## **Índice**

**1.**[**Introducción** 2](#_Toc170377514)

[**1.1 Motivación** 3](#_Toc170377515)

[**1.2 Objetivos** 3](#_Toc170377516)

[**1.3 Planificación** 4](#_Toc170377517)

[**1.4 Estructura** 4](#_Toc170377518)

**2.**[**Desinformación y Procesamiento del Lenguaje Natural** 5](#_Toc170377519)

[**2.1 Desinformación** 5](#_Toc170377520)

[**2.2 Procesamiento del Lenguaje Natural** 6](#_Toc170377521)

[**2.2.1 Transformers** 10](#_Toc170377522)

[**2.2.2 Modelos del Lenguaje** 11](#_Toc170377523)

[**2.2.3 RoBERTa** 12](#_Toc170377524)

**3.**[**Detección de Fake News** 14](#_Toc170377525)

[**3.1 Datasets disponibles** 14](#_Toc170377526)

[**3.1 Análisis y Estudio del conjunto de datos** 15](#_Toc170377527)

[**3.2 Fine-Tuning del modelo RoBERTa** 18](#_Toc170377530)

[**3.2.1 RoBERTa-large-bne** 20](#_Toc170377531)

[**3.2.2 Xlm-RoBERTa-large** 22](#_Toc170377532)

[**3.2.3 RoBERTalex** 24](#_Toc170377533)

[**3.2.4 Conclusiones** 26](#_Toc170377534)

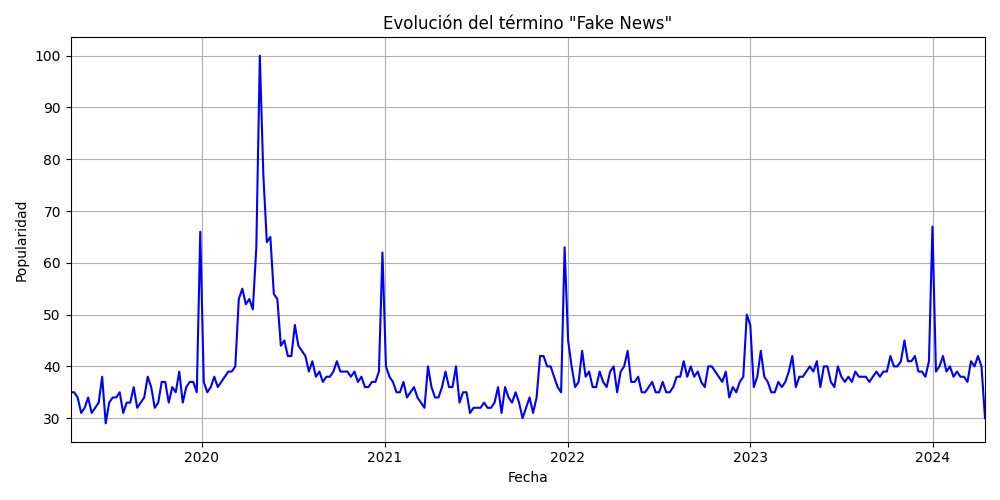
[**Referencias** 27](#_Toc170377535)

Capítulo 1

## **Introducción**

En plena era digital, el fenómeno de las noticias falsas, o “*fake news”,* ha emergido como una poderosa fuerza que trasciende fronteras lingüísticas, influenciando de manera significativa la percepción pública, la toma de decisiones o la estabilidad democrática en todo el mundo. Este impacto se ve aumentado por la rapidez con la que la desinformación se propaga a través de las redes sociales y otros canales de información.

Uno de los principales objetivos de las *fake news* es la manipulación de la opinión de la sociedad para fines políticos o sociales. Actualmente, podemos apreciarlo con claridad con el conflicto entre Israel y Palestina, donde a diferencia de otros conflictos bélicos en el pasado las nuevas armas toman forma de ‘post’ o publicación cuya crudeza interpela directamente a los sentimientos de los usuarios. [1]



**Fig 1 - Evolución del término "Fake News" durante los años**

En la siguiente gráfica podemos observar el crecimiento de la popularidad del término *fake news,* destacando 3 escenarios históricos relevantes. El primero es en plena pandemia y se extiende hasta finales de 2021. El segundo en 2022, el conflicto entre Ucrania y Rusia. Por último, a finales de 2023, cuando el conflicto entre Israel y Palestina estaba en su punto álgido.

Aunque a día de hoy la mayoría de *fake news* que vemos provienen de usuarios anónimos con un bajo nivel cultural, detrás de la propagación de bulos y desinformación se encuentran partidos políticos o grandes empresas.

Un ejemplo claro fue el caso de *Cambridge Analytica*, una empresa que saltó a la fama por su supuesta participación en la campaña electoral presidencial de Estados Unidos en 2016 o con el Brexit en Europa. La propia empresa tiene en la portada de su página web el slogan “Data drives all we do. Cambridge Analytica uses data to change audience behaviour” [2]

En el siguiente artículo del periódico EL PAÍS se hace especial énfasis en la comunidad hispanohablante en Estados Unidos señalando que esta población es la más afectada por la desinformación en el país. Las traducciones al español pueden transformar un contexto en otro totalmente diferente, algo de lo que se aprovechan los medios de comunicación en la zona. Temas muy críticos como la salud pública o los derechos de inmigración son los que más preocupan a los habitantes hispanohablantes. [3]

Con estos ejemplos de la actualidad social, subrayan la importancia para los ciudadanos de saber diferenciar entre información verídica y falsa, ya que como se ha visto tiene un gran impacto tanto en la sociedad como en el individuo.

## **1.1 Motivación**

El desafío de distinguir entre noticias falsas y verdaderas se complica actualmente por la falta de tiempo. Cada día recibimos más información proveniente de diversos medios como redes sociales y medios de comunicación tradicionales entre otros. Las personas no disponen del tiempo que se necesita para contrastar que la noticia que estamos leyendo es falsa o no.

Esta situación se agrava más si hablamos de las noticias en español. La comunidad hispanohablante se enfrenta a barreras adicionales en el acceso a información confiable en su propio idioma debido a que el idioma que predomina en estos casos es el inglés.

La creación de un sistema automático de detección de *fake news* que se enfoque en el español se vuelve crucial. El objetivo es abordar el problema con el texto como el principal criterio para determinar la veracidad de la información. Esto se llevará a cabo desde la perspectiva del Procesamiento del Lenguaje Natural (PLN), con el fin de analizar y evaluar la autenticidad del texto. No se emplearán otros métodos, como las fuentes o la red subyacente de comunicación entre usuarios que comparten la noticia falsa.

## **1.2 Objetivos**

El objetivo principal de este trabajo fin de grado es el desarrollo de un sistema de detección de desinformación basado en RoBERTa.

A continuación, se enumeran los objetivos específicos necesarios para la realización del objetivo principal:

1. **Analizar el estado del arte del problema**: Investigar el panorama actual en cuanto a la detección de desinformación, *Deep Learning* y el Procesamiento del lenguaje Natural (PLN), revisando las contribuciones más destacadas en la literatura.
2. **Evaluar los conjuntos de datos más relevantes:** Analizar los conjuntos más relevantes y actualizados disponibles para abordar el problema de detección de noticias falsas.
3. **Diseñar e implementar un sistema de aprendizaje profundo:** Se desarrollará un sistema que se enfoque en la detección de noticias falsas, utilizando exclusivamente el contenido textual de las noticias (título y cuerpo) sin depender de datos adicionales.
4. **Desarrollar una interfaz web interactiva:** Implementar una interfaz que permita visualizar y demostrar de manera efectiva el funcionamiento del sistema desarrollado.

## **1.3 Planificación**

## **1.4 Estructura**

Capítulo 2

## **Desinformación y Procesamiento del Lenguaje Natural**

En este capítulo, se establece el marco teórico del trabajo de investigación, enfocado en el estudio y comprensión de las *fake news,* Procesamiento del Lenguaje Natural, *Transformers,* BERT y RoBERTa.

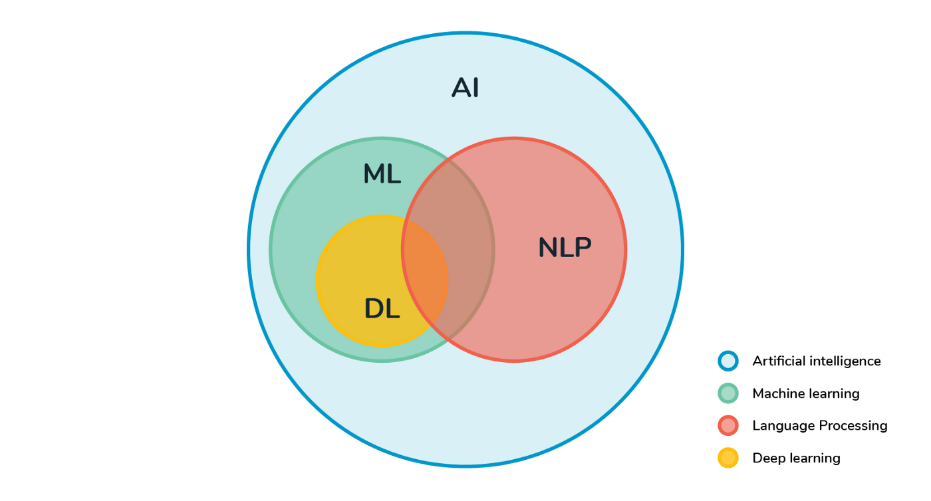
## **2.1 Desinformación**

La desinformación es un tipo de información falsa o parcialmente falsa, la cual se difunde de manera intencionada para manipular al lector. “Este fenómeno hace referencia tanto al contenido informativo fraudulento (*fake news),* al engañoso (*misleading content),* los discursos de odio (*hate speech),* los discursos falso deliberados (*false speech)* o los errores informáticos no deliberados de medios o periodistas (*misinformation)”*. [4]

* **Fake News**: “Se conocen como ***fake news*** al tipo de bulo que consiste en un contenido seudoperiodístico difundido a través de portales de noticias, prensa escrita, radio, televisión y redes sociales y que tiene como objetivo desinformar a un público en específico”. [5]
* **Misleading Content:** Es el tipo de noticia más difícil de descubrir. El *misleading content* o contenido engañoso en español, puede aparecer en muchísimas noticias reales, y la razón por la que es tan difícil de descubrir es porque se requiere algún tipo de experiencia o conocimiento sobre un tema determinado para determinar si los hechos y los detales de cualquier artículo de noticias están siendo tergiversados. [6]
* **Discurso de odio:** Discurso público que expresa odio o fomenta la violencia hacia una persona o grupo por motivos de raza, religión, sexo u orientación sexual. [7]
* **False Speech:** Declaraciones o discursos intencionalmente falsos o engañosos pronunciados por individuos con la intención de manipular la opinión pública o alcanzar algún objetivo específico. Puede incluir declaraciones políticas falsas, promesas incumplidas o afirmaciones engañosas para obtener ventaja personal o política. [4]
* **Misinformation:** Este término abarca todos los tipos de información incorrecta, ya sea intencional o no. Incluye errores involuntarios, información desactualizada o malinterpretada, así como contenido falso o engañoso [8]. La desinformación puede propagarse fácilmente en línea y puede tener consecuencias significativas en la percepción pública y en la toma de decisiones.

