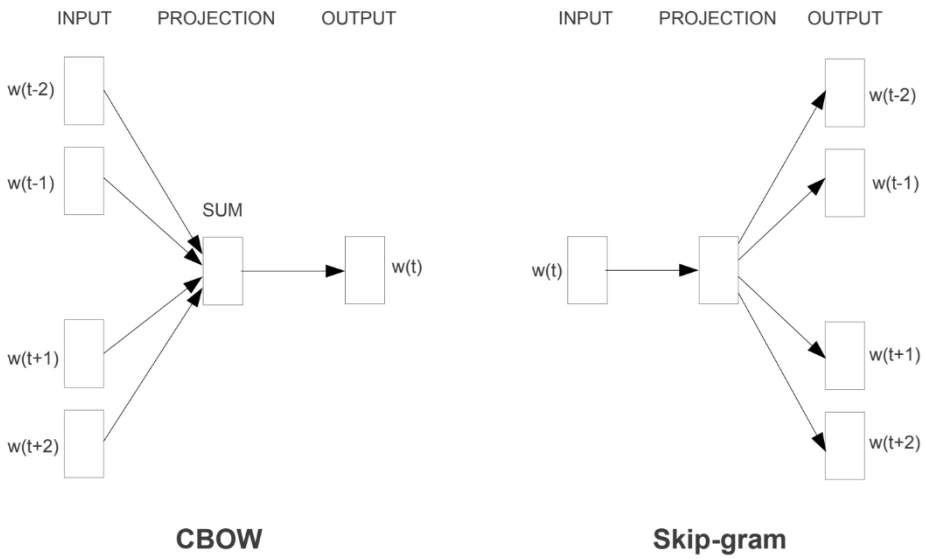

Ilustración 4. Procesos CBOW y Skip-Gram de Word2Vec. Fuente: https://studymachinelearning.com/introduction-to-word-embeddings/

El modelo CBOW tiende a ser más rápido y presenta resultados más precisos con palabras frecuentes, mientras que *Skip-Gram* trabaja bien con conjuntos de datos grandes y es capaz de capturar mejor las palabras raras o infrecuentes (Mikolov *et al.*, 2013).

Vamos a centrarnos en el método *Skip-Gram.* En dicho método, la función de la red neuronal consiste en, dada una palabra de entrada en una frase, predecir la probabilidad de que cada palabra del vocabulario se encuentre en la vecindad contextual de dicha palabra de entrada. Cuando el vector de características asignado a una palabra no logra predecir con precisión el contexto de esa palabra, los componentes de dicho vector son ajustados. En este contexto, cada palabra en el corpus sirve como un referente que envía señales de error retroactivas para ajustar el vector de características. Los vectores de palabras que son evaluados como similares por su contexto se acercan mediante la modificación de los valores en el vector.

Para que la red neuronal pueda entrenar los datos, es imperativo representar las palabras de una forma numérica. Para este propósito, se emplean vectores *one-hot*, donde la posición de la palabra de entrada es representada por "1" y todas las otras posiciones son "0". Así, las entradas de la red neuronal son simplemente vectores *one-hot*, y la salida es también un vector, con la misma dimensión que el vector *one-hot*, que contiene, para cada palabra del vocabulario, la probabilidad de que una palabra "cercana" seleccionada al azar corresponda a esa palabra del vocabulario. A modo de

ilustración, si utilizamos un vocabulario de tamaño V y una capa oculta de tamaño N, la arquitectura de la red se puede visualizar como se muestra en la Ilustración 5:

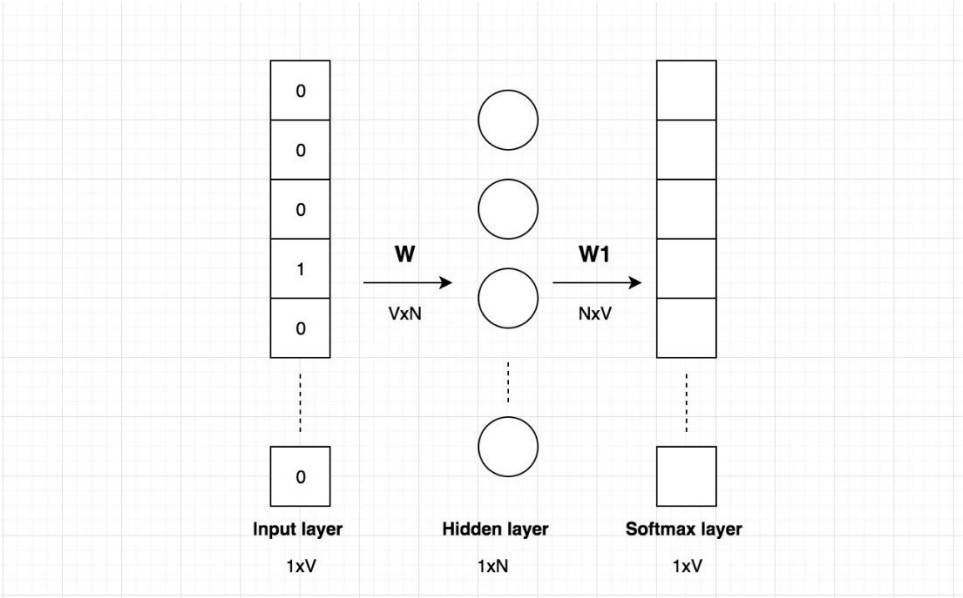

Ilustración 5. Arquitectura de la red neuronal para el modelo Skip-Gram de Word2Vec. Fuente: https://shuzhanfan.github.io/2018/08/understanding-word2vec-and-doc2vec/

La entrada es un vector *one-hot* con una dimensión 1xV. La dimensión de la matriz de pesos de la capa oculta es VxN. Si los multiplicamos, obtendremos un vector de dimensión 1xN.

$$[0\,0\,0\,0\,1\,0] \times \begin{bmatrix} 10 & 23 & 15 \\ 3 & 14 & 9 \\ 18 & 26 & 2 \\ 10 & 17 & 7 \\ 12 & 23 & 8 \\ 9 & 10 & 12 \end{bmatrix} = [12\ \ 23\ \ 8]$$

Ilustración 6. Cálculo matricial en Word2Vec.

Tal como se comprueba en el cálculo matricial de la Ilustración 6, la matriz de pesos de la capa oculta actúa como una tabla de búsqueda. De manera efectiva, esta tabla seleccionará únicamente la fila de la matriz correspondiente al valor "1", produciendo como salida el vector de incrustación de la palabra de entrada.

Dicha matriz de pesos contiene V filas, cada una asociada a un vector de palabras del vocabulario. Por consiguiente, el aprendizaje se centra predominantemente en la determinación de la matriz de pesos de la capa oculta, también conocida como 'Word Embeddings'. La capa de salida, por otro lado, es configurada como una capa *softmax* de dimensiones 1×V, donde cada elemento denota la probabilidad de que la palabra

correspondiente sea seleccionada de manera aleatoria en la cercanía de la palabra de entrada.

Como se ha mencionado anteriormente, el enfoque de CBOW (*Continuous Bag-Of-Words*) opera de manera inversa a *Skip-Gram* (ver Ilustración 4). En este caso, la tarea de la red neuronal consiste en predecir la palabra central basándose en el contexto de palabras circundantes. Dada una frase, la red predecirá la probabilidad de que cada palabra del vocabulario sea esa palabra.

Doc2Vec

Doc2Vec es una extensión de Word2Vec, de manera que Word2Vec extrae vectores de características de palabras mientras que Doc2Vec extrae vectores de características de textos añadiendo, para ello, otro vector (ID de texto) a la entrada. La arquitectura del modelo Doc2Vec se muestra en la Ilustración 7:

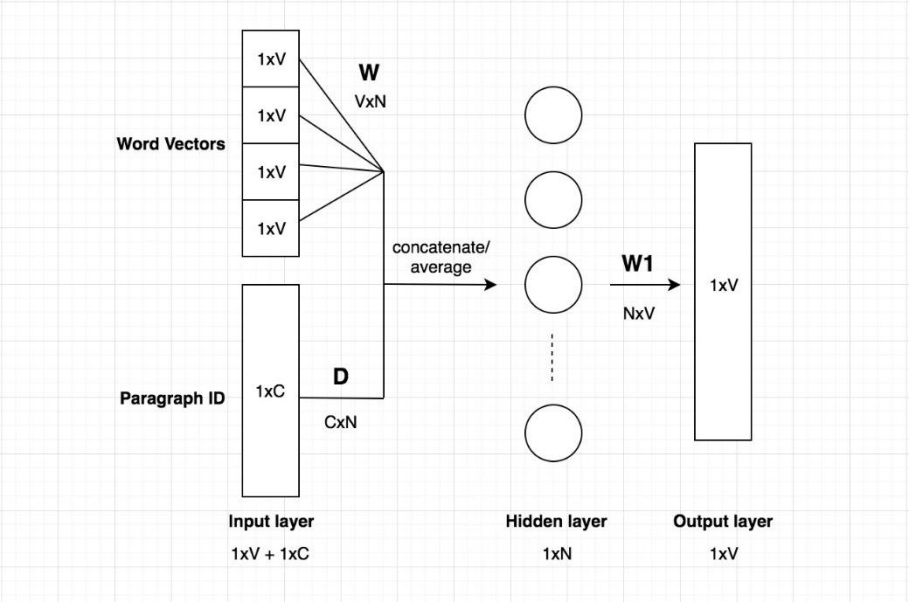

Ilustración 7. Arquitectura del modelo Doc2Vec. Fuente: https://shuzhanfan.github.io/2018/08/understanding-word2vec-and-doc2vec

El diagrama de la Ilustración 7 se basa en el modelo CBOW, pero en lugar de utilizar sólo las palabras "cercanas" para predecir la palabra, también se añade otro vector de características, que es único para el texto. Así, al entrenar los vectores de palabras W, se entrena también el vector de textos D, que al final del entrenamiento contiene una representación numérica del texto. Las entradas consisten en vectores de palabras y vectores de identificación del texto. El vector de palabras es un vector *one-hot* con una dimensión 1xV. El vector Id del texto tiene una dimensión de 1xC, donde C es el número de textos totales. La dimensión de la matriz de pesos W de la capa oculta es VxN. La dimensión de la matriz de pesos D de la capa oculta es CxN. El modelo anterior se denomina *Memory version of Paragraph Vector* (PV-DM). Existe también

otro algoritmo Doc2Vec que se basa en *Skip-Gram*, denominado *Distributed Bag of Words version of Paragraph Vector* (PV-DBOW).

3.2.3. Modelo clasificador

Una vez transformado el texto en vectores numéricos mediante técnicas como TF-IDF o Word2Vec, ya es posible construir un clasificador de aprendizaje automático entrenado con datos previamente etiquetados tal y como se mostró en la Ilustración 1.

Un algoritmo muy habitual usado para la tarea de clasificación se sentimientos es el *Random Forest* (Karthika *et al*., 2019) (Al Amrani *et al*., 2018). Los clasificadores *Random Forest* tienen una muy buena capacidad para manejar un gran volumen de características y se adaptan bien a diferentes distribuciones de datos, lo cual es común en el análisis de sentimientos. Este modelo consiste en un conjunto de árboles de decisión que trabajan conjuntamente para mejorar la precisión y la robustez del modelo, mitigando así el riesgo de sobreajuste, un problema común en el análisis de texto. Además, *Random Forest* permite evaluar la importancia de las diferentes características (variables relevantes), ofreciendo *insights* sobre qué palabras o conjuntos de palabras son más determinantes al inferir el sentimiento.

3.3. Grandes Modelos de Lenguaje

Los Grandes Modelos de Lenguaje (LLMs) han resultado ser uno de los avances más significativos en la inteligencia artificial. Los LLMs son modelos de propósito general que se entrenan utilizando grandes conjuntos de datos y que pueden adaptarse fácilmente a varias tareas de Procesamiento del Lenguaje Natural (NLP), como son la traducción, la generación de lenguaje natural, resumen de textos, o su clasificación. Los LLMs se basan principalmente en dos pilares: el mecanismo de atención y el aprendizaje por transferencia.

El mecanismo de atención (Vaswani *et al*., 2017) permite obtener los conocidos como *embeddings*, que son representaciones numéricas de palabras, frases o documentos en un espacio vectorial. Estas representaciones capturan características semánticas y sintácticas, lo que permite a los modelos de NLP comprender y comparar el significado y las relaciones entre las palabras, a pesar de ser datos textuales (unidades semánticamente similares tienen representaciones similares). La atención resuelve problemas lingüísticos importantes relacionados con la polisemia y la desambiguación de palabras, y permite que el modelo genere o comprenda el texto.

Por otro lado, el aprendizaje por transferencia (Bozinovski, 2020) permite que los LLMs se adapten para resolver tareas específicas; es decir, proporcionan capacidades de comprensión del lenguaje de propósito general que se pueden ajustar para resolver tareas en otros dominios en los que el lenguaje o la información de fondo sean más precisos.

Algunos de los primeros y más populares LLMs son BERT (Devlin *et al.*, 2019), RoBERTA (Liu *et al.*, 2019) o ALBERT (Chiang *et al.*, 2020) y, más recientemente, los modelos desarrollados por OpenAI. Como es fácilmente deducible, la versión original de estos modelos estaba disponible solo para el idioma inglés; sin embargo, pronto aparecieron variantes multilingües o LLMs entrenados para idiomas distintos al inglés, como el español.

Una de las principales desventajas de los LLMs es que son computacionalmente costosos. De hecho, el uso de estos modelos no es razonable en computadoras sin GPUs o TPUs, ni durante el entrenamiento ni durante la inferencia. Sin embargo, algunas técnicas se centran en simplificar los LLMs. Por ejemplo, DistilBERT (Sanh *et al.*, 2019) es una versión más ligera de BERT entrenada mediante una técnica llamada destilación, que permite reducir los recursos computacionales necesarios para desplegar el modelo, manteniendo un rendimiento similar en términos de comprensión del lenguaje y generación de texto.

Otra desventaja de los LLMs es su naturaleza como "cajas negras". A pesar de su impresionante capacidad para generar texto coherente y contextualmente relevante, el funcionamiento interno y los procesos de toma de decisiones de estos modelos a menudo carecen de transparencia por lo que sus resultados son difíciles de interpretar.

3.4. Análisis de sentimientos en el ámbito financiero

En el campo de las finanzas, el propósito del AS es capturar los sentimientos y emociones del texto analizado (normalmente redactado por inversores o medios especializados y recogido de redes sociales o medios de comunicación) hacia el mercado financiero, y cuantificar estos sentimientos como variables numéricas que podrían ser predictores del mercado de valores (Goodell *et al.*, 2023).

Sin embargo, diversos factores dificultan la efectividad del AS en el ámbito financiero, tal como se enumeraron en la página 10: lenguaje frecuentemente complejo, fuerte dependencia del contexto, dificultad para encontrar el principal objeto económico referenciado, diferentes entes tienen sentimientos diferentes para un mismo texto y dificultad para saber de antemano qué modelo LLM es el más adecuado para el ámbito financiero (Pan *et al.*, 2023).

Uno de los principales problemas de los anteriormente mencionados, y más específicamente en el contexto del mercado de valores, es que los enfoques actuales no han sido diseñados para tener en cuenta los sentimientos desde diferentes puntos de vista (diferentes entes económicos), algo que podría ser relevante para respaldar las decisiones de inversión. Existen diversos estudios sobre la relación entre el sentimiento público/experto y los precios de las acciones (Wang *et al.*, 2013) (Li *et al.*, 2014), sin embargo, debido a la complejidad del ámbito financiero, los resultados de los enfoques propuestos no son lo suficientemente buenos. Sin embargo, los últimos avances en el campo, incluyendo los ya mencionados *embeddings* de palabras y *transformers*, han permitido mejorar el rendimiento de las soluciones de AS.

