



**UNIVERSIDAD
DE GRANADA**

TRABAJO FIN DE MÁSTER
MÁSTER EN CIENCIA DE DATOS E INGENIERÍA DE
COMPUTADORES

Procesamiento de Lenguaje Natural Explicable para Análisis de Desinformación

Autor

Santiago González Silot

Directores

Juan Gómez Romero

Eugenio Martínez Cámara



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, 9 de julio 2023



Procesamiento de Lenguaje Natural Explicable para Análisis de Desinformación

Autor

Santiago González Silot

Directores

Juan Gómez Romero

Eugenio Martínez Cámara

Procesamiento de Lenguaje Natural Explicable para Análisis de Desinformación

Santiago González Silot

Palabras clave: Noticias falsas, Deep Learning, PLN, XAI, Explicabilidad, SHAP, Desinformación

Resumen

La difusión de noticias falsas constituye un reto de envergadura. El procesamiento del lenguaje natural (PLN) puede contribuir a la detección de noticias falsas aunque, en ocasiones, desde la comunidad científica se afronta este problema sin tener en cuenta la explicabilidad y ausencia de sesgos que necesitan los modelos en este tipo de tareas tan sensibles. En este trabajo se parte de un modelo de *deep learning* basado en modelos de lenguaje, al cual se le va a añadir una capa de explicabilidad usando SHAP para analizar a qué atiende el modelo durante el proceso de aprendizaje. Gracias a SHAP se pudo identificar que el modelo atendía a características espurias y entidades nombradas. Finalmente se propone una metodología que mejora el preprocesamiento del modelo y que sustituye dichas entidades por su categoría de entidad nombrada. Con este *pipeline* se dota al modelo de clasificación de una mayor interpretabilidad, ausencia de sesgos y capacidad de generalización. Para comprobar la mejora de la capacidad de generalización y efectividad de la metodología presentada se evaluó el rendimiento del modelo original y el resultante de este trabajo en un *dataset* externo y tras la aplicación del *pipeline* se observó un 24,95 % de mejora en Macro-F1.

Explainable Natural Language Processing for Disinformation Analysis

Santiago González Silot

Keywords: Deep Learning, NLP, XAI, Explainability, SHAP, Fake News

Abstract

The spread of fake news is a major challenge. Natural language processing (NLP) can contribute to the detection of fake news although, sometimes, the scientific community faces this problem without taking into account the explainability and absence of bias that models need in this kind of sensitive tasks. In this work we start from a deep learning model based on language models, to which we are going to add a layer of explainability using SHAP to analyze what the model attends to during the learning process. Thanks to SHAP it was possible to identify that the model catered to spurious features and named entities. Finally, a methodology is proposed that improves the model preprocessing and replaces these entities by their named entity category. With this methodology, the classification model is provided with better interpretability, absence of biases and generalization capability. To check the improvement of the generalization capacity and effectiveness of the presented methodology, the performance of the original model and the one resulting from this work were evaluated in an external dataset and after the application of the methodology an improvement of 24.95 % in Macro-F1 was observed.

Yo, **Santiago González Silot**, alumno de la titulación Máster en Ciencia de Datos e Ingeniería de Computadores de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 77034478A, autorizo la publicación de la siguiente copia de mi Trabajo Fin de Máster en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Santiago González Silot

Granada a 9 de julio de 2023.

D. **Juan Gómez Romero**, Profesor del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada.

D. **Eugenio Martínez Cámara**, Profesor del Departamento de Informática de la Universidad de Jaén.

Informan:

Que el presente trabajo, titulado ***Procesamiento de Lenguaje Natural Explicable para Análisis de Desinformación***, ha sido realizado bajo su supervisión por **Santiago González Silot**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 9 de julio de 2023.

Los directores:

Juan Gómez Romero Eugenio Martínez Cámara

Agradecimientos

En primer lugar, me gustaría agradecer a Juan y a Eugenio por todas las reuniones, la disponibilidad, paciencia y apoyo absoluto que han tenido conmigo. Sin su guía y mentoría no hubiese podido hacer un trabajo tan completo y correcto como lo ha sido. Gracias por los conocimientos aportados y la oportunidad.

También quiero agradecer a mis padres y pareja y en general a toda mi familia y amigos por haberme apoyado durante todos estos años de grado y máster. Sin un hombro en el que apoyarse este camino hubiese sido mucho más duro.

Finalmente quisiera agradecer a todos los profesores del máster por todo el conocimiento que me han transmitido estos años y su dedicación.

Índice general

1. Introducción	19
1.1. Motivación y trabajo previo	20
1.2. Objetivos	21
1.3. Nuevas aportaciones	21
1.4. Estructura de la memoria	22
2. Contexto	23
2.1. Desinformación	23
2.2. Procesamiento del Lenguaje Natural	25
2.2.1. <i>Transformers</i>	27
2.2.2. Mecanismos de atención	28
2.2.3. Modelos del lenguaje	29
2.3. Inteligencia Artificial Explicable (XAI)	32
2.3.1. SHAP	35
3. Experimentación	39
3.1. Punto de partida	40
3.1.1. Análisis Exploratorio de los datos	42
3.2. Aplicación de SHAP	43
3.2.1. Análisis de algunas frases	44
3.2.2. Resúmenes obtenidos de los Shapley Values	45
3.2.3. Cambios en el preprocesamiento y re-aplicación de SHAP.	48
3.2.4. Conclusiones parciales	50
3.3. Sentiment Analysis (SA) de las palabras más relevantes.	53
3.4. Aplicación de Named Entity Recognition (NER) y sustitución de entidades por un token comodín.	54
3.5. Desarrollo de la metodología final y experimentación con <i>dataset</i> externo.	55
4. Conclusiones y trabajo futuro	57
4.1. Conclusiones	57
4.2. Trabajo futuro	59

Bibliografía

60

Índice de figuras

2.1. Diagrama PLN. Fuente: Pinterest.	25
2.2. Arquitectura de <i>Transformers</i> , <i>Multi-Head-Attention</i> y <i>Attention</i> . Fuente: Medium.	28
2.3. Arquitectura de BERT junto con la arquitectura <i>Transformers</i> . Fuente: ResearchGate.	30
2.4. Representación de la fase de entrenamiento de BERT. Fuente: Medium.	31
3.1. Metodología a seguir en este trabajo.	39
3.2. Distribución del número de palabras del <i>dataset</i>	43
3.3. Ejemplos de frases analizadas usando SHAP.	44
3.4. Palabras con mayor impacto de media.	46
3.5. Palabras con mayor impacto total.	48
3.6. Ejemplos de frases analizadas usando SHAP tras mejorar el preprocesamiento.	49
3.7. Palabras con mayor impacto de media tras mejorar el preprocesamiento.	50
3.8. Ejemplos de análisis con SHAP de Sentiment Analysis	52
3.9. Ejemplo de palabras más relevantes con SHAP de Sentiment Analysis	52
3.10. Ejemplos de textos procesados usando NER.	54
3.11. Metodología desarrollada en este trabajo	56

Índice de tablas

3.1. Comportamientos extraños detectados durante el TFG. . . .	41
3.2. Análisis Básico del conjunto de datos CONSTRAINT AAAL.	42
3.3. Entidades a detectar en el proceso de NER.	55
3.4. Comparativa modelo original-resultante	56

Capítulo 1

Introducción

La era de la información y de la conectividad que actualmente vivimos gracias en gran parte a internet nos ha traído grandes beneficios, y ha impulsado enormemente el progreso de la humanidad. Así mismo, la facilidad y rapidez de propagar y comunicar información ha actualizado, ampliado y fortalecido desafíos y peligros de la comunicación de información.

En este sentido, el reto más relevante es el de discernir si una información es veraz o falsa, y así evitar la desinformación. Ésta se manifiesta a través de la propagación de falsedades, bulos y en muchas ocasiones de mensajes maliciosos con la intención de orientar la opinión pública hacia un determinado punto de vista. La desinformación ha evolucionado de forma paralela a la propia humanidad, y se ha visto potenciada por el propio progreso de la comunicación a través de internet y las redes sociales, como fue evidente durante la campaña del referéndum sobre el Brexit [36] o durante la pandemia de Covid-19.

Esto evidencia el necesario desarrollo de tecnología que asista a las personas a discernir si un mensaje es susceptible de ser falso. Al expresarse las *fake news* principalmente de forma escrita o hablada, es el área del procesamiento del lenguaje natural (PLN) la que inició el desarrollo de modelos, métodos y metodologías encaminadas a la detección de mensajes que puede conducir a la desinformación [53, 38]. En ocasiones la comunidad científica organiza eventos específicos de tratamiento de la desinformación como CONSTRAINT AAAI [51] o FakeDes [29].

Generalmente la comunidad científica se centra únicamente en las métricas de precisión, dejando a un lado factores claves como la interpretabilidad, explicabilidad y ausencia de sesgos en una temática tan sensible como es la desinformación. La Inteligencia Artificial Explicable (XAI) es cada vez más importante para introducir la inteligencia artificial en la sociedad y para adaptarse a las nuevas leyes [24, 33].

La XAI cada vez es más importante para un correcto uso de la Inteligencia Artificial en la sociedad. La población suele ser reticente a adoptar técnicas que no son directamente interpretables, manejables y fiables [79]. Esto se intensifica cuando se utiliza la Inteligencia Artificial de ciertas temáticas delicadas como la sanidad, economía o política.

Además, el uso de XAI para la detección de *fake news* puede ayudar a identificar sesgos y errores que pueden tener los sistemas de clasificación e intentar actuar en consecuencia. Un sistema de clasificación de *fake news* con sesgos políticos, ideológicos o sociales puede ser un peligro para la sociedad, por ello la aplicación de XAI se vuelve indispensable en este tipo de sistemas tan críticos.

Por tanto, en la Sección 1.1 se justifica la necesidad de la realización de este trabajo. En la Sección 1.2 se detallan los objetivos de este trabajo. En la Sección 1.3 se mencionan las nuevas aportaciones de este trabajo. Finalmente en la Sección 1.4 se detalla brevemente la estructura que tiene la memoria de este trabajo.

1.1. Motivación y trabajo previo

Este Trabajo Fin de Máster (TFM) es una continuación del Trabajo Fin de Grado (TFG) de Ingeniería Informática de la Universidad de Granada, desarrollado durante el curso académico 2021-2022 [30]. En el TFG, tras la experimentación entre 28 arquitecturas distintas de modelos de lenguaje para la detección de *fake news* tanto en español como inglés, se presentó flifA, un sistema de detección de *fake news* que consta de 2 modelos basados en BERT/RoBERTa, uno para español y otro para inglés, los cuales obtuvieron resultados que representan el estado del arte en sus respectivas competiciones internacionales de alto prestigio [30]. En concreto se obtuvo un 98,41 de Macro-F1 y un quinto puesto para el modelo en inglés y un 73,77 y un octavo puesto para el modelo en español. En el TFM se continúa el trabajo previo del TFG, tomando el modelo en inglés como punto de partida.

Aun con las buenas excelentes de clasificación obtenidas, sería muy ingenuo pensar que este sistema roza la perfección, pues como bien sabido es los sistemas de *Machine Learning* encuentran un patrón el cual se ajusta a los datos, pero que quizás no tiene un sentido lógico para un ser humano o incluye patrones indeseables, presentes en los datos, los cuales provocan un sesgo peligroso a la hora de usar un sistema de este tipo en un entorno real.

Es por esto que el principal objetivo de este trabajo es la aplicación de métodos de XAI al sistema de detección de *fake news* desarrollado durante mi Trabajo Fin de Grado, para así poder analizarlo en profundidad, conocer en que se fija para tomar una decisión, ver si hay sesgos en el modelo e

intentar paliarlos.

1.2. Objetivos

Si bien el objetivo principal de este TFM es la aplicación de XAI al sistema de detección de *fake news* desarrollado durante el TFG, a continuación se enumeran los objetivos específicos necesarios para la realización del objetivo principal:

1. **Analizar el estado del arte del problema:** Se estudiará el estado del arte del problema de detección de *fake news*, *Deep Learning*, Procesamiento Del Lenguaje Natural (PLN) e Inteligencia Artificial Explicable (XAI).
2. **Estudio y análisis del comportamiento del modelo mediante el uso de técnicas de explicabilidad:** Una vez estudiado el estado del arte en las temáticas anteriormente mencionadas y especialmente en Inteligencia Artificial Explicable, se decidirán que técnicas de explicabilidad aplicar al sistema y así poder estudiar y analizar el comportamiento del modelo.
3. **Propuesta de técnicas para mejorar el modelo desde un punto de vista ético y seguro:** Tras estudiar y analizar los posibles fallos del sistema así como en que partes del texto presta más atención el modelo, se analizarán los problemas que este presenta para así intentar solventarlos.

1.3. Nuevas aportaciones

A continuación se enumeran las nuevas aportaciones obtenidas como resultado del trabajo realizado:

1. Se ha observado que **el modelo de lenguaje para clasificación de *fake news* en ocasiones se fijan en características espurias como símbolos de \$, emoticonos o URLs.**
2. Se ha observado que **es complicado encontrar un razonamiento humano para las palabras en las más que se fija el modelo. Se ha visto que no es sencillo encontrar una interpretación directa de algunas de las palabras que el modelo más presta atención ni de forma indirecta usando técnicas externas como *Sentiment Analysis*.**
3. Se ha observado que **orientando el preprocesamiento del texto**

a una mayor explicabilidad, se soluciona el problema relacionado a la atención prestada en características espurias.