## **2.2 Procesamiento del Lenguaje Natural**

El procesamiento del lenguaje natural es un subcampo de la lingüística, ciencia de computadores e inteligencia artificial dedicado a diseñar métodos y algoritmos que toman como entrada o salida lenguaje natural no estructurado [9]. El objetivo es que un ordenador sea capaz de generar y “entender” los contenidos de un texto para así poder interactuar con él.



**Fig 2 - Diagrama NLP**

En los últimos años, el procesamiento de grandes cantidades de información a través de texto está cada vez más presente, surgiendo multitudes de dificultades para poder tratarlos. Es por ello que esta disciplina está alcanzando un importante papel en la actualidad. Entre las tareas más comunes donde se usa el procesamiento del lenguaje natural se encuentran las siguientes:

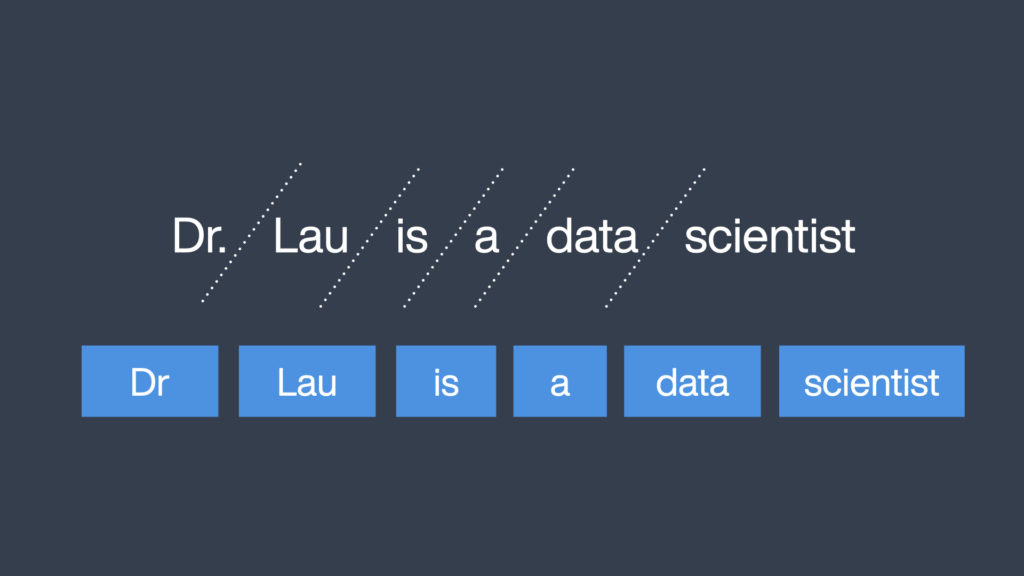
* **Reconocimiento de voz**. Dado un audio de una persona hablando, convertirlo en texto.
* ***Text-to-speech****.* Al contrario que el reconocimiento de voz, transforma el texto en audio.
* **Clasificación de texto.** Asignar a un texto o un documento una o más categorías.
* **Traducción automática.** Traducción automática de texto de un idioma a otro.
* **Generación automática de texto.** Dadas unas características generar un texto que las cumpla.
* **Análisis de sentimientos.** Caso específico de la clasificación de texto en el que se le asigna un sentimiento al texto.

Cuando tenemos un problema de PLN, podemos distinguir dos aspectos esenciales:

* La parte “**lingüística**”, que consiste en preprocesar y transformar la información de entrada en una serie de datos explotables.
* La parte de “**aprendizaje automático**”, que se basa en la aplicación de modelos de *Machine Learning* o *Deep Learning* a ese conjunto de datos.

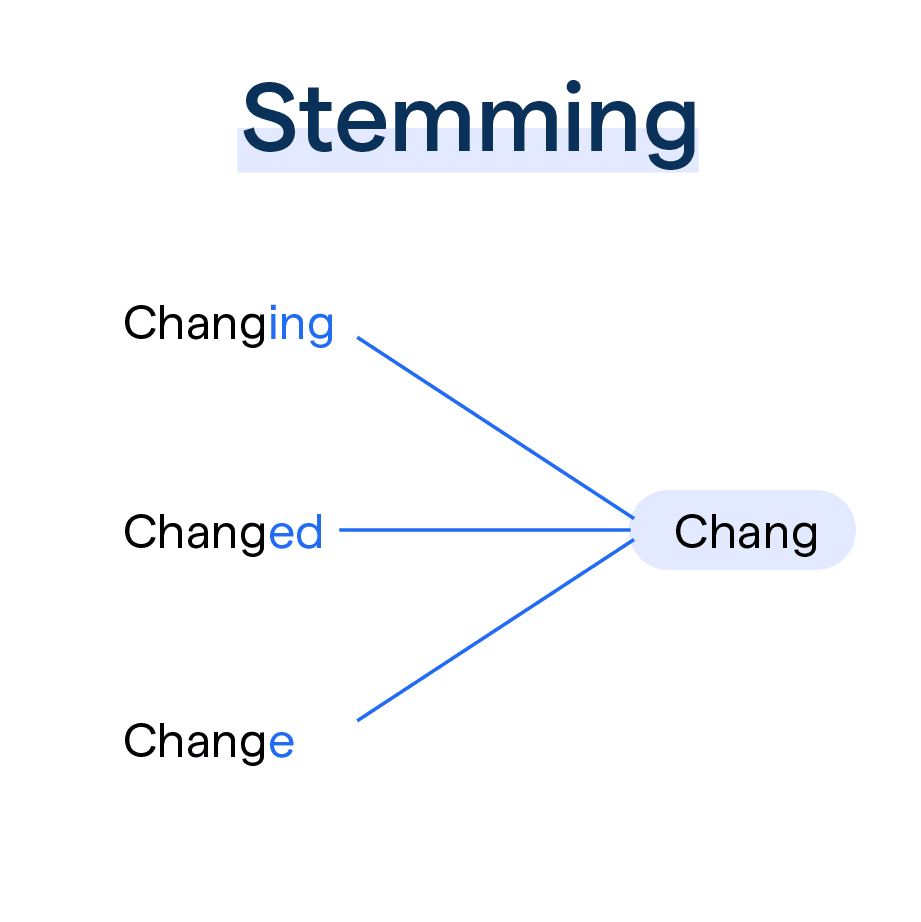
La fase de preprocesamiento consiste en limpiar y preparar el texto para que un algoritmo PLN pueda analizarlo. Entre las principales etapas que se siguen, encontramos:

* **Limpieza del texto**: eliminar signos de puntuación, números, eliminar etiquetas HTML, símbolos, cambio a minúsculas.
* **Tokenización**: proceso de dividir la cadena de texto en unidades más pequeñas llamadas **tokens** usando el espacio como seperardor**.** Los tokens incluyen palabras, caracteres y subpalabras.



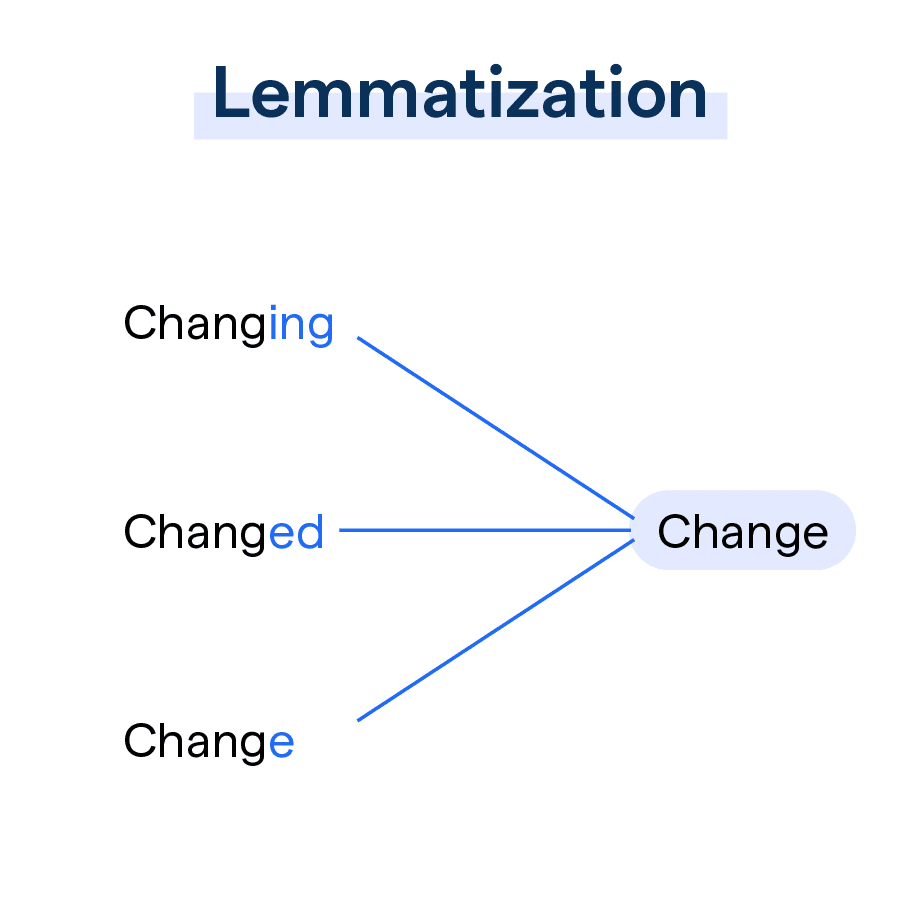
**Fig 3 - Ejemplo de tokenización de una frase**

* **Stemming**: una misma palabra se puede encontrar de diferentes formas dependiendo del género, del número, la persona etc. El stemming designa el simple proceso heurístico de cortar el final de las palabras para mantener solo la raíz de la palabra.