Más recientemente Daudert (2021) utilizó una red neuronal prealimentada (*feed-forward*) con un enfoque novedoso que aprovecha el texto y la información contextual de un registro para el análisis de sentimiento detallado. Kilimci *et al*. (2019) utilizaron una red neuronal profunda y un transformador BERT para predecir la dirección de los precios de las acciones en el mercado de valores turco (BIST100) empleando textos en turco extraídos de redes sociales. García-Díaz *et al*. (2023) exploraron el impacto de la combinación de diferentes conjuntos de características en la precisión del AS en textos financieros en español, compilando un corpus con 15.915 tweets que se anotaron manualmente con el sentimiento de positivo, negativo o neutral.

4. Desarrollo del proyecto y resultados

En este capítulo se detalla la metodología empleada para la planificación y el desarrollo del trabajo. Además, se describe el conjunto de datos usados, el preproceso realizado, la definición del problema a resolver y las herramientas y librerías usadas para su resolución.

4.1. Metodología de resolución del problema

En este apartado se describe la metodología usada para la resolución del problema, CRISP-DM, identificando y explicando sus distintas fases desde un punto de vista teórico y describiendo las distintas tareas a realizar propuestas por la guía de la metodología, las cuales se han llevado a cabo posteriormente en el desarrollo del proyecto.

4.1.1. CRISP-DM

La metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) es una de las guías de referencia más utilizada en proyectos de minería de datos. Como metodología, incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas. Como modelo de proceso, CRISP-DM ofrece un resumen del ciclo vital de minería de datos. A continuación, en la Ilustración 8 se describen cada una de las fases en las que se divide el ciclo vital de la metodología CRISP-DM (IBM, 2012) (Arancibia, 2009):

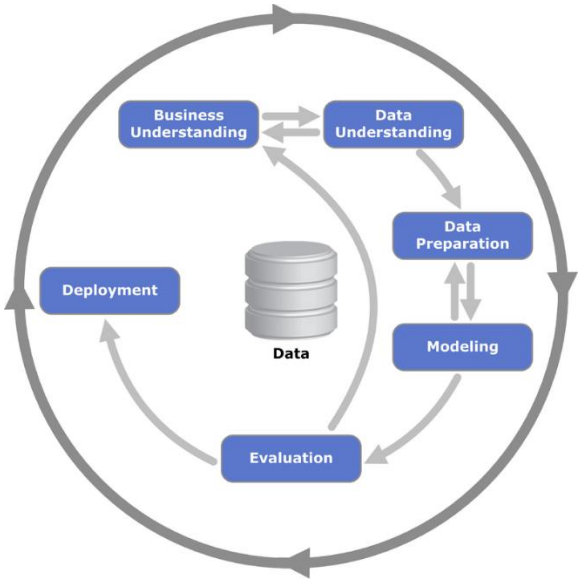

Ilustración 8. Ciclo CRISP-DM. Fuente: http://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/

- Fase de comprensión del negocio: La primera fase de la guía de referencia CRISP-DM, denominada fase de comprensión del problema o del negocio es una de las más importante ya que incluye las tareas de comprensión de los objetivos y los requisitos del proyecto. Se aborda esta fase desde un punto de vista empresarial, con la idea de convertir los objetivos empresariales en objetivos técnicos y en un plan de proyecto. La clave de esta fase es comprender el problema desde el mismo punto de vista que la persona o entidad que usa los datos, para seleccionar la información y los datos que permitan resolverlo. Para la realización de esta fase, la metodología CRISP-DM propone seguir las siguientes tareas: determinar la finalidad del negocio, valorar la situación, determinar el propósito de la minería de datos y por último realizar un plan de proyecto.

- Fase de comprensión de datos: La segunda fase es la fase de comprensión de los datos. Abarca la recopilación inicial de datos con la finalidad de tener un primer contacto con el problema para familiarizarse con los datos, determinar su calidad y detectar las relaciones más evidentes que permitan definir las primeras suposiciones o hipótesis. Esta fase junto a las siguientes dos fases, son las que demandan mayor trabajo y tiempo en un proyecto de *Data Mining*. En primer lugar, se debe identificar la calidad de los datos y definir una primera hipótesis en base a éstos. Las principales tareas propuestas por la guía en esta fase son: la recolección de datos iniciales, la descripción de los datos, la exploración de los datos y la verificación de la calidad de los datos.

- Fase de preparación de los datos: Con los datos ya recopilados, se procede a su preparación para ajustarlos a la técnica de minería de datos que se desea utilizar y construir un conjunto de datos que se ajuste al problema. Esta fase incluye las tareas de selección de datos a los que se aplicará una técnica de modelado, y la limpieza y generación de datos y variables adicionales. Las principales tareas a realizar en esta fase son: la selección de datos, limpieza de datos, la estructuración de los datos, la integración de los datos y el formateo de los datos.

- Fase de modelado: En esta fase de CRISP-DM, se seleccionan las técnicas de modelado más apropiadas para el proyecto de *Data Mining* específico. Las técnicas a utilizar en esta fase se eligen en función de los siguientes criterios:

    - Ser apropiada al problema.

    - Disponer de datos adecuados.

    - Cumplir los requisitos del problema.

    - Tiempo adecuado para obtener un modelo.

    - Conocimiento de la técnica

Las tareas propuestas por la guía son las siguientes: selección de la técnica de modelado, generación de un plan de prueba, construcción del modelo, y evaluación del modelo.

- Fase de evaluación: En esta fase se evalúa el modelo buscando el cumplimiento de los criterios de rendimiento del problema. Es preciso revisar el proceso, teniendo en cuenta los resultados obtenidos, para poder repetir algún paso anterior si fuera necesario. Se pueden emplear múltiples herramientas para la interpretación de los resultados, como por ejemplo las matrices de confusión, que son muy empleadas en problemas de clasificación. Si finalmente se concluye que el modelo seleccionado es válido para el estudio, y cumple con los criterios de éxito, se procede a la explotación del modelo. Las tareas que componen esta fase son las siguientes: evaluación de los resultados, revisión del proceso y determinar próximos pasos.

- Fase de implementación: Una vez que el modelo ha sido seleccionado, construido y validado, en esta fase se transforma el conocimiento y los resultados obtenidos en acciones para el negocio. Identificaremos las acciones a realizar por la empresa o cliente, basándonos en los resultados obtenidos a partir del modelo. En esta fase también se documentarán los resultados obtenidos de manera comprensible para el usuario. Las tareas que se llevan a cabo en esta fase son: plan de implantación, plan de monitoreo y mantenimiento, realización del informe final y revisión del proyecto.

4.2. Entendimiento del dominio del problema

El problema se divide en dos tareas:

Tarea 1: Análisis de sentimiento financiero

Esta tarea consiste, en primer lugar, en identificar el principal objetivo económico (POE) a partir de los titulares de noticias financieras y, en segundo lugar, en determinar la polaridad del sentimiento (positiva, neutral o negativa) hacia dicho objetivo en el texto procesado.

Tarea 2: Análisis de sentimiento financiero para empresas y consumidores

Esta tarea consiste en determinar la polaridad del sentimiento de cada titular de noticias hacia las empresas y los consumidores.

4.3. Entendimiento de los datos

El conjunto de datos para este proyecto está compuesto por titulares de noticias escritas en español recopiladas de periódicos digitales especializados en noticias económicas, financieras y políticas. Algunos de estos periódicos especializados son Expansión[5], El Economista[6], Modaes[7] o El Financiero[8]. Vale la pena indicar que estos periódicos tienen su base en diferentes países de habla hispana.

[5] https://www.expansion.com/
[6] https://www.eleconomista.es/
[7] https://www.modaes.com/
[8] https://www.elfinanciero.com.mx/

Para la preparación del conjunto de datos por parte de la organización, se realizó un proceso de filtrado de dos etapas. En primer lugar, identificaron las secciones específicas de los periódicos que contienen noticias relacionadas con contenido económico (por ejemplo, *elconfidencial.com/mercados*). Además, los titulares recopilados fueron preprocesados para descartar aquellos que no estaban dentro del ámbito financiero. En segundo lugar, un curador de contenido revisó manualmente los titulares restantes y eliminó los irrelevantes.

Cada titular se ha etiquetado manualmente por la organización con la entidad objetivo (POE: principal objeto económico) y la polaridad de sentimiento en tres dimensiones diferentes: entidad objetivo, empresas y consumidores. Es decir, dado un titular, se clasificó manualmente como positivo, neutral o negativo para tres entidades específicas: (1) la entidad objetivo (es decir, la empresa o activo específico donde se aplica el hecho económico), (2) las empresas (es decir, las entidades que producen los bienes y servicios que otros consumen) y (3) los consumidores (es decir, hogares/individuos).

Cada titular fue anotado por tres miembros del comité de organización y comparados. En caso de desacuerdo, los anotadores discutieron sobre el caso especial y, si no se llegó a un acuerdo, se descartó el titular. Durante este primer paso, se recopilaron alrededor de 14.000 titulares, se filtraron los titulares de corta longitud o aquellos que no especificaron un POE. El conjunto de datos final está compuesto por 6.359 titulares de noticias.

Titular	Target (POE)	target_sentiment	companies_sentiment	consumers_sentiment
El Estado recaudará 2.000 millones de euros menos de lo que espera por la subida del IVA	Estado	NEG	NEG	NEG
Empresas Banmédica se retira de la Asociación de Clínicas y eleva tensión en el sector salud	Empresas Banmédica	NEG	NEU	NEU
En estos países Netflix subirá los precios de su suscripción; esto costará el acceso a la plataforma	Netflix	POS	NEU	NEG

Tabla 3. Ejemplos de titulares anotados con las entidades y la polaridad de sentimiento para el objetivo, las empresas y los consumidores.

La distribución total del AS es la siguiente:

Tabla 4 Distribución total del análisis de sentimiento

Sentimiento	Total	Porcentaje
POS	4355	22,8%
NEU	8617	45,2%
NEG	6094	32,0%

Dividido por entes:

Tabla 5 Distribución por entes del análisis de sentimiento

POE (target)			Empresas			Consumidores		
POS	**NEU**	**NEG**	**POS**	**NEU**	**NEG**	**POS**	**NEU**	**NEG**
2815	606	2935	645	3841	1870	895	4170	1289
44,3%	9,5%	46,2%	10,1%	60,4%	29,4%	14,1%	65,6%	20,3%

Se observa que hay un desbalanceo en las diferentes categorías de sentimientos. Por un lado, en los totales, hay clara mayoría del sentimiento neutral. Luego, dividiendo por ente, encontramos que en el POE los sentimientos positivos y negativos están balanceados, pero con un porcentaje muy bajo de sentimientos neutrales, algo lógico ya que cada titular tiende a valorar un POE y no a mantenerse neutral. Respecto al sentimiento de las empresas, hay una clara mayoría neutral, un valor intermedio de negativas y muy pocas positivas. En cuanto a los consumidores, de forma muy similar, hay una clara mayoría neutral y mínima de positivas.

4.4. Preprocesado y transformación de datos

Con respecto a las tareas de preprocesamiento se tendrán en consideración las dos propuestas diferentes con las que se van a dar al problema: el uso de librerías clásicas como NLTK y spaCy, y la utilización de Modelos de Lenguaje con Transformers (LLMs) preentrenados.

Para el preprocesamiento con librerías estándar como NLTK y spaCy, las tareas de preprocesado clásicas son las siguientes:

1. Eliminación de ruido: Consiste en identificar y eliminar caracteres especiales, signos de puntuación o cualquier otro ruido que no sea relevante para el análisis.

2. Eliminación de *stopwords*: Aunque muchas librerías eliminan *stopwords*, es posible personalizar dicha para una tarea específica.

3. Normalización: Esto podría incluir la conversión de texto a minúsculas para garantizar la coherencia en el análisis.

4. Manejo de datos faltantes o erróneos: Si los datos contienen información faltante o errores tipográficos, habría que decidir cómo abordar estos problemas.

Como se ha comentado anteriormente, el conjunto de datos disponible ha sido creado y revisado de forma exhaustiva, por lo que no es de esperar encontrar ruido o datos erróneos. El preprocesamiento se centrará por tanto en la eliminación de caracteres especiales y *stopwords*.

En el caso de los Grandes Modelos de Lenguaje no es necesario a priori ningún tipo de preprocesado, ya que estos modelos son capaces de entender la totalidad de los titulares en su forma natural.

4.5. Modelado

Tal como se indicó en el apartado Objetivos específicos, se ha planteado explorar diversas técnicas de preprocesamiento y modelos de NLP con el fin de proponer una solución comparativamente óptima al problema propuesto.

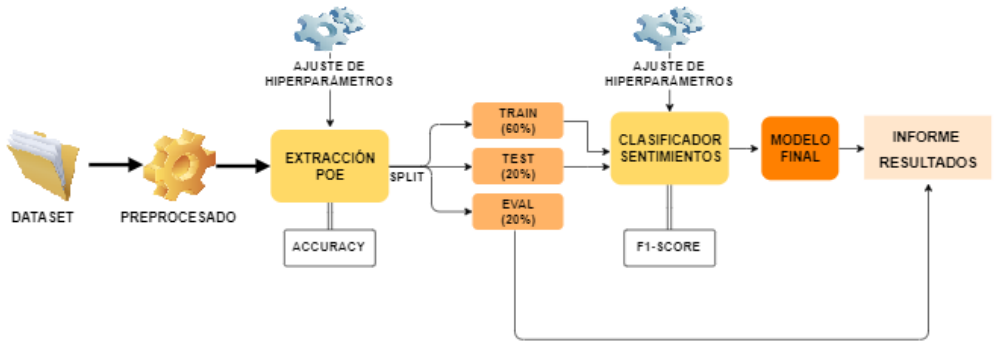

Ilustración 9. Arquitectura general del modelo de análisis de sentimiento financiero. Fuente: elaboración propia.

La Ilustración 9 muestra el esquema general propuesto para el modelo. Los datos son, en primer lugar, preprocesados para los modelos NLTK y spaCy. Este preprocesamiento consiste principalmente en la conversión a minúsculas de todo el texto, eliminación de tildes y otros signos y la eliminación de *stopwords* (palabras comunes y repetitivas que no aportan significado).

Una vez realizado el preprocesamiento, se procede a extraer el POE (Principal Objeto Económico) del titular usando las herramientas mencionadas de NLP. Aquí, calcularemos la exactitud (*accuracy*) que hemos obtenido. El *accuracy* (Ecuación 4) ofrece una medida directa y fácilmente interpretable del grado en que el modelo acierta al identificar el objeto económico en cada titular.

$$accuracy = \frac{núm.\,de\ aciertos}{\text{núm. total de observaciones}}$$

Ecuación 4 – Accuracy (exactitud)

Dado que estamos lidiando con una amplia variedad de objetos económicos en su mayor parte únicos o con muy pocas repeticiones, no existe un desequilibrio significativo de clases que pueda sesgar la métrica de *accuracy*. En otras palabras, no estamos especialmente preocupados por minimizar los falsos positivos o los falsos negativos,

sino más bien por maximizar la tasa de aciertos. Por lo tanto, en este escenario específico, el uso de la métrica de *accuracy* puede proporcionar una visión clara y objetiva de la eficacia del modelo en la tarea de extraer el POE más relevante de una serie de titulares financieros variados.

Posteriormente dividimos los datos en un conjunto de entrenamiento, de test y de validación. Con los datos de entrenamiento procedemos a entrenar tres modelos clasificadores, uno por cada ente económico: POE, empresas y consumidores. Cada uno de ellos tiene tres posibles estados: positio, neutral o negativo. Para ello, usaremos como entrada el titular preprocesado y el POE extraído, convertidos en vectores numéricos para que puedan ser procesados por el clasificador.

