Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Modelos de Transformers para la Clasificación de Texto

Autor(a): Guillem García Subies
Tutor(a): Francisco Serradilla García

Madrid, Julio 2021

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Título: Modelos de Transformers para la Clasificación de Texto

Julio 2021

Autor(a): Guillem García Subies
Tutor(a): Francisco Serradilla García
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Agradecimientos

A Francisco Serradilla, por aceptar dirigir el presente trabajo y por sus consejos durante estos meses.

Al Instituto de Ingeniería del Conocimiento, mi participación en el Máster no hubiera sido posible sin el apoyo del IIC, que me permitió dedicar parte de mis horas al estudio del mismo. Asimismo, tanto el IIC como el Grupo de Aprendizaje Automático de la UAM me han proporcionado los recursos computacionales para realizar los experimentos necesarios en este trabajo.

To everyone at huggingface, for creating and maintaining the amazing library of transformers, and to the whole open source community for their disinterested contributions to the open science. Without your contributions, none of this work would have been possible.

A mi *Sensei*, Álvaro Barbero, por confiar en mí desde el primer momento y por todas las cosas que me has enseñado a lo largo de estos años. Me has ayudado mucho a crecer tanto personal como profesionalmente y sin tu ayuda no hubiera llegado tan lejos.

A Paula, es difícil expresar con palabras todo lo que significas para mí y cuánto me has ayudado estos últimos años. Gracias por aguantarme y apoyarme incondicionalmente, tanto en los buenos momentos, como en los malos.

Als meus pares, Amparo i José, per tota la paciència que heu tingut tots estos anys, per fer l'esforç de permetre'm estudiar el que jo volia, per estar sempre ahí i suportar-me sempre sense importar el què. Gràcies per l'educació que m'heu proporcionat, pels consells que m'heu donat, per cada xicotet moment que hem compartit, gràcies per tot.

A todos vosotros, gracias.

Resumen

Este trabajo tiene como objetivo el estudio minucioso del estado del arte de la clasificación de textos usando modelos del lenguaje basados en Transformers para aplicar las técnicas más novedosas a problemas de relevancia social.

Para ello, primero se estudia el estado del arte tanto de los modelos del lenguaje, las técnicas de clasificación de texto y análisis del sentimiento, y las técnicas de *Data Augmentation* para Procesamiento del Lenguaje Natural (PLN).

A continuación se describe la participación en las tareas compartidas de la edición de 2021 del *Iberian Languages Evaluation Forum* (IberLeF), enmarcado en el congreso anual de la Sociedad Española de Procesamiento del Lenguaje Natural (SEPLN). En esta participación se exploran las mejores técnicas estudiadas, centrándose en técnicas de bajos recursos computacionales, y se obtienen resultados del estado del arte. Asimismo, se propone una nueva técnica que mejora el estado del arte en los corpus plurilingües.

Los resultados obtenidos en este trabajo, son de alta trascendencia científica ya que se aplican a problemas muy relevantes dado el contexto de cambio social actual, al énfasis en el uso del software libre y modelos eficientes computacionalmente, en contra de la tendencia actual de crear modelos cada vez más grandes y caros, solamente asumibles para las grandes multinacionales.

Fruto de este trabajo, se han publicado tres *papers* en solitario y un cuarto como coautor mostrando los resultados de las competiciones.

Abstract

The main objective of this work is a thorough study of the state-of-the-art in text classification using Transformer-based language models in order to apply the most novel techniques to problems of social relevance.

To do this, we first study the state-of-the-art of language models, text classification techniques and sentiment analysis, and Data Augmentation techniques for Natural Language Processing (NLP).

Next we describe our participation in the shared tasks of the 2021 edition of the Iberian Languages Evaluation Forum (IberLeF), framed within the annual congress of the Spanish Society for Natural Language Processing (SEPLN). In this participation, the best studied techniques are explored, focusing on techniques that require low computational resources, and state-of-the-art results are obtained. In addition, a new technique is proposed that improves the state-of-the-art in multilingual corpus.

The results obtained in this work are of high scientific significance since they are applied to very relevant problems given the context of current social change, the emphasis on the use of open source software and computationally efficient models, contrary to the current trend of creating increasingly large and expensive models, only affordable for large companies.