**Fig 4 - Ejemplo de stemming**

* **Lematización**: consiste en realizar la misma tarea que el stemming pero utilizando un vocabulario y un análisis minucioso de la construcción de las palabras. La lematización permite eliminar únicamente las terminaciones inflexibles y de ese modo aislar la forma canónica de la palabra, es decir, el lema.



**Fig 5 - Ejemplo de lematización**

Para poder aplicar métodos de Machine Learning a los problemas relativos al lenguaje natural, es indispensable transformar los datos textuales en datos digitales.

Alguna de las técnicas clásicas de PLN utiliza el **tf-idf** (Term Frequency-Inverse Document Frequency) como entrada para modelos clásicos de *Machine Learning* como pueden ser las máquinas de vectores soporte y los árboles de decisión.

El *Machine Learning,* dentro del campo de la Inteligencia Artificial, busca capacitar a las máquinas para aprender sin una programación explítica (Arthur Samuel, 1959) [10]. Desde una perspectiva más contemporánea, según Tom Mitchell en 1997 [10]: “Un programa de computador se dice que aprende de la experiencia *E* con respecto a alguna clase de tareas de *T*, medido por *P*, mejora con la experiencia *E*.”

Dentro del *Machine Learning,* se distinguen varios tipos de aprendizaje, siendo los más comunes el aprendizaje supervisado y el no supervisado. En el aprendizaje supervisado, se aprende una función utilizando datos etiquetados, donde las etiquetas guían el proceso de aprendizaje. Por otro lado, el aprendizaje no supervisado, se aprende una función únicamente a partir de los datos, sin etiquetas predefinidas que guíen el proceso.

El TF-IDF (*Term Frequency-Inverse Document Frequency)* [11]es una medida numérica que indica la relevancia de una palabra en un documento específico dentro de un conjunto de documentos. Este método se calcula contando las apariciones de cada palabra en un documento (*Term Frequency),* y luego ponderando esa frecuencia por la frecuencia inversa de documentos que contienen ese término (*Inverse Document Frequency).* La formulación matemática es la siguiente:

Siendo ***tf (t, d)*** la frecuencia del término *t* en un documento *d:*

Donde:

* es el número de veces que aparece *t* en un documento *d*.
* es el número total de términos en el documento *d*

Siendola frecuencia de documento inversa, es decir, una medida de cuanta información proporciona una palabra, sobre como de común o específica es en todos los documentos.

Donde:

* *N* es el número total de documentos en el corpus,
* es el número de documento en los que el término *t* aparece.

Por tanto, el valor de *tf-idf* de las palabras de un texto, sirve como entrada para los modelos de *Machine Learning* tales como las Máquinas de Soporte Vectorial, Árboles de Decisión o Regresión Logística entre otros. Si bien estos modelos logran un buen resultado a la hora de resolver los problemas de PLN y más en concreto de clasificación de texto, el estado del arte viene marcado por distintos modelos basados en *Deep Learning*, entre los que destacan:

* **CNN,** Redes Neuronales Convolucionales
* **RNN,** Redes Neuronales Recurrentes.
* **Modelos con mecanismos de atención.**
* **Redes de memoria aumentada.**
* **Transformers y Modelos de lenguaje pre-entrenados.**

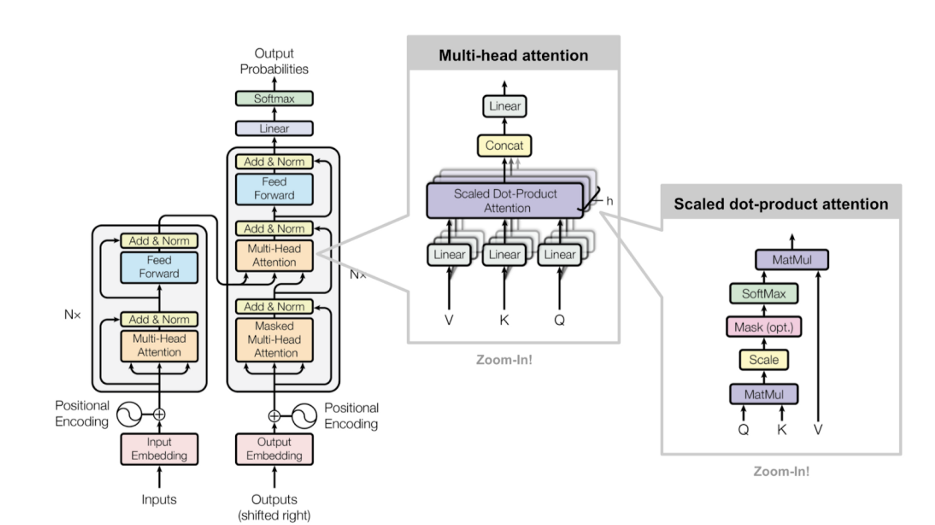
## **2.2.1 Transformers**

Un Transformer es una arquitectura de red neuronal desarrollada para tareas de procesamiento de lenguaje natural (PLN). Esta arquitectura se ha convertido en una de las más dominantes en el campo del PLN, superando a modelos neuronales alternativos como las redes neuronales convolucionales y recurrentes en el rendimiento para tareas de comprensión y generación de lenguaje natural [12].

Esta arquitectura soluciona los problemas de dependencias largas que presentenan las redes neuronales recurrentes, las cuales procesan el texto palabra a palabra y de manera secuencial. En luhar de depender de la recurrencia para procesar secuencias de textos, los Transformers utilizan mecanismos de atención para permitir que cada token en la secuencia se relacione con todos los demás tokens de manera simultánea [13]. Esta capacidad de atención global les permite capturar relaciones complejas y de largo alcance en el texto, lo que ha demostrado ser fundamental para el éxito en tareas como clasificación, traducción automática y generación de texto.

Además, los Transformers eliminan esa dependencia que tenía las redes recurrentes. Esta arquitectura tiene la capacidad de procesar secuencias de entrada más rápidas y de forma efectiva, lo que los hace ideales para tareas que involucran grandes cantidades de datos.

La arquitectura de los Transformers tiene una estructura *encoder-decoder,* esta estructura se basa en codificar la secuencia de entrada para posteriormente descodificarla. La peculiaridad de esta arquitectura es el uso únicamente de mecanismos de atención sin la inclusión de redes recurrentes o redes convolucionales. [14]



ENCODER

DECODER

**Fig 6 - Arquitectura de Transformers**

El *encoder* consta de 6 bloques idénticos. Cada bloque del *encoder* tiene una estructura específica que incluye diferentes subcomponentes:

* **Bloque Atencional (*Multi-head self attention mechanism)***

El mecanismo de atención en un Tranformer [13] permite al modelo calcular la importancia relativa de cada token en la secuencia con respecto a un token de consulta dado, lo que ayuda al modelo a capturar relaciones y dependencias entre los tokens de manera más efectiva.

Para expresar numéricamente las relaciones entre las partes según los grados de asociación entre palabras se utiliza el mecanismo de atención *Scaled Dot-Product Attention.*

Recibe como entrada tres conjuntos de vectores: *query*(Q), *key*(K) y *value*(V) [14], que se obtienen mediante transformaciones lineales de los embeddings de entrada. Se calcula el producto escalar y posteriormente se divide entre para controlar la escala de los vectores resultantes y evitar que se vuelvan demasiado grandes o demasiado pequeños a medida que aumenta la dimensión de los vectores.

* **Bloque Residual (*Add & Norm*)**

Recibe tanto la entrada como la salida del bloque atencional, ya que la red al ser muy profunda, enviar solo los datos de la salida del bloque atencional puede causar degradación progresiva de la información, dificultando el entrenamiento y desempeño de la red. [15]

Este bloque suma los datos de la entrada y salida para después normalizarlos ya que estos datos han sido “deformados” tras el resto de capas. Al sumar la entrada original a la salida transformada, se crea una ruta directa para el flujo de información, mejorándola eficiencia del aprendizaje.

* **Bloque Feedforward**

Este bloque transforma la salida del bloque residual mediante dos capas lineales y una función de activación no lineal [14]. Esto permite al modelo aprender relaciones más complejas entre los datos de entrada.

La entrada y la salida de este bloque son llevadas a otro bloque residual que preparará los datos para el siguiente *encoder.*

El *decoder* tambiénconsta 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 una capa *multi-head self attention mechanism* enmascarada para que tan solo utilice las palabras que ya ha procesado.

## **2.2.2 Modelos del Lenguaje**

Los Modelos del Lenguaje (LM) son herramientas diseñadas para distinguir entre secuencias gramaticales y no gramaticales. Dada una frase, un modelo de lenguaje ha de identificar si esta frase es plausible o no dada la gramática del lenguaje [16]. En los últimos años se han publicado una gran cantidad de modelos basados en redes neuronales profundas, entre los más conocidos en la literatura se encuentran:

* BERT
* RoBERTa
* GPT-4
* GPT-3.5
* ELECTRA

Para este trabajo se ha decidido utilizar RoBERTa, desarrollado por Facebook, ya que es una versión mejorada el modelo BERT, desarrollado por Google. Este modelo ha sido entrenado en conjuntos de datos mucho más grandes y multilingües en comparación a BERT, incluyendo el español. Además, el uso de enmascaramiento dinámico, que se explicará con más detalle en la siguiente sección.

## **2.2.3 RoBERTa**

RoBERTa (*Robustly Optimized BERT pre-training Approach)* es una variante de BERT (*Bidirectional Encoder Representations from Transformers*), que fue desarrollada por investigadores de Facebook AI [17]. Al igual que BERT, RoBERTa es un modelo de lenguaje basado en Transformers que utiliza la autoatención para procesar secuencias de entrada y generar representaciones contextualizadas de palabras en una frase.

A continuación, se explicará la arquitectura BERT, ya que, como se ha mencionado, RoBERTa es una variante de BERT.

El modelo BERT surgió de 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 capas adicionales para especializarse en una tarea en concreto [18] . Una de las principales diferencias de este modelo con los modelos de lenguaje tradicionales es que BERT utiliza un enfoque bidireccional, considerando el contexto izquierdo y derecho de las palabras de una oración. En lugar de analizar el texto secuencialmente, BERT analiza todas las palabras de una oración simultáneamente.

BERT emplea una arquitectura de solo codificador. En la arquitectura *Transformer* original, hay módulos codificadores y decodificadores. La decisión de utilizar una arquitectura de solo codificador en BERT sugiere un enfoque principal en comprender secuencias de entrada en lugar de generar las secuencias de salida.