En este caso, calcularemos la métrica F1-score sobre el conjunto de test. F1-score (Ecuación 5) es una métrica de evaluación que proporciona una medida balanceada entre la precisión y la exhaustividad (*recall*) de un modelo.

$$F1 - score = 2 \cdot \frac{precisión \cdot exhaustividad}{\text{precisión} + \text{exhaustividad}}$$

Ecuación 5 - F1-score

La métrica se calcula tomando en cuenta la "precisión" y la "exhaustividad" de cada clase, lo que significa que evalúa cuán bien el modelo clasifica los titulares como "positivo", "neutral" o "negativo" frente a todas las veces que debería haberlo hecho, y cuán bien evita clasificaciones incorrectas. Dado que el conjunto de datos está desbalanceado (ver Tabla 5), el F1-score, al ponderar de manera equilibrada la precisión y la exhaustividad para cada clase, ofrece una métrica robusta y completa para evaluar el rendimiento del modelo.

Finalmente, se integran todos los modelos anteriores en un único modelo y se evalúan sobre el conjunto de datos de evaluación.

4.5.1. Modelo NLTK

En este primer modelo recurrimos al *Natural Language Toolkit* (NLTK) para preprocesar los titulares. NLTK es una biblioteca que ofrece herramientas de procesamiento de lenguaje natural, desde la tokenización hasta el análisis sintáctico, y es particularmente útil para lidiar con las complejidades del lenguaje natural en múltiples idiomas.

El primer paso de este modelo es elegir la forma de extracción del POE en cada titular. Se ha optado por utilizar métricas de frecuencia de palabras para identificar este ente, lo cual resulta un método práctico y eficiente para nuestro caso. Esta aproximación, aunque simple, establece un punto de referencia interesante sobre la precisión que podemos esperar de modelos más complejos.

A continuación, enfrentamos el núcleo del análisis de sentimientos. Aquí, en lugar de limitarnos a usar el texto del titular como única característica, también incorporamos el

POE que hemos extraído previamente. Con esto se pretende capturar la relación contextual entre la entidad económica y el sentimiento expresado en el titular.

Para transformar nuestro texto en algo que una máquina pueda entender, empleamos *TfidfVectorizer* de la biblioteca *scikit-learn*. TF-IDF (*Term Frequency-Inverse Document Frequency*) es una de las técnicas más conocidas para convertir texto en un formato numérico y es especialmente buena para manejar grandes colecciones de documentos. La razón para usarla en este contexto es su eficiencia y su capacidad para resaltar la importancia de términos específicos en cada documento, lo cual es esencial para un análisis preciso del sentimiento.

Después de la vectorización, utilizamos un *RandomForestClassifier* para realizar la clasificación de sentimientos. Los bosques aleatorios son conocidos por su flexibilidad y son particularmente buenos para evitar el sobreajuste, lo cual es crucial en tareas de NLP donde el número de características puede ser muy grande.

Para optimizar tanto el vectorizador como el clasificador, recurrimos a *GridSearchCV*. Este es un método de búsqueda exhaustiva que trabaja probando todas las posibles combinaciones de los parámetros que le damos. Lo usamos para ajustar varios aspectos de nuestro *pipeline* de manera simultánea, lo que nos ahorra el esfuerzo de hacerlo manualmente y nos garantiza una solución óptima dentro del espacio de búsqueda definido.

La exploración de hiperparámetros se realiza sobre el siguiente conjunto:

```
parameters = {
    'features__processed_text__tfidf__max_df': [0.5, 1.0],
    'features__processed_text__tfidf__min_df': [1, 2],
    'features__processed_text__tfidf__ngram_range': [(1, 1), (1, 2)],
    'features__predicted_target__tfidf__max_df': [0.5, 1.0],
    'features__predicted_target__tfidf__min_df': [1, 2],
    'features__predicted_target__tfidf__ngram_range': [(1, 1), (1, 2)],
    'clf__n_estimators': [50, 100],
    'clf__max_depth': [None, 20],
    'clf__min_samples_split': [2, 5],
    'clf__min_samples_leaf': [1, 2],
}
```

features__processed_text__tfidf__ y *features__predicted_target__tfidf__* son prefijos que señalan a qué parte del *pipeline* o *ColumnTransformer* se están aplicando los parámetros de TF-IDF. *clf__* es un prefijo que indica que estos parámetros están asociados con el clasificador *RandomForestClassifier* en el *pipeline*.

Parámetros TF-IDF:

1. *max_df*: Ignora los términos que tienen una frecuencia de documento superior al valor dado. Puede ser un valor en punto flotante en el rango [0.0, 1.0] representando una proporción de documentos o un entero absoluto representando la cantidad de documentos.

2. *min_df*: Ignora los términos que tienen una frecuencia de documento inferior al valor dado. Puede ser un valor en punto flotante representando una proporción de documentos o un entero absoluto.

3. *ngram_range*: Define el rango de n-gramas que se extraerán. (1, 1) significa solo unigramas, (1, 2) significa unigramas y bigramas, y (1, 3) significa unigramas, bigramas y trigramas.

Parámetros del *RandomForestClassifier*:

1. *n_estimators*: El número de árboles en el bosque de *RandomForestClassifier*. Cuantos más árboles, más robusto es el modelo, pero también es más computacionalmente intensivo.

2. *max_depth*: Indica la máxima profundidad del árbol. Si tiene el valor *None*, los nodos se expanden hasta que todas las hojas son puras o hasta que contienen menos de *min_samples_split* muestras.

3. *min_samples_split*: Es el número mínimo de muestras requeridas para dividir un nodo interno.

4. *min_samples_leaf*: Es el número mínimo de muestras requeridas para ser un nodo hoja.

Las mejores combinaciones encontradas por *GridSearchCV* optimizadas para la métrica *f1_weighted* son las siguientes:

Clasificador de sentimientos del POE:

```
best_parameters={'clf__max_depth': None,
'clf__min_samples_leaf': 1,
'clf__min_samples_split': 5,
'clf__n_estimators': 50,
'features__predicted_target__tfidf__max_df': 0.5,
'features__predicted_target__tfidf__min_df': 1,
'features__predicted_target__tfidf__ngram_range': (1, 2),
'features__processed_text__tfidf__max_df': 1.0,
'features__processed_text__tfidf__min_df': 2,
'features__processed_text__tfidf__ngram_range': (1, 2)}
```

F1-score for target_sentiment: 0.6070

F1-score for target_sentiment on validation data: 0.6194

Clasificador de sentimientos de las compañías:

```
best_parameters= {'clf__max_depth': None,
'clf__min_samples_leaf': 2,
'clf__min_samples_split': 5,
'clf__n_estimators': 100,
'features__predicted_target__tfidf__max_df': 1.0,
```

```
'features__predicted_target__tfidf__min_df': 2,
'features__predicted_target__tfidf__ngram_range': (1, 1),
'features__processed_text__tfidf__max_df': 1.0,
'features__processed_text__tfidf__min_df': 1,
'features__processed_text__tfidf__ngram_range': (1, 1)}
```

F1-score for companies_sentiment: 0.5938

F1-score for companies_sentiment on validation data: 0.6013

Clasificador de sentimientos de los consumidores:

```
best_parameters={'clf__max_depth': None,
'clf__min_samples_leaf': 2,
'clf__min_samples_split': 5,
'clf__n_estimators': 100,
'features__predicted_target__tfidf__max_df': 0.5,
'features__predicted_target__tfidf__min_df': 2,
'features__predicted_target__tfidf__ngram_range': (1, 2),
'features__processed_text__tfidf__max_df': 0.5,
'features__processed_text__tfidf__min_df': 1,
'features__processed_text__tfidf__ngram_range': (1, 2)}
```

F1-score for consumers_sentiment: 0.6320

F1-score for consumers_sentiment on validation data: 0.6468

Análisis de las variables más relevantes

Las variables o características más relevantes en un *Random Forest* son aquellas que, en promedio, resultan más informativas o útiles para hacer predicciones precisas en el conjunto de árboles de decisión que componen el bosque.

La importancia de una variable se calcula generalmente mediante el promedio de la disminución en la impureza (como la entropía o el índice de Gini) que aporta dicha variable en todos los árboles del bosque, o mediante el promedio de la mejora en la precisión o en la reducción del error que aporta. En esencia, si una variable frecuentemente divide los datos de manera efectiva y mejora la precisión del modelo, tendrá una importancia alta.

El atributo *feature_importances* en los modelos *Random Forest* proporciona directamente esta medida de importancia para cada variable del modelo, permitiendo identificar cuáles son las variables más influyentes para las decisiones del clasificador.

Feature_target_sentiment	Feature_companies_sentiment	Feature_consumers_sentiment
menos	empresas	despidos
pierde	guerra	ere
compra	pymes	millones
lanza	pib	pib
empresas	inflacion	espana
millones	millones	personas
mas	ley	lanza
noticias	sector	sube
caida	competencia	inflacion
cae	ve	erte
riesgo	ibex	hipotecas
hodar	puntos	empleo
baja	espanola	huelga
deja	competir	despido
reduce	economia	comunidad
sube	fed	deficit
mejor	prohibe	precios
perdidas	espana	clientes
ere	amenaza	tipos
cotizacion	deficit	pensiones

Tabla 6. TOP 20 Variables más relevantes en los clasificadores del modelo NLTK

Tal como se observa en la Tabla 6, algunas de las palabras más relevantes para el modelo son conceptos o términos genéricos que no aportan valor al sentimiento de la noticia. Por ejemplo, las palabras “espana” (España), “pib” o “millones” no aportan un sentimiento al objetivo *per se*.

Por ello, se van a eliminar del top 20 de cada modelo aquellas palabras que no aportan sentimiento y se van a volver a ejecutar los modelos para comprobar si se han producido mejoras.

Palabras para eliminar:

Clasificador POE: empresas, millones, noticias.

Clasificador Empresas: empresas, pib, millones, ley, sector, ve, Ibex, puntos, española, fed, espana.

Clasificador Consumidores: millones, pib, espana, personas, comunidad.

Una vez eliminadas las palabras, comparamos los resultados obtenidos:

Comparación métricas	Clasificador POE	Clasificador Empresas	Clasificador Consumidores
Anterior F1-score	0.6194	0.6013	0.6468
Nuevo F1-score	0.6193	0.5984	0.62929

Tabla 7. Comparación de las métricas tras extracción de palabras no significativas en el modelo NLTK

En la Tabla 7 se observa que ninguno de los clasificadores experimenta mejoría al eliminar las palabras eliminadas.

4.5.2. Modelo spaCy

En este modelo el preprocesamiento se lleva a cabo mediante un modelo de lenguaje de spaCy específico para el español, denominado "*es_core_news_sm*". spaCy es una biblioteca para procesamiento del lenguaje natural que es ampliamente utilizada para tareas complejas como el etiquetado gramatical, la identificación de entidades y la tokenización, entre otras. La elección de spaCy para uno de los modelos, además de por su popularidad, se debe a su eficiencia y precisión en el manejo de idiomas distintos al inglés, lo cual es crucial aquí dado que estamos trabajando con un conjunto de datos en español.

Se lleva a cabo una limpieza del texto mediante la eliminación de acentos y la conversión a minúsculas. Además, se aplica una tokenización que retira las *stopwords*. Este preprocesamiento resulta esencial para cualquier modelo de NLP porque ayuda a reducir la complejidad del texto y a destacar las características que son realmente importantes para las tareas de clasificación.

Para le extracción del POE, este modelo se aprovecha de las capacidades de spaCy en el análisis de dependencias gramaticales y el reconocimiento de entidades nombradas (NER). Para ello, primero se intenta extraer entidades nombradas del tipo "ORG"

(Organización), y si no encuentra ninguna, se utiliza un análisis de dependencias para encontrar el núcleo del sujeto o el objeto directo.

Después del preprocesamiento, se utiliza de nuevo la técnica de vectorización del texto conocida como TF-IDF (*Term Frequency-Inverse Document Frequency*) para convertir las palabras en vectores numéricos. Se escoge `*TfidfVectorizer*` sobre otros métodos como el conteo de palabras, como *CountVectorizer*, debido a su habilidad para dar menos peso a palabras comunes que podrían ser irrelevantes para nuestro análisis. TF-IDF permite destacar la importancia de las palabras en función no solo de su frecuencia en un documento particular, sino también en función de su presencia en todo el conjunto de datos.

Para la fase de modelado, se recurre de nuevo a un clasificador de bosques aleatorios *RandomForestClassifier*. Este es un algoritmo de aprendizaje supervisado que es eficiente para la clasificación y generaliza bien para conjuntos de datos complejos.

El ajuste de hiperparámetros para los tres clasificadores se realiza de nuevo con *GridSearchCV*. Con esto se realiza una búsqueda exhaustiva dentro de un subconjunto específico del espacio de hiperparámetros del modelo.

La búsqueda se realiza sobre el siguiente conjunto de hiperparámetros:

```
parameters = {
    'tfidf__max_df': [0.5, 0.75, 1.0],
    'tfidf__min_df': [1, 2, 3],
    'tfidf__ngram_range': [(1, 1), (1, 2), (1, 3)],
    'clf__n_estimators': [50, 100, 150],
    'clf__max_depth': [None, 10, 20, 30],
    'clf__min_samples_split': [2, 5, 10],
    'clf__min_samples_leaf': [1, 2, 4]}
```

De forma muy similar a la búsqueda realizada en el Modelo NLTK, lo términos de exploración son los siguientes:

1. *tfidf__max_df*: Es el máximo *document frequency* permitido para el *TfidfVectorizer*. Se usa para ignorar términos que tienen una frecuencia de documento muy alta, ya que estos términos suelen ser comunes y menos informativos. Especificado como una fracción entre [0.0, 1.0].

2. *tfidf__min_df*: Es el mínimo *document frequency* permitido. Se usa para eliminar términos con una frecuencia de documento muy baja, ya que son raras y, a menudo, menos informativas. Puede ser un entero absoluto o una fracción.

3. *tfidf__ngram_range*: Especifica el rango de n-gramas a extraer. Un n-grama es una secuencia continua de n palabras. `(1, 1)` significa solo unigramas, `(1, 2)` significa unigramas y bigramas, y `(1, 3)` significa unigramas, bigramas y trigramas.

4. *clf__n_estimators*: El número de árboles en el bosque de *RandomForestClassifier*. Cuantos más árboles, más robusto es el modelo, pero también es más computacionalmente intensivo.

5. *clf__max_depth*: Es la máxima profundidad de los árboles. *None* significa nodos que se expanden hasta que todas las hojas sean puras. Una profundidad limitada puede ayudar a evitar el sobreajuste.

6. *clf__min_samples_split*: Es el mínimo número de muestras requeridas para dividir un nodo interno. Cuanto más alto es el número, más restrictiva es la expansión del árbol, ayudando a evitar el sobreajuste.

7. *clf__min_samples_leaf*: Es el mínimo número de muestras requeridas para ser en una hoja (nodo terminal). Un número más alto hace que el modelo sea más conservador.

GridSearchCV realiza una búsqueda exhaustiva sobre todas las combinaciones posibles de estos parámetros y selecciona la combinación que ofrezca el mejor rendimiento según una métrica de evaluación específica, en este caso *f1_weighted* ya que es un clasificador desbalanceado. La combinación óptima obtenida es:

Las mejores combinaciones encontradas por *GridSearchCV* optimizadas para la métrica *f1_weighted* son las siguientes:

Clasificador de sentimientos del POE:

```
best_parameters={'clf__max_depth': None,
'clf__min_samples_leaf': 1,
'clf__min_samples_split': 5,
'clf__n_estimators': 150,
'tfidf__max_df': 1.0,
'tfidf__min_df': 2,
'tfidf__ngram_range': (1, 3)}
```

F1-score for target_sentiment: 0.6013

F1-score for target_sentiment on validation data: 0.6175

Clasificador de sentimientos de las compañías:

```
best_parameters={'clf__max_depth': None,
'clf__min_samples_leaf': 2,
'clf__min_samples_split': 5,
'clf__n_estimators': 50,
'tfidf__max_df': 0.5,
'tfidf__min_df': 1,
'tfidf__ngram_range': (1, 2)}
```

F1-score for companies_sentiment: 0.5956

F1-score for companies_sentiment on validation data: 0.6163

Clasificador de sentimientos de los consumidores:

```
best_parameters={'clf__max_depth': None,
'clf__min_samples_leaf': 1,
'clf__min_samples_split': 10,
'clf__n_estimators': 150,
'tfidf__max_df': 0.75,
'tfidf__min_df': 3,
'tfidf__ngram_range': (1, 1)}
```

F1-score for consumers_sentiment: 0.6475

F1-score for consumers_sentiment on validation data: 0.6458

Análisis de las variables más relevantes

Como ya se ha comentado, en un clasificador *Random Forest*, las variables más relevantes (o características importantes) son aquellas que, en promedio, resultan más informativas o útiles para hacer predicciones precisas en el conjunto de árboles de decisión que componen el bosque.