As a result of this work, three papers have been published and a fourth one as a co-author showing the results of the competitions.

Tabla de contenidos

Índice de cuadros

Capítulo 1

Introducción

El Procesamiento del Lenguaje Natural (en adelante, PLN) es una rama multidisciplinar de la Inteligencia Artificial que se dedica a transmitir los conocimientos humanos del lenguaje natural a las máquinas para que se puedan realizar una gran cantidad de tareas como la recuperación de información [1]. resúmenes automáticos [2], traducción automática [3], reconocimiento de voz [4], corrección de errores gramaticales [5], sistemas de pregunta-respuesta [6], reconocimiento de entidades nombradas [7] y clasificación de texto [8], entre otras [9] [10].

En los últimos años esta disciplina ha recibido mucha atención tanto de la academia como de la industria [11], sobre todo desde la aparición de los modelos de atención como el Transformer [12], que revolucionó la forma de crear modelos del lenguaje y el estado del arte. Desde su aparición, y gracias a librerías como huggingface/transformers [11], se ha conseguido (en parte) democratizar el PLN de calidad para todos los investigadores. Pese a ello, aun existen algunos problemas que complican la completa democratización de estas técnicas al público general, como se verá durante este trabajo.

Durante este trabajo se va a hacer hincapié en las técnicas de PLN para resolver problemas de clasificación de texto. Concretamente, se usarán modelos neuronales del lenguaje basados en la arquitectura Transformers y en BERT [13] para dicho fin.

Asimismo, también se quiere hacer un énfasis en los modelos del lenguaje para la lengua española. Pese a ser el segundo idioma con más hablantes nativos y el tercero con más contenido en internet [14], carece de los recursos lingüísticos y computacionales existentes para otros idiomas como el inglés, idioma para el que se crean la gran mayoría de modelos del lenguaje en el estado del arte.

1.1. Motivación

La principal motivación de este trabajo es la continuación del trabajo empezado en los Trabajos de Fin de Grado del autor; “Estudio teórico sobre modelos de secuencias con redes neuronales recurrentes para la generación de texto” [15] y “Estudio práctico sobre modelos de secuencias con redes neuronales recurrentes para la generación de texto” [16]. En ellos se estudia exhaustivamente tanto los modelos del lenguaje neuronales que eran estado el arte en su momento, las redes LSTM, como las mejores implementaciones software de los mismos. Finalmente, se deja como trabajo

futuro seguir con el estado del arte y los, entonces incipientes modelos basados en el Transformer y en BERT [13]. Dicho trabajo pues, se realizará en el presente Trabajo de Fin de Máster.

Dada la gran cantidad de ramas que tiene el PLN, y para concretar, este trabajo se centrará en la clasificación de texto y el análisis del sentimiento. El principal motivo de esta decisión es la gran cantidad de atención que ha recibido esta disciplina, tanto nacional [17] [18], como internacionalmente [19] [20] [21] en los últimos años.

Se prestará especial atención a los modelos para la lengua española ya que solo existe un modelo del lenguaje general para la misma [22] y otro concreto para tweets [23].

Finalmente, dada la naturaleza cerrada de algunos de los nuevos modelos del lenguaje [24] y que solo las grandes multinacionales como Facebook [25] [26], Google [27] [28] o Microsoft [29] [30] tienen presupuesto suficiente para crearlos (cuanto más grandes son los modelos, mejores resultados obtienen [31]), se hará hincapié en modelos de tamaño más modesto y abiertos.

1.2. Objetivos

El objetivo principal de este trabajo es obtener técnicas de clasificación de texto que alcancen el estado del arte, pero que no requieran de una cantidad abusiva de recursos computacionales y sean de código abierto. Para alcanzar dicho objetivo, se participará en varias de las competiciones de foro *Iberian Languages Evaluation Forum* (en adelante, IberLeF).

El foro anual IberLef, creado en 2019 por la Sociedad Española de Procesamiento del Lenguaje Natural (SEPLN), es una serie de competiciones de PLN organizadas por universidades iberoamericanas. Dichas competiciones tienen como objetivo dar pasos adelante en el estado del arte de PLN en la escena iberoamericana. De esta forma, se fija como objetivo secundario de este trabajo la obtención de buenos resultados en dichas competiciones.