El nombre de la arquitectura en concreto que utiliza BERT es *multi-layer bidirectional Transformer encoder* [18]. Como se observa en la figura 7, se trata de una concatenación de varias capas, siendo cada una de estas capas un bloque *transformer* como los explicados anteriormente.



**Fig 7 - Arquitectura BERT con el bloque Transformer**

El entrenamiento de BERT consta de 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 *BookCorpus* [18]. En concreto BERT se entrena en las tareas de *Masked Language (MLM) y* *Next Sentence Prediction (NSP),* lo cual ayuda a BERT a entender el lenguaje. Aquí se muestra una imagen del funcionamiento de la fase de entrenamiento de BERT.

En la tarea de *Masked Language* (Modelo de Lenguaje enmascarado en español), BERT enmascara algunas palabras de una secuencia de entrada y se entrena para predecir las palabras originales basándose en el contexto de las palabras circundantes [18]. Este enfoque ayuda a comprender mejor el significado y contexto de las palabras en un texto.

Por otro lado, en la tarea de *Next Sentence Prediction* (Predicción de la siguiente frase en español) [18], BERT predice si una segunda frase está conectada con la primera. Para ello, transforma la salida del token *[CLS]* en un vector y calcula la probabilidad de que la segunda oración siga a la primera usando una función *softmax.* Con esta tarea se entrena al modelo para entender la relación entre pares de frases.



**Fig 8 - Representación de la fase de pre-entrenamiento de BERT**

En la segunda fase, se realiza un *fine-tunning* 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.

Una vez explicada la arquitectura de BERT, vamos a ver los principales puntos donde RoBERTa gana a su predecesor:

* **Mayor tamaño de datos de entrenamiento:** RoBERTa se entrenó con un conjunto de datos mucho más grande y utilizó un procedimiento de entrenamiento más efectivo. Este modelo fue entrenado con 160 Gb de texto, que es 10 veces más grande que el conjunto de datos con el que se entrenó BERT. Ese conjunto además incluye textos en varios idiomas, lo que lo hace un modelo muy versátil para tareas en diferentes lenguajes.
* **Enmascaramiento dinámico**: A diferencia de BERT, que utiliza un enmascaramiento estático (donde las mismas partes del texto están enmascaradas en cada época), RoBERTa utiliza un enmascaramiento dinámico. Esto significa que se enmascaran diferentes partes del texto en cada época, lo cual hace que el modelo sea mucho más robusto.
* **Eliminación de NSP**: En el pre-entrenamiento de RoBERTa, solo se utiliza la tarea de MLM (*Masked Language Modeling)* y se elimina la tarea NSP *(Next Sentence Prediction).* Los autores experimentaron con la inclusión y exclusión de la pérdida de NSP en varias versiones y concluyeron que eliminar la tarea del NSP iguala o mejora ligeramente el rendimiento en tareas posteriores.
* **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. Esto resulta en una mejora tanto en la velocidad como en el rendimiento del modelo.

Capítulo 3

## **Detección de Fake News**

Para realizar un estudio científico sobre la detección de noticias falsas en español, primero se examinarán los conjuntos de datos disponibles en la literatura. Se identificarán aquellos conjuntos de datos que sean más relevantes y actualizados, con un enfoque particular en aquellos que contengan noticias en español.

Después de este análisis, se presentarán unos modelos bases que servirán como referencia. Finalmente, se presentarán los modelos propuestos para el estudio y sus resultados correspondientes.

## **3.1 Datasets disponibles**

Para resolver el problema de la detección de *fake news,* se ha realizado un estudio de los conjuntos de datos más reconocidos y utilizados en este ámbito. A continuación, se muestra un listado de los conjuntos de datos analizados y sus características.

* **Liar** [19]**:** Este conjunto contiene un total de 12.836 declaraciones ciertas recogidas de Politifact. Cada declaración se mide en una escala de seis opciones de veracidad y también incluye información sobre la temática, partido político, contexto y orador. Los datos están en inglés.
* **FakesNewsNet** [20]**:** Consiste en la cabecera y cuerpo del texto de artículos de noticias falsas obtenidas en BuzzFeed y Politifact. Los datos están en inglés.
* **CONSTRAINT@AAI 2021** [21]**:** Consiste en 10.700 publicaciones en inglés, con 5.100 noticias falsas y 5.600 verdaderas. Las noticias verdaderas se recogen en X (antiguo Twitter), mientras que las falsas provienen de varias fuentes, incluyendo X, Facebook y sitios de fact-checking como Politifact, NewsChecker y Boomlive.
* **Spanish Fake News Corpus** [22]**:** Este conjunto contiene 971 noticias en español sobre 9 dominios distintos, clasificadas en verdadero y falso. El corpus contiene noticias de 9 tópicos diferentes: ciencia, deporte, economía, educación, entretenimiento, política, salud, seguridad y sociedad.
* **FNC-1 dataset** [23]**:** *Fake News Challenge* es un conjunto de datos en inglés que contiene 49.972 artículos provenientes del projecto EMERGENT sobre política, sociedad y tecnología. El conjunto de datos está preparado para detectar si la noticia es falsa o no según si está relacionado el título de la noticia con el cuerpo.
* **Fact checking dataset** [24]**:** Es el primer conjunto de datos públicos para la detección de fake news. Los datos provienen de Politifact y Channel4. El conjunto de datos en inglés contiene 221 declaraciones, con la fecha en la que fueron hechas, el orador y la URL. La veracidad se mide en una escala entre 5 opciones: verdadero, mayoritariamente verdadero, medio verdad, mayoritariamente falso y falso.

Para este estudio se ha escogido el conjunto de datos de **Spanish Fake News Corpus** ya que ha participado en IberLEF [25], el cual se celebra anualmente en el congreso internacional de la Sociedad Española de Procesamiento del Lenguaje Natural. IberLEF se enfoca en la evaluación de sistemas de Procesamiento del Lenguaje Natural en español y otros idiomas ibéricos.

## **3.1 Análisis y Estudio del conjunto de datos**

Antes de profundizar en el análisis detallado del corpus, primero se examinarán algunas medidas básicas para establecer una comprensión inicial del conjunto de datos. Estas medidas incluyen número total de palabras del corpus, el balance de clases, tamaño total del conjunto de datos y la división del conjunto de datos en subconjuntos de entrenamiento y prueba.

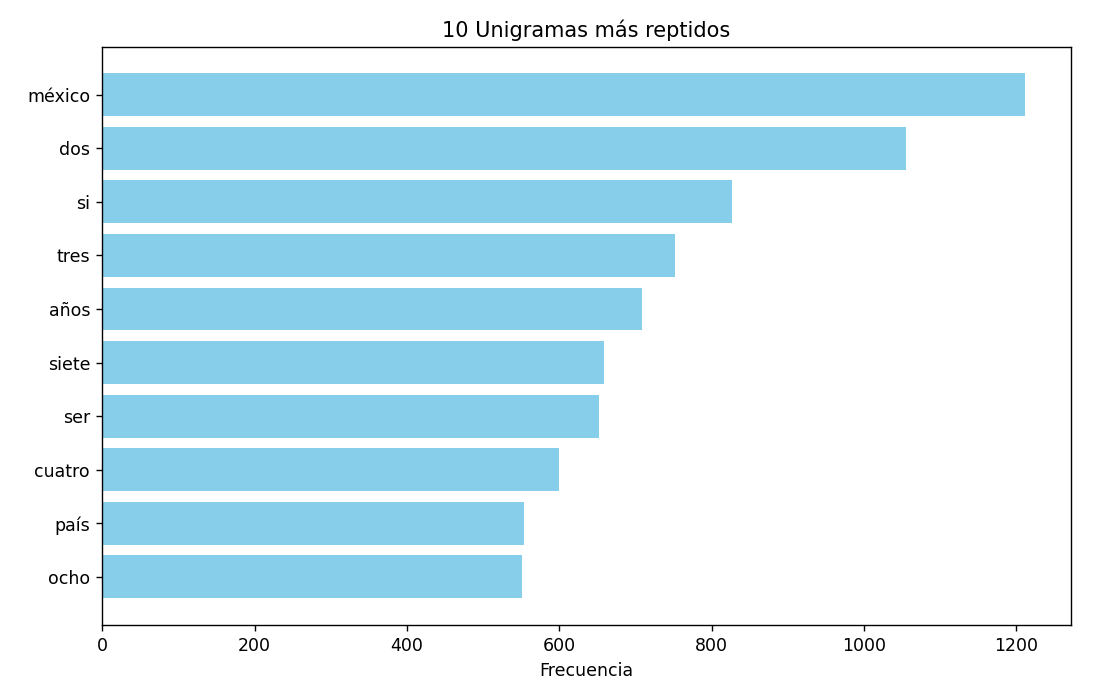
|  |  |
| --- | --- |
| Medida | Spanish Fake News Corpus |
| Tamaño total del conjunto de datos | **1.248** |
| Total de palabras del corpus | **68.616** |
| Tamaño del conjunto de entrenamiento | **676** |
| Tamaño del conjunto de prueba | **572** |
| Número de noticias verdaderas | **491** |
| Número de noticias falsas | **480** |

**Tabla 1 - Medidas básicas del Spanish Fake News Corpus**

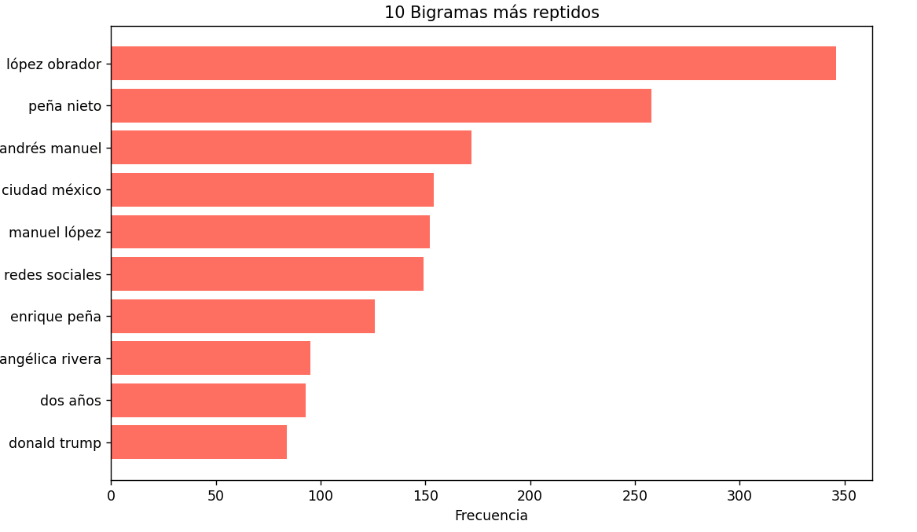
El corpus presenta un conjunto de datos bien equilibrado para el análisis de noticias en español. Aunque no sea un conjunto muy grande, ofrece una muestra significativa para el desarrollo de la investigación. Con 491 noticias verdaderas y 480 falsas, el corpus exhibe un balance notable entre ambas clases, asegurando que los modelos pueden aprender de manera equilibrada las características distintivas de noticias verdaderas y falsas.