A partir del atributo *feature_importances*, podemos extraer directamente una medida de importancia para cada variable del modelo, permitiendo identificar cuáles son las variables más influyentes para las decisiones del clasificador.

Extraemos de esta forma el top 20 de las variables más relevantes de cada clasificador:

feature_target_sentiment	feature_companies_sentiment	feature_consumers_sentiment
empresas	empresas	ere
lanza	guerra	millones
millones	pymes	despidos
pierde	inflacion	pib
compra	pib	inflacion
noticias	millones	espana
cae	sector	lanza
caida	ibex	hipotecas
cotizacion	espanola	empleo
baja	ley	erte
reduce	economia	precios
deja	deficit	personas
riesgo	prohibe	pensiones
espana	competencia	huelga
ere	sube	sube
sube	ve	deficit
ibex	puntos	empleos
charts hodar	empleados	clientes
hodar	opa	gobierno
aprueba	accionistas	economia

Tabla 8. TOP 20 Variables más relevantes en los clasificadores del modelo spaCy

Al igual que en el Modelo NLTK, lo que se observa en la Tabla 8 es que algunas de las palabras son conceptos o términos genéricos que no aportan valor al sentimiento de la noticia. Por ejemplo, las palabras “empresas”, “España” o “ve” no tienen ninguna incidencia en el tipo de sentimiento del titular.

Por ello, de nuevo se van a eliminar del top 20 de cada modelo aquellas palabras que no aportan sentimiento y se van a volver a ejecutar los modelos para comprobar si se han producido mejoras.

Palabras para eliminar:

Clasificador POE: empresas, millones, noticias, espana, ibex.

Clasificador Empresas: empresas, pymes, millones, pib, sector, ibex, espanola, ley, ve, puntos.

Clasificador Consumidores: millones, pib, espana, gobierno.

Una vez eliminadas las palabras, comparamos los resultados obtenidos:

Comparación métricas	Clasificador POE	Clasificador Empresas	Clasificador Consumidores
Anterior F1-score	0.6175	0.6163	0.6458
Nuevo F1-score	0.6004	0.5988	0.6668

Tabla 9. Comparación de las métricas tras extracción de palabras no significativas en el modelo spaCy.

La Tabla 9 muestra que solo se ha experimentado mejora en el clasificador del sentimiento para los consumidores. Por tanto, mantendremos el conjunto de palabras completo para los clasificadores de sentimientos del POE y de las empresas y filtraremos en el de consumidores.

4.5.3. Modelo openAI

Este modelo emplea la API de OpenAI para automatizar la extracción de información y el análisis de sentimiento de titulares financieros. OpenAI es un modelo LLM preentrenado (ver apartado Grandes Modelos de Lenguaje). Estos modelos son capaces de entender y generar texto en lenguaje natural, realizando tareas como traducción automática, respuesta a preguntas, resumen de texto, análisis de sentimiento, entre otras, y se entrenan utilizando grandes cantidades de texto para aprender patrones, gramática, hechos sobre el mundo y otras características del lenguaje.

Dado que a fecha actual la API de openAI es de pago y no ofrece ningún tipo de prueba gratuita, se ha optado por usar en las pruebas un muestreo del dataset del 5%, lo que equivale a algo más de 300 titulares de noticias.

El modelo se inicia el proceso con una configuración de clave de API específica y la carga el muestreo del conjunto de datos financiero con pandas, el cual se someterá a análisis.

En cuanto al preprocesamiento, se formula un *prompt* base, detallando ejemplos específicos de titulares y sus correspondientes objetos económicos principales para guiar al modelo en la extracción de objetos económicos de otros titulares dentro del conjunto de datos.

Prompt base del modelo para extraer el POE:

```
prompt_base = """Contesta con una única palabra o palabra compuesta.
Te voy a dar tres ejemplos de titulares de noticias financieras y de
cuál es su principal objeto económico. Ejemplo 1: 'Bayer presenta un
ERE para 75 personas en Sant Joan Despí (Barcelona)'. En este primer
titular, el principal ente económico es 'Bayer'. Ejemplo 2: 'Banc
Sabadell vende su gestora a Amundi con 351M en plusvalías'. En este
segundo titular, el principal objeto económico es 'Banc Sabadell'.
Ejemplo 3: 'Los datos sobre el uso de los ascensores arrojan una caída
del 45% en la afluencia a la oficina por ómicron'. En este tercer y
último ejemplo, el principal objeto es 'uso de los ascensores'. Ahora
debes tú extraer el principal objeto económico de este titular: '{}'.
Basado en los ejemplos previos sobre titulares financieros y sus
objetos económicos principales, identifica el principal objeto
económico del anterior titular y responde siempre con una única
palabra o palabra compuesta."""
```

A lo largo del proceso de predicción de objetos económicos, se itera sobre el conjunto de datos, utilizando el modelo *DaVinci* de OpenAI, con ajustes específicos en los parámetros, para obtener respuestas enfocadas y determinadas.

Ajuste del modelo:

```
response = openai.Completion.create(
        engine="text-davinci-002",
        prompt=prompt,
        max_tokens=10,
        temperature=0.2,
        top_p=0.9,)
```

engine especifica el motor de OpenAI que se utilizará para generar el texto. En este caso, se ha seleccionado "*text-davinci-002*", basado en el modelo GPT-3.

max_tokens limita la longitud del texto generado, especificando el número máximo de tokens (palabras, puntuaciones, etc.) que la respuesta debería contener. En este caso, el modelo generará un texto de no más de 10 tokens de longitud.

temperature controla el grado de creatividad o variabilidad en las respuestas del modelo. Un valor bajo como 0.2 produce respuestas más determinísticas y enfocadas, mientras que valores más altos generarían respuestas más creativas y diversas.

top_p es un parámetro que controla la diversidad de la salida generada al aplicar un muestreo "*nucleus*". Un valor de 0.9 significa que el modelo seleccionará sus respuestas de entre el 90% de los tokens más probables en cada paso de predicción, eliminando las opciones menos probables y contribuyendo a la coherencia y relevancia de la respuesta generada.

Las respuestas obtenidas, que representan los objetos económicos extraídos (POE), se almacenan y añaden al conjunto de datos original. Posteriormente, se lleva a cabo un proceso de limpieza y estandarización de los datos.

La precisión del modelo en esta tarea se evalúa calculando el *accuracy* entre las predicciones y los valores reales, proporcionando así un indicador cuantitativo del rendimiento del modelo en la extracción de objetos económicos.

Para el análisis de sentimiento, se genera un segundo *prompt* que incorpora tanto el titular original como el objeto económico predicho, solicitando al modelo que determine el sentimiento del titular respecto a los diferentes entes económicos.

Prompt para el AS del POE:

```
sentiment_prompt_poe = """El titular es: '{}'. El principal objeto
económico identificado es: '{}'. ¿Cuál es el sentimiento del titular
respecto al principal objeto económico? Responde con la palabra exacta
'positivo', 'neutral' o 'negativo'."""
```

Prompt para el AS de las Empresas:

```
sentiment_prompt_companies = """El titular es: '{}'. El principal
objeto económico identificado es: '{}'. ¿Cuál es el sentimiento del
titular con respecto a las empresas (definiendo a las empresas como
aquellas entidades que producen los bienes y servicios que otros
consumen)? Considera el sentimiento con respecto a las empresas en
general, no específicamente a la empresa nombrada en el titular.
Responde con la palabra exacta 'positivo', 'neutral' o 'negativo'. Si
no hay una relación clara entre el titular y el sentimiento que
provoca en las empresas en general, devuelve la respuesta
'neutral'."""
```

Prompt para el AS de los Consumidores:

```
sentiment_prompt_consumers = """El titular es: '{}'. El principal
objeto económico identificado es: '{}'. ¿Cuál es el sentimiento del
titular con respecto a los consumidores (definiendo a los consumidores
como los hogares e individuos que consumen lo que producen las
empresas)? Considera como afecta a los consumidores en general, no
específicamente a las personas nombradas en el titular si las hubiese.
Responde con la palabra exacta 'positivo', 'neutral' o 'negativo'. Si
no hay una relación clara entre el titular y el sentimiento que
provoca en los consumidores en general, devuelve la respuesta
'neutral'."""
```

Los sentimientos predichos se añaden al conjunto de datos original y, tras un mapeo y limpieza de las etiquetas de sentimiento, se calcula el F1-score para evaluar el rendimiento del modelo en la predicción de sentimiento.

4.5.4. Modelo spaCy + BERT

Este es un modelo híbrido que utiliza spaCy, una biblioteca para procesamiento de lenguaje natural que ya se ha usado anteriormente para la parte de extracción del POE, y BERT, que es una técnica de aprendizaje profundo para NLP desarrollada por Google. BERT, que significa "Transformador de codificación bidireccional", es un modelo preentrenado que se utiliza para tareas de comprensión del lenguaje natural, como el análisis de sentimientos, pudiendo entender el contexto de una palabra en una frase.

El modelo *es_core_news_sm* de *spaCy*, especializado en el procesamiento del lenguaje natural en español, se carga y emplea para tokenizar y preprocesar el texto, permitiendo la extracción de palabras o términos individuales del texto original, filtrando palabras vacías y términos no alfabéticos.

Tras la normalización del texto, el texto es procesado mediante la aplicación del modelo *es_core_news_sm*, para obtener una lista de tokens representativos. El código utiliza la función *extract_keywords* para identificar el token más común en el texto procesado como palabra clave. Se calcula el *accuracy* del modelo comparando las palabras clave predichas con las procesadas.

Posteriormente, el código integra el modelo BERT preentrenado *nlptown/bert-base-multilingual-uncased-sentiment*. Este modelo multilingüe de análisis de sentimientos, proporcionado por NLP Town, categoriza opiniones en una escala de 1 a 5 estrellas, donde 1 es el sentimiento más negativo y 5 el más positivo. Dichas calificaciones son luego mapeadas a categorías de sentimientos: '*negative*', '*neutral*' y '*positive*'.

Como entrada para la clasificación la función *predict_sentiment* combina el texto procesado y la palabra clave predicha antes de pasar al clasificador BERT, de esta forma se refuerza el análisis del sentimiento para el POE. Si el clasificador no devuelve resultado, se asigna un sentimiento '*neutral*'.

```
def predict_sentiment(processed_text, predicted_target):
    combined_text = ' '.join(processed_text) + ' ' + predicted_target
    result = classifier(combined_text)
    if not result:
        return 'neutral'
    stars_label = result[0]['label']
    sentiment = stars_to_sentiment(stars_label)
    return sentiment
```

Este modelo BERT se aplica para inferir el sentimiento de los textos procesados sobre el POE, ofreciendo un análisis contextualizado y preciso gracias a su capacidad para entender el contexto y la relación entre palabras.

Sin embargo, en este problema no es posible el uso de BERT para inferir el análisis de sentimientos que los titulares financieros tienen sobre las empresas o sobre los consumidores en general, ya que no hay contexto explícito en el titular al respecto.

4.5.5. Modelo híbrido

Este último modelo se plantea como un híbrido utilizando las partes de los anteriores modelos que mejores métricas han obtenido tal como se resumen en la Tabla 25. En este caso, tanto en la extracción del POE como en el AS del POE se usa openAI. Para la clasificación del AS de las Empresas de y los Consumidores se usará spaCy en el preprocesamiento de los textos y *Random Forest* en la clasificación tal como se muestra en la Ilustración 10.

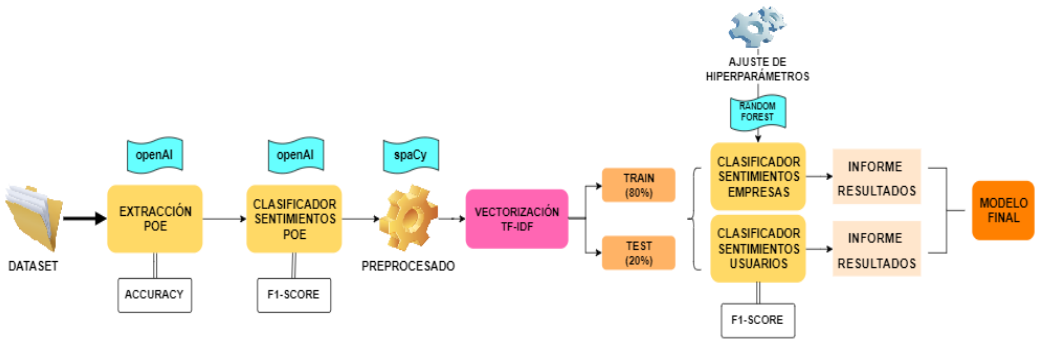

Ilustración 10. Esquema del modelo híbrido seleccionado

La primera parte de este modelo utiliza openAI para la extracción del POE y para el análisis del sentimiento tal como se indicó en el apartado 0

Modelo openAI. En esta ocasión se ha optado por usar en las pruebas un muestreo del dataset del 10%, lo que equivale a algo más de 600 titulares de noticias. Al ampliar el número de datos, se modificarán levemente las métricas obtenidas para este modelo anteriormente, siendo las nuevas un reflejo más fiel de la capacidad del modelo.

El proceso principal, al igual que en el apartado 0, consta de un *prompt* base que sirve como plantilla para las solicitudes que se enviarán a OpenAI. Este *prompt* está diseñado para que el modelo de OpenAI extraiga de forma apropiada el principal objeto económico de los titulares de noticias financieras.

```
prompt_base = """Contesta con una única palabra o palabra compuesta.
Te voy a dar tres ejemplos de titulares de noticias financieras y de
cuál es su principal objeto económico. Ejemplo 1: 'Bayer presenta un
ERE para 75 personas en Sant Joan Despí (Barcelona)'. En este primer
titular, el principal ente económico es 'Bayer'. Ejemplo 2: 'Banc
Sabadell vende su gestora a Amundi con 351M en plusvalías'. En este
segundo titular, el principal objeto económico es 'Banc Sabadell'.
Ejemplo 3: 'Los datos sobre el uso de los ascensores arrojan una caída
del 45% en la afluencia a la oficina por ómicron'. En este tercer y
último ejemplo, el principal objeto es 'uso de los ascensores'. Ahora
debes tú extraer el principal objeto económico de este titular: '{}'.
```

```
Basado en los ejemplos previos sobre titulares financieros y sus
objetos económicos principales, identifica el principal objeto
económico del anterior titular y responde siempre con una única
palabra o palabra compuesta."""
```

Es consulta se itera sobre cada fila del *DataFrame*, formateando el prompt con el texto del titular correspondiente en cada fila. Para cada iteración, se realiza una solicitud a OpenAI para obtener una predicción del principal objeto económico presente en el titular. La respuesta obtenida se procesa para extraer el texto predicho y se añade a una lista de predicciones, calculando al final del proceso la precisión del modelo.

Subsecuentemente, se inicia un nuevo proceso para predecir el sentimiento asociado a cada titular financiero con respecto al objeto económico identificado previamente. Para ello, se prepara un nuevo *prompt* de sentimiento y se itera nuevamente sobre el *DataFrame*. En cada iteración, se formula una nueva solicitud a OpenAI, utilizando el texto del titular y la predicción del objeto económico como inputs, para obtener una predicción del sentimiento del titular:

```
sentiment_prompt_poe = """El titular es: '{}'. El principal objeto
económico identificado es: '{}'. ¿Cuál es el sentimiento del titular
respecto al principal objeto económico? Responde con la palabra exacta
'positivo', 'neutral' o 'negativo'."""
```

Para finalizar, se realiza una serie de operaciones de preprocesamiento y mapeo para asegurar la consistencia en las etiquetas de sentimiento y se calcula el F1 Score para evaluar el rendimiento del modelo en la predicción del sentimiento.