Finalmente, se fija otro objetivo de estudio y ampliación de los modelos del lenguaje para el español. Esta última tarea es desafiante dado que el español es una lengua más compleja que el inglés y que existe una menor cantidad de corpus disponibles. Sin embargo, los incentivos para desarrollar dichos modelos son grandes; como se ha mencionado anteriormente, el español es una de las lenguas más habladas en el mundo por lo que existe gran interés académico e industrial en desarrollar dichos recursos.

1.3. Estructura del documento

El estudio se organizará de la siguiente manera: En el siguiente Capítulo se llevará a cabo un estudio del estado del arte del Procesamiento del Lenguaje Natural en general y, en concreto, de los métodos de clasificación de texto y de *Data Augmentation*. En el Capítulo 3 se detallarán los métodos elegidos para cada dataset, y los experimentos realizados. Después, en el Capítulo 4 se analizaran en profundidad los resultados obtenidos en dichos experimentos. Finalmente, en el Capítulo 5 se expondrán las principales conclusiones de este estudio y se propondrán futuras líneas de trabajo en continuación a lo aquí expuesto y alineadas con las últimas ideas del estado del arte.

En el los Anexos se expondrán los *papers* publicados durante el transcurso de este trabajo y las aceptaciones de los mismos.

Capítulo 2

Estado del arte

2.1. Introducción

En este capítulo se repasarán las técnicas más novedosas relacionadas con el Procesamiento del Lenguaje Natural y su aplicación a la clasificación de textos.

Como las máquinas no pueden trabajar directamente con texto [32] [33], se requiere de algún tipo de representación vectorial apropiada para el texto, de ahí surge la idea de modelo del lenguaje.

La mayoría de los modelos del lenguaje clásicos funcionan aprendiendo la distribución de probabilidades de cada palabra en el lenguaje para poder "adivinar" la siguiente palabra más probable. Para aprender sobre el lenguaje, los modelos son entrenados con grandes corpus como OSCAR [34] (cuanto más grandes, mejor [35]). Para cada línea del corpus, una palabra o varias son enmascaradas y tratadas como una etiqueta para que el modelo las prediga. De esta manera, con solo datos sin etiquetar, el modelo puede aprender el lenguaje de una forma auto-supervisada.

Una vez el modelo del lenguaje está entrenado en el idioma elegido (preentrenamiento), la última capa de la red neuronal puede ser cambiada para que solucione distintos tipos de problemas (*fine-tuning*). Este proceso se llama *transfer learning*.

Bengio et al. [36] en el 2003 crearon el primer modelo (también conocido como modelo de n-gramas) del lenguaje usando redes neuronales y ya predijeron cual sería uno de los mayores problemas de los modelos del lenguajes; la maldición de la dimensionalidad [37] [38] [39] [40]. Gran parte de los modelos y mejoras presentados en este capítulo se centran en reducir el efecto negativo de este problema al mismo tiempo que son capaces de tratar con datos cada vez más complejos.

A continuación se realizará un rápido resumen del estado del arte en el campo del procesamiento de texto natural para seguir explicando el estado del arte de la clasificación de textos automática.

2.2. Selección de los artículos

Los artículos de este resumen del estado del arte, aparte de haberse buscado en las típicas plataformas, se han buscado en arXiv y en el Hub de modelos de Huggingface.

Esto se debe principalmente a que la gran mayoría de estos artículos son del 2019 en adelante.

Asimismo, también se ha usado la web *paperswithcode.com* ya que recopila los mejores *papers* de inteligencia artificial de forma abierta y clasifica los problemas y *datasets* de manera que sea muy fácil ver qué modelos y qué *papers* son los que mejor lo hacen en cada momento.

2.2.1. Hub de modelos de Huggingface

Se ha convertido en un standard *de facto* el publicar los nuevos modelos del lenguaje de forma abierta en el Hub de modelos de huggingface/transfromers [11], con más de nueve mil modelos a fecha de mayo del 2021. Por lo tanto, es una muy buena fuente para buscar modelos con gran impacto, tanto en la academia como en la industria.