El análisis que se va a llevar a cabo, se examinarán 3 tipos de N-gramas: los unigramas, bigramas y trigramas. Un N-grama es una secuencia de n elementos consecutivos en un texto [26]. Por ejemplo, si estamos analizando los unigramas, serán las secuencias compuestas por una sola palabra. Los bigramas son secuencias de dos palabras consecutivas, y los trigramas son secuencias de tres palabras consecutivas. Esto se hará con el fin de analizar alguna peculiaridad del corpus.

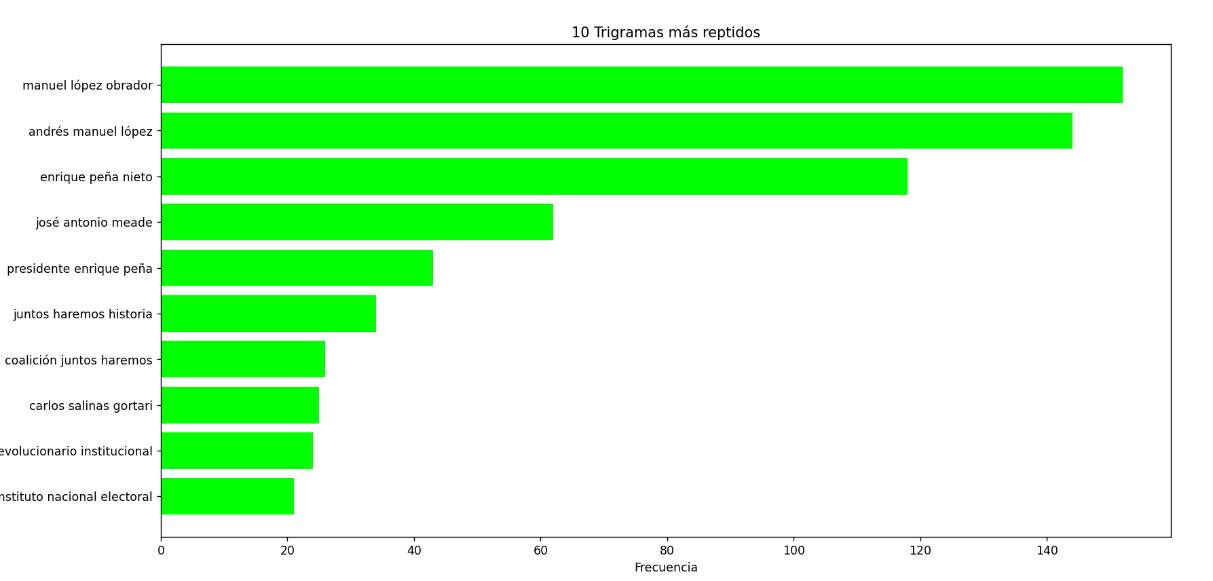
Para concluir, se mostrará una nube de palabras del conjunto de datos. Estas nubes de palabras representan gráficamente las palabras más repetidas en todo el conjunto de datos, ajustando su tamaño a la frecuencia con la que aparecen. Las palabras más repetidas ocuparán más tamaño que las que no se repiten tanto. Además, se analizarán las nubes de palabras del conjunto de noticias falsas y verdaderas, con el objetico de identificar diferencias entre ambos grupos.

****

**Fig 9 - Unigramas más repetidos**

****

**Fig 10 - Bigramas más repetidos**

****

**Fig 11 - Trigramas más repetidos**

Observamos en las figuras 9, 10 y 11 los N-gramas más repetidos junto al número de veces que se repiten en el total del corpus. Podemos ver que entre los N-gramas más repetidos aparecen términos como “México”, “país”, “López obrador”, “peña nieto”. Es decir, todos los términos parecen estar bastante relacionados con la política y haciendo hincapié en la política mexicana.

Si observamos la nube de palabras del corpus, como se muestra en la figura 12, podemos identificar los términos más repetidos por su tamaño. Muchas de estas palabras ya han sido destacadas como los N-gramas más frecuentes en las figuras 9, 10 y 11. Al igual que en las gráficas anteriores, notamos que los términos más repetidos incluyen “México”, “país” o “López obrador” entre otros.

****

**Fig 12 - Nube de palabras del corpus**

## 

## 

**Fig 13 - Nube de palabras del conjunto de noticias falsas**

**Fig 14 - Nube de palabras del conjunto de noticias verdaderas**

Como vemos en las figuras 14 y 15, al comparar las nubes de palabras de las noticias verdaderas y falsas, observamos que en ambas destacan “México” y “país”. Sin embargo, las noticias verdaderas incluyen palabras más específicas y concretas como “año” o “López Obrado”, reflejando un enfoque en eventos y figuras políticas concretas.

Por otro lado, las noticias falsas presentan términos más generales y vagos como “persona”, “ser” y “así”, junto con “además”, lo que sugiere un estilo menos preciso y más general.

## **3.2 Fine-Tuning del modelo RoBERTa**

Para realizar la resolución de la tarea de clasificación de noticias falsas, se han propuesto tres modelos base basados en RoBERTa. A estos tres modelos se le aplicará un *fine-tunning* [27], esto consiste en adaptar un modelo base previamente entrenado para una tarea específica utilizando un conjunto de datos adicional y más específico. Esto permite que el modelo refine sus parámetros para mejorar su rendimiento en esa tarea particular.

Además, se usarán tres optimizadores diferentes. Los optimizadores se emplean para ajustar los parámetros del modelo durante el entrenamiento, minimizando la función de pérdida y mejorando la precisión del modelo en la tarea de clasificación. Los diferentes optimizadores pueden influir en la rapidez y efectividad del entrenamiento.

Los 3 modelos base basados en RoBERTa son los siguientes:

* **PlanTL-GOB-ES/roberta-large-bne** [28]**:** Es el modelo más grande del conjunto de modelos del lenguaje del Plan de Gobierno de España de Tecnologías del Lenguaje. Ha sido entrenado en el corpus más grande conocido en español, incluye 570GB de texto limpio, recopilado por la Biblioteca Nacional de España entre 2009 y 2019. Su dominio es independiente, es decir, abarca datos desde el ámbito de la literatura, prensa hasta ciencia tecnología y otros ámbitos de conocimiento general.
* **FacebookAI/xlm-roberta-large** [29]**:** Es una versión multilingüe de RoBERTa, preenetrenada con 2,5 TB de datos de CommonCrawl filtrados, que abarcan 100 idiomas. Este modelo abarca una amplia variedad de dominios, como artículos de noticias, blogs, foros, literatura y sitios web generales.
* **PlanTL-GOB-ES/RoBERTalex** [30]**:** Este modelo también del Plan de Gobierno de España de Tecnologías del Lenguaje está entrenado específicamente en el dominio legal español. Utiliza un corpus de 8,9 GB de datos legales, incluyendo textos como legislación, sentencias judiciales y documentos administrativos.

En cuanto a los optimizadores, se usarán los siguientes:

* **Adam** [31]**:** Este optimizador es ampliamente usado en clasificación binaria debido a su capacidad para adaptar dinámicamente las tasas de aprendizaje según la magnitud de los gradientes de los parámetros. Combina las ventajas del método del gradiente descendente estocástico con un término de *momentun* (método que acelera la convergencia del entrenamiento acumulando el gradiente de iteraciones anteriores para moverse en la dirección adecuada con mayor consistencia), acelerando la convergencia del entrenamiento y mejorando la estabilidad. Adam inicializa y actualiza los momentos de primer y segundo orden durante el proceso de optimización.
* **AdamW** [32]**:** Es una variante de Adam diseñada para mejorar la regularización durante el entrenamiento. A diferencia de Ada, AdamW incorpora un término de regularización directamente en el proceso de optimización mediante la norma L2 de los pesos del modelo. Esto ayuda a controlar el sobreajuste al penalizar los pesos más grandes, mejorando así la generalización del modelo.
* **Adamax** [31]**:** Es una variante de Adam que simplifica el cálculo al usar solo el momento de primer orden y adapta las tasas de aprendizaje según la norma infinito de los gradientes. A diferencia de Adam, que utiliza momentos de primer y segundo orden, Adama mejora la eficiencia computacional al eliminar el cálculo del segundo momento. En comparación con AdamW, que incorpora regularización para mejorar la generalización del modelo, Adamax se enfoca exclusivamente en la adaptación dinámica de las tasas de aprendizaje sin agregar términos de regularización adicionales.

Para los 3 modelos se han utilizado los siguientes hiperparámetros: el mismo *learning rate* de 2e-5 recomendado[17], un tamaño del lote de entrenamiento de 8 y un tamaño de lote de validación de 16. Se han seleccionado estos tamaños debido a la disponibilidad de una GPU lo suficientemente potente, como es una L4 de Nvidia proporcionada por los servidores de Google Colab.

Todos los modelos se entrenarán durante 20 épocas para asegurar la consistencia de los valores obtenidos. Usando un 70% de los datos para el entrenamiento, un 20% para validación y un 10% de test. Durante el proceso, se estudiarán los valores de pérdida (*loss)* tanto en el conjunto de entrenamiento como el de validación. La pérdida en el conjunto de entrenamiento indicará como el modelo se ajusta a los datos utilizados para el entrenamiento, mientras que la pérdida en el conjunto de validación proporcionará una medida de la capacidad del modelo para generalizar a datos nuevos. Además, se analizará la precisión (*accuracy*) en ambos conjuntos para evaluar cuán de bien el modelo está clasificando ambos conjuntos de datos.

Para los 3 optimizadores se usarán como parámetros una épsilon de 1e-8 y un decaimiento del peso de 0,01. Estos parámetros han demostrado ser efectivos en varios estudios[31] sobre el rendimiento de los optimizadores Adam y sus variantes.