4. Se ha observado que **el modelo presta mucha atención a entidades que aparecen en el texto** (como pueden ser países, personas, etc), lo cual **causa un gran sesgo** en el modelo con las consecuencias éticas correspondientes.
5. **Se ha presentado una metodología de trabajo la cual utiliza los conocimientos obtenidos durante este trabajo para aportar al modelo una mayor explicabilidad y capacidad de generalización. Para afirmar este aumento de la capacidad de generalización se ha evaluado con *dataset* externo distinto al de entrenamiento, logrando una mejora de un 24,95 % de Macro-F1.**

1.4. Estructura de la memoria

Para facilitar el seguimiento de la memoria al lector, esta se ha estructurado en tres capítulos. En la primera sección se introduce al lector en el contexto necesario para la lectura de la memoria, posteriormente se explica toda la metodología y experimentación llevada a cabo. Finalmente se presentan las conclusiones obtenidas junto a posibles trabajos futuros. A continuación se detalla más en profundidad el contenido de cada una de estas partes.

1. **Contexto:** En este capítulo se hará una breve introducción a los distintos conceptos necesarios para la correcta comprensión del trabajo realizado. Esto es, tanto la definición de *fake news* y otros conceptos relacionados como la definición de varios conceptos relacionados con el *Deep Learning*, PLN y XAI. Las secciones relativas a desinformación y PLN fueron explicadas durante el TFG, aun así de nuevo se explican en este TFM para que sea auto contenido.
2. **Experimentación:** En este capítulo se explicará toda la experimentación realizada durante el TFM, tanto la aplicación de SHAP al modelo original, como los resultados obtenidos y las técnicas utilizadas para paliar este problema.
3. **Conclusiones y trabajo futuro:** En el capítulo final se analizarán las conclusiones obtenidas de este trabajo además de mencionar posibles mejoras y trabajo futuro para seguir avanzando en la detección de *fake news* y la Inteligencia Artificial Explicable aplicada en este tipo de dominios.

Todo el código utilizado para el desarrollo de este trabajo se encuentra disponible en el siguiente repositorio.

Capítulo 2

Contexto

Para que el lector pueda entender correctamente el trabajo de investigación e implementación realizado, en este capítulo se presenta el contexto teórico sobre las distintas temáticas específicas de las cuales tratará este trabajo. Estas temáticas son las de *fake news*, Procesamiento del Lenguaje Natural, *Transformers* e Inteligencia Artificial Explicable.

Por tanto en la Sección 2.1 se realizará una introducción a las *fake news* y conceptos relacionados, en la Sección 2.2 sobre Procesamiento del Lenguaje Natural y finalmente en la Sección 2.3 sobre Inteligencia Artificial Explicable.

2.1. Desinformación

La desinformación es un tipo de información falsa o parcialmente falsa, la cual se difunde generalmente de manera intencionada para manipular al lector. En inglés se hace la diferencia entre *disinformation* para cuando la desinformación es de manera intencionada, y se utiliza *misinformation* para cuando se trata de falta de información la cual se comparte de forma no maliciosa aparentemente.

El término desinformación es muy amplio y en ocasiones algunos de los términos relacionados no tienen una definición sólida por la comunidad científica ni periodística. Es por esto por lo que a continuación se presenta una clasificación de los distintos tipos de desinformación según [78].

- **Fake News:** El término *fake news* en los últimos años se ha convertido en el término más comunmente utilizado para identificar información falsa en medios de comunicación, Internet y redes sociales. En [3] se propone una definición más formal de las *fake news*: “*fake news* es un artículo de prensa que es intencionalmente y de manera verificable

falso”. Con esta definición se esclarece que las *fake news* son noticias escritas intencionadamente para desinformar y las cuales se puede verificar la veracidad de la noticia.

Dentro del concepto de *fake news* se pueden diferenciar Varios términos como fabricaciones serias, engaños a gran escala y sátira. Las fabricaciones serias son la forma prototípica de *fake news*. Por otro lado los engaños a gran escala suelen estar más organizados que un artículo y suelen estar dirigidos hacia ciertas figuras públicas. Finalmente la sátira son noticias humorísticas con la intención de entretener. En estas últimas se considera que el lector es consciente de que estas noticias están creadas en un tono humorístico.

- **Rumores:** En la literatura, los rumores son el mayor caso de estudio de información falsa que se propaga por Internet. Estos hacen referencia a información la cual no se ha confirmado su veracidad por medios oficial y la cual se propaga fácilmente en redes sociales como Facebook o Twitter. Aun así los rumores no son producto únicamente de las redes sociales, si no que estas han ayudado a que se propaguen con mayor facilidad [12].

Existen diversas definiciones de rumor según distintas interpretaciones. En [22] se definen los rumores como “declaraciones informativas sin verificar y potencialmente relevantes en circulación”. Por otro lado en [81] se definen los rumores como: “historia en circulación de veracidad cuestionable, la cual es aparentemente creíble pero difícil de verificar y produce suficiente escepticismo y/o ansiedad”.

Ambas definiciones tienen como característica principal la idea de que es información sin verificar. Dentro de la categoría de los rumores, [80] separa los rumores en dos categorías principales. Por un lado están los rumores a gran escala los cuales circulan durante un gran periodo de tiempo, como por ejemplo, las leyendas urbanas y teorías de la conspiración. Por otro lado están los rumores de noticias de última hora, los cuales tienen una gran importancia en la literatura ya que pueden ser muy peligrosos en un corto periodo de tiempo y se han de identificar para parar su difusión.

- **Otros tipos de desinformación:** Si bien las *fake news* y los rumores representan la temática principal de interés en el ámbito de la desinformación, existen otros tipos de desinformación en la web. Algunos de estos tipos son: *Clickbait*, *Social spammers* y opiniones falsas en sitios web de comercio electrónico.

2.2. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN), es un subcampo que pertenece tanto a las ciencias de la computación e inteligencia artificial como a la lingüística. El PLN se dedica al diseño de métodos y algoritmos que tienen como entrada (o salida) lenguaje natural no estructurado, teniendo como objetivo dotar a un ordenador la capacidad procesar y “comprender” un texto para poder interactuar con él. En la figura 2.1 se puede observar donde se encuentra el PLN junto a la Inteligencia Artificial, *Machine Learning* y *Deep Learning*.

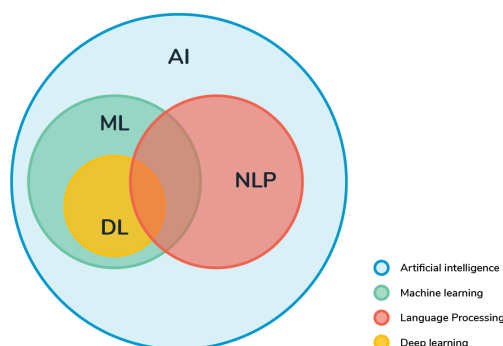


Figura 2.1: Diagrama PLN. Fuente: Pinterest.

A continuación se listan algunas de las tareas más comunes que se resuelven utilizando PLN:

- **Text-to-Speech (TTS):** La tarea de TTS consiste en convertir texto escrito en voz sintetizada. Utilizando técnicas de PLN, se procesa el texto para generar una salida de voz realista y comprensible.
- **Reconocimiento de voz (ASR):** El reconocimiento automático de voz es la tarea de convertir señales de voz habladas en texto escrito. Utilizando técnicas de PLN y *Machine Learning*, se busca transcribir y comprender el contenido de la voz.
- **Clasificación de texto:** Esta tarea implica asignar una o varias etiquetas o categorías a un texto dado. Por ejemplo, clasificar correos electrónicos como spam o no spam, o categorizar noticias en diferentes temas como deportes, política o entretenimiento.
- **Generación de lenguaje:** Esta tarea involucra generar texto coherente y comprensible en lenguaje natural. Puede abarcar desde la generación automática de respuestas en sistemas de chatbots hasta la

generación de descripciones de imágenes o la redacción automática de informes.

- **Análisis de sentimiento:** Esta tarea se refiere a determinar el tono emocional o la polaridad de un texto, es decir, si es positivo, negativo o neutral. Se utiliza para analizar opiniones y emociones en redes sociales, reseñas de productos, comentarios de clientes, etc.
- **Extracción de información:** Esta tarea implica identificar y extraer información estructurada de un texto. Por ejemplo, extraer nombres de personas, fechas, ubicaciones o eventos relevantes de noticias o documentos.
- **Traducción automática:** Se trata de traducir automáticamente el texto de un idioma a otro. Utilizando técnicas de PLN, los sistemas de traducción automática intentan capturar el significado y la intención del texto en un idioma y generar una traducción equivalente en otro idioma.
- **Resumen automático:** Esta tarea implica resumir automáticamente un texto largo en un formato más conciso y comprensible. Puede ser útil para resumir artículos de noticias, documentos técnicos o extractos de libros.
- **Reconocimiento de entidades nombradas (NER):** El NER consiste en identificar y clasificar entidades nombradas, como nombres de personas, organizaciones, ubicaciones, fechas, cantidades, etc., en un texto dado.

Algunas de las técnicas clásicas de PLN usan el tf-idf (Term frequency - inverse document) como entrada para modelos clásicos de Machine Learning como SVM o Árboles de decisión. Si bien estos modelos logran un buen resultado a la hora de resolver los problemas de PLN [43, 19] y más en concreto de clasificación de texto, el estado del arte viene marcado por distintos modelos basados en *Deep Learning* [76, 1], entre los que destacan los siguientes:

- **Recurrent Neural Networks (RNN):** Las RNN son una arquitectura de modelo que procesa datos secuenciales teniendo en cuenta la dependencia temporal. Son especialmente útiles para modelar el contexto a largo plazo en el procesamiento del lenguaje. Sin embargo, las RNN pueden sufrir del problema de desvanecimiento o explosión del gradiente en secuencias muy largas, lo que ha llevado al desarrollo de variantes como las Long Short-Term Memory (LSTM) y las Gated Recurrent Unit (GRU), que solucionan ese problema.

- **Convolutional Neural Networks (CNN):** Las CNN son arquitecturas populares en el procesamiento de imágenes, pero también se han utilizado con éxito en tareas de PLN, especialmente en la clasificación de texto. Las capas convolucionales capturan características locales en un texto, mientras que las capas de agrupación reducen la dimensionalidad. Las CNN son eficientes en términos computacionales y pueden manejar tareas de clasificación de texto de manera efectiva.
- **LSTM (Long Short-Term Memory):** Las LSTM son una variante de las RNN que abordan el problema del desvanecimiento y la explosión del gradiente. Las LSTM utilizan unidades de memoria para mantener información relevante a largo plazo y se han utilizado con éxito en tareas como el etiquetado de secuencias y la generación de texto.
- **Transformers y Modelos de Lenguaje pre-entrenados:** Los Transformers son una arquitectura de modelo que ha revolucionado el campo del PLN. Utiliza mecanismos de atención para capturar las relaciones entre las palabras en un texto. Los Transformers son altamente paralelos y permiten un procesamiento eficiente de secuencias largas. Han demostrado un rendimiento sobresaliente en tareas como traducción automática, generación de lenguaje y análisis de sentimiento.

En concreto, los siguientes apartados se centrarán en el mecanismo de atención, *transformers* y modelos de lenguaje pre-entrenados ya que suponen el estado del arte para problemas de PLN y serán los modelos utilizados como punto de partida en este trabajo.

2.2.1. *Transformers*

En primer lugar, un *transformer* es una arquitectura de red neuronal considera como estado del arte [76] en los modelos secuenciales. En el ámbito del PLN soluciona los problemas de dependencias largas que presentan las redes neuronales recurrentes [73] las cuales procesan el texto palabra a palabra y debido a su propia arquitectura empeoran con frases largas las cuales necesitan de contexto.

La arquitectura de *transformers* fue propuesta por primera vez en el artículo “Attention is All You Need” [76]. Esta arquitectura tiene una estructura *encoder-decoder*, esta estructura se basa en codificar la secuencia de entrada para posteriormente decodificarla, es una estructura típica en la tarea de traducción automática [15]. La peculiaridad de esta arquitectura es el uso únicamente de mecanismos de atención sin la inclusión de RNN o CNN [76]. En la figura 2.2 se puede observar con detalle la arquitectura *transformers*.