2.3. Modelos del lenguaje

2.3.1. *Embeddings* de texo

Los *embeddings* son un tipo de representación vectorial de un texto donde las palabras con significado similar, tienen vectores similares [41] y cada palabra es representada por un único vector $\in \mathbb{R}$ [42]. Estos *embeddings* suelen aprender solo la semántica de las palabras, no la sintaxis ni la gramática [43]. Los *embeddings* de palabras también pueden llamarse espacio vectorial semántico [44] o modelo semántico [45] [46].

Los primeros intentos más relevantes de usar las redes neuronales en el PLN son los modelos de *embeddings* [47] ya que la representación vectorial de las palabras puede ser mapeada fácilmente en una red neuronal [48]. Su propósito es reemplazar los modelos clásicos como la bolsa de palabras [49] o Tf-idf [50] con *embeddings* más inteligentes que aprenden sobre una palabra en base a las palabras que la rodean y no solo usando la frecuencia de las palabras en el texto, como hacían los modelos antiguos.

Los modelos más relevantes [51] que usan esta técnica son GLoVe [52], Word2Vec [53] y FastText [47] [54] [55].

Para aprender, estos modelos usan las técnicas auto supervisadas de predicción de una palabra enmascarada en mitad de la frase y predicción de las palabras enmascaradas alrededor de una palabra fija dentro de una ventana (también conocido como *skip-gram*). También existen versiones de *skip-gram* aplicadas a frases enteras [56] para conseguir embeddings de frases en vez de palabras.

A través de este proceso, se obtienen *embeddings* para cada palabra del lenguaje. Con estos *embeddings* se pueden crear representaciones vectoriales de los textos para que estén listos para ser utilizados en cualquier tipo de modelo de aprendizaje automático.

Este tipo de *embeddings* se consideran estáticos [57] ya que, una vez creados, no cambian aunque cambie el contexto. Esta característica hace que sean menos potentes que otras alternativas, sin embargo, estos tipos de *embeddings* siguen siendo útiles para ciertos tipos de problemas más simples o para la creación de *baselines*.

2.3.1.1. Word2Vec

Word2Vec [44] [53] es una red neuronal de dos capas en la que solo se dispone de una capa oculta y una capa de salida. Su función es situar cada palabra del vocabulario en un espacio 300-dimensional intentando juntar las palabras que tienen parecidos significados semánticos. Con estos vectores, se puede calcular la similaridad coseno entre palabras.

También se puede calcular los vectores de las palabras de un texto y hacer su media para tener una representación semántica de un texto. Con estas representaciones (y con etiquetas para las mismas), se podría entrenar un modelo de aprendizaje automático para resolver tareas concretas.

2.3.1.2. GLoVe

GLoVe nace para intentar solucionar el principal problema de Word2Vec, que no aprende el contexto global de cada palabra [52]. En el caso de GLoVe, los embeddings aprendidos se generan a partir del contexto local y las estadísticas globales de cada palabra.

Para cada texto, GLoVe calcula la matriz de co-ocurrencias en la que aparece el número de veces que cada palabra aparece en ese mismo texto. Después es entrenado usando esa matriz y la información local similar a Word2Vec.

2.3.1.3. FastText

FastText [47] [54] [55] representa las palabras con vectores construidos en base a las sumas de las representaciones aprendidas de los n-gramas que contiene. Esto es muy útil cuando aparecen palabras que el modelo no ha visto. En otros modelos no se puede obtener una representación vectorial de una palabra que el modelo no haya visto durante el entrenamiento. Sin embargo, FastText puede construir una representación de esa palabra nueva usando las representaciones que sí ha aprendido de sus n-gramas.

2.3.2. *Embeddings* contextuales

Los embeddings contextuales se generan de forma dinámica, esto significa que una palabra puede tener diferentes representaciones dependiendo de su contexto. Esto se consigue añadiendo alguna información de los embeddings del resto de palabras en el texto a la palabra en cuestión. El modelo más notable de embeddings contextuales es ELMo [58].

2.3.2.1. ELMo

ELMo (Embeddings from Language Models, por sus siglas en inglés) [58] es un modelo del lenguaje de redes neuronales profundas y bidireccionales.

El hecho de que sea bidireccional ayuda a leer el texto tanto de izquierda a derecha como de derecha a izquierda para entender mejor esa palabra dentro de la frase y las redes profundas, en contraposición a los modelos que se han descrito anteriormente, le ayudan a conseguir mejores representaciones de las palabras.