## **3.2.1 RoBERTa-large-bne**

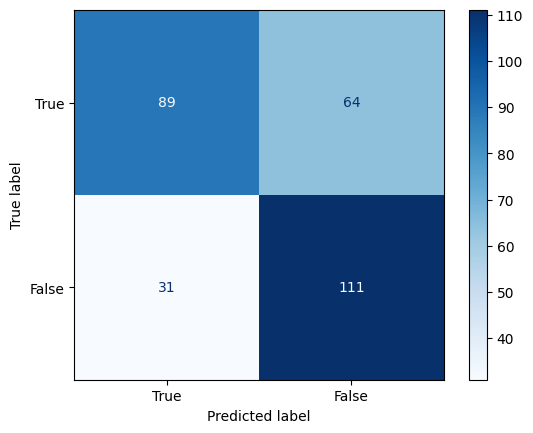
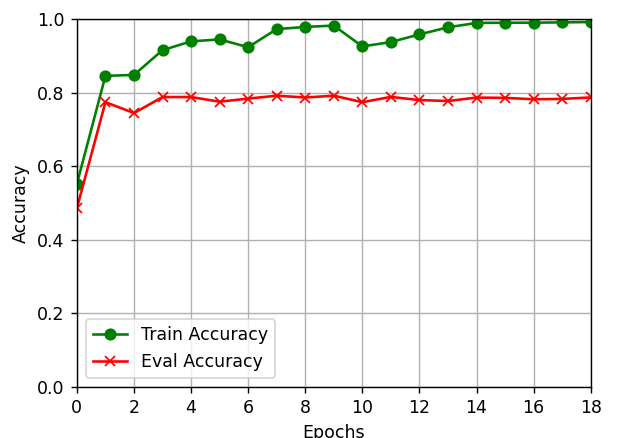
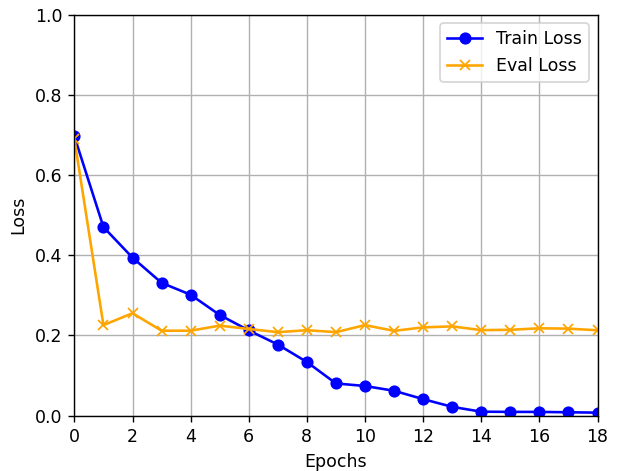
En las siguientes gráficas observamos como en las curvas de *accuracy* el optimizador AdamW es el que ha obtenido mayor F1-Score con un 84,13%. En cuanto a las curvas de pérdida notamos que los 3 optimizadores disminuyen a casi cero mientras que las pérdidas de evaluación se estabilizan, lo que se traduce en un sobreajuste del modelo sobre los datos.

|  |  |
| --- | --- |
| Modelo | F1-Score |
| RoBERTa\_large\_bne + Adam | 79.4 % |
| RoBERTa\_large\_bne + Adamax | 70.8 % |
| RoBERTa\_large\_bne + AdamW | 84.13 % |

**Tabla 2 - F1-Score RoBERTa-large-bne**

Las matrices de confusión muestran que Adam tiene un alto número de verdaderos positivos y falsos positivos, Adamax presenta menos falsos positivos, pero más falsos negativos y AdamW presenta un buen equilibrio entre todas las métricas.

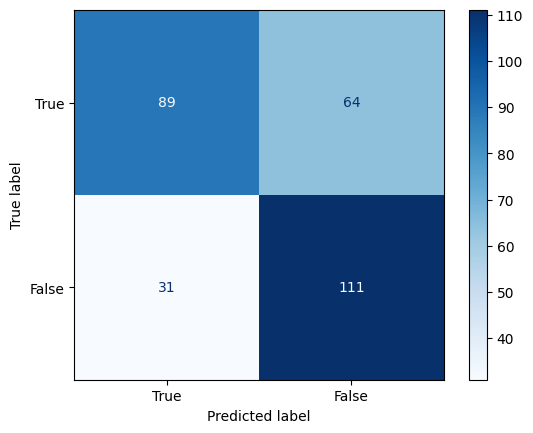
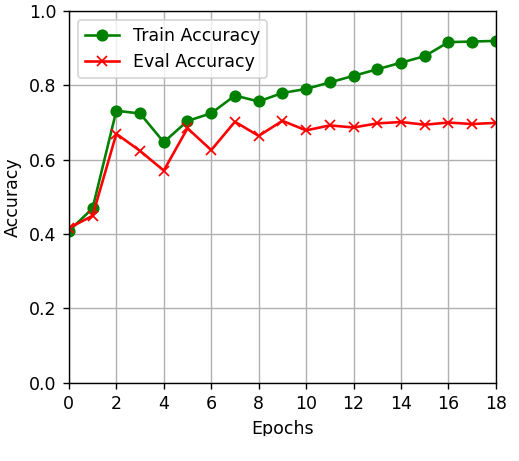
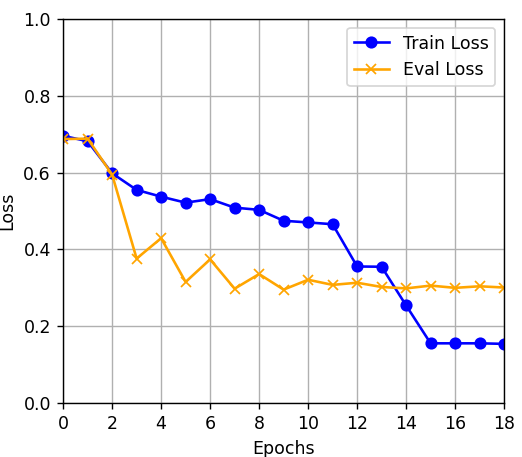
Matriz de confusión RoBERTa-large-bne + Adam



**Fig 15 - RoBERTa-large-bne + Adam**

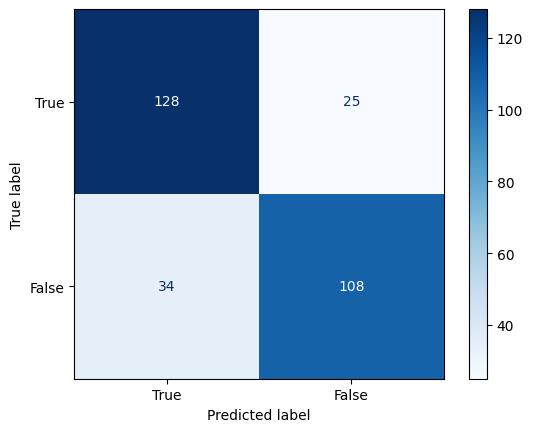
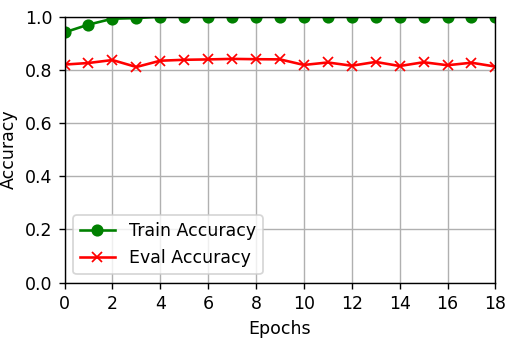
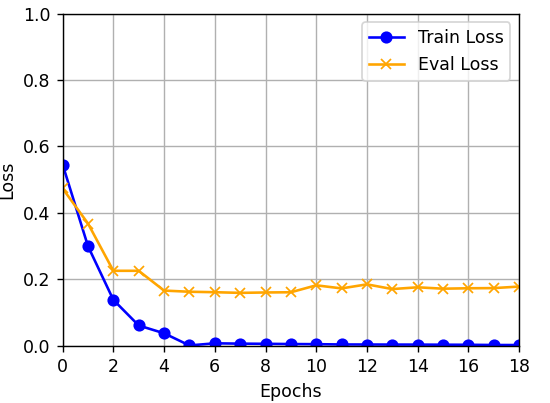
**Fig 16 - RoBERTa-large-bne + Adamax**

Matriz de confusión RoBERTa-large-bne + Adamax



**Fig 16 - RoBERTa-large-bne + Adamax**

Matriz de confusión RoBERTa-large-bne + AdamW



**Fig 17 - RoBERTa-large-bne + AdamW**

## **3.2.2 Xlm-RoBERTa-large**

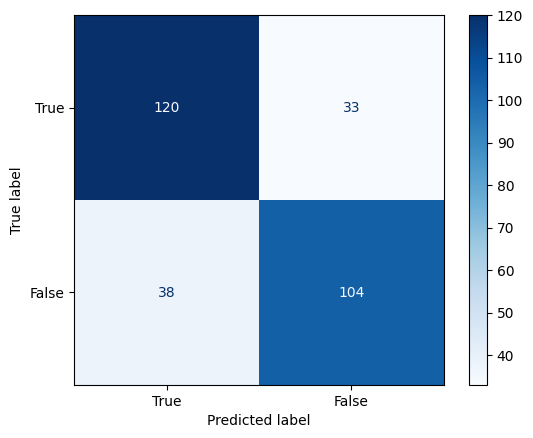
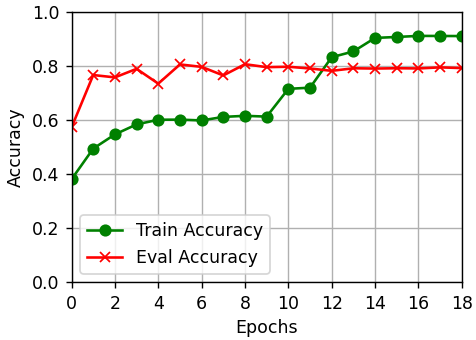
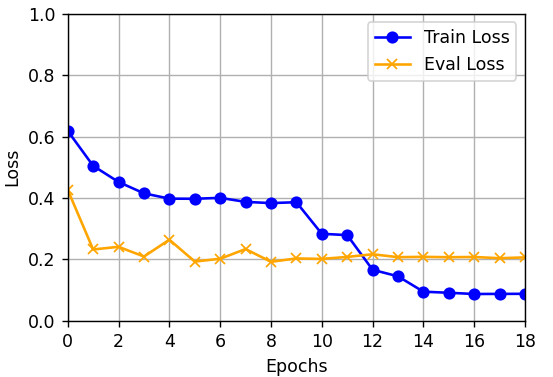
En el caso de Xlm-RoBERTa-large observamos que los 3 optimizadores mejoran sus respectivos F1-Score respecto a RoBERTa-large-bne, siendo AdamW otra vez el que mejor se adapta a este modelo obteniendo un F1-Score de 88,6%. Al igual que el modelo anterior, tanto la pérdida como el *accuracy* del conjunto de validación se estabilizan, pero el de entrenamiento sigue mejorando tanto en la pérdida como en el *accuracy,* lo que vuelve a mostrar un sobreajuste del modelo a los datos.