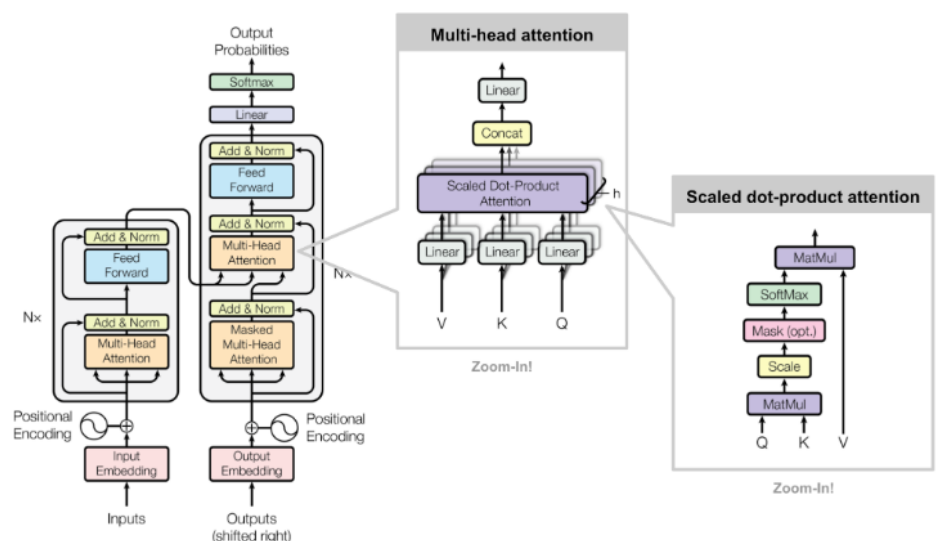


Figura 2.2: Arquitectura de *Transformers*, *Multi-Head-Attention* y *Attention*. Fuente: Medium.

El *encoder* se compone de 6 bloques idénticos, cada uno de ellos se compone de una capa de *multi-head self attention mechanism* (la cual se explica en profundidad en la sección 2.2.2) seguido de una capa *Fully Connected*, es decir una capa de neuronas totalmente conectadas. Además después de tanto el mecanismo de atención, como de la capa *Fully Connected* se encuentra una capa residual [35] y una capa de normalización [6]. La primera de estas se utiliza para que no se pierda ni se degrade la información mientras que la capa de normalización se utiliza para normalizar los datos los cuales han sido “deformados” tras el resto de capas, ayudando así al proceso de aprendizaje.

El *decoder* también se compone de 6 bloques idénticos. Estos bloques tienen la misma estructura que los bloques del *encoder*, con la diferencia de que el *decoder* introduce al inicio del bloque una capa de *multi-head self attention mechanism* enmascarada para que tan solo utilice las palabras que ya ha procesado.

2.2.2. Mecanismos de atención

Los mecanismos de atención [8] permiten a la red centrarse en una parte de la entrada y darle menos atención a otras. Se encarga de analizar la totalidad de la secuencia de entrada y encontrar relaciones entre las distintas partes de la secuencia. Para ello expresa numéricamente las relaciones entre las partes según los grados de asociación entre palabras.

Existen distintos tipos de mecanismos de atención, entre ellos el que se utiliza en la arquitectura *transformers* es llamado *Scaled Dot-Product Attention* el cual tiene la siguiente formulación:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Además, en vez de utilizar una única función de atención se utilizan varias y posteriormente se concatenan, lo cual lo hace un modelo de *Multi-Head-Attention*. Este mecanismo de atención recibe como entrada *queries* y *keys* [8] de dimensión d_k y valores de dimensión d_v . Se calcula el producto escalar QK^T y posteriormente se divide entre $\sqrt{d_k}$. La única diferencia entre *Dot-Product Attention* y *Scaled Dot-Product Attention* es el factor de escala $\frac{1}{\sqrt{d_k}}$ lo cual hace que este tipo de atención sea más eficiente en tiempo y en espacio [73].

2.2.3. Modelos del lenguaje

Los Modelos del Lenguaje (LM) son *word embeddings* contextuales, es decir, los modelos del lenguaje son distribuciones de probabilidad sobre las palabras o secuencias de palabras. Dada una frase, un modelo del lenguaje ha de identificar si esta frase es plausible o no dada la gramática del lenguaje [54]. En los últimos años se han publicado una gran cantidad de modelos basados en redes neuronales profundas, entre los más populares en la literatura destacan los siguientes:

- BERT [21].
- RoBERTa [39].
- GPT2 [58].
- GPT3 [13].
- ELECTRA [16].

Para este trabajo se ha decidido utilizar BERT y RoBERTa ya que son modelos del lenguaje muy potentes desarrollados y respaldados por grandes instituciones como lo son Google y Facebook respectivamente. Además estos modelos a diferencia de otros tienen acceso libre desde diversas APIs como la de HuggingFace y Tensorflow, lo cual facilita su uso.

BERT

BERT (Bidirectional Encoder Representations from Transformers) [21] surgió con la idea de crear un modelo base que aprende a interpretar el

lenguaje en general y una vez pre-entrenado el modelo de lenguaje base, añadir capa/s adicionales para especializarse en una tarea en concreto. Si bien la arquitectura *Transformers* se basa en una fase de codificación y otra de decodificación, en BERT tan solo hay codificación.

El nombre de la arquitectura en concreto que utiliza BERT es *multi-layer bidirectional Transformer encoder* [21]. Como se puede observar en la figura 2.3, se trata de una concatenación de varias capas, siendo cada una de una estas capas un bloque *transformers* como los explicados anteriormente en el apartado 2.2.1.

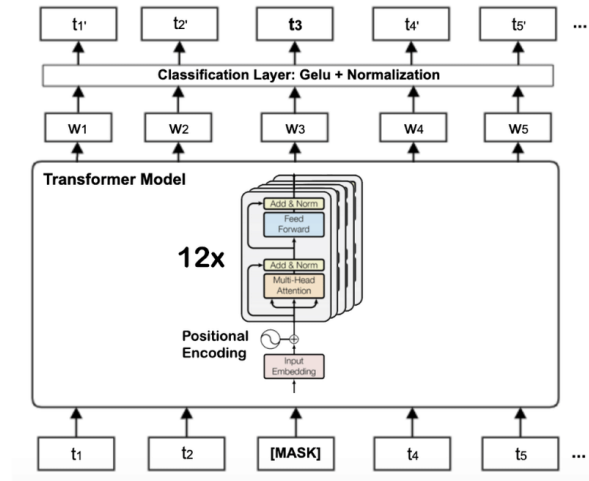


Figura 2.3: Arquitectura de BERT junto con la arquitectura *Transformers*. Fuente: ResearchGate.

El entrenamiento de BERT se basa en 2 fases, un pre-entrenamiento en el que BERT “aprende” que es el lenguaje y su contexto. Para ello BERT es entrenado con 2.500 millones de palabras provenientes de la Wikipedia y 850 millones provenientes de *BooksCorpus* [21]. En concreto BERT se entrena en las tareas de *Masked Language* y *Next Sentence Prediction*, lo cual ayuda BERT a entender el lenguaje. En la figura 2.4 se puede observar de forma gráfica este proceso de entrenamiento.

Posteriormente en la segunda fase, se realiza un *fine-tuning* donde BERT aprende a resolver el problema en concreto que se trata de resolver. Para ello se deben añadir las capas de salida que requiera la tarea en concreto a resolver.

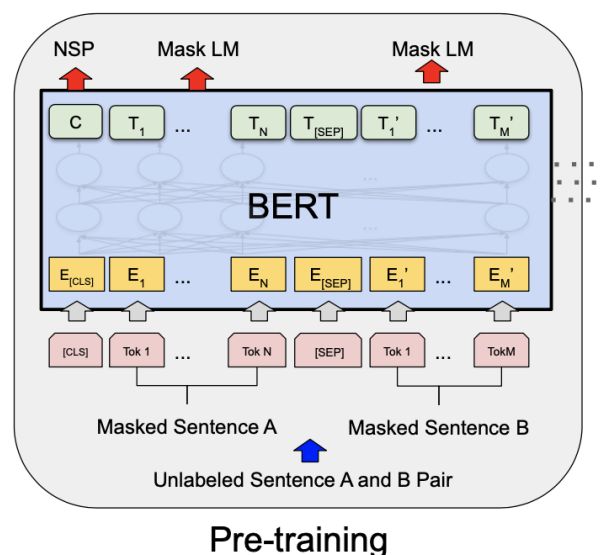


Figura 2.4: Representación de la fase de entrenamiento de BERT. Fuente: Medium.

RoBERTa

RoBERTa: Robustly Optimized BERT pre-training Approach es una variante de BERT introducida por *Facebook AI*¹ en 2019 [39]. Algunas de las principales diferencias con BERT son las siguientes:

- **Enmascaramiento dinámico:** Mientras que BERT utiliza un enmascaramiento estático, es decir, la misma parte del texto está enmascarada en cada época, en RoBERTa se enmascaran distintas partes del texto en cada época, lo cual hace que el modelo sea mucho más robusto.
- **Eliminación de NSP:** A diferencia de BERT, RoBERTa durante el pre-entrenamiento tan solo se entrena para la tarea de MLM (*Masked Language*) y se elimina la tarea de NSP (*Next Sentence Prediction*) ya que se observó que no era muy útil.
- **Entrenado con más datos:** Mientras que la cantidad total de datos de BERT es de 16GB, RoBERTa también es entrenado con otros conjuntos de datos como *CC-NEWS*, *OPENWEBTEXT* y *STOREIS*, lo cual eleva el tamaño de los datos a 160GB.

¹<https://ai.facebook.com/>

- **Mayor tamaño del *batch*:** Mientras que BERT utiliza un tamaño del *batch* de 256 con 1 millón de pasos, RoBERTa utiliza un tamaño de *batch* de 8.000 con 300.000 pasos, lo cual tiene como consecuencia una mejora en la velocidad y rendimiento del modelo.

2.3. Inteligencia Artificial Explicable (XAI)

Si bien los primeros sistemas de Inteligencia Artificial eran fácilmente interpretables (por ejemplo, regresión logística, *k-nearest neighbors* o árboles de decisión), en los últimos años ha habido un gran aumento del uso de sistemas de decisión más opacos tales como las Redes Neuronales Profundas debido a su gran rendimiento en distintas tareas. Estas grandes redes se caracterizan principalmente por una combinación de sus algoritmos de aprendizaje eficiente y su enorme espacio paramétrico. Este espacio se compone de cientos de capas y millones de parámetros, lo que hace que estas Redes Neuronales Profundas sean consideradas como modelos de caja negra [20].

En general, los humanos son reticentes a usar técnicas que no son interpretables de forma directa y fiable y este problema se resalta en mayor medida si se trata de ciertas temáticas más delicadas como la medicina, finanzas o desinformación. Además la interpretabilidad de un modelo puede mejorar su usabilidad asegurando una decisión imparcial, añadiendo robustez frente a ataques adversarios que puedan cambiar la predicción y asegurar que únicamente las variables importantes afectan en el resultado, garantizando que existe una causalidad veraz subyacente en el razonamiento del modelo.

A la hora de definir la terminología se ha de tener en cuenta la diferencia entre interpretabilidad y explicabilidad. La interpretabilidad es una característica pasiva de un modelo referida al nivel que ese modelo tiene sentido para el humano. Por otro lado la explicabilidad se puede considerar como una característica activa de un modelo que denota cualquier acción o procedimiento realizado por un modelo con la intención de aclarar o detallar sus funciones internas. Un modelo se puede explicar, pero su interpretabilidad es algo que proviene del diseño del modelo en sí mismo. Teniendo todo esto en cuenta, en [20] se propone la siguiente definición para XAI:

Dada una audiencia, una Inteligencia Artificial Explicable es aquella que produce detalles o razones para que su funcionamiento sea claro o fácil de entender.

A continuación se enumeran y explican brevemente los objetivos que tiene la Inteligencia Artificial Explicable:

1. **Fiabilidad:** Es la certeza de que un modelo actuará según lo previsto ante un problema determinado.
2. **Causalidad:** Encontrar causalidad entre las variables de los datos. Los modelos de Machine Learning encuentran relaciones de correlación entre los datos, las cuales no implicación una causalidad.
3. **Transferabilidad:** Medida en que puede aplicarse a otros contextos o estudios. La explicabilidad favorece la capacidad de transferencia ya que puede facilitar la tarea de comprensión e implementación de los modelos.
4. **Informatividad:** Los modelos de Machine Learning se usan principalmente para la toma de decisiones, aun así no es el mismo problema el que resuelve el humano que el modelo. Por lo que se necesita de mucha información para relacionar la decisión del usuario con la solución dada por el modelo.
5. **Confidencia:** Se trata de una generalización de la robustez y la estabilidad y se debe de evaluar siempre que al modelo se le espere fiabilidad.
6. **Equidad:** Desde un punto de vista social, la explicabilidad puede considerarse la capacidad de alcanzar y garantizar la equidad en los modelos de *Machine Learning*, permitiendo un análisis ético o de equidad y resaltar el sesgo en los datos a los que se expuso un modelo. Este punto es cada vez más clave al usarse la IA cada vez más en campos donde se implican vidas humanas.
7. **Accesibilidad:** Capacidad de implicación de los usuarios finales en el proceso de mejora y desarrollo de un determinado modelo.
8. **Interactividad:** Capacidad de un modelo para ser interactivo con el usuario final.
9. **Conocimiento de la privacidad:** No ser capaz de entender lo que ha sido capturado por el modelo y almacenado en su representación interna puede suponer una violación de privacidad a la hora de explicar las relaciones internas de un modelo.