Las anteriores características, junto a que sus embeddings no son estáticos y se generan de manera dinámica en función del contexto de cada palabra, consiguen que ELMo obtenga muy buenos resultados en los *benchmarks*.

2.3.3. Modelos de atención

Los modelos *sequence-to-sequence* [59] toman una secuencia de objetos y devuelven otra secuencia de objetos. Esto se utilizaba mucho en traducción automática pero se ha expandido a otro tipo de modelos. La estructura de estos modelos está compuesta por un codificados que genera el contexto y un decodificador que lo coge y genera la salida [60].

En base a la estructura codificador-decodificador, los modelos de atención [61] [62] permiten al decodificador centrarse en las partes importantes de los datos. Esto se consigue pasándole todos los estados ocultos al codificador, no solo el último.

2.3.4. El modelo Transformer

El modelo Transformer [12] intenta juntar las mejores ideas del estado del arte y añade de algunas mejoras nuevas. Los cambios más notables son el uso de la autoatención multicabeza que permite un mejor comportamiento en las tareas de tipo Winograd [63].

Esto significa que el modelo consigue un entendimiento más profundo de la gramática del lenguaje y no solo de la semántica de palabras concretas. Casi todos los modelos que llegan al estado del arte en tareas de PLN están inspirados, de una forma u otra en el Transformer.

También cabe que destacar que los modelos basados en el Transformer son mucho más rápidos que sus alternativas, como pueden ser las redes neuronales convolucionales autoregresivas [64] [65]. Esto se debe al extenso uso de la multiplicación de matrices en las capas de autoatención, lo que permite al Transformer funcionar mucho más rápido en las GPUs.

2.3.5. Modelos basados en el Transformer

Se ham creado miles de modelos inspirados en el Transformer original, estos modelos suelen llamarse Transformers [11]. El más conocido y que sigue usándose hoy en día, pese a haber sido creado algunos años atrás, es BERT (Bidirectional Encoder Representations from Transformers, por sus siglas en inglés) [13].

2.3.5.1. BERT

BERT [13] es un modelo que usa solo la parte del codificador y con capas bidireccionales. Su mayor novedad es el modelo del lenguaje enmascarado (MLM por sus siglas en inglés, *masked language model*). El MLM es una forma de entrenar un modelo autosupervisado que enmascara aleatoriamente un porcentaje de las palabras en el texto, no solo una o las que estén dentro de la ventana. Esto ayuda al modelo a aprender, no solo lo que precede a una palabra, sino también lo que va detrás de ella, dándole un mejor entendimiento general del lenguaje.

En la segunda parte del preentrenamiento de un modelo BERT, se pasa del MLM a la predicción de la siguiente frase. Esta técnica también autosupervisada se basa en dar dos frases a un modelo y que el modelo aprenda si la segunda frase va después de la primera en el texto total.

2.3.5.2. GPT

Otra familia muy significativa de modelos de Transformers son los GPT (Generative Pretrained Transformer, por sus siglas en inglés) [66] [24] [31]. Estos modelos aprenden usando un modelado del lenguaje autorregresivo, al igual que el modelo original de Bengio et al. [36].

Este aprendizaje se basa en dar como *input* al modelo una palabra y hacer que prediga la siguiente (como se conoce la siguiente, se puede calcular el error). En el siguiente paso se hace lo mismo con la palabra que viene a continuación y se repite hasta haberlo hecho con todas las palabras del texto.

Aparte de eso, estos modelos han demostrado que cuantos más datos se usan y más grandes son los modelos (la única diferencia entre los tres GPT es el gran aumento de tamaño), mejor es el resultado [24]. De hecho, uno de los mayores hitos de GPT-3 es su capacidad de hacer tareas concretas sin *fine-tuning*, lo que es llamado *zero-shot learning* [31]. Asimismo, GPT-3 obtiene unos muy buenos resultados en tareas complicadas como las de WinoGrande [67] (una extensión de Winograd), demostrando una profunda comprensión de la gramática y sintaxis de la lengua escrita.