En una segunda parte del código se aborda el análisis de sentimientos de cada titular para las empresas y los consumidores.

Al igual que en el apartado 4.5.2, se usa el modelo de lenguaje español pre-entrenado "*es_core_news_sm*" de spaCy. Se realiza un preprocesamiento previo del texto implica la eliminación de acentos y la tokenización del texto, donde se aplican funciones específicas para normalizar y tokenizar el texto en las columnas relevantes, eliminando *stopwords*, y creando nuevas representaciones textuales procesadas en el *dataframe*.

Para los conjuntos de entrenamiento y prueba, en esta ocasión se realiza una división del 80% para entrenar y un 20% para test. El motivo de no usar en este caso un conjunto de validación, como excepción, es que estamos probando con sólo un 10% del conjunto de datos para no tener un sobrecoste económico, ya que la API de openAI es de pago y no ofrece ningún tipo de prueba gratuita.

En cuanto al modelado, se define un *pipeline* que incluye un vectorizador TF-IDF y un clasificador basado *Random Forest*, al igual que se realizó en el apartado 4.5.2. Se implementa una búsqueda de cuadrícula (*GridSearchCV*) con validación cruzada para optimizar los hiperparámetros del modelo, utilizando como métrica el promedio ponderado del F1-score. Esta búsqueda de cuadrícula se realiza por separado para los modelos de sentimiento de empresas y consumidores.

La búsqueda se realiza sobre el siguiente conjunto de hiperparámetros:

```
param_grid = {
    'tfidf__max_df': [0.85, 0.9, 0.95],
    'tfidf__min_df': [2, 3, 5],
    'clf__n_estimators': [50, 100, 200],
    'clf__max_depth': [None, 10, 20, 30],
}
```

Los hiperparámetros son los siguientes:

1. *tfidf__max_df*: Es el máximo umbral de frecuencia de documento para eliminar términos que aparecen demasiado frecuentemente. Si un término aparece en más del max_df% de los documentos, se descarta. En este caso, se están probando los valores 0.85, 0.9 y 0.95.

2. *tfidf__min_df*: Es el mínimo umbral de frecuencia de documento necesario para incluir un término en el vocabulario. Si es un entero, un término debe aparecer al menos en *min_df* documentos. Si es un *float*, el valor representa una fracción de documentos. Aquí, se están probando los valores 2, 3 y 5.

3. *clf__n_estimators*: Es el número de árboles en el bosque del modelo *RandomForest*. Más árboles pueden resultar en un modelo más robusto, pero computacionalmente más intensivo. Los valores en este caso son 50, 100 y 200.

4. *clf__max_depth*: Es la profundidad máxima de cada árbol en el *RandomForest*. Si su valor es *None*, los nodos se expanden hasta que todas las hojas sean puras o hasta que contengan menos de *min_samples_split* muestras. Se están probando los valores None, 10, 20 y 30.

Se indican a continuación los mejores hiperparámetros obtenidos en cada modelo.

Clasificador de sentimientos para las Empresas:

```
best_companies={'clf__max_depth': None, 'clf__n_estimators': 50,
'tfidf__max_df': 0.95, 'tfidf__min_df': 3}
```

f1-score companies: 0.5852

Clasificador de sentimientos para los Consumidores:

```
best_consumers={'clf__max_depth': 30, 'clf__n_estimators': 50,
'tfidf__max_df': 0.95, 'tfidf__min_df': 5}
```

f1-score consumers: 0.5542

4.6. Evaluación

Una vez generados los modelos, el siguiente paso consiste en analizar las salidas de los diferentes modelos y comparar las métricas obtenidas para cada modelo.

4.6.1. Modelo NLTK

Extracción del POE:

Accuracy for predicted_target: 0.5529

En la extracción del principal ente económico de la noticia con *NLTK* hemos obtenido un *accuracy* del 55%.

Ejemplos de algunos aciertos obtenidos:

Titular	POE real	Predicción
Bankinter considera que la opa de KKR sobre Telepizza tiene un precio "atractivo"	bankinter	bankinter
Daimler y la china BYD lanzarán este año una marca de eléctricos de alta gama	daimler	daimler
La digitalización, clave también para atraer talento	digitalizacion	digitalizacion

Tabla 10. Ejemplos de aciertos obtenidos en la predicción del POE con NLTK

Ejemplos de algunos errores obtenidos:

Titular	POE real	Predicción
Vialegis Dutilh incorpora como socia a Marta Sanz, de Ramón y Cajal, para Inmobiliario	marta sanz	vialegis
Dorna Sports se alía con LaLiga para luchar contra la piratería audiovisual	dorna sports	dorna
La fotovoltaica española despierta el apetito de los inversores extranjeros	fotovoltaica espanola	fotovoltaica
Credit Suisse y ataque ruso: “Los mercados aprenderán a vivir con la realidad de un nuevo orden mundial”	mercados	credit
Restaurant Brands nombra a Jorge Carvalho nuevo director general de Burger King en España y Portugal	jorge carvalho	restaurant

Tabla 11. Ejemplos de errores obtenidos en la predicción del POE con NLTK

Un análisis de la Tabla 11 nos muestra que el modelo tiende a priorizar entidades nombradas que son más frecuentemente asociadas con contextos económicos en el entrenamiento, como “Vialegis” o “Credit”, por encima de nombres de individuos como “Marta Sanz” o “Jorge Carvalho”.

Además, se observa que el modelo prioriza palabras que aparecen al inicio del titular, como se observa en el caso de “Dorna Sports”, lo que sugiere una posible influencia del posicionamiento de las palabras en el texto.

En otros casos, en los que el POE es una palabra compuesta, el modelo tiende a escoger sólo la primera palabra clave, como en “fotovoltaica española”.

Clasificadores de sentimientos:

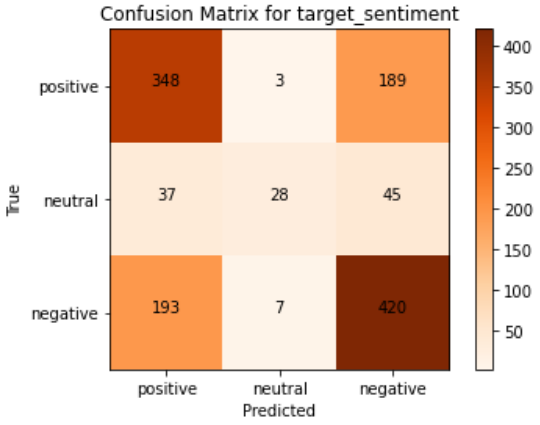

Ilustración 11. Matriz de confusión clasificación sentimientos POE - NLTK

F1-score: 0.6194

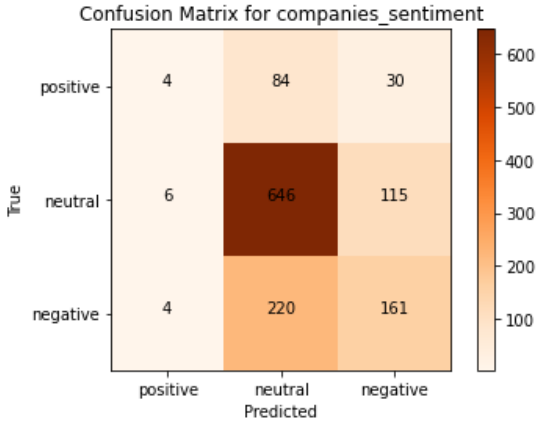

Ilustración 12. Matriz de confusión clasificación sentimientos Empresas - NLTK

F1-score: 0.6013

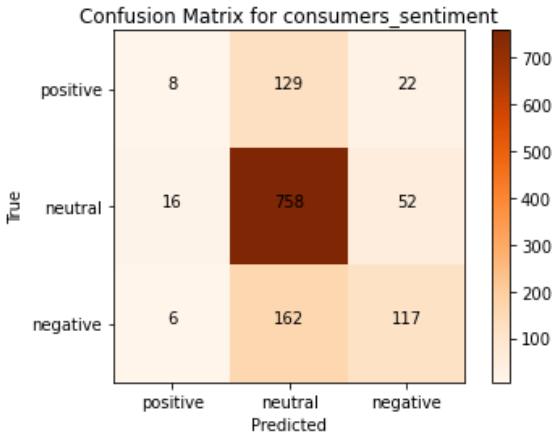

Ilustración 13. Matriz de confusión clasificación sentimientos Consumidores - NLTK

F1-score: 0.6468

Los resultados obtenidos con los clasificadores del AS NLTK en los tres agentes económicos queda reflejado en las matrices de confusión.

En la Ilustración 11 podemos observar la matriz de confusión del clasificador de sentimientos del POE. En este clasificador las clasificaciones neutrales apenas representan un 9,5% de las clasificaciones, por lo que es lógico que la mayoría de las predicciones sean positivas o neutras. El modelo captura correctamente un 65% de las clasificaciones positivas y un 68% de las negativas, pero apenas un 25% de las neutrales.

LIME (*Local Interpretable Model-agnostic Explanations*) es una técnica para interpretar modelos de *machine learning*, y tiene implementaciones específicas para diferentes tipos de datos, incluyendo texto y datos tabulares. Usamos esta herramienta para

analizar algunas de las clasificaciones realizadas en los tres clasificadores empezando por el AS del POE:

TITULAR	POE	AS etiquetado	AS modelo
Nissan prevé que más del 90% de sus vehículos esté electrificado en menos de un año	nissan	positive	negative
Cameron quiere financiar con peajes las nuevas carreteras en Inglaterra	cameron	negative	positive

Tabla 12. Ejemplos de AS del POE errados por NLTK

El análisis de explicabilidad LIME de los dos ejemplos anteriores muestra lo siguiente:

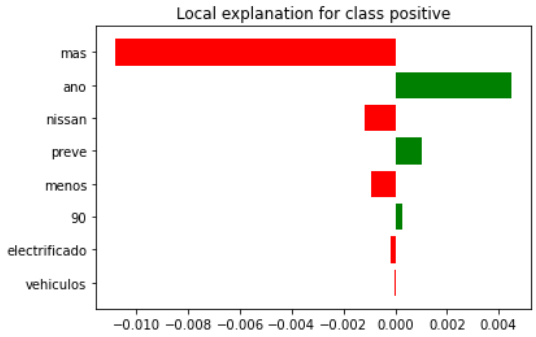

Ilustración 14. Ejemplo 1 de explicabilidad del AS en el POE en el mod. NLTK por LIME

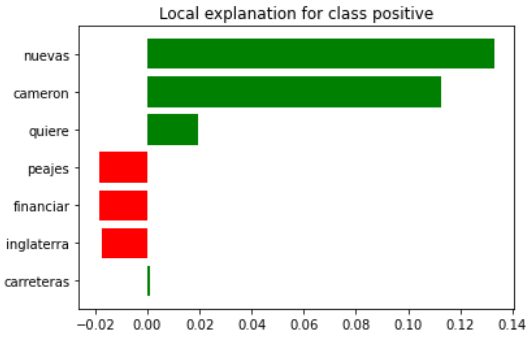

Ilustración 15. Ejemplo 2 de explicabilidad del AS en el POE en el mod. NLTK por LIME

En estos dos casos errados en la clasificación del AS del POE se observa que el modelo se equivoca, o bien asignando pesos negativos a ciertas palabras que no necesariamente lo son, o dando un peso positivo significativo a palabras positivas, pero en el contexto en el que se encuentran generan realmente un sentimiento negativo.

Así, por ejemplo, en la Ilustración 14 se observa que el siguiente titular “Nissan prevé que más del 90% de sus vehículos esté electrificado en menos de un año”. Este titular esta etiquetado con un sentimiento positivo para el POE (Nissan), pero se clasifica con sentimiento negativo ya que el modelo les da un peso negativo a las palabras “mes”, “Nissan”, “menos”, “electrificado” y “vehículos”. En el segundo ejemplo de la Tabla 12, el titular “Cameron quiere financiar con peajes las nuevas carreteras en Inglaterra” con un sentimiento negativo para el POE (Cameron), es interpretado como positivo por el modelo al darle un peso significativo a las palabras “nuevas”,” Cameron”, “quiere” y “carreteras” tal como se observa en la Ilustración 15.

En lo referente a la clasificación del AS de las compañías, el 60,4% de las etiquetas de sentimiento son neutrales, por lo que el modelo realiza la mayoría de sus predicciones de este tipo tal como se observa en la Ilustración 12. Apenas un 10% de los sentimientos etiquetados son positivos y eso hace que el clasificador apenas logre capturar etiquetas de este sentimiento. Sin embargo, logra capturar un 84% de las etiquetas neutrales y un 42% de las negativas.

La Tabla 13 muestra dos ejemplos errados por el clasificador del AS para las Empresas:

TITULAR	POE	AS etiquetado	AS modelo
Avante pone a disposición de las empresas sus agendas personalizadas en el exterior	avante	positive	negative
Ferrovial a la CNMV: "El Caso Palau supone un riesgo reputacional"	ferrovial	neutral	negative

Tabla 13. Ejemplos de AS de las Empresas errados por NLTK.

El análisis de explicabilidad LIME de los dos ejemplos anteriores muestra lo siguiente:

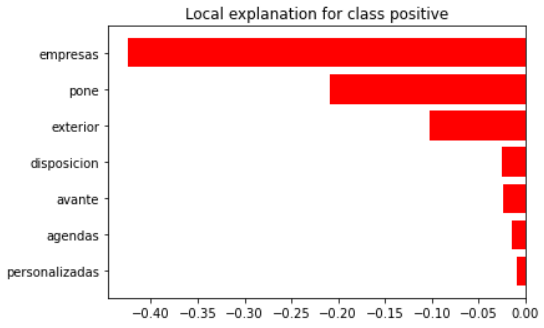

Ilustración 16. Ejemplo 1 de explicabilidad del AS en las Empresas en el mod. NLTK por LIME

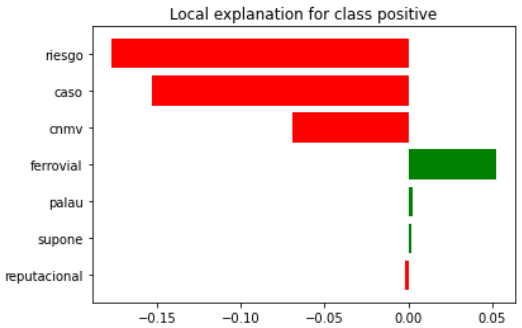

Ilustración 17. Ejemplo 2 de explicabilidad del AS en las Empresas en el mod. NLTK por LIME

En estos dos ejemplos se observa de nuevo que el clasificador erra cuando, o bien asigna pesos negativos a palabras que no necesariamente lo son (Ilustración 16), o bien cuando las sumas de pesos decantan el clasificador hacia un sentido u otro de forma equivocada (Ilustración 17).

La Ilustración 13 muestra la matriz de confusión en el AS de los consumidores. De nuevo, en este clasificador hay mayoría de neutrales (65,6%) lo que hace que el modelo sea capaz de etiquetar correctamente el 92% de estas etiquetas. Sin embargo, solo

logra etiquetar correctamente un 5% de las etiquetas positivas y un 41% de las negativas.

La Tabla 14 muestra dos ejemplos errados por el clasificador del AS para los Consumidores:

TITULAR	POE	AS etiquetado	AS modelo
Este argentino promete ganarle a la inflación con su app y ya le "sacó" 100 mil clientes a los bancos	argentino	neutral	negative
Bruselas exige a Ryanair que devuelva 8,5 millones de ayuda ilegal de Francia	bruselas	positive	neutral

Tabla 14. Ejemplos de AS de los Consumidores errados por NLTK.