Además, dentro de la Inteligencia Artificial Explicable se puede hacer una taxonomía según el tipo de explicabilidad. Por un lado hay dos grandes grupos de XAI, los modelos transparentes y la explicabilidad *a posteriori* o *post-hoc*. Un modelo se considera transparente si por si mismo es comprensible. A continuación se muestran algunos ejemplos de modelos de esta categoría junto a un breve resumen de porqué son explicables:

- **Regresión Lineal/Logística:** Los predictores de este modelo y sus interacciones son legibles para el humano.
- **Árboles de decisión:** Un humano puede simular la predicción de un árbol de decisión fácilmente. Además las reglas que aprende son fácilmente legibles para el humano.
- **K-Nearest Neighbors:** Es un modelo realmente muy sencillo el cual un humano puede simular fácilmente.
- **Rule Based Learners:** Las variables incluidas en las reglas son legibles y el conjunto de reglas es manejable por un humano.
- **Modelos Bayesianos:** Las relaciones estadísticas modeladas entre las variables se pueden comprender directamente por la audiencia objetivo.

Por otro lado, la explicabilidad *a posteriori* o *post-hoc* se centra en aportar una capa de explicabilidad a los modelos más complejos los cuales no son interpretables por si mismos. Dentro de la explicabilidad *post-hoc* se pueden diferenciar 2 grandes grupos:

- Técnicas **agnósticas al modelo**, las cuales pueden aplicarse a cualquier modelo de *Machine Learning* sin tener en cuenta su procesamiento interno o representaciones internas.
- Técnicas de explicabilidad *post-hoc* las cuales están **hechas a medida** o diseñadas para explicar ciertos modelos de *Machine Learning*.

En el contexto de este trabajo, se explican más en detalle los distintos tipos de técnicas agnósticas al modelo (ya que son las utilizadas en este trabajo):

- **Explicación por simplificación:** Son la técnicas más amplias dentro de esta categoría de métodos *post-hoc*. También se incluyen las explicaciones locales en esta categoría ya que los modelos simplificados sólo son representativos de ciertas secciones de un modelo. Casi todas las técnicas que siguen este camino para la simplificación de modelos se basan en técnicas de extracción de reglas. Una de las técnicas más conocidas es LIME (Local Interpretable Model-Agnostic Explanations) [59]. Lime construye modelos localmente lineales alrededor de las predicciones de un modelo de caja negra para explicarlo.
- **Feature relevance explanation:** Estas técnicas pretenden describir el funcionamiento de un modelo opaco, clasificando o midiendo la influencia, relevancia o importancia que tiene cada característica en la

predicción del modelo que se quiere explicar. Una de las contribuciones más fructíferas en este campo es SHAP (SHapley Additive exPlanations), la cual se explica en detalle en la Sección 2.3.1 ya que es la usada en este trabajo. Otros enfoques para abordar la contribución de cada característica ha sido la teoría de juegos coalicional y los gradientes locales [69], técnica en la cual se prueban los cambios necesarios a cada característica para producir un cambio en el resultado del modelo.

- **Técnicas de explicación visual:** Son una forma de lograr explicaciones agnósticas del modelo. En algunos trabajos como en [18] se presentan distintas técnicas de visualización para ayudar en la explicación de un modelo ML de caja negra construido sobre el conjunto de técnicas mencionadas anteriormente. Este tipo de técnicas son menos comunes ya que el diseño de métodos que garantizar que puedan aplicarse sin problemas a cualquier modelo *Machine Learning* sin tener en cuenta su estructura interna, crear visualizaciones complejas a partir de las entradas y salidas de un modelo opaco es una tarea compleja.

2.3.1. SHAP

SHAP (SHapley Additive exPlanations) [42] es un enfoque teórico de juegos para explicar el resultado de cualquier modelo de *Machine Learning*. Es un punto de conexión entre la asignación óptima de créditos con explicaciones locales usando los valores de Shapley clásicos de teoría de juegos. SHAP tiene una base matemática en los valores de Shapley de teoría de juegos, inventados por Lloyd Shapley en 1953 [65] para medir la contribución de cada jugador en un juego o de una manera más formal para proporcionar una solución a la siguiente pregunta:

Si tenemos una coalición C que colabora para producir un valor V .

¿Cuándo contribuye cada miembro en ese valor final?

En pocas palabras, un valor de Shapley es la contribución media marginal de una instancia de una característica entre todas las coaliciones posibles. Con esto se podría saber cuánto ha influido los beneficios de una empresa cada empleado o cuando ha de pagar cada uno en la cuenta de un restaurante.

La respuesta a esta pregunta se vuelve compleja cuando existen efectos de interacción entre los miembros, cuando ciertas permutaciones hacen que los miembros contribuyan más que la suma de sus partes. Por tanto para tener en cuenta los efectos de interacción de un determinado miembro, se calcula el valor de Shapley para la coalición con ese miembro y sin el, siendo la diferencia la contribución marginal de ese miembro para esa coalición. Este proceso se repite para cada permutación posible en la que la única

diferencia es la presencia o no de este miembro. El valor de Shapley es la media de todas estas contribuciones. De forma matemática se define como:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

Donde:

- $\phi_i(v)$ representa el valor de Shapley del jugador i .
- $v(S)$ es el valor o ganancia del juego para una coalición S .
- N es el conjunto de jugadores en el juego cooperativo.
- $|S|$ y $|N|$ representan el tamaño de las coaliciones S y N , respectivamente.

Esta fórmula establece que el valor de Shapley para un jugador se calcula sumando las contribuciones marginales que hace el jugador en todas las posibles coaliciones en las que puede participar. La contribución marginal se determina restando el valor del juego de una coalición con la inclusión del jugador a la coalición y el valor del juego de la misma coalición sin la presencia del jugador [47].

El objetivo de SHAP es explicar la predicción de una instancia x calculando la contribución de cada característica a la predicción. En el contexto de *Machine Learning* un jugador puede ser una característica individual, un grupo de valores de características, los píxeles de una imagen o palabras en un texto. Una de las innovaciones propuestas en [42] con SHAP es que el valor de Shapley se representa como un método de atribución de características aditivas, es decir, como un modelo lineal.

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j$$

donde:

- g es el modelo que va a explicar el modelo de *machine learning* f .
- $z' \in \{0, 1\}^M$ es el vector de coalición o características simplificadas, siendo 1 que la característica está presente y 0 que no lo esta.
- M es el tamaño máximo de la coalición.
- $\phi_j \in R$ es la atribución de características para una característica j .

Otro punto a destacar sobre los valores de Shapley (y por tanto SHAP) es que son la única solución que satisface las siguientes 3 propiedades deseables de un método aditivo de atribución de características:

1. **Precisión local:**

$$f(x) = g(x') = \phi_0 + \sum_{j=1}^M \phi_j x'_j$$

Esta propiedad establece que la suma de las contribuciones de cada característica debe sumar el resultado total menos el resultado promedio del grupo de comparación. En otras palabras, **la predicción debe atribuirse equitativamente a los valores de las características.**

2. **Ausencia:**

$$x'_j = 0 \Rightarrow \phi_j = 0$$

Dado que en el modelo simplificado g las características representan la presencia de características, la ausencia de características en la entrada original no tienen impacto, por tanto, **una característica faltante obtiene una atribución de cero.**

3. **Consistencia:** Deje que $f_x(z') = f(h_x(z'))$ y $z_{\setminus j'}$ indiquen que $z'_j = 0$. Para cualquiera de los dos modelos f y f' que satisfacen:

$$f'_x(z') - f'_x(z'_{\setminus j}) \geq f_x(z') - f_x(z'_{\setminus j})$$

para todas las características $z' \in \{0, 1\}^M$, entonces:

$$\phi_j(f', x) \geq \phi_j(f, x)$$

Es decir, esta propiedad dice que **cuando el modelo cambie de modo que la contribución marginal de un valor se aumente o se mantenga (independiente del resto de características) su valor de Shapley también debe aumentar o permanecer igual.**

Capítulo 3

Experimentación

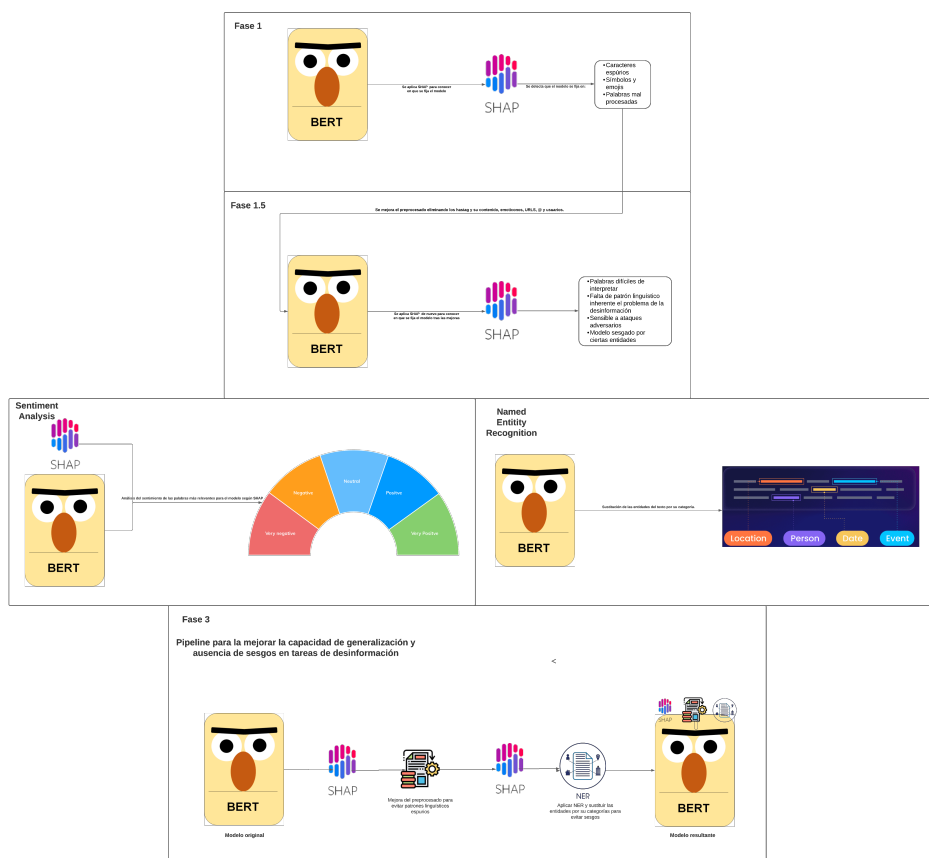


Figura 3.1: Metodología a seguir en este trabajo.

En la figura 3.1 se puede observar la metodología planteada y seguida en este trabajo. En la Fase 1 de la experimentación, se parte de un modelo

de detección de *fake news*, el cual obtiene resultados del estado del arte en la tarea. Se le aplica una técnica de explicabilidad como lo es SHAP para así conocer a que presta atención y detectar posibles errores y sesgos. Adicionalmente, en la fase 1.5 se mejora el preprocesamiento del modelo con la información obtenida con SHAP para arreglar algunos de estos errores.

En la Fase 2 de la experimentación, con la información obtenida previamente, se estudian las distintas técnicas lingüísticas automáticas disponibles y se aplican las más adecuadas para así obtener más información sobre el conocimiento que aprende el modelo y para arreglar los posibles fallos que este tenga. En concreto las técnicas utilizadas fueron *Sentiment Analysis* y *Named Entity Recognition*.

Finalmente, en la Fase 3, se presenta una metodología para solucionar los errores y sesgos detectados, aumentar la capacidad de generalización del modelo y finalmente se evalúa el rendimiento de esta metodología utilizando un *dataset* externo distinto al de entrenamiento.

Por tanto, la Sección 3.1 pondrá en contexto al lector sobre el punto de partida iniciado en el TFG [30] y como se continua en este TFM. En la Sección 3.2 se aplica SHAP al modelo del trabajo, para posteriormente hacer un análisis aplicando *Sentiment Analysis* y *Named Entity Recognition* en la Secciones 3.3 y 3.4 respectivamente. Finalmente en la Sección 3.5 se muestra la metodología final resultante de este trabajo.

3.1. Punto de partida

Como ya se ha mencionado en la Sección 1.1, este trabajo parte de un trabajo previo desarrollado durante el Trabajo Fin de Grado del Grado, titulado “fIlfA: Modelado computacional de la desinformación”¹, en el cual tras una extensa experimentación, probando diferentes modelos, optimizadores y parámetros, se obtuvieron unos resultados del estado del arte en clasificación de *fake news* con el dataset de la competición internacional CONSTRAINT AAI, en concreto con un F1-score de 98,41 %, con un quinto puesto en la competición y una diferencia de 2,89 puntos con el ganador de esta.

El modelo que obtuvo este resultado en el TFG es un Fine-Tuning del modelo base DigitalEpidemiology-V2. Es un modelo pre-entrenado con 97 millones de *tweets*, creado por DigitalEpidemiologyLab en colaboración con FISABio (Valencia) [48]. Además cabe destacar que fue el utilizado en el *Ensemble* que ganó la competición [28] con un 98,61 % de Macro-F1. Por tanto durante el TFG para el Fine-Tuning de este modelo se preprocesaron los *tweets* tal y como indicaban los autores, es decir:

¹Memoria y código del trabajo: <https://github.com/sgonzalezsilot/Thesis-FakeNewsDetection>

- Pasando todas las palabras a minúsculas.
- Sustituyendo los URL por el token \$URL\$.
- Sustituyendo los *hashtag* por el token \$HASHTAG\$.