|  |  |
| --- | --- |
| Modelo | F1 |
| xlm\_RoBERTa\_large + Adam | 80.8 % |
| xlm\_RoBERTa\_large + Adamax | 74.1 % |
| xlm\_RoBERTa\_large + AdamW | 88.6 % |

**Tabla 3 - F1-Score Xlm-RoBERTa-large**

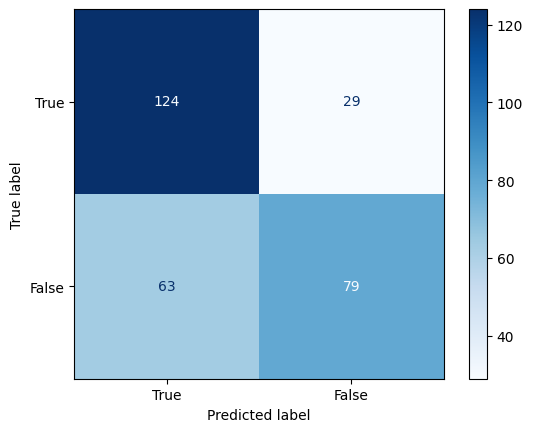
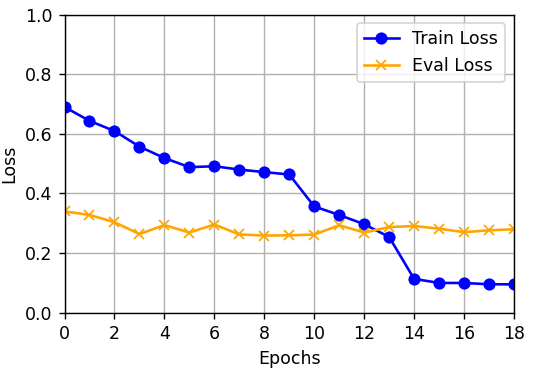
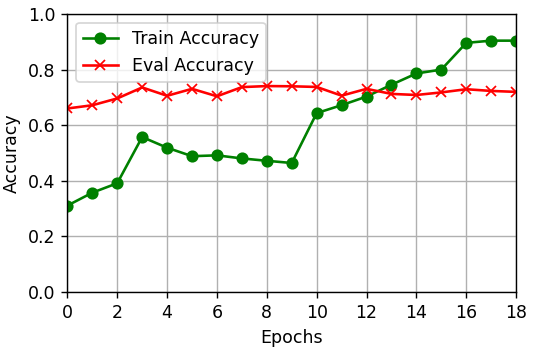
En cuanto a las matrices de confusión, Adamax logró un alto número de verdaderos positivos, pero su tasa de falsos negativos es bastante alta. El optimizador Adam presentó un rendimiento similar en términos de verdaderos posititos y negativos. Sin embargo, tuvo menos falsos positivos y negativos que Adamax. Por último, AdamW vuelve a ser el mejor en este caso, aunque tuvo algunos casos más de falsos positivos (32) que Adam y Adamax.

Matriz de confusión Xlm-RoBERTa-large + Adam



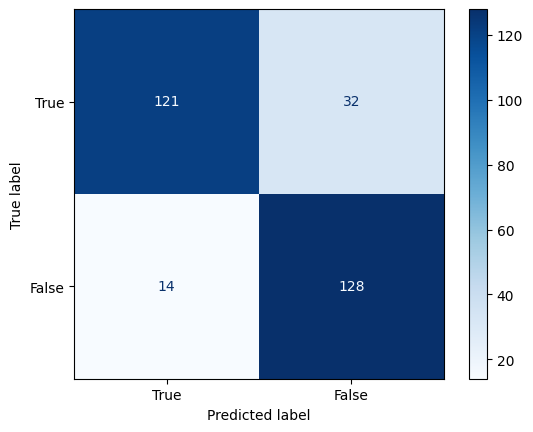
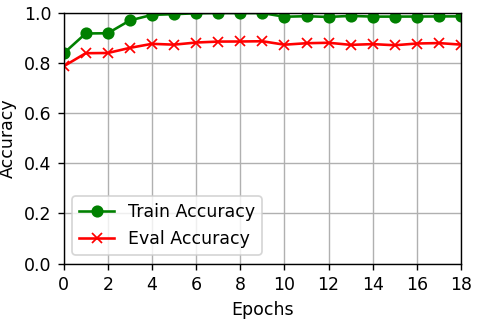
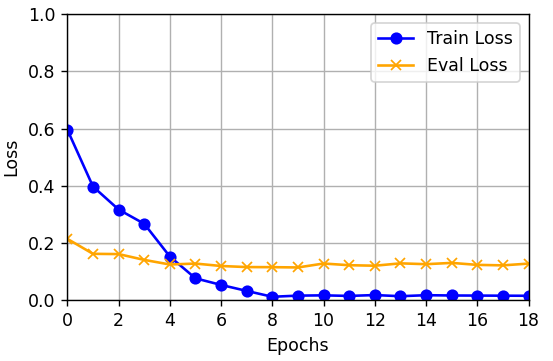
**Fig 18 - Xlm-RoBERTa-large + Adam**

Matriz de confusión Xlm-RoBERTa-large + Adamax



**Fig 19 - Xlm-RoBERTa-large + Adamax**

Matriz de confusión Xlm-RoBERTa-large + AdamW



**Fig 20 -** **Xlm-RoBERTa-large + AdamW**

## **3.2.3 RoBERTalex**

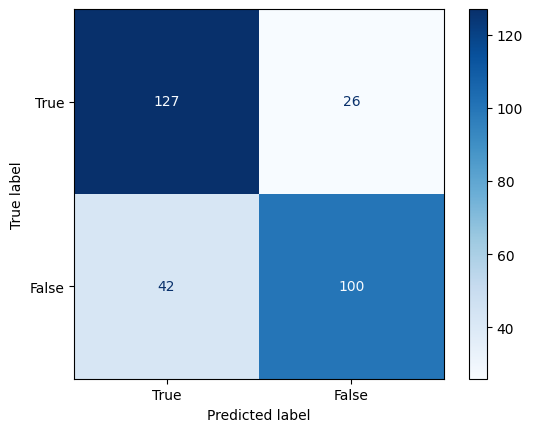
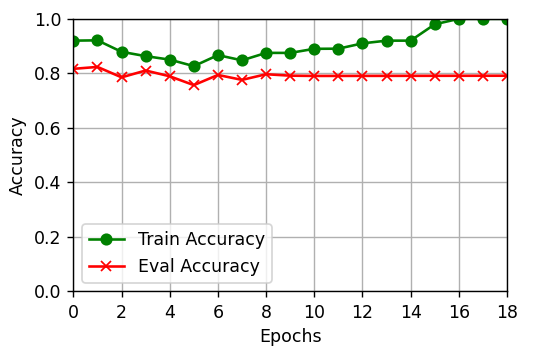
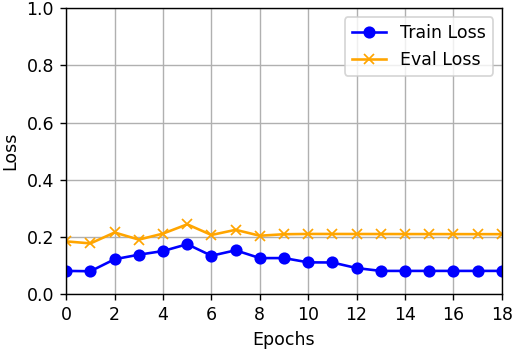
En las curvas de pérdida y *accuracy* observamos que el optimizador AdamW proporciona el mejor rendimiento con un F1-Score de 83,6%. Se puede notar que, al igual que ha pasado en los otros modelos, a partir de cierto punto las curvas del conjunto de validación se estabilizan, pero las de entramiento siguen mejorando, indicando nuevamente un sobreajuste del modelo a los datos.

|  |  |
| --- | --- |
| Modelo | F1 |
| RoBERTalex + Adam | 82.3 % |
| RoBERTalex + Adamax | 70.7 % |
| RoBERTalex + AdamW | 83.6 % |

**Tabla 4 - F1-Score de RoBERTalex**

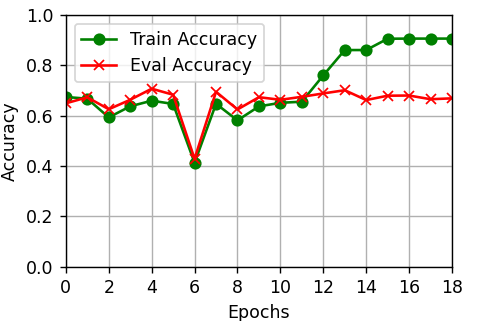
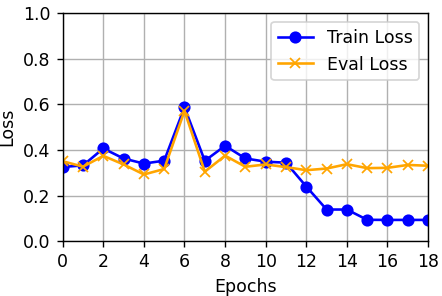
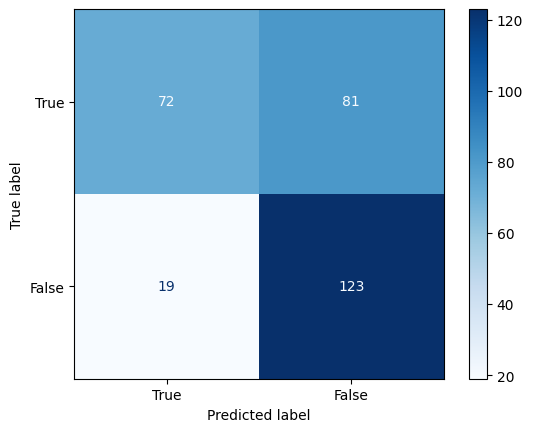
En las matrices de confusión también notamos que el optimizador AdamW presenta el mejor equilibrio, contando con una baja cantidad de falsos positivos y negativos. Por otro lado, Adam tiende a clasificar incorrectamente los negativos como positivos, mientras que Adamax tiene una tasa más alta de falsos negativos que de falsos positivos.