El análisis de explicabilidad LIME de los dos ejemplos anteriores muestra lo siguiente:

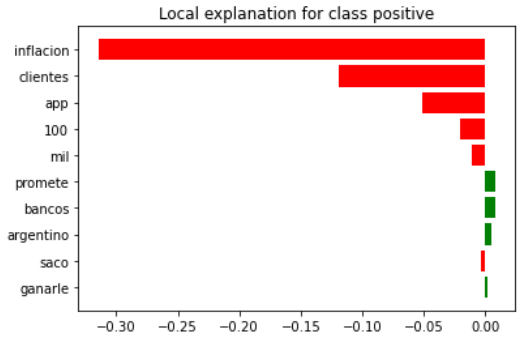

Ilustración 18. Ejemplo 1 de explicabilidad del AS en los Consumidores en el mod. NLTK por LIME

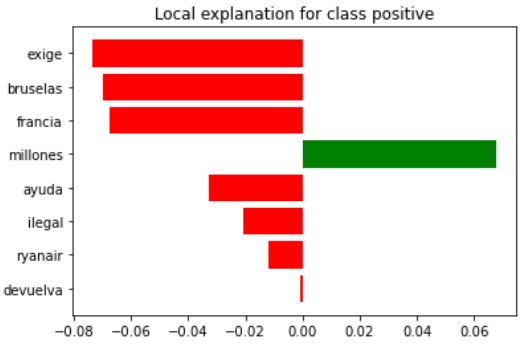

Ilustración 19. Ejemplo 2 de explicabilidad del AS en los Consumidores en el mod. NLTK por LIME

En este caso, para el clasificador del AS de los Consumidores, observamos como los pesos asignados por el clasificador a ciertas palabras decantan la clasificación hacía un sentimiento incorrecto. Así, por ejemplo, el titular “Este argentino promete ganarle a la inflación con su app y ya le "sacó" 100 mil clientes a los bancos” que está etiquetado como neutral para con los consumidores, es clasificado como negativo (Ilustración 18) por el fuerte peso negativo que se le asocia a la palabra “inflación”, entre otras. Y en el titular “Bruselas exige a Ryanair que devuelva 8,5 millones de ayuda ilegal de Francia” que está etiquetado como positivo (Ilustración 19) para los consumidores, se clasifica como neutral por encontrar el modelo un equilibrio neutral entre palabras negativas como “exige” y “bruselas” y positivas como “millones”.

En general, lo que se observa analizando las clasificaciones erradas con el análisis de explicabilidad LIME, es que el contexto en que se encuentran las palabras de los titulares, que es fundamental para identificar el sentimiento, no es interpretado correctamente en algunos casos por el modelo, que evalúa el sentimiento de las palabras de forma individual perdiendo el contexto.

4.6.2. Modelo spaCy

Extracción del POE:

Accuracy for predicted_target: 0.5649

En la extracción del principal ente económico de la noticia con *spaCy* hemos obtenido un *accuracy* del 56%.

Ejemplos de algunos aciertos obtenidos:

Titular	POE	Predicción
El IBEX (-0,9%) transita en rojo por un miércoles santo que deja pérdidas en resto de Bolsas europeas	ibex	ibex
Acciona ve imposible adjudicar el contrato de ATLL a Agbar	acciona	acciona
El juez liquida una inmobiliaria de Banco de Valencia, Gesfesa, Igsa y Planea	juez	juez

Tabla 15. Ejemplos de aciertos obtenidos en la predicción del POE con spaCy

Ejemplos de algunos errores obtenidos:

Titular	POE real	Predicción
Fitch avisa de que la unión bancaria no tranquiliza a los inversores	union bancaria	fitch
La confianza empresarial cae un 2,5% por las peores expectativas ante ómicron	confianza empresarial	confianza
Los inspectores de Trabajo preparan huelga en marzo ante el “abandono” de Díaz y Montero	inspectores trabajo	inspectores
Trump: "La paciencia estratégica con Pyongyang se ha acabado"	pyongyang	trump
Bestinver: "Las buenas compañías son caras y las malas no son baratas"	companias	bestinver

Tabla 16. Ejemplos de errores obtenidos en la predicción del POE con spaCy

Un análisis de la Tabla 16 nos muestra que el modelo tiende a fallar (en la mayoría de las ocasiones) en la extracción del ente principal cuando se dan dos situaciones muy concretas:

1- Cuando el POE es una palabra compuesta (por ejemplo ‘inspectores trabajo’ o ‘confianza empresarial’) suele extraer solo la primera palabra.

2- Cuando el titular consta de una cita u opinión de otro sujeto económico, pero este no es en sí mismo el principal ente económico del titular. Por ejemplo, en el titular “Trump: "La paciencia estratégica con Pyongyang se ha acabado"” el POE es Pyongyang y no el citado Trump, pero el modelo suele escoger el primer ente que aparece.

En general, se observa que el modelo tiende a priorizar entidades que son reconocidas como actores principales en el sector financiero en lugar del verdadero POE del titular.

Clasificadores de sentimientos:

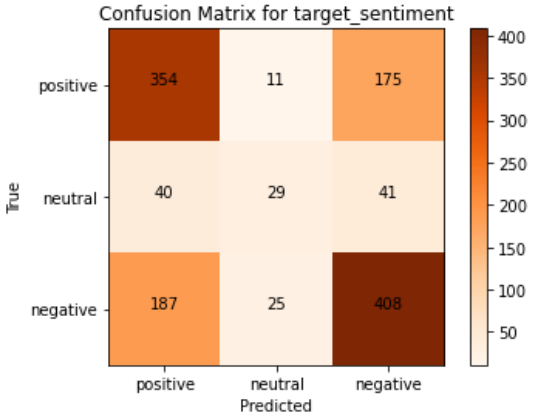

Ilustración 20. Matriz de confusión clasificación sentimientos POE - spaCy

F1-score: 0.6175

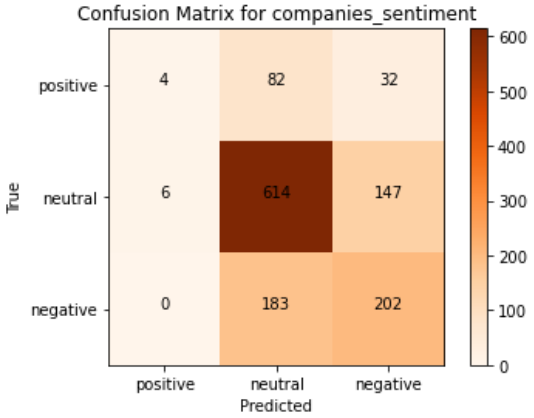

Ilustración 21. Matriz de confusión clasificación sentimientos Empresas - spaCy

F1-score: 0.6163

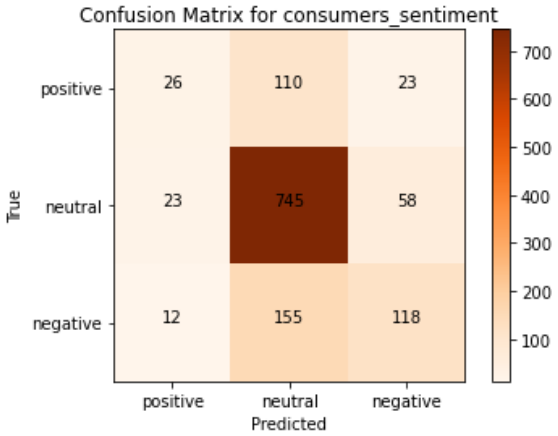

Ilustración 22. Matriz de confusión clasificación sentimientos Consumidores - spaCy

F1-score: 0.6668

Los resultados obtenidos con los clasificadores del AS spaCy en los tres agentes económicos queda reflejado en las matrices de confusión.

En primer lugar, para el AS del POE, se observa en la Ilustración 20 que la mayoría de las clasificaciones son positivas o negativas, lo cual es lógico teniendo en cuenta que están muy desbalanceadas con apenas un 9,5% de clasificaciones neutrales (ver apartado 4.3). El modelo captura correctamente un 65,5% de las clasificaciones positivas y un 65,8% de las negativas.

Algunos ejemplos de clasificaciones erradas:

TITULAR	POE	AS etiquetado	AS modelo
La gran banca paga 32,1 millones a sus consejeros hasta junio, un 15,4% más	banca	positive	negative
El Popular confirma lo que desmintió el jueves: venta urgente	popular	negative	positive

Tabla 17. Ejemplos de AS para el POE errados por spaCy

El análisis de explicabilidad LIME nos muestra lo siguiente para los dos ejemplos anteriores:

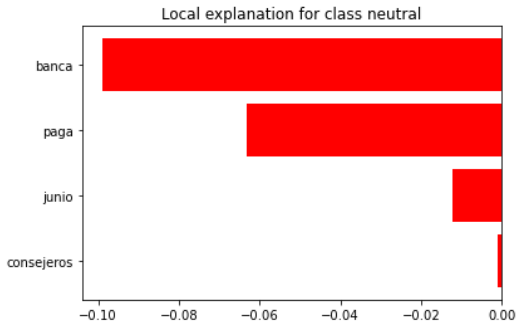

Ilustración 23. Ejemplo 1 de explicabilidad del AS para el POE en el mod. spaCy por LIME

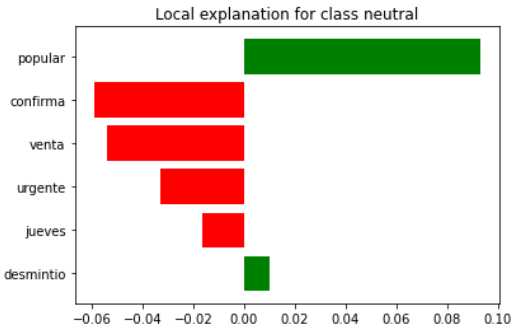

Ilustración 24. Ejemplo 2 de explicabilidad del AS para el POE en el mod. spaCy por LIME

Se observa en ambos casos que el modelo asigna pesos a ciertas palabras sin tener en cuenta el contexto en el que se encuentran, lo que lleva a una clasificación incorrecta. En el primer ejemplo de la Tabla 17, el titular “La gran banca paga 32,1 millones a sus consejeros hasta junio, un 15,4% más” etiquetado como positivo para el POE (banca), es clasificado negativo ya que el modelo asigna pesos negativos (Ilustración 13) a las palabras “banca” y “paga”, entre otras. En el segundo ejemplo de la Tabla 17, el titular “El Popular confirma lo que desmintió el jueves: venta urgente” que en este caso tiene etiquetado un sentimiento negativo para su POE “popular”, es clasificado como positivo (Ilustración 24) ya que el modelo da un peso alto positivo a la palabra “popular”.

Con respecto a la clasificación del AS de las compañías, el 60,4% son neutrales por lo que el modelo realiza la mayoría de sus predicciones como neutrales como se observa en la Ilustración 21. En este clasificador, apenas un 10% de los sentimientos etiquetados son positivos y eso hace que el clasificador apenas logre capturar etiquetas de este sentimiento. Sin embargo, logra capturar un 80% de las etiquetas neutrales y un 35% de las negativas.

Algunos ejemplos de clasificaciones erradas:

TITULAR	POE	AS etiquetado	AS modelo
Las cementeras advierten de cierres de plantas por la escalada del precio de la luz	cementeras	neutral	negative
La gran distribución baja los salarios un 1,5% a sus 230.000 empleados	distribucion	positive	neutral

Tabla 18. Ejemplos de AS de las Empresas errados por spaCy

El análisis de explicabilidad LIME nos muestra lo siguiente para los dos ejemplos anteriores:

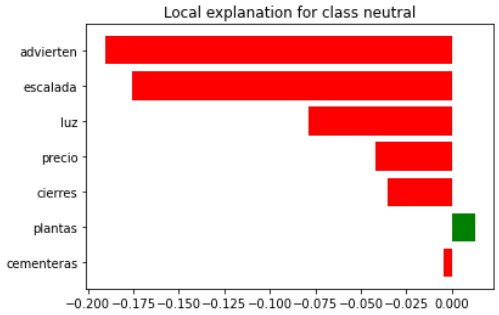

Ilustración 25. Ejemplo 1 de explicabilidad del AS para las Empresas en el mod. spaCy por LIME

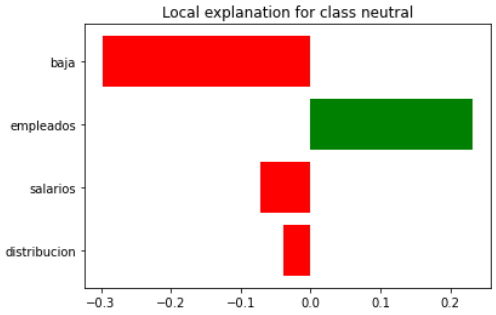

Ilustración 26. Ejemplo 2 de explicabilidad del AS para las Empresas en el mod. spaCy por LIME

De nuevo se observa que los pesos que asigna el modelo a ciertas palabras no tienen en cuenta el contexto en los dos casos analizados. En el primer ejemplo de la Tabla 18, el titular “Las cementeras advierten de cierres de plantas por la escalada del precio de la luz” que tiene un sentimiento neutral para las Empresas, se clasifica como negativo (Ilustración 25) por términos como “advierten”, “escalada” o “cierres”, entre otras. EN el segundo ejemplo, el titular “La gran distribución baja los salarios un 1,5% a sus 230.000 empleados” que está etiquetado como positivo para las empresas en general, obtiene una clasificación neutral ya el modelo considera que el conjunto los pesos negativos y positivos de las palabras (Ilustración 26) queda equilibrado.

En la Ilustración 22 observamos la clasificación de sentimientos de los consumidores. De nuevo, en este clasificador hay mayoría de neutrales (65,6%) lo que hace que el modelo sea capaz de etiquetar correctamente el 90% de estas etiquetas. Sin embargo,

solo logra etiquetar correctamente un 16% de las etiquetas positivas y un 41% de las negativas.

Algunos ejemplos de clasificaciones erradas:

TITULAR	POE	AS etiquetado	AS modelo
Transporte y empresas del sector público muy afectados por la huelga general	transporte	neutral	negative
Suiza prohíbe la venta del Porsche Cayenne diésel de 3.0 litros por manipulación de emisiones	suiza	negative	neutral

Tabla 19. Ejemplos de AS de los Consumidores errados por spaCy

El análisis de explicabilidad LIME nos muestra lo siguiente para los dos ejemplos anteriores:

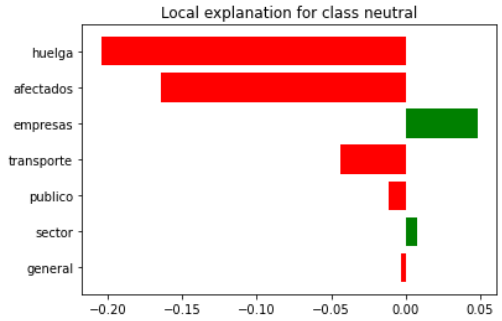

Ilustración 27. Ejemplo 1 de explicabilidad del AS para los Consumidores en el mod. spaCy por LIME

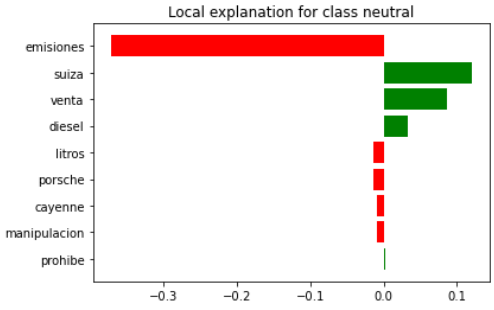

Ilustración 28. Ejemplo 2 de explicabilidad del AS para los Consumidores en el mod. spaCy por LIME

En este último clasificador, se observa como las diferentes interpretaciones que puede tener un mismo titular afecta a la predicción del modelo. En el primer ejemplo de la Tabla 19, el titular “Transporte y empresas del sector público muy afectados por la huelga general” está etiquetada como neutral para los consumidores, pero el clasificador la considera negativa (Ilustración 27) ya que palabras como “huelga” o “afectados” tienen un sentido negativo para los consumidores. En el segundo ejemplo, el titular “Suiza prohíbe la venta del Porsche Cayenne diésel de 3.0 litros por manipulación de emisiones” está etiquetado como negativo para los consumidores, pero el clasificador

considera que hay equilibrio (Ilustración 28) entre términos negativos como “emisiones” o “manipulación” y términos positivos como “suiza”, “venta” o “diesel”.