Aún con los buenos resultados en las distintas métricas de clasificación, se pudo observar e intuir que el sistema tenía ciertas carencias como puede ser una gran falta de interpretabilidad y transparencia, además de unos posibles sesgos y patrones indeseables que fueron observados analizando en el conjunto de datos. A continuación en la tabla 3.1 se muestran algunos de estos comportamientos no deseados.

Anomalía en los datos	Posibles consecuencias
La palabra “Rwanda” solo aparece una vez en todo el <i>dataset</i> y lo hace en una noticia falsa.	Puede que todas las noticias que mencionen este país el sistema las clasifique como falsas.
Hay muchos <i>tweets</i> con el hashtag “#CoronaVirusUpdates” los cuales se tratan del mismo <i>tweet</i> notificando la situación actual de covid en una zona y lo único que cambia entre <i>tweets</i> es la fecha y el número de afectados.	Gran sesgo en la red con cierto tipo de mensajes, además de ser muy vulnerable a ataques adversarios y a que los usuarios aprendan a camuflar noticias falsas como verdaderas.
Gran cantidad de emoticonos, <i>hashtags</i> , nombres de usuario, números y enlaces.	Puede que el modelo no esté encontrando un buen patrón subyacente para diferenciar entre noticias falsas y verdaderas y quizá solo este encontrando patrones sin sentido.

Tabla 3.1: Comportamientos extraños detectados durante el TFG.

Por tanto en este trabajo se ha decidido continuar este trabajo, aplicando técnicas de explicabilidad con este modelo, en concreto la aplicación de SHAP en la Sección 3.2 para poder conocer en qué se fija el modelo a la hora de tomar una decisión, ver si hay algún patrón o sesgo indeseable o si la red se está fijando en las palabras importantes y no en caracteres espurios.

A continuación, en la sección 3.1.1 se hará un análisis del conjunto de datos para introducir al lector con los datos que se van a utilizar durante todo el trabajo para posteriormente en las siguientes secciones ir mostrando el resto de la experimentación.

3.1.1. Análisis Exploratorio de los datos

El conjunto de datos a utilizar, es el presentado en la competición internacional CONSTRAINT AAAI 2021 [51]. Este conjunto de datos consiste en 10.700 *tweets* en inglés de las cuales 5.100 son falsas y 5.600 verdaderas. Las noticias verdaderas están recogidas de Twitter y aportan información útil sobre el COVID-19. Por otro lado las noticias falsas provienen de Twitter, Facebook y Whatsapp además de distintos sitios de fact-checking como Politifact, NewsChecker, Boomlive, etc [52].

En primer lugar se va realizar un análisis básico del conjunto de datos. Este análisis sirve como toma de contacto con los datos, aportando información relevante para su futuro estudio y resolución. Para ello se medirá el tamaño del conjunto de datos, como están distribuidos los datos entre los subconjuntos de entrenamiento y de test, cual es el número de palabras del conjunto de entrenamiento y finalmente la media, mediana y máxima del número de palabras en el conjunto de entrenamiento. En la tabla 3.2 se pueden observar estas medidas las cuales nos sirven para conocer de mejor forma los datos y sus peculiaridades.

Medida	CONSTRAINT AAAI
Tamaño del conjunto de datos	10.700
Nº de palabras del conjunto de entrenamiento	227.724
Tamaño del conjunto de entrenamiento	8.560
Tamaño del conjunto de test	2.140
Media del número de palabras en el conjunto de entrenamiento	26,61
Mediana del número de palabras en el conjunto de entrenamiento	25
Máximo número de palabras en el conjunto de entrenamiento	130

Tabla 3.2: Análisis Básico del conjunto de datos CONSTRAINT AAAI.

Como se puede observar en la figura 3.2 el conjunto de datos de CONSTRAINT AAAI tiene una gran cantidad instancias, es decir, 10.700 noticias únicas. Por otro lado tiene 227.724 palabras o tokens el conjunto de entrenamiento. Como se puede observar en la figura 3.2, el histograma del número de palabras está distribuido homogéneamente entre las 10 y 50 palabras aproximadamente por *tweet*. Además hay un total de 21 tokens de media en cada *tweet* (hay que tener en cuenta que Twitter tiene una limitación en el número de caracteres que puede contener como máximo un *tweet*). Esto es una buena noticia en el ámbito del Procesamiento del Lenguaje Natural usando *Deep Learning* ya que los clasificadores basados en *embeddings* como BERT o RoBERTa tienen un tamaño máximo de entrada de 512 tokens. Por tanto la mayoría de *tweets* no tendrán que ser truncados y no habrá una

gran pérdida de información.

Una vez situado al lector en un conocimiento pleno de las características del conjunto de datos a usar en este TFM, se procede a continuación, en la Sección 3.2 a comenzar la aplicación de SHAP para conocer mejor el modelo y en que partes del texto presta más atención. La aplicación de SHAP para conocer más sobre el modelo viene motivada por alguna peculiaridades que se observaron en el modelo y conjunto datos y que se han explicado en profundidad en la Sección 3.1.

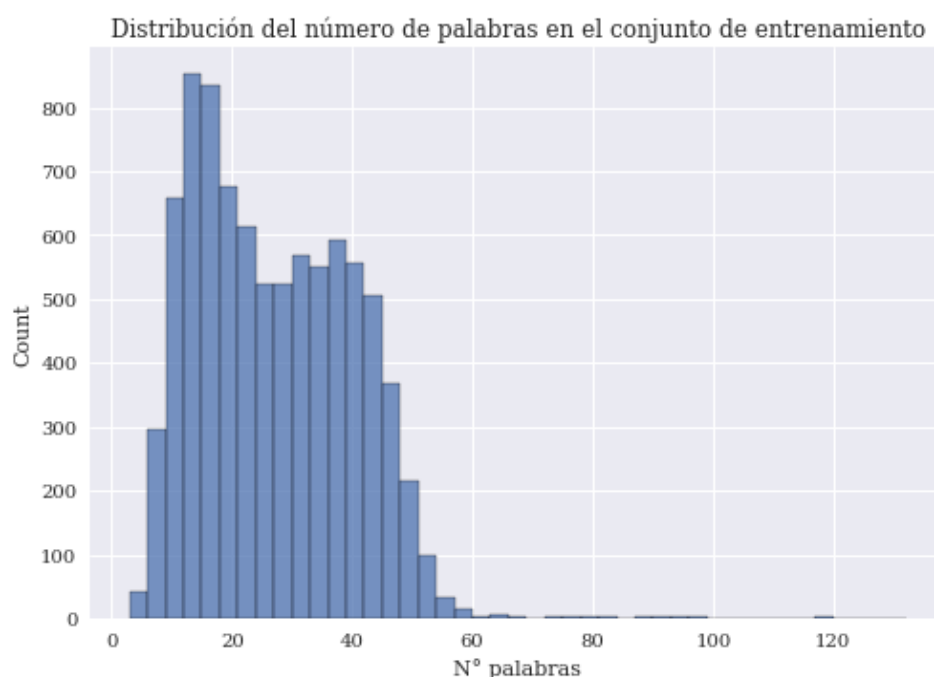


Figura 3.2: Distribución del número de palabras del *dataset*.

3.2. Aplicación de SHAP

En esta sección se aplicará SHAP (SHapley Aditive exPlanations) al modelo resultante del trabajo previo para así conocer más sobre en que se fija y ver su funcionamiento. Para ello se va a utilizar la librería SHAP ² con la cual se puede añadir una capa de explicabilidad a cualquier modelo Machine Learning, tanto para datos tabulares, imágenes o texto.

Para ello primero se han de generar los shap values usando el *Explainer* de la librería SHAP, lo cual debido a la complejidad de cómputo que tiene

²<https://shap.readthedocs.io/en/latest/>

el cálculo de los valores de shapley y el número de tokens de cada texto, llevará una gran cantidad de tiempo de ejecución aún usando GPUs con gran capacidad de cómputo.

3.2.1. Análisis de algunas frases

Una vez generados los valores de shapley procedemos a analizar algunos *tweets* del conjunto de datos para ver si podemos ir conociendo poco a poco en que se fija el modelo. En rojo se pueden observar las palabras que el modelo determina como más relevantes de la clase verdadera y en azul las más relevantes de la clase falsa para ese texto en concreto. La intensidad del color determina la relevancia de la palabra para la predicción dentro de ese texto.

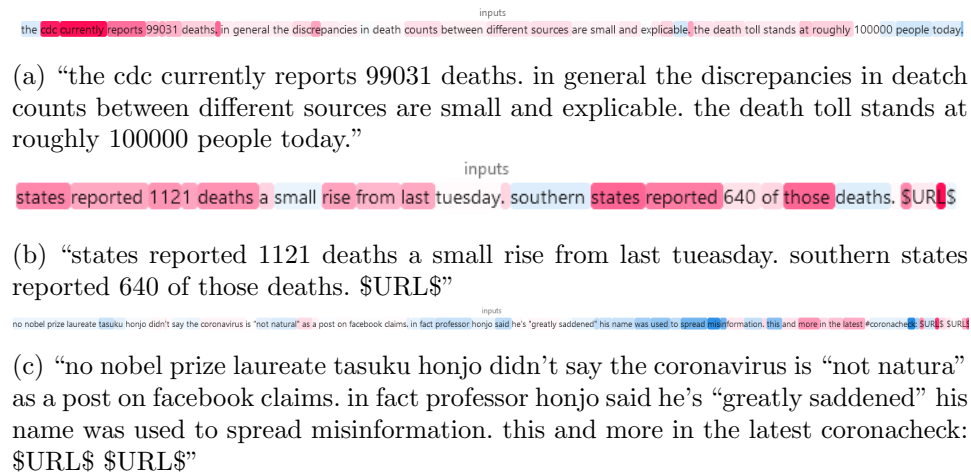


Figura 3.3: Ejemplos de frases analizadas usando SHAP.

Como se puede observar en el primer ejemplo de la figura 3.3, el modelo predice el texto correctamente como verdadero con una alta probabilidad y se fija principalmente en las palabras “cdc”, “currently” y “reports”. “CDC” son las siglas de los centros para el control y prevención de enfermedades, es decir, la agencia nacional pública de Estados Unidos. Por tanto por un lado parece que el modelo identifica correctamente que esta palabra indica que el texto es verdadero pero a la vez muestra que el modelo es sensible a ataques adversarios, puesto que si una persona averiguase esto y añadiese estas siglas al *tweet* ayudaría a que se identificase como cierto.

Por otro lado las palabras “currently” y “reports” *a priori* no parecen tener un significado claro interpretable por el ser humano que nos haga pensar que sea un texto verdadero o falso.

Seguidamente, en el segundo ejemplo de la figura 3.3 podemos encontrar

otra frase que el modelo ha predicho correctamente como verdadera. En este caso lo que más destaca es que el modelo por algún motivo, lo que más le hace creer que la noticia es verdadera es la letra “L” que aparece en “URL”. También le da relativamente bastante relevancia al token \$.

En ambos casos, como es lógico, no tiene ningún sentido que el modelo use con tanta importancia este tipo de tokens para discernir la veracidad de una noticia. Con esto se está empezando a ver una de las hipótesis que se presentaban al inicio de este trabajo. El modelo no está aprendiendo realmente a discernir entre noticias verdaderas y falsas, sino que ha aprendido un patrón textual (inapreciable e ilógico por el ser humano) presente en el conjunto de datos para lograr diferenciar las 2 clases.

Finalizando con los ejemplos individuales, en el tercer ejemplo de la figura 3.3 se puede ver que de nuevo el modelo se fija bastante en lo mencionado anteriormente respecto a las URL pero además se fija en la palabra “misinformation” lo cual, si bien por una lado *a priori* parece lógico, cuando nos fijamos detenidamente se puede observar que lo hace solo en ciertas partes de la palabra, en concreto en “mis” y en “in” por separado. Esto se debe a que esta palabra el modelo la está tokenizando incorrectamente y esto tiene implicaciones negativas a la hora de buscar patrones en el texto. Este fenómeno también ocurre con el *hashtag* “coronacheck”.

3.2.2. Resúmenes obtenidos de los Shapley Values

Tras analizar algunos ejemplos interesantes, se pasa a analizar el conjunto de datos al completo usando las herramientas de resúmenes que nos proporciona SHAP. En primer lugar se pueden ver las palabras (tokens) que de media en todo el *dataset* han tenido más influencia a la hora de tomar una decisión.

En la figura 3.4 se pueden observar las 20 palabras que más impacto tienen de media. Como se puede apreciar, en general no parece que ninguna tenga mucho sentido, ni serían las que un experto identificaría como palabras clave a la hora de identificar una noticia falsa. A continuación se analizan algunas de ellas:

- **dashboard**: Esta palabra es la que más impacto tiene de media y *a priori* no parece tener mucho sentido, hace referencia a los cuadros de mando o gráficas y sería complicado relacionarlo con noticias falsas o verdaderas.
- **allegheny**: Este token representa a una entidad geográfica, en concreto al condado de Allegheny, Pensilvania. Como se puede observar este token causa un gran impacto en la predicción del modelo ya que puede incurrir en sesgos sociales o demográficos.