Matriz de confusión RoBERTalex + Adam



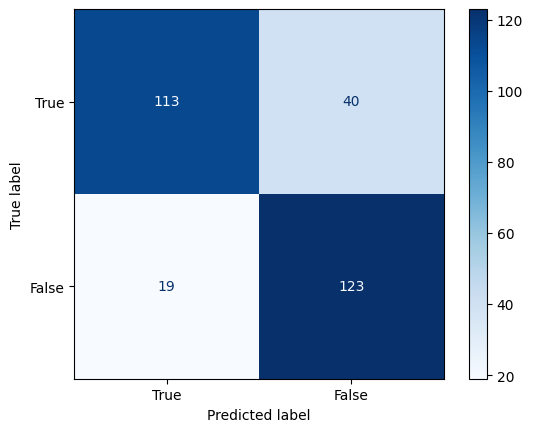
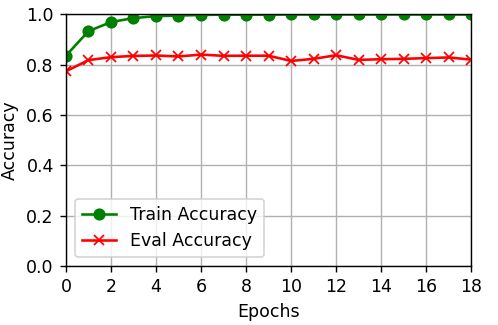
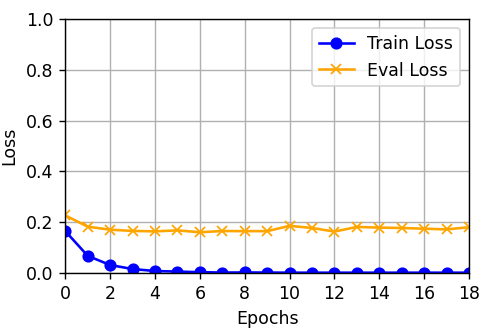
**Fig 21 - RoBERTalex + Adam**

Matriz de confusión RoBERTalex + Adamax



**Fig 23 - RoBERTalex + Adamax**

Matriz de confusión RoBERTalex + AdamW



**Fig 22 - RoBERTalex + AdamW**

## **3.2.4 Conclusiones**

Tras entrenar todos los modelos, obtenemos la siguiente clasificación donde la mejor combinación modelo + optimizador es la compuesta por Xlm-RoBERTa-large + AdamW con un F1-Score de 88,6%. Podemos observar una diferencia considerable de 4 puntos entre el primer clasificado y el segundo, marcando este la diferencia con el resto.

|  |  |
| --- | --- |
| Modelo | F1 |
| Xlm-RoBERTa-large + AdamW | 88,6 % |
| RoBERTa-large-bne + AdamW | 84,13 % |
| RoBERTalex + AdamW | 83,6% |
| RoBERTalex + Adam | 82,3% |
| Xlm-RoBERTa-large + Adam | 80,8% |
| RoBERTa-large-bne + Adam | 79,4% |
| Xlm-RoBERTa-large + Adamax | 74,1% |
| RoBERTa-large-bne + Adamax | 70,8% |
| RoBERTalex + Adamax | 70,7% |

AdamW se destaca como el mejor optimizador gracias a su capacidad para prevenir el sobreajuste mediante el decaimiento de peso, lo cual se refleja en un rendimiento superior en los tres primeros puestos.

En comparación, los modelos con dominio específicos como RoBERTalex no alcanzan el mismo nivel de desempeño que los modelos entrenados en corpus más diversos y extensos como Xlm-RoBERTa-large. Esto justifica la ventaja de utilizar modelos grandes y generalizados para tareas de clasificación complejas, donde capturar una mayor diversidad de patrones y contextos es crucial.

Estos resultados sugieren que, para maximizar el rendimiento en futuro estudios, es aconsejable optar por modelos grandes como Xlm-RoBERTa-large y los optimizadores avanzados como AdamW, siempre considerando cuidadosamente los costos computacionales y los tiempos de entrenamiento asociados. Este enfoque puede proporcionar resultados significativamente mejores en términos de métricas, aprovechando la capacidad robusta y la precisión de estos modelos.

Sin embargo, cuando los recursos son limitados, una alternativa viable es considerar el modelo clasificado en segundo lugar, RoBERTa-large-bne + AdamW y el tecercer lugar, RoBERTalex + AdamW. RoBERTa-large-bne, con un F1-Score de 84,13%, ofrece una opción viable para aquellos con restricciones de recursos. Aunque este modelo es menos complejo y más pequeño en escala en comparación con el primer clasificado, la diferencia de rendimiento de 4 puntos en el F1-Score se compensa con su tamaño.

Por otro lado, RoBERTalex, aunque ocupe el tercer puesto en la clasificación con un F1-Score de 83,6%, tiene una diferencia de puntuación muy pequeña en comparación con el segundo lugar, a pesar de la notable diferencia de tamaños entre ambos modelos. Esta opción es recomendable considerar en problemas dentro del ámbito jurídico, dado que el modelo está preentrenado en ese dominio. Sin embargo, si tratamos problemas de ámbito general, la mejor opción sería RoBERTa-large-bne al no tener un dominio de datos específico.

## **Referencias**

|  |  |
| --- | --- |
| [1] | J. Figueras, P. Enériz y A. Solà, «¿Por qué surgen ‘fake news’ en los conflictos armados?,» *elPeriódico,* 26 Octubre 2023. |
| [2] | «Cambridge Analytica – Data drives all that we do», https://web.archive.org/web/20171113180809/https://cambridgeanalytica.org. |
| [3] | A. J. ARRATIBEL, «El peligroso impacto de los bulos en las vidas de los hispanohablantes de Estados Unidos,» *EL PAÍS,* 26 Diciembre 2023. |
| [4] | C. Rodríguez Pérez, «No diga fake news, di desinformación: una revisión sobre el fenómeno de las noticias falsas y sus implicaciones,» *Dialnet,* 2019. |
| [5] | «Fundéu BBVA: "noticias falsas" o "falseadas", mejor que "fake news",» *LA VANGUARDIA,* 28 09 2017. |
| [6] | «Evaluating Online Information: Types of Misinformation,» *IOWA University Libraries.* |
| [7] | «Cambridge Dictionary,» [En línea]. Available: https://dictionary.cambridge.org/us/dictionary/english/hate-speech. |
| [8] | «Misinformation and disinformation».*American Psychological Association.* |
| [9] | J. Holdsworth, «IBM.com,» 6 6 2024. https://www.ibm.com/topics/natural-language-processing. |
| [10] | M. Awad y R. Khanna, Efficient Learning Machines. Theories, Concepts, and Applications for Engineers and System Designers. |
| [11] | J. Barnard, «IBM.com,» 23 1 2024. https://www.ibm.com/topics/word-embeddings#:~:text=One%20example%20of%20frequency%2Dbased,significant%20for%20a%20particular%20document.. |
| [12] | T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cisac y T. Rault, «Transformers: State-of-the-Art Natural Language Processing». |
| [13] | M. Oren, M. Hassid, Y. Adi y R. Schwartz, «Transformers are Multi-State RNNs,» 2024. |
| [14] | A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones y A. N. Gomez, «Attention Is All You Need,» 2017. |
| [15] | X. Z. S. R. a. J. S. Kaiming He, «Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778,» 2016. |
| [16] | M. taher y J. Camacho-Collados, «Embeddings in natural language processing: Theory and advnaces in vector representation of meaning. Synthesis Lectures on Human Language Technologies, 13(4):1–175,» 2020. |
| [17] | Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis y L. Zettlemoyer, «Roberta: A robustly optimized BERT pretraining approach.,» CoRR,abs/1907.11692, 2019. |
| [18] | J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, «Bert: Pre-training of deep bidirectional transformers for language understanding.,» arXiv preprint arXiv:1810.04805, 2018. |
| [19] | W. Yang Wang, «”liar, liar pants on fire”: A new benchmark dataset for fake news detection.,» arXiv preprint arXiv:1705.00648, 2017. |
| [20] | K. Shu, D. Mahudeswaran, D. Lee y H. 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. |
| [21] | P. Patwa, M. Bhardwaj, V. Guptha, G. Kumari, S. Sharma, S. PYKL, A. Das, A. Ekbal, S. Akhtar y T. 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. |
| [22] | J.-P. Posadas-Durán, H. Gómez-Adorno, G. Sidorov y J. J. Moreno Esobar, «Detection of fake news in a new corpus for the spanish language.,» Journal of Intelligent & Fuzzy Systems,36(5):4869–4876, 2019. |
| [23] | B. Riedel, I. Augenstein, G. P Spithourakis y S. Riedel, «A simple but tough-to-beat baseline for the fake news challenge stance detection task.,» arXiv preprint arXiv:1707.03264, 2017. |
| [24] | A. Vlachos y S. 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. |
| [25] | https://sites.google.com/view/iberlef2022. |
| [26] | «Wikipedia», https://en.wikipedia.org/wiki/N-gram. |
| [27] | D. Bergmann, «IBM.com,» 12 1 2023. https://www.ibm.com/es-es/topics/fine-tuning. |
| [28] | https://huggingface.co/PlanTL-GOB-ES/roberta-large-bne. |
| [29] | https://huggingface.co/FacebookAI/xlm-roberta-large. |
| [30] | https://huggingface.co/PlanTL-GOB-ES/RoBERTalex. |
| [31] | D. P.Kingma y J. Ba, «Adam: A Method for Stochastic Optimization,» 2014. [En línea]. Available: https://doi.org/10.48550/arXiv.1412.6980. |
| [32] | I. Loshchilov y F. Hutter, «Decoupled Weight Decay Regularization,» 2017. [En línea]. Available: https://arxiv.org/abs/1711.05101v3. |
| [33] | M. Bhardwaj, V. Guptha, G. Kumari, S. Sharma, S. PYKL, A. Das, A. Ekbal, S. Akhtar y T. 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. |