4.6.3. Modelo openAI

Resultados obtenidos sobre una muestra del 5% del dataset (~300 titulares).

Extracción del POE:

Accuracy for predicted_target: 0.5786

En la extracción del principal ente económico de la noticia con *openAI* hemos obtenido un *accuracy* del 58%.

Ejemplos de algunos aciertos obtenidos:

Titular	POE	Predicción
Google deja a Huawei sin Android ni Gmail. China podría aplicar represalias	google	google
Sorigué ampliará el Hospital de Granollers por 24 millones	sorigué	sorigué
4 obstáculos que pueden penalizar a BBVA en los mercados, pese a sus crecientes perspectivas	bbva	bbva

Tabla 20. Ejemplos de aciertos obtenidos en la predicción del POE con openAI

Ejemplos de algunos errores obtenidos:

Titular	POE real	Predicción
El golf europeo amplía su green, pero no los premios	golf europeo	golf
El mercado da por sentada una reducción de las compras de deuda del BCE	mercado	bce
Aerolíneas, la agonía que no cesa: Lufthansa “sólo” pierde un millón cada dos horas y KLM hará 1.000 despidos	lufthansa	aerolíneas
El petróleo no se recuperará hasta 2023, según Bank of America	petróleo	bank of america
Lecciones que ha dejado la realización de eventos virtuales y que las empresas deben aprender	lecciones	eventos virtuales

Tabla 21. Ejemplos de errores obtenidos en la predicción del POE con openAI

Los resultados obtenidos con OpenAI revelan una tendencia a seleccionar términos más generales, entidades financieras reconocidas, y entidades específicas en lugar de términos más amplios o abstractos. Así, por ejemplo, el modelo seleccionó “BCE” como el POE en lugar de “mercado”. El modelo podría haber interpretado que el BCE, por ser una entidad financiera prominente, era el sujeto principal, omitiendo que el enfoque de la noticia está en la reacción del “mercado”.

También hay que considerar la subjetividad inherente en la tarea de etiquetar el Principal Objeto Económico (POE) manualmente, ya que diferentes individuos pueden interpretar el foco principal de un titular de maneras distintas. Así, por ejemplo, en el titular “El golf europeo amplía su green, pero no los premios” aunque “golf europeo” proporciona un contexto geográfico específico, la selección de “golf” por parte del modelo puede también ser válida si se interpreta que el contexto europeo no es central para el entendimiento de la noticia. Y en la noticia “Lecciones que ha dejado la realización de eventos virtuales y que las empresas deben aprender” “eventos virtuales” puede ser una interpretación válida como POE si se considera que el contexto de la realización de estos eventos es más crucial que las “lecciones” aprendidas.

Clasificadores de sentimientos:

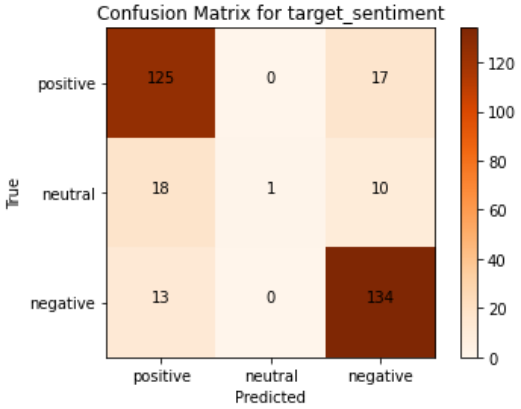

Ilustración 29. Matriz de confusión clasificación sentimientos POE - openAI

F1-score: 0.7829

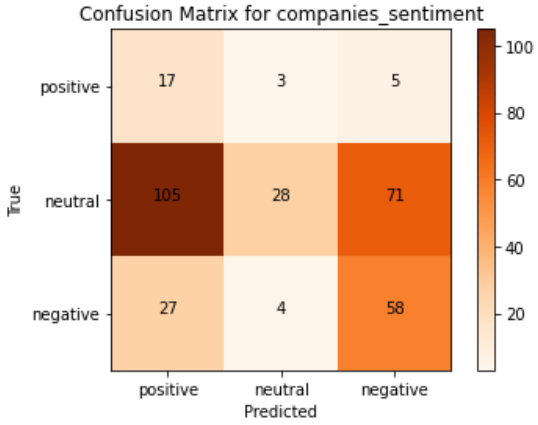

Ilustración 30. Matriz de confusión clasificación sentimientos Empresas - openAI

F1-score: 0.3113

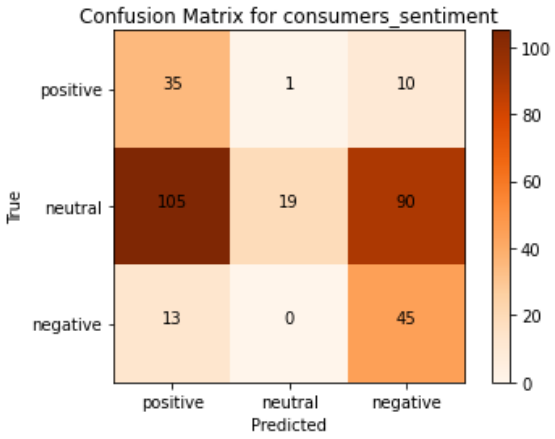

Ilustración 31. Matriz de confusión clasificación sentimientos Consumidores - openAI

F1-score: 0.2410

En la Ilustración 29 observamos como el modelo solo ha realizado una predicción neutral en la clasificación de sentimientos del POE, siendo el resto de las clasificaciones positivas o negativas. Dado que 'neutral' es la categoría claramente minoritaria, esto ha llevado a que se capturen correctamente el 88% de las clasificaciones positivas y el 91% de las negativas.

Para comprender porque se producen las diferencias en las clasificaciones de sentimientos vamos a analizar algunos de ejemplos:

TITULAR	POE	AS etiquetado	AS modelo
Brasil despeja este jueves si su economía entra en una nueva recesión técnica	brasil	neutral	negative
Soltec configura un nuevo consejo de administración para su próxima salida a Bolsa	soltec	neutral	positive
Los recortes no deben terminar aún	recortes	positive	negative
Mercadona cede cuota por el empuje de los frescos en el confinamiento	mercadona	negative	positive

Tabla 22. Ejemplos de AS del POE errados por openAI

"*Brasil despeja este jueves si su economía entra en una nueva recesión técnica*": En el primer titular de la Tabla 22 observamos que la discrepancia entre considerar que el titular es neutral o negativo para Brasil podría deberse a cómo el modelo procesa las palabras asociadas con conceptos económicos negativos, como "recesión técnica". El etiquetado del titular como "neutral" posiblemente fue debido a una interpretación del titular como una mera declaración de hechos, sin una connotación negativa inherente.

"*Soltec configura un nuevo consejo de administración para su próxima salida a Bolsa*": El segundo titular parece indicar claramente que el modelo interpreta términos como "nuevo consejo de administración" y "próxima salida a Bolsa" como indicativos de crecimiento o mejora, lo cual es generalmente considerado positivo en el ámbito financiero. Sin embargo, en el etiquetado se debió considerar que el titular simplemente proporciona información sin un tono positivo ni negativo claro.

"*Los recortes no deben terminar aún*": El término "recortes" suele tener connotaciones negativas en muchos contextos, lo que podría haber influenciado al modelo a etiquetar como negativo. En cambio, en el etiquetado manual, se podría haber interpretado el sentido de que "no deben terminar aún" como algo positivo, quizás en el contexto de recortes de gastos o costos. En cualquier caso, faltaría contexto para determinar el sentimiento en este titular.

"*Mercadona cede cuota por el empuje de los frescos en el confinamiento*": El concepto de "cede cuota" puede haber sido interpretado en el etiquetado como una pérdida de mercado para Mercadona, mientras que el modelo podría haber dado más valor a los términos "empuje de los frescos" durante el confinamiento como un desarrollo positivo general.

En general, las discrepancias observadas entre el etiquetado manual y las predicciones del modelo pueden atribuirse a diferencias en la interpretación del contexto y de los términos específicos utilizados en los titulares, así como a las connotaciones asociadas con ciertas palabras en el ámbito financiero y económico. Además, la subjetividad inherente al análisis de sentimiento puede resultar en variaciones significativas en la interpretación del tono y el sentimiento, tanto por parte de expertos humanos como de modelos de inteligencia artificial.

La Ilustración 30 muestra la matriz de confusión para el AS de las Empresas. En esta clasificación, la mayoría de las etiquetas son neutrales (60,4%). Sin embargo, el modelo muestra una tendencia clara a darles el sentimiento positivo o negativo. De esta forma, el modelo logra capturar un 68% de las clasificaciones positivas y un 65% de las negativas, pero apenas logra capturar un 14% de las neutrales.

Analizamos un par de ejemplos:

TITULAR	POE	AS etiquetado	AS modelo
Magna aumentó un 135% su beneficio neto en el primer trimestre	magna	neutral	positive
Sharp podría reducir un tercio su plantilla	sharp	neutral	negative

Tabla 23. Ejemplos de AS para las Empresas errados por openAI

La Tabla 23 muestra dos titulares neutrales que han sido etiquetados por el modelo como positivo (el primero) y negativo (el segundo).

"*Magna aumentó un 135% su beneficio neto en el primer trimestre*": El etiquetado manual de esta noticia como neutral ha sido posiblemente realizado al interpretarse el titular como una simple presentación de hechos financieros sin relevancia para el resto de las empresas. Por otro lado, el modelo ha interpretado el significativo aumento en el beneficio neto como un evento positivo, probablemente reflejando una percepción general de que un aumento en los beneficios es positivo para las empresas en general.

"*Sharp podría reducir un tercio su plantilla*": De manera similar, el etiquetado de este titular como neutral puede deberse al ser considerado como una declaración neutral de eventos futuros posibles, mientras que el modelo ha interpretado la posible reducción

de plantilla como una señal negativa para las empresas en general. Esta interpretación negativa puede reflejar la percepción generalizada de que reducciones en la plantilla son eventos negativos para las empresas en términos de moral, estabilidad y capacidad productiva.

En ambos casos, las discrepancias entre el AS etiquetado y el AS del modelo pueden reflejar diferencias en la interpretación de los eventos descritos y sus implicaciones para las empresas en general. Mientras que el etiquetado humano parece haber adoptado una postura más descriptiva y neutral, el modelo de *OpenAI* ha inferido el sentimiento general de las empresas basándose en las implicaciones típicamente asociadas con los eventos descritos en los titulares.

Por último, la Ilustración 31 muestra la matriz de confusión para los consumidores. Igual que en el caso anterior, aquí hay una mayoría de etiquetas “neutrales”, pero el modelo tiende a clasificar como “positivo“ o “negativo”. En este caso se logra capturar un 76% de las clasificaciones positivas y un 78% de las negativas, pero apenas se capturan un 9% de las neutrales.

Se analizan un par de ejemplos:

TITULAR	POE	AS etiquetado	AS modelo
Telefónica consolida su racha alcista y alcanza máximos de dos meses	telefónica	neutral	positive
Grifols pospone tres días la publicación de resultados prevista para hoy	grifols	neutral	positive

Tabla 24. Ejemplos de AS para los Consumidores errados por openAI

En la Tabla 24 se muestran dos titulares neutrales que han sido etiquetados como positivos por el modelo.

“*Telefónica consolida su racha alcista y alcanza máximos de dos meses*”: El etiquetado humano ha asignado un valor neutral a este titular, tal vez interpretando que la noticia no tiene un impacto directo aparente sobre los consumidores. En cambio, el modelo ha interpretado que alcanzar "máximos de dos meses" es una señal positiva, que podría reflejar una interpretación de estabilidad o fortaleza empresarial que podría beneficiar a los consumidores a largo plazo, posiblemente en términos de inversión o de continuidad de los servicios de la empresa.

"*Grifols pospone tres días la publicación de resultados prevista para hoy*": Nuevamente, el etiquetado humano ha otorgado un valor neutral, interpretando que posponer la publicación de resultados puede no tener un impacto claro o inmediato en los consumidores. Sin embargo, el modelo ha evaluado esto como positivo para los consumidores. Esto podría deberse a una interpretación de que el aplazamiento proporciona a la empresa más tiempo para preparar y presentar resultados, lo que potencialmente podría traducirse en comunicaciones más claras y precisas a los consumidores y accionistas.

En conclusión, las divergencias entre el AS etiquetado y el AS del modelo para los Consumidores reflejan diferentes interpretaciones de los impactos potenciales de las noticias en los consumidores. Mientras que el etiquetado humano opta por una interpretación más neutral y cautelosa, posiblemente debido a la falta de detalles concretos sobre cómo estas noticias afectan directamente a los consumidores, el modelo de *OpenAI* parece haber inferido implicaciones positivas basándose en interpretaciones potenciales de los eventos descritos en los titulares.

4.6.4. Modelo spaCy + BERT

Extracción del POE:

Accuracy for predicted_target: 0.5649

En este modelo híbrido la primera parte es idéntica a la empleada en el Modelo spaCy, por lo que el resultado de la extracción del POE es el mismo.

Clasificadores de sentimientos:

En este caso solo se realiza un análisis de sentimientos sobre el POE, tal como se indicó en el apartado 4.5.4:

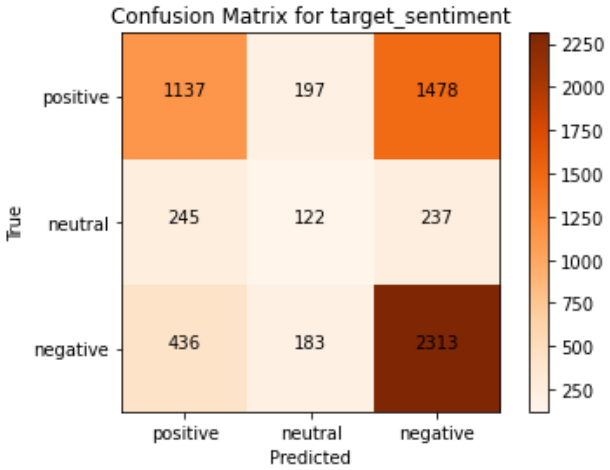

Ilustración 32. Matriz de confusión clasificación sentimientos POE - spaCy+BERT

F1-Score for target_sentiment: 0.5455

En la Ilustración 32 se puede observar, al igual que en los anteriores modelos, un mayor número de clasificaciones positivas y negativas que neutrales, debido principalmente el desbalanceo existente de las etiquetas neutrales. Este modelo logra capturar un 40% de las clasificaciones positivas, un 20% de las neutrales y un 79% de las negativas.

4.6.5. Modelo híbrido

Resultados obtenidos sobre una muestra del 10% del dataset (~600 titulares).

Extracción del POE:

Accuracy for predicted_target: 0.5729

En la extracción del principal ente económico de la noticia con *openAI* hemos obtenido un *accuracy* del 57%.

Para la extracción del POE, se ha utilizado openAI de la misma forma que en el apartado 4.5.3, por lo que la evaluación es la misma que en el apartado 4.6.3 Modelo openAI.