2.3.5.3. ALBERT

ALBERT (A Lite BERT, de sus siglas en inglés) [68] intenta solucionar el problema del coste computacional de los grandes modelos del lenguaje. ALBERT usa la misma arquitectura que BERT, salvo que utiliza dos técnicas para disminuir el número de parámetros de la red neuronal.

La primera es la parametrización factorizada de los *embeddings* que divide las matrices de palabras en matrices más pequeñas y la segunda son los parámetros compartidos a través de las capas, esto previene a la red de generar un número excesivo de parámetros a medida que la complejidad aumenta. De esta manera se obtienen resultados similares a los de BERT con un tiempo de entrenamiento significativamente menor.

Otra diferencia que existe entre ALBERT y BERT es que sustituye la predicción de la siguiente frase con la predicción del orden de frases en la que se presentan dos frases consecutivas, que pueden estar en orden o no, y el modelo debe decir si están ordenadas o las han desordenado.

2.3.5.4. BART

BART [69] es un modelo basado en la arquitectura original del Transformer que se centra en aprender a quitar el ruido del texto.

Este ruido lo añaden con varias estrategias de entrenamiento:

- MLM: De la misma forma que en BERT

- Eliminación de tokens: Se eliminan aleatoriamente algunos tokens y el modelo debe decidir en qué posiciones faltan los tokens.
- Rellenado de texto: Inspirado en SpanBERT [70], funciona igual que el MLM solo que las máscaras que se quitan pueden ser de varias palabras, no solo una como en el MLM:
- Desordenado de frases: Se separa en frases un documento y se desordenan. El modelo debe aprender a reordenarlas.
- Rotación de documentos: Se elige un token aleatorio del texto y se rotan todos los tokens de manera que el token elegido es el primero del texto. El modelo debe aprender donde empieza el texto originalmente.

El modelo de BART consigue resultados del estado el arte en generación de texto y resúmenes automáticos.

2.3.5.5. PEGASUS

PEGASUS [71] centra su tarea de aprendizaje en eliminar frases de un texto y aprender a generarlas. Como se puede observar, esta tarea de preentrenamiento es bastante similar a lo que podría ser una tarea de resúmenes automáticos. Los autores del *paper* teorizan que, de esta manera, cuando se vaya a hacer *fine-tuning* en esa tarea en concreto, el modelo funcionará muy bien.

Los resultados que obtiene son muy destacables en varias tareas se *sequente-to-sequence*.

2.3.5.6. DeBERTa

DeBERTa (Decoding-enhanced BERT with disentangled attention en inglés) [72] consigue grandes resultados en el estado del arte, incluso mejora los resultados humanos en el dataset SuperGLUE [73]. Aparte, DeBERTa propone una serie de mejoras que consiguen hacer que el modelo sea más eficiente a la hora de entrenar en comparación a otros modelos, lo que resulta en un mejor modelo con los mismos recursos. Las principales novedades de DeBERTa son su mecanismo de atención desenrollada y un decodificador mejorado.

La atención desenrollada hace que la representación de cada palabra sea con dos vectores, uno para el contenido de la palabra y otro para su posición. Asimismo, los pesos de la atención entre las palabras se calculan usando matrices desenrolladas sobre sus contenidos y las posiciones relativas.

Por otra parte, la capa decodificadora recibe como *input* extra las posiciones absolutas de las palabras y usa un método adversarial para hacer el entrenamiento en las tareas de *fine-tuning* y mejorar la generalización.

2.3.5.7. Otros modelos relevantes

Dada la gran cantidad de modelos existentes y, para no desviar este resumen del objetivo principal, a continuación se listan algunos modelos que han mostrado ser relevantes en alguno de sus aspectos: XLM [74], XLNet [75], RoBERTa [76], DistilBERT [77], XLM-RoBERTa [78]. ELECTRA [79] y T5 [35].

2.3.5.8. Modelos multimodales

En los últimos años también se está experimentado con modelos multimodales, que pueden aprender de distintos tipos de datos al mismo tiempo. De esta forma pueden, por ejemplo, describir una imagen con lenguaje natural o *vice versa*. Como ejemplos de esos modelos, se pueden destacar ViLBERT [80] (imágenes-lenguaje), VideoBERT (video-lenguaje) [81] o Wav2Vec [82] (audio-lenguaje).