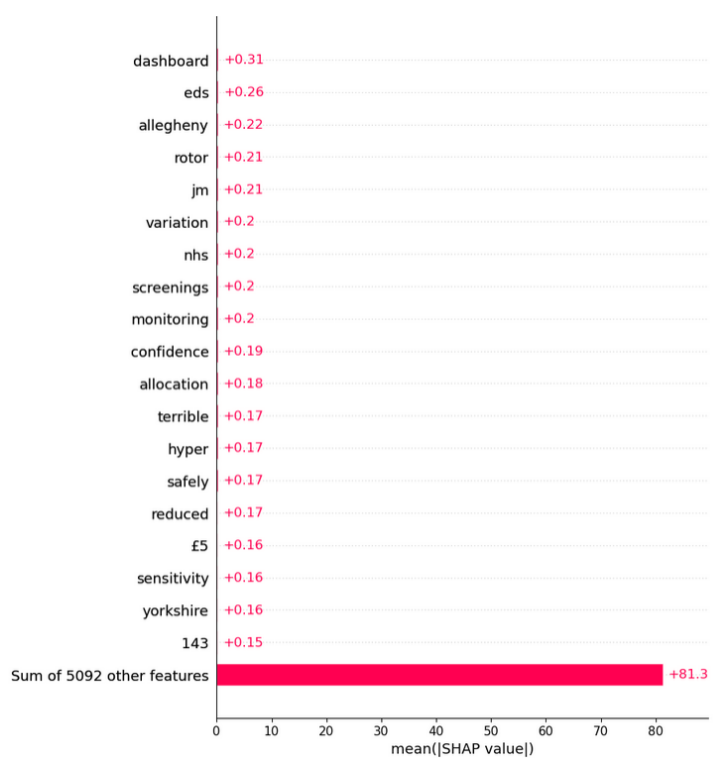


Figura 3.4: Palabras con mayor impacto de media.

- **rotor**. Este token proviene de una mala tokenización de Rotorua, una ciudad de Nueva Zelanda. De nuevo una entidad geográfica.
- **screenings**: Esta palabra significa cribado, en este caso este token si tiene bastante sentido que se relacione con *fake news* ya que este dataset es de *fake news* de covid.
- **monitoring**: Esta palabra significa el control médico de un paciente, al igual que con el término anterior, tiene sentido en este contexto.

En general, el modelo de media en lo que más se fija es en palabras mal procesadas y encuentra patrones extraños en ellas, palabras que representan una entidad, lo cual es peligroso y palabras con un sentido dentro del contexto, las cuales un experto también identificaría.

También cabe destacar un término que aparece con bastante impacto, el cual es “jm”. Buscando el *dataset*, no hay ninguna palabra la cual contenga estas letras, pero analizando detenidamente los *tweets*, en algunos de ellos contienen enlaces y estos enlaces se componen de letras aleatorias. En bastantes de estos enlaces, da la casualidad de que se encuentran estas 2 letras juntas y el modelo las procesa como un token cuando se encuentra el enlace, por ejemplo, en el enlace: <https://t.co/k7u0CjpAjM> .

Con esto vemos que el modelo no está encontrando patrones lógicos para un ser humanos, sino que encuentra patrones dentro de enlaces, que contienen conjuntos de letras sin sentido, las cuales se repiten en el conjunto de datos y el modelo usa estos patrones para separar entres las 2 clases. El modelo no sigue ningún tipo de lógica aparente, simplemente busca separar los datos para seguir mejorando la función de perdida durante el entrenamiento.

Parte del problema de este fenómeno reside en que esto no es nada generalizable, si se evaluase el modelo con otros *datasets* o se pusiese en producción y se evaluase el rendimiento del modelo con textos que no tienen estos patrones, se vería como probablemente el rendimiento decae.

Para finalizar este análisis obtengamos un resumen de la sumatoria de los valores de shapley de cada término, así tendremos en cuenta tanto el número de veces que se repite como el peso que tiene. En la figura 3.5 se puede observar que algunos de los términos que más impacto tienen son símbolos como \$, # o “: ”. Además también aparecen palabras muy comunes de la lengua inglesa como “the” y “of” junto a las distintas partes de la palabra “covid-19”, la cual el modelo la procesa como varios tokens.

Tras todo este análisis, se ha podido observar que por un lado el modelo si se fija en ciertos términos o palabras las cuales parecen lógicas y son parecidas a las que se fijaría un humano para discernir entre una noticia falsa y una verdadera. En cambio por otro lado le da mucho valor a términos

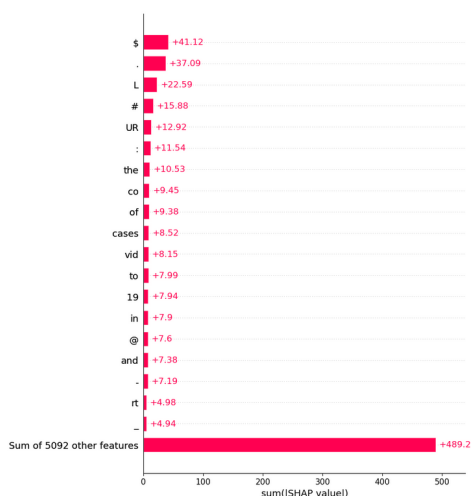


Figura 3.5: Palabras con mayor impacto total

sin sentido, a palabras mal procesadas, a símbolos espurios y patrones sin sentido.

Aun así, antes de aseverar la calidad del modelo, se ha considerado que para hacer un juicio justo del modelo usando SHAP, primero se tendrían que intentar solucionar durante el procesamiento y entrenamiento los problemas que hemos visto que tiene el modelo gracias a SHAP para posteriormente re-analizar el comportamiento del modelo, lo cual se hará en la siguiente sección.

3.2.3. Cambios en el preprocesamiento y re-aplicación de SHAP.

Tal y como se ha mencionado anteriormente, antes de emitir un juicio sobre el uso de modelos de lenguaje para tareas de clasificación de desinformación, primero se va a usar todo el conocimiento adquirido durante el análisis inicial, para intentar paliar con un mejor preprocesamiento de los datos. Si bien los autores del modelo original, recomiendan el preprocesamiento que fue usado durante el TFG y durante el análisis inicial con SHAP, se modificará el preprocesamiento para así intentar lograr un modelo que intente ser más lógico e interpretable. Para ello se aplicará el siguiente preprocesamiento a los textos:

- Eliminar los *hashtag* y su contenido.
- Eliminar emoticonos.
- Eliminar URLs.

- Eliminar @ y sus respectivos usuarios.

Una vez aplicado el nuevo preprocesamiento, se vuelve a entrenar el modelo y se aplica de nuevo el mismo proceso con SHAP para ver como ha afectado a lo que aprende el modelo y su interpretabilidad. Como se puede observar los resultados han empeorado un poco con respecto al modelo original el cual tiene un Macro-F1 de 98,41 % y ahora tiene un 97,80 %. Esto es lógico ya que esto se debe a que algunos de los patrones que usaba el modelo para discernir ya no los tiene y si bien se dificulta la tarea, el objetivo es que busque patrones más lógicos. Para ver el funcionamiento del nuevo modelo, se van a analizar nuevamente una serie de ejemplos.

inputs

A complex Sri Lankan herbal drink was said to remedy all virus infections

(a) “A complex Sri Lankan herbal drink was said to remedy all virus infections”

inputs

Bolivia approved the use of chlorine dioxide amid the fight against covid-19.

(b) “Bolivia approved the use of chlorine dioxide amid the fight against covid-19.”

Figura 3.6: Ejemplos de frases analizadas usando SHAP tras mejorar el preprocesamiento.

Como se puede observar en los ejemplos de la figura 3.6, al aplicar el nuevo preprocesamiento, parte de este sesgo y patrones indeseados se han eliminado ya que ahora únicamente se fija en los tokens relevantes del texto. Aun así como se puede observar en el caso de “Sri Lanka” o “Bolivia”, podemos observar que se le está dando mucho peso a estos términos que representan entidades geográficas y que quizá no deberían de tener tanta relevancia.

Para continuar con el análisis de la explicabilidad del modelo, podemos de nuevo consultar los resúmenes de información que nos proporciona SHAP. En la figura 3.7 se puede observar los términos que mayor impacto tienen de media. Si bien ahora las palabras que aparecen si pertenecen al diccionario inglés y esta lista no se compone de términos espurios, siguen sin ser palabras que a simple vista determinen que una noticia es falsa o verdadera. Además se sigue confirmando que hay grandes sesgos con ciertas entidades como pueden ser “Rwanda”, “Tanzania”, “Netflix” o “Nicaragua” entre otros.



Figura 3.7: Palabras con mayor impacto de media tras mejorar el preprocesamiento.

3.2.4. Conclusiones parciales

Llegado a este punto del trabajo, tras la aplicación de técnicas de explicabilidad como SHAP a un clasificador de *fake news* basado en LLMs el cual resultó en una gran puntuación y ranking en una competición internacional de relevancia, como es el CONSTRAINT AAAI, se ha podido analizar la explicabilidad del modelo, en que se fija y presencia de sesgos.

Se ha visto que el modelo en ocasiones le daba más importancia de la debida a caracteres espurios, emojis, URLs o incluso a patrones dentro de las propias URLs, por tanto, para paliar este problema se ha propuesto un mejor preprocesado, el cual si bien empeora las métricas de clasificación del modelo, mejora su explicabilidad y hace que el modelo sea más seguro al prestar atención a las partes del texto que son más importantes.

También se ha visto que entre las palabras que para el modelo son más importantes, hay varias las cuales son difíciles de interpretar por un ser humano como relacionadas a *fake news* o noticias verdaderas. Por un lado esto es normal, ya que quizá parte del problema de los sistemas de clasificación de *fake news*, reside en que no existe o es extremadamente complejo para un humano, visualizar un patrón lingüístico que identifique noticias falsas. En ocasiones este patrón que encuentra el modelo ni siquiera tiene por qué ser inherente a las noticias falsas frente a las verdaderas, sino en algún sesgo proveniente de los datos, la forma de obtenerlos y de etiquetarlos. Si se analizan varias instancias del conjunto de datos, se puede observar que hay sesgos claros en el conjunto de datos, entre los que destacan los siguientes:

- Si bien las noticias falsas parecen no tener ningún patrón presente, en las verdaderas se puede observar que sí lo hay, generalmente están obtenidas de organismo públicos de salud y por tanto la mayoría de textos son en forma de declaración informativa o informes diarios sobre la actualidad sanitaria del covid en las distintas zonas geográficas.
- Por otro lado hay una cantidad relevante de textos los cuales son prácticamente idénticos, siendo una plantilla de *tweet* la cual se utiliza para ir cambiando la fecha y número de incidentes relacionados con el covid-19 en una zona geográfica o varias.

Siguiendo con el análisis de interpretabilidad de los tweets, puede llegar a ser complejo encontrar un patrón lingüístico que utilice el modelo o palabras relevantes que se fije el modelo y que el humano corrobore como relevantes, ya que, en comparación con otras tareas de procesamiento del lenguaje, la identificación de estos patrones lingüísticos se vuelve muy compleja para el ser humano y poco intuitiva. Pongamos como ejemplo el análisis de sentimientos que se presenta en SHAP como ejemplo de aplicación de valores de shapley a LLMs. En la figura 3.8 se pueden observar los valores de shapley de 3 textos. En los 2 primeros textos la clase predicha es “sadness” (tristeza) y las palabras que el modelo da más importancia para que la predicción sea tristeza son “humiliated”, “feeling” y “hopeless”, lo cual *a priori* para el lector parece lógico y parece que el modelo presta atención donde debería. De igual forma ocurre con la clase “anger” (ira) del tercer texto y las palabras identificadas como más relevantes “greedy” y “wrong”.

Si se observan en la figura 3.9 los resúmenes de este mismo modelo de análisis de sentimientos, se pueden ver las palabras más relevantes para identificar un texto como “joy” (alegre) o como lo contrario. Las palabras que más destacan tanto para esta clase como para la opuesta son fáciles de interpretar para el lector, muestran claramente la idea mental que tiene el lector de este concepto o de lo opuesto a este concepto.