Clasificadores de sentimientos:

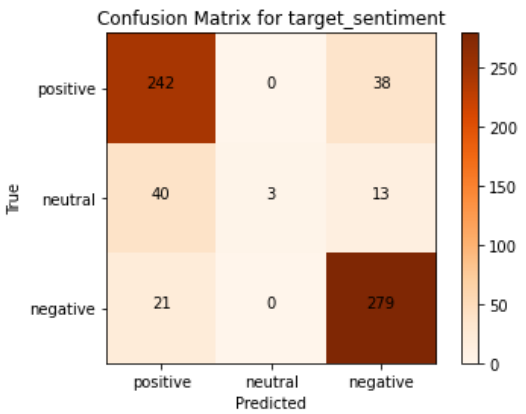

Ilustración 33. Matriz de confusión clasificación sentimientos POE – mod. híbrido

F1-score: 0.7922

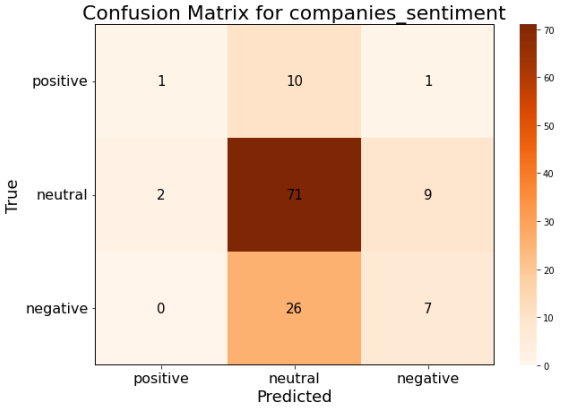

Ilustración 34. Matriz de confusión clasificación sentimientos Empresas – mod. híbrido

F1-score: 0.5852

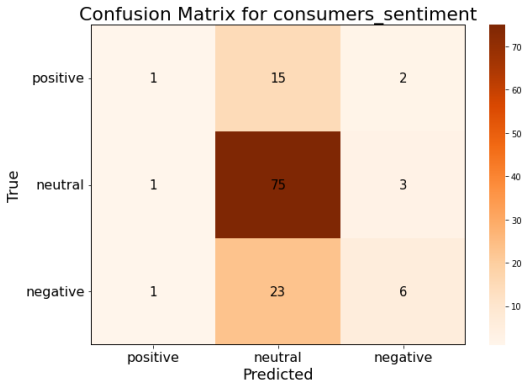

Ilustración 35. Matriz de confusión clasificación sentimientos Consumidores – mod. híbrido

F1-score: 0.5542

Los resultados obtenidos en el AS muestran un clasificador para el AS del POE muy similar (Ilustración 33), como era de esperar, al obtenido para el modelo *openAI* en el apartado 4.6.3. Sin embargo, los resultados obtenidos para el AS de Empresas (Ilustración 34) y Consumidores (Ilustración 35) son más pobres de lo esperado. Esto puede ser debido a dos principales factores. El primero de ellos es que estamos usando solo un 10% del conjunto de datos para entrenar y para pruebas, dado el hándicap económico que tiene la API de openAI. Al entrenar con menos datos, los modelos aprenden obviamente peor. Y el segundo motivo puede ser que el POE extraído por spaCy en el modelo del apartado 4.5.2 sea más significativo para el sentimiento de las Empresas y Consumidores que el extraído por openAI, a pesar de tener este último una superior tasa de aciertos con respecto al POE etiquetado manualmente.

4.6.6. Tabla resumen de resultados

A continuación, se presenta una tabla en forma de resumen con los mejores resultados obtenidos sobre el conjunto de validación.

	Extracción	**Análisis de sentimientos**		
MODELO	**Accuracy POE**	**F1-Score POE**	**F1-Score Empresas**	**F1-Score Consumidores**
NLTK	0.5529	0.6194	0.6013	0.6468
spaCy	0.5649	0.6175	0.6163	0.6668
openAI	0.5786	0.7829	0.3113	0.2410
spaCy+BERT	0.5649	0.5455	-	-
Híbrido	0.5729	0.7922	0.5852	0.5686

Tabla 25. Resumen de resultados obtenidos

5. Conclusión y trabajos futuros

En este proyecto se han explorado distintas soluciones a un problema doble: la extracción del principal objeto económico de un titular financiero, y el análisis de sentimientos de dicho titular frente a tres entes económicos diferentes. Para ello, se ha cumplido con el objetivo general establecido en el apartado 2: explorar diferentes modelos y herramientas y, a partir de ahí, presentar la mejor solución encontrada como solución al problema. Además, se han abarcado los cinco puntos especificados como objetivos específicos, consistentes en la exploración de diferentes modelos y el uso de herramientas de explicabilidad.

Para el primero de los problemas, la extracción del principal objeto económico del titular, se han explorado diferentes opciones. Por un lado, las herramientas "clásicas" como las bibliotecas NLTK o spaCy han mostrado un desempeño moderado en esta tarea. Dichos modelos tienden a priorizar entidades nombradas (que son más frecuentemente asociadas con contextos económicos) por encima de nombres de individuos que muchas veces son los principales entes económicos de un titular. Además, se ha observado que estos modelos priorizan palabras que aparecen al inicio del titular, lo que sugiere una posible influencia del posicionamiento de las palabras en el texto.

Por otro lado, los LLM's han demostrado un desempeño tan solo ligeramente mejor en la tarea de extracción del principal objeto económico. En este aspecto, *openAI* ha mostrado una tendencia a seleccionar de los titulares términos más generales, entidades financieras reconocidas, y entidades específicas en lugar de términos más amplios o abstractos que muchas veces era el principal objeto seleccionado por los expertos. Sin embargo, en este punto es preciso recalcar que hay una importante subjetividad inherente en la tarea de identificar el principal objeto económico manualmente, ya que diferentes individuos pueden interpretar el foco principal de un titular de maneras distintas. Y, de hecho, en muchos casos en los que ha habido discrepancia, el principal objeto identificado por *openAI* podría sustituir el anotado por los expertos de forma bien razonada.

El segundo problema para afrontar ha sido el análisis de sentimientos del titular bajo tres puntos de vista: para el principal objeto económico y para las empresas y para los consumidores en general. La mayor dificultad que ha presentado esta tarea ya se identificó en el apartado 1.1: los titulares de noticias financieras suelen presentar un lenguaje frecuentemente complejo con una fuerte dependencia del contexto, donde diferentes entes tienen sentimientos diferentes para un mismo texto y además no existe, de antemano, un modelo LLM que sea el más adecuado para el ámbito financiero.

Para resolver esta tarea, se han probado tanto clasificadores *Random Forest* como LLM's. Los clasificadores *Random Forest* han mostrado un rendimiento moderado prediciendo el sentimiento para los tres entes económicos, teniendo más problemas para clasificar el sentimiento minoritario en cada clasificador. La mayor parte de predicciones erróneas detectadas en estos modelos han sido por la falta de capacidad

de interpretar el contexto del titular, definiendo el sentimiento solo por las palabras analizadas.

En cuando a los LLM's, *openAI* ha mostrado un rendimiento excelente en el análisis de sentimientos del principal objeto económico y, en general, las discrepancias observadas entre el etiquetado manual y las predicciones de *openAI* se deben a diferencias subjetivas en la interpretación del contexto y de los términos específicos utilizados en los titulares, así como a las connotaciones asociadas con ciertas palabras en el ámbito financiero y económico.

Sin embargo, en el análisis de sentimientos para las empresas y los consumidores, open AI ha mostrado un rendimiento bajo. De nuevo la subjetividad es aquí un factor importante. Mientras que el etiquetado humano de sentimientos en estos entes parece adoptar una postura muy descriptiva y neutral (la mayoría de las etiquetas son neutrales), el modelo de *OpenAI* busca interpretaciones más profundas de los eventos descritos en los titulares y les asigna un sentimiento más frecuentemente positivo o negativo.

En conclusión, tras la evaluación efectuada en este proyecto, se ha observado que los modelos clasificadores "clásicos" demuestran una superioridad en términos de clasificación. Por otro lado, los Grandes Modelo de Lenguaje probados (LLM) muestran una mayor capacidad para explicar de forma razonada sus decisiones y para identificar matices con mayor precisión.

Dadas estas características distintivas y complementarias de ambos tipos de modelos, se postula que una estrategia viable y prometedora podría ser la implementación de modelos clásicos para la filtración o selección inicial de mensajes, para posteriormente someterlos a un proceso de análisis y refinamiento más detallado mediante el uso de LLMs.

5.1. Trabajo futuro

Teniendo en cuenta el trabajo realizado y los resultados obtenidos, se proponen tres líneas principales como trabajo futuro:

1- La vectorización del texto en los modelos que lo han requerido se ha realizado con TF-IDF (ver apartado 3.2.2). Sin embargo, otra forma más eficiente de convertir el texto en vectores numéricos son los *embeddings*. Los *embeddings*, son representaciones vectoriales densas y de baja dimensión de palabras, frases o documentos que capturan el significado semántico y la relación entre las palabras. A diferencia de los métodos de vectorización como TF-IDF, los *embeddings* mapean palabras a vectores de manera que palabras con significados similares estén cercanas en el espacio vectorial. Se deja como trabajo futuro el realizar la conversión del texto a vectores con *embeddings* como *Word2Vec* o *openAI embeddings*.

2- En este proyecto se han usado dos LLM's: BERT (apartado 4.6.4) y openAI (4.6.3). Sin embargo, existen otros LLM's que podrían proporcionar buenos resultados. Se proponen como trabajo futuro dos de ellos:
    a. LLaMA2[9] es un LLM *open source* desarrollado por Meta AI que está entrenado con un conjunto de datos masivo de texto y código, y es capaz de generar texto, traducir idiomas, escribir diferentes tipos de contenido creativo y responder a sus preguntas de forma informativa. Se puede usar de forma local, aunque su instalación puede resultar compleja para su uso con una GPU.
    b. FinBERT[10] es un modelo preentrenado para analizar el sentimiento en textos financieros. Está construido sobre el LLM BERT de base, realizando un entrenamiento adicional con un gran corpus financiero y, por ende, ajustándolo para la clasificación de sentimientos financieros. Está inglés y necesitarían hacerse pruebas para su uso con los titulares en castellano.
3- Se propone una exploración más profunda de la hibridación entre modelos clasificadores clásicos y Grandes Modelos de Lenguaje (LLM). Una estrategia prometedora podría ser la implementación de modelos clásicos para la filtración o selección inicial de mensajes, para posteriormente someterlos a un proceso de análisis y refinamiento más detallado mediante el uso de LLMs. Otra línea futura interesante es la utilización de LLMs para extraer representaciones semánticas de los textos y añadirlas como variables adicionales a los modelos clásicos, buscando sinergizar la eficiencia de los clasificadores tradicionales con la riqueza semántica que proporcionan los LLMs.

[9] https://ai.meta.com/llama/
[10] https://github.com/ProsusAI/finBERT

6. Referencias

Alexander Ligthart, C. C. (2021). Systematic reviews in sentiment analysis: a tertiary study. *Artificial Intelligence Review*.

Ashima Yadav, D. K. (2019). Sentiment analysis using deep learning architectures: a review. *Artificial Intelligence Review*.

Ashish Vaswani, N. S. (2017). Attention Is All You Need. *Conference on Neural Information Processing Systems (NIPS 2017)*.

Bozinovski, S. (2020). Reminder of the First Paper on Transfer Learning in Neural Networks, 1976. *Informatica*.

Cheng-Han Chiang, S.-F. H.-y. (2020). Pretrained Language Model Embryology: The Birth of ALBERT. *Association for Computational Linguistics*.

Chuan-Ju Wang, M.-F. T.-T. (2013). Financial Sentiment Analysis for Risk Prediction. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*.

Daudert, T. (2021). Exploiting textual and relationship information for fine-grained financial sentiment analysis. *Knowledge-Based Systems*.

Frank Z. Xing, E. C. (2018). Natural language based financial forecasting: a survey. *Artificial Intelligence Review*.

Gallardo Arancibia, J. A. (2009). Metodología para la definición de requisitos en proyectos de data mining. *Facultad de Informática (UPM)* .

Goodell, J. W., Kumar, S., Rao, P., & Verma, S. (2023). Emotions and stock market anomalies: A systematic review. *Journal of Behavioral and Experimental Finance*.

Gorod, N. (2021). Top Librerías de Python para NLP. *Data & Artificial Intelligence (https://noeliagorod.com/2021/11/25/top-librerias-de-python-para-nlp-2/)*.

Gupta, A. (2021). Top 8 Python Libraries For Natural Language Processing (NLP) in 2021. *Analytics Vidhya (https://www.analyticsvidhya.com/blog/2021/05/top-8-python-libraries-for-natural-language-processing-nlp-in-2021/)*.

Honnibal, M., & Montani, I. (2016). spaCy: A Fast and Efficient Toolkit for Natural Language Processing in Python. *EMNLP*.

IBM. (2012). *ibm.com.* Obtenido de Manual CRISP-DM de IBM SPSS Modeler: https://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/15.0/es/CRISP-DM.pdf

Jacob Devlin, M.-W. C. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers). Cedarville: Association*.

Jain, A. P., & Dandannavar, P. (2018). Application of machine learning techniques to sentiment analysis. *2018 International Conference on Orange Technologies (ICOT)*.

Jireh Yi-Le Chan, K. T. (2022). State of the art: a review of sentiment analysis based on sequential transfer learning. *Artificial Intelligence Review*.

John W. Goodell, S. K. (2023). Emotions and stock market anomalies: A systematic review. *Journal of Behavioral and Experimental Finance*.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*.

José Antonio García-Díaz, F. G.-S.-G. (2023). Smart Analysis of Economics Sentiment in Spanish Based on Linguistic Features and Transformers. *IEEE Access (Volume: 11)*.

Karthika, P., Murugeswari, R., & Manoranjithem, R. (2019). Sentiment Analysis of Social Media Network Using Random Forest Algorithm. *IEEE*.

Katikapalli Subramanyam Kalyan, A. R. (2021). AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing. *Arxiv.org*.

Kumar, N. V., & Mehrotra, S. (2022). A Comparative Analysis of word embedding techniques and text similarity Measures. *5th International Conference on Contemporary Computing and Informatics (IC3I).*

Liu Y, O. M. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Liu, B. (2020). Sentiment analysis: mining opinions, sentiments, and emotions. *Cambridge University Press*.

Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. *Association for Computational Linguistics*.

Pan, S. J., & Yang, Q. (2009). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*.

Prashant Johri, S. K.-T. (2021). Natural Language Processing: History, Evolution, Application, and Future Work. *Lecture Notes in Networks and Systems (LNNS,volume 167)*.

Ronghao Pan, J. A.-D.-S.-G. (2023). Evaluation of transformer models for financial targeted sentiment analysis in Spanish. *PeerJ Computer Science*.

Salton, G., & McGill, M. (1971). The SMART Retrieval System. *Englewood Cliffs, NJ: Prentice-Hall*.

Singh, G., & Singh, A. (2022). A Comparative Study of Word Embeddings for Similarity Measures. *Information Processing & Management*.

Tomas Mikolov, I. S. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*.

Victor SANH, L. D. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv preprint*.

Xiaodong Li, H. X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*.

Yaakov HaCohen-Kerner, D. M. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *plos.org*.

Yassine Al Amrani, M. L. (2018). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. *PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING IN DATA SCIENCES, ICDS2017*.

Zeynep Hilal Kilimci, D. O. (2019). Financial Sentiment Analysis for Predicting Direction of Stocks using Bidirectional Encoder Representations from Transformers (BERT) and Deep Learning Models. *International Conference on Innovative and Intelligent Technologies (ICIIT-19)*.

Apéndice – repositorio GitHub

Los *scripts* de los modelos usados en este proyecto en encuentran en el siguiente repositorio de GitHub:

https://github.com/runciter2078/tfm

Los datos de entrenamiento usados se encuentran en la web de la competición *IBERLEF 2023 Task - FinancES. Financial Targeted Sentiment Analysis in Spanish* de *CodaLab*:

https://codalab.lisn.upsaclay.fr/competitions/10052