2.3.6. Modelos del lenguaje en otros idiomas

La mayoría de modelos del lenguaje que se han explicado están solo en inglés salvo algunas excepciones como la versión plurilingüe de BERT [13] o los embeddings de fastText [55].

Por regla general, los modelos del lenguaje en otro idioma que no sea el inglés, no suelen introducir novedades significativas al estado del arte salvo la disponibilidad de ese modelo en una lengua nueva (algo que, como como se ha explicado anteriormente, puede ser muy caro).

A continuación se listan algunos de los modelos del lenguaje más importantes en lenguas no inglesas: CamemBERT [83] y FlauBERT [84] para el francés, BERTje [85] para el holandés, FinBERT [86] para el finés, o BERTeus [87] para el euskera, German BERT [88] para el alemán, varios modelos en chino de Cui et Al. [89], BERTimbau [90] para el portugués, RuBERT para el ruso [91], Arabic BERT [92] para el Árabe, ParsBERT [93] para el persa, etc. Este último año se han publicado modelos del lenguaje para casi todos los idiomas, de forma que esta lista podría seguir de forma casi interminable.

2.3.6.1. BETO

Para el castellano, recientemente Cañete et al. publicaron BETO [22], el primer y único modelo del lenguaje basado en el Transformer en español. BETO está basado en BERT y en RoBERTa y se ha entrenado con el con el corpus Spanish Unannotated Corpora (SUC) [94], también de Cañete et al.

El modelo ha conseguido resultados notablemente mejores que los del BERT plurilingüe De esta forma, es el modelo del lenguaje de referencia para cualquier tarea de procesamiento del lenguaje natural en castellano.

2.4. Clasificación de texto

Esta sección se va a centrar en métodos neuronales para clasificar textos, usualmente apoyados en los modelos explicados anteriormente. No se va a centrar en métodos heurísticos, basados en glosarios o en características lingüísticas.

2.4.1. Redes Neuronales Recurrentes

Las primeras aproximaciones neuronales para clasificar textos se basaban en el Perceptrón [95] Multicapa (MLP, *multilayer perceptron*) [96]. Sin embargo tenían el problema de que muy pronto olvidaban lo primero que habían leído, esto es el problema

de los gradientes desvanecientes [97], generado al derivar repetidamente las funciones de activación en la fase de *backpropagation* de la red neuronal [98].

Las redes neuronales recurrentes (RNN, *recurrent neural networks*) [99] [100] intentan solucionar este problema. Su principal novedad funcional es procesar datos con forma de secuencias como, por ejemplo, listas de palabras o series temporales. Esto se consigue creando parámetros compartidos a través de las diferentes capas del modelo a modo de "memoria".

Sin embargo, las RNN básicas, no consiguen atajar del todo el problema de los gradientes desvanecientes [101].

Normalmente, este tipo de redes neuronales, se suelen usar después de alguna vectorización previa del texto, como las explicadas en la sección 2.3.1.

2.4.2. LSTM

Las redes de larga memoria a corto plazo (LSTM, por sus siglas inglesas *Long Short Term Memory*), son un tipo especial de redes recurrentes diseñadas para recordar dependencias a largo plazo [102] e intentar paliar el problema de los gradientes desvanecientes.

El redescubrimiento de estas redes, aplicadas al PLN, ha supuesto una revolución que, pese a llevar varios años usándose, siguen consiguiendo algunos resultados cercanos al estado del arte [103] [104].

2.4.3. GRU

Las redes con unidades de puertas recurrentes (GRU, por sus siglas inglesas *Gated Recurrent Units*) [105] [106] son una modificación de las LSTM que entrenan más rápido y que pueden funcionar muy bien en algunos contextos aunque generalmente son equivalentes [107].

En algunas aplicaciones, siguen consiguiendo resultados en el estado del arte [103] [104].

2.4.4. Redes bidireccionales

Las redes recurrentes bidireccionales [108] son simplemente dos redes neuronales recurrentes juntas. Las entradas de una van en orden normal mientras que los de la otra van en orden inverso. Después, las salidas de ambas se concatenan.

Esta estructura permite a las redes tener información tanto hacia adelante en la secuencia de texto como hacia atrás. Esta