Esto no ocurre con la clasificación de *fake news* ya que quizá, no existan palabras o términos clave que identifiquen a una noticia como falsa o ni siquiera patrones lingüísticos claro que lo hagan. Por tanto si no existiesen dichos patrones, el modelo no los encontraría. Ya que, como se ha mencionado anteriormente, las palabras que nos proporcionan los resúmenes de SHAP como más relevantes no aportan una idea clara de a que clase pertenecen, se va analizar el sentimiento que aportan esas palabras usando diversas técnicas de Sentiment Analysis en la Sección 3.3. Así, ya que no se observa una idea clara de la interpretación de estas palabras, usando técnicas de análisis de sentimientos en las palabras relevantes, quizá se observe un gran sentimiento negativo para las palabras que representen las noticias falsas o en general una gran polaridad de sentimientos entre las palabras

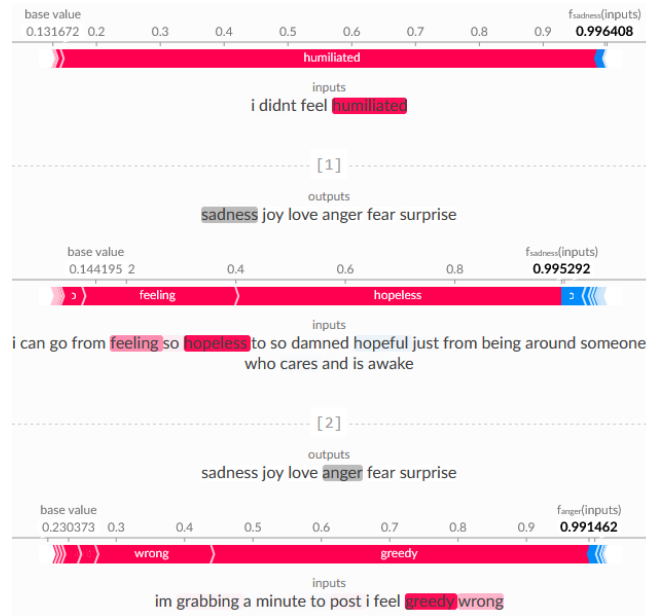


Figura 3.8: Ejemplos de análisis con SHAP de Sentiment Analysis

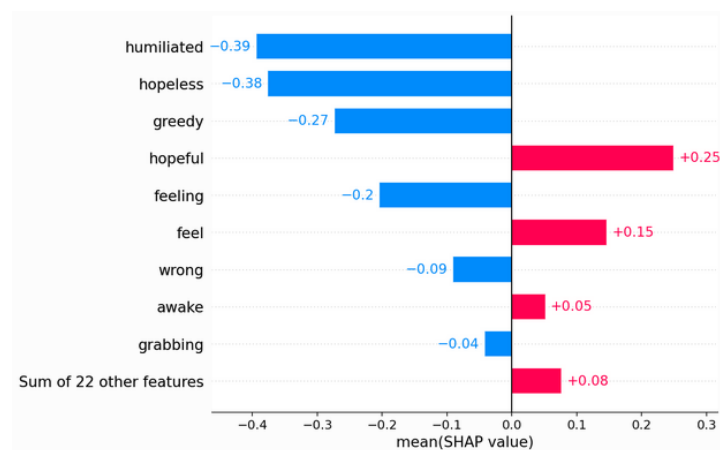


Figura 3.9: Ejemplo de palabras más relevantes con SHAP de Sentiment Analysis

más importantes.

Para finalizar estas conclusiones parciales, también se destaca que tras aplicar SHAP para observar las palabras más relevantes del conjunto de datos, se ha observado que muchas de estas representan a entidades, ya bien sean países, ciudades, personas o empresas. Esto supone un gran problema a la hora de crear un clasificador de noticias falsas ya que hace que este tenga un gran sesgo.

Más en concreto se pueden observar ejemplos como “Rwanda” en los cuales tiene un gran valor de shapley hacia la clase negativa lo cual haría que con una gran probabilidad cualquier noticia que trate sobre este país sería identificada como falsa. También ocurre con otros términos como “CDC” (agencia nacional de salud pública) los cuales tienen un gran valor de shapley para la clase positiva, lo cual se podría usar en ataques adversarios para que se detecten noticias falsas como verdaderas [77]. Es por ese motivo por el cual en la en la Sección 3.4 se va a aplicar un proceso de Named Entity Recognition (NER) para detectar dichas entidades y tratarlas correctamente para desarrollar un modelo con menos sesgo y más generalizable.

3.3. Sentiment Analysis (SA) de las palabras más relevantes.

Debido a que las palabras más influyentes que podemos observar en los resúmenes aportados por SHAP no tienen una fácil interpretación, se ha decidido aplicar técnicas de Sentiment Analysis (SA) para intentar observar si existe algún tipo de relación entre la diferencia de noticias falsas y verdaderas y una polaridad negativa y positiva.

Para obtener el sentimiento que aportan estas palabras, se va a utilizar SentiWordnet [7, 64] y NRC Lexicon [46, 45]. SentiWordNet asigna a cada synset de WordNet una puntuación de sentimiento en 3 dimensiones (positivo, negativo y objetivo) basado en un enfoque de minería de opiniones y generado en el análisis de corpus de texto etiquetado manualmente. NRC Word-Emotion Association Lexicon es un recurso léxico desarrollado por la Universidad de Toronto para el análisis de sentimientos. NRC Lexicon proporciona información sobre la asociación de palabras con ocho emociones básicas y un sentimiento positivo y negativo.

Tras analizar las palabras más relevantes manualmente se ha observado que la mayoría de estas palabras son prácticamente neutras en un totalidad, no aportan apenas sentimiento positivo o negativo según ninguna de las herramientas de Sentiment Analysis usadas, por lo que por un lado se descarta *a priori* una correlación entre sentimiento positivo o negativo con desinformación a nivel de palabra y por otro se refuerza la hipótesis de que

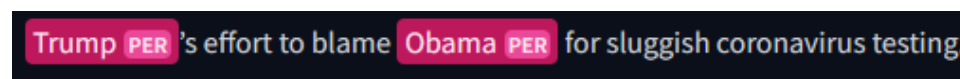
no hay un patrón lingüístico claro y sencillo para el humano que se pueda observar con técnicas de explicabilidad como SHAP.

Por tanto, para seguir este análisis desde el punto de vista del análisis de sentimientos se tendría que optar por usar el sentimiento de toda la frase y añadir este sentimiento como una característica adicional o basar la detección de *fake news* en el análisis de sentimientos y emociones ya que en ocasiones los creadores de *fake news* utilizan diversos trucos estilísticos para excitar los sentimientos de los receptores [4, 10].

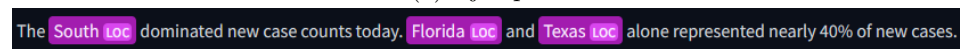
3.4. Aplicación de Named Entity Recognition (NER) y sustitución de entidades por un token comodín.

Como se ha visto a lo largo de este trabajo, el modelo de detección de *fake news* tiene un grave problema con ciertas entidades (países, ciudades, personas y empresas), debido a que estas aparecen en pocas ocasiones en el conjunto de datos y en ocasiones de una forma sesgada (apareciendo mayoritariamente en una clase). Esto es un gran problema a la hora de afrontar el problema de la desinformación desde un punto de vista de la justicia y la ética.

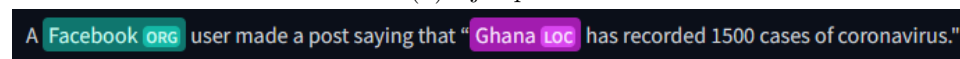
Es por este motivo por el que se ha decidido aplicar NER al conjunto de datos y sustituir cada entidad por su categoría correspondiente, para así re-evaluar el modelo una vez no tenga la información sobre las entidades en concreto y solo sobre su categoría (persona, país, etc). Para ello se va a usar un modelo³ de BERT entrenado para reconocimiento de entidades (NER) con el dataset “CoNLL-2003 Named Entity Recognition” [72].



(a) Ejemplo 1



(b) Ejemplo 2



(c) Ejemplo 3

Figura 3.10: Ejemplos de textos procesados usando NER.

En la figura 3.10 se pueden observar distintos tweets de ejemplo del con-

³<https://huggingface.co/dslim/bert-large-NER>

junto de datos y como el modelo de reconocimiento de entidades, reconoce correctamente las distintas entidades presentes en el texto, ya bien sean personas, localizaciones u organizaciones. Las entidades serán sustituidas por los tokens de la tabla 3.3.

Token	Significado	Ejemplo
PER	Persona	Donald Trump
ORG	Organización	CDC
LOC	Localización	California
MISC	Entidades diversas (eventos, nacionalidades, productos, etc)	Spanish

Tabla 3.3: Entidades a detectar en el proceso de NER.

Una vez aplicado este proceso, se procede a re-entrenar el modelo con los parámetros originales. Una vez re-entrenado, las métricas en el conjunto de test son levemente peores, en concreto se pasa de un 97,80 % a un 96,34 % de Macro-F1. Son entorno a 2 puntos peores de Macro-F1 peores que las obtenidas con el modelo original del TFG. Aun así, merece la pena esta pérdida de métrica de precisión, a cambio de obtener un modelo más transparente, seguro y ausente de sesgos peligrosos para la sociedad.

3.5. Desarrollo de la metodología final y experimentación con *dataset* externo.

Durante el desarrollo de este trabajo, se ha acabado desarrollando una metodología que se puede observar en la figura 3.11. En esta metodología no se incluye los resultados obtenidos en el proceso de análisis de sentimientos de las palabras más relevantes ya que estos no fueron prometedores.

Toda esta metodología tiene como objetivo, pasar de un modelo con una gran precisión en un conjunto de datos relevante a un modelo con una mayor interpretabilidad, más transparente, con menos sesgos y que se fije en patrones lingüísticos coherentes. Esta mejora a nivel de explicabilidad es rentable aun teniendo en cuenta que todo este proceso ha resultado en un leve de las métricas de clasificación del modelo original, pasando de un 98,41 % de Macro-F1 en el conjunto de test a un 96,71 %.

Aun así, se ha podido ver que todas estas técnicas aplicadas resultado en una metodología para la mejora de la explicabilidad y eliminación de sesgos en el modelo ha resultado en un modelo con una mayor capacidad de generalización, por lo que para comprobar que esto es así, se compararán los resultados del modelo original del TFG y del modelo resultante con un

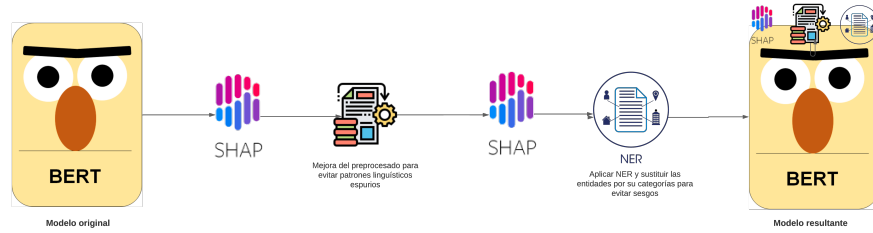


Figura 3.11: Metodología desarrollada en este trabajo

dataset externo de *fake news* sobre covid-19 para ver si esto se cumple.

El conjunto de datos escogido ha sido CTF (**C**OVID-19 **T**witter **F**ake News), conjunto de datos propuesto en [50]. Se ha escogido este conjunto de datos ya que es muy similar al usado para el entrenamiento del modelo, son datos de *fake news* (y no otro tipo de desinformación), de Twitter (por tanto usa el mismo lenguaje) y sobre Covid-19 (la misma temática). En concreto se usará únicamente el conjunto de test el cual consta de 774 tweets, de los cuales verdaderos son 387 y falsos 387, por lo cual está perfectamente balanceados.

En la tabla 3.4 se pueden observar los resultados de ambos modelos tanto en el *dataset* interno (CONSTRAINT AAAI) como en el *dataset* externo (CTF). Si bien tras aplicar todas las técnicas mencionadas anteriormente durante este trabajo, el rendimiento del modelo frente al *dataset* interno ha empeorado alrededor de 1,7 puntos en Macro-F1 a cambio de una mayor interpretabilidad y menor sesgo, en el caso del *dataset* externo (para el cual no ha sido entrenado) obtiene 11,03 puntos más de Macro-F1 frente al modelo original.

Por tanto, se puede afirmar que esta metodología es un buen mecanismo para no solo nutrir a los modelos de una mayor capacidad de interpretabilidad, transparencia, coherencia y ausencia de sesgos, sino que también con el procedimiento presentado en este trabajo se logra una de las tareas más difícil en el área de *Machine Learning*, la capacidad de generalización.

Modelo	Dataset interno (Constraint AAAI)	Dataset externo (CTF)
Modelo original	98,41	44,2
Modelo resultante	96,71	55,24

Tabla 3.4: Comparativa modelo original-resultante

Capítulo 4

Conclusiones y trabajo futuro

Para finalizar el trabajo se han de analizar los resultados obtenidos y compararlos en función de los objetivos, con esto se podrá saber por un lado si se han cumplido y por otro poder analizar cual es el trabajo futuro que se ha de realizar para seguir avanzando en la tarea de detección de *fake news* y la explicabilidad aplicada al área de la desinformación.

En este capítulo final del trabajo se comentarán las conclusiones del trabajo en la Sección 4.1 además de posibles trabajos futuros en la Sección 4.2.

4.1. Conclusiones

Si bien la tarea de detección de *fake news* y la aplicación de explicabilidad en el campo de la desinformación constituyen una tarea amplia y en constante avance e investigación, en este trabajo se han analizado los distintos puntos clave de este problema para el avance de este área de investigación.

En concreto, como **punto clave de este trabajo, se ha desarrollado una metodología de preprocesamiento la cual dota al modelo de una mayor explicabilidad, ausencia de sesgos y capacidad de generalización. Se ha evaluado el modelo tras aplicar esta metodología con un *dataset* externo (distinto al de entrenamiento) y se ha logrado una mejora del 24,95 %.** Para lograr este punto final se han analizado y tratado los siguientes puntos a destacar:

- En primer se ha partido de un modelo de clasificación de *fake news* con una gran precisión pero una gran ausencia de

interpretabilidad. Al finalizar el TFG se observaron ciertos patrones indeseables en el conjunto de datos que debido al funcionamiento del modelo podía repercutir en problemas de sesgos y capacidad de generalización.

- En segundo lugar se ha aplicado SHAP como técnica de explicabilidad *post-hoc* para analizar el comportamiento del modelo y observar en que partes del texto presta más atención para discernir entre noticias falsas y verdaderas.

Gracias al análisis realizado con SHAP se han podido identificar y analizar ciertos errores y comportamientos indeseables que presenta el modelo. **Los principales errores detectados con SHAP son el sesgo en ciertas entidades relevantes como países o personas y que el modelo presta demasiada atención a características espurias del texto como pueden ser URLs o emoticonos.**

- En tercer lugar se ha mejorado el preprocesamiento del modelo para que deje de prestar atención a características espurias y tan solo lo haga a las palabras del texto. Con esto se ha logrado una mejora de la explicabilidad del modelo a cambio de un leve empeoramiento de las métricas de clasificación.

Tras este preprocesamiento se han analizado las palabras más relevantes de todo el *dataset* para el modelo a la hora de tomar una decisión. Estas palabras principalmente eran de nuevo entidades relevantes y palabras difíciles de interpretar y relacionar con el ámbito de las *fake news* por un ser humano.

Por tanto para intentar analizar más en profundidad estas palabras se ha aplicado un proceso de *Sentiment Analysis* para conocer el sentimiento que transmiten estas palabras usando herramientas como *SentiWordnet* y *NRC Emotion Lexicon* que asignan grado de sentimiento y emoción a cada palabra. **Este análisis ha resultado en la conclusión de que estas palabras más relevantes no aportan apenas ningún tipo de sentimiento positivo ni negativo así como ninguna emoción.** Por tanto se cree que parte del problema de interpretabilidad que tienen estas palabras reside en la propia complejidad de las *fake news* y su identificación.

- En cuarto lugar, para solucionar el problema relacionado con la atribución de una gran importancia a entidades relevantes como localizaciones o personajes públicos se ha decidido aplicar un proceso de reconocimiento de entidades (NER). En concreto se han sustituido estas entidades por un token el cual representa su categoría. Esto ha resultado en una gran mejora

respecto a los sesgos que puede contar el modelo a cambio de nuevo de una leve disminución de las métricas de clasificación.

Finalmentese ha condensado el proceso de mejora del preprocesamiento y de NER en una metodología con la cual se adapta el preprocesamiento del texto a la tarea de *fake news*. Con esto se logra una mayor explicabilidad del modelo, ausencia de sesgos y capacidad de generalización.

Para comprobar esta afirmación se ha evaluado el modelo tras aplicar este proceso con un *dataset* externo (distinto al de entrenamiento pero similar), comparando así el rendimiento del modelo con datos distintos a los usados en el entrenamiento pero en la misma tarea. Se logra un 24,95% de mejora tras aplicar estas buenas prácticas en el *dataset* externo, lo cual valida la capacidad de esta metodología de dotar al modelo de una mayor capacidad de generalización en clasificación de *fake news*.

Esta propuesta metodológica podría aplicarse a otros problemas basados en clasificación, en particular en dominios complejos que involucran complejas interacciones sociales.

4.2. Trabajo futuro

En esta sección se abordan los posibles trabajos futuros relacionados con la investigación de modelos de *Deep Learning* para la tarea de detección de *fake news* y aplicación de explicabilidad en estos. A continuación se describen estos posibles trabajos futuros.

- **Replicación de los experimentos con más *datasets*.** Para poder afianzar la validez de los resultados obtenidos por la metodología final presentada en este trabajo, se debería replicar el experimento que evalúa el rendimiento del modelo antes y después de aplicar la metodología con el *dataset* de entrenamiento (interno) y con el *dataset* externo. Las pruebas podrían ser realizadas con pares de *datasets* lo más similares posibles en el tipo de desinformación (*fake news* o rumores), idioma, temática y origen de los datos (noticias periodísticas o redes sociales). También se podría llegar a probar con distintos tipos de desinformación, ya bien sean *fake news* o rumores, ya que esta metodología mejora ciertas carencias que tienen los modelos a nivel de sesgos y carencia de interpretabilidad presente en cualquier tipo de desinformación.
- **Creación de un modelo de lenguaje propio.** Si bien durante el

TFG se utilizaron distintos modelos de BERT entrenados por distintas empresas, universidades y organizaciones, otro paso para intentar lograr mejores resultados a nivel de explicabilidad y capacidad de generalización sería la creación de un modelo propio de BERT. Con esto se lograría paliar algunos de los problemas que no se han solucionado en este trabajo como la incorrecta tokenización de algunos términos claves del dominio de las *fake news* y la Covid-19 como por ejemplo la propia palabra “Covid-19”. Para esto se debería de utilizar una gran cantidad de datos y hacer un entrenamiento no supervisado del modelo base de BERT/RoBERTa con estos datos durante un gran periodo de tiempo, lo cual es el principal motivo por el que no se ha decidido incluir este apartado en este trabajo. Durante este proceso, también se deberán de estudiar y aplicar técnicas para evitar sesgos peligrosos heredados de los datos de entrenamiento, ya que como se ha observado en algunos casos existe cierto sesgo en los modelos de lenguaje por sexo [11] o por etnia [27].

- **Uso de distintas técnicas de explicabilidad.** El análisis hecho de los modelos de clasificación de *fake news*, se podría haber efectuado usando otros métodos de explicabilidad como Integrated Gradients o LIME. Para un trabajo futuro se destaca el uso de Integrated Gradients (IG) ya que este método es considerablemente distinto a SHAP. Si bien SHAP se basa en las contribuciones de cada atributo, IG explica la predicción del modelo usando el gradiente del resultado respecto a las características de entrada.
- **Creación de un modelo *cross-lingue*.** Ya que en la sociedad actual cada vez es más común el conocimiento de más de un idioma, y más en concreto del inglés y el español en conjunto, se propone como posible trabajo futuro la implementación de un modelo *cross-lingue* el cual sea capaz de detectar noticias tanto en inglés como en español. Una posible aproximación sería utilizando un modelo *XLNet*.

Bibliografía

- [1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [2] Sarah A Alkhodair, Steven HH Ding, Benjamin CM Fung, and Junqiang Liu. Detecting breaking news rumors of emerging topics in social media. *Information Processing & Management*, 57(2):102018, 2020.
- [3] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.
- [4] Miguel A Alonso, David Vilares, Carlos Gómez-Rodríguez, and Jesús Vilares. Sentiment analysis for fake news detection. *Electronics*, 10(11):1348, 2021.
- [5] Mariette Awad and Rahul Khanna. *Machine Learning*, pages 1–18. Apress, Berkeley, CA, 2015.
- [6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [7] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani, et al. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec*, volume 10, pages 2200–2204, 2010.
- [8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [9] Iz Beltagy, Arman Cohan, and Kyle Lo. Scibert: Pretrained contextualized embeddings for scientific text. *CoRR*, abs/1903.10676, 2019.
- [10] Bhavika Bhutani, Neha Rastogi, Priyanshu Sehgal, and Archana Purwar. Fake news detection using sentiment analysis. In *2019 twelfth*

- international conference on contemporary computing (IC3)*, pages 1–5. IEEE, 2019.
- [11] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.
 - [12] Alessandro Bondielli and Francesco Marcelloni. A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38–55, 2019.
 - [13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.
 - [14] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*, 2020.
 - [15] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
 - [16] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555, 2020.
 - [17] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
 - [18] Paulo Cortez and Mark J Embrechts. Opening black box data mining models using sensitivity analysis. In *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 341–348. IEEE, 2011.
 - [19] Daelemans, Walter and Hoste, Veronique. Evaluation of machine learning methods for natural language processing tasks. In *LREC 2002 : third international conference on language resources and evaluation*, page 6. European Language Resources Association (ELRA), 2002.
 - [20] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *CoRR*, abs/2006.11371, 2020.

- [21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [22] Nicholas DiFonzo and Prashant Bordia. Rumor, gossip and urban legends. *Diogenes*, 54(1):19–35, 2007.
- [23] Salma El Anigri, Mohammed Majid Himmi, and Abdelhak Mahmoudi. How bert’s dropout fine-tuning affects text classification? In *International Conference on Business Intelligence*, pages 130–139. Springer, 2021.
- [24] European-Commision. Ethics guidelines for trustworthy AI., 2019.
- [25] European-Commision. Ethics guidelines for trustworthy AI., 2019.
- [26] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL, 2016.
- [27] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.
- [28] Anna Glazkova, Maksim Glazkov, and Timofey Trifonov. g2tmn at constraint@aaai2021: Exploiting CT-BERT and ensembling learning for COVID-19 fake news detection. *CoRR*, abs/2012.11967, 2020.
- [29] Helena Gómez-Adorno, Juan Pablo Posadas-Durán, Gemma Bel Enuguix, and Claudia Porto Capetillo. Overview of fakedes at iberlef 2021: Fake news detection in spanish shared task. *Procesamiento del Lenguaje Natural*, 67:223–231, 2021.
- [30] Santiago González Silot. filfa: Modelado computacional de la desinformación, 2022.
- [31] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378, 2019.
- [32] Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquin Silveira-Ocampo, Casimiro Pio Carrino, Carme Armentano-Oller, Carlos Rodriguez-Penagos, Aitor Gonzalez-Agirre, and Marta Villegas. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68(0):39–60, 2022.

- [33] Balint Gyevnar, Nick Ferguson, and Burkhard Schafer. Get your act together: A comparative view on transparency in the ai act and technology, 2023.
- [34] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- [35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [36] Maximilian Höller. The human component in social media and fake news: the performance of uk opinion leaders on twitter during the brexit campaign. *European Journal of English Studies*, 25(1):80–95, 2021.
- [37] Benjamin Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 759–766, 2017.
- [38] Ian Kelk, Benjamin Basseri, Wee Lee, Richard Qiu, and Chris Tanner. Automatic fake news detection: Are current models “fact-checking” or “gut-checking”? In *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, pages 29–36, Dublin, Ireland, May 2022. Association for Computational Linguistics.
- [39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [40] Ricardo Llugsi, Samira El Yacoubi, Allyx Fontaine, and Pablo Lupeira. Comparison between adam, adamax and adam w optimizers to implement a weather forecast based on neural networks for the andean city of quito. In *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6, 2021.
- [41] Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. Timelms: Diachronic language models from twitter. *arXiv preprint arXiv:2202.03829*, 2022.
- [42] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
- [43] Lluís Marquez and Jordi Girona Salgado. Machine learning and natural language processing, 2000.

- [44] Tanushree Mitra and Eric Gilbert. Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 258–267, 2015.
- [45] Saif Mohammad and Peter Turney. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA, June 2010. Association for Computational Linguistics.
- [46] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.
- [47] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.
- [48] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitterbert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*, 2020.
- [49] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020.
- [50] William Scott Paka, Rachit Bansal, Abhay Kaushik, Shubhashis Sen-gupta, and Tanmoy Chakraborty. Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection. *Applied Soft Computing*, 107:107393, 2021.
- [51] Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas PYKL, Amitava Das, Asif Ekbal, Shad Akhtar, and Tanmoy Chakraborty. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT)*. Springer, 2021.
- [52] Parth Patwa, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. Fighting an infodemic: Covid-19 fake news dataset, 2020.
- [53] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages

- 3391–3401, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.
- [54] Mohammad Taher Pilehvar and Jose Camacho-Collados. Embeddings in natural language processing: Theory and advances in vector representations of meaning. *Synthesis Lectures on Human Language Technologies*, 13(4):1–175, 2020.
 - [55] Juan-Pablo Posadas-Durán, Helena Gómez-Adorno, Grigori Sidorov, and Jesús Jaime Moreno Escobar. Detection of fake news in a new corpus for the spanish language. *Journal of Intelligent & Fuzzy Systems*, 36(5):4869–4876, 2019.
 - [56] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
 - [57] J Ross Quinlan. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, 28(1):71–72, 1996.
 - [58] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
 - [59] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.
 - [60] Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*, 2017.
 - [61] Aránzazu Román-San-Miguel, Nuria Sánchez-Gey Valenzuela, and Rodrigo Elías Zambrano. Las fake news durante el estado de alarma por covid-19. análisis desde el punto de vista político en la prensa española. *Revista latina de comunicación social*, (78):359–391, 2020.
 - [62] Victoria L Rubin, Yimin Chen, and Nadia K Conroy. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015.
 - [63] James Rumbaugh, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education, 2004.
 - [64] Fabrizio Sebastiani and Andrea Esuli. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th international conference on language resources and evaluation*, pages 417–422. European Language Resources Association (ELRA) Genoa, Italy, 2006.

- [65] Lloyd S Shapley et al. A value for n-person games. 1953.
- [66] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188, 2020.
- [67] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36, sep 2017.
- [68] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer, 2019.
- [69] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.
- [70] Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*, 2017.
- [71] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.
- [72] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.
- [73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [74] Andreas Vlachos and Sebastian Riedel. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22, 2014.
- [75] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.
- [76] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface’s transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

- [77] Wei Emma Zhang, Quan Z. Sheng, and Ahoud Alhazmi. Generating textual adversarial examples for deep learning models: A survey. *CoRR*, abs/1901.06796, 2019.
- [78] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40, 2020.
- [79] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G. Michael Youngblood. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, 2018.
- [80] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. Detection and resolution of rumours in social media: A survey. *ACM Comput. Surv.*, 51(2), feb 2018.
- [81] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Kalina Bontcheva, and Peter Tolmie. Towards detecting rumours in social media. In *Workshops at the Twenty-Ninth AAAI conference on artificial intelligence*, 2015.
- [82] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989, 2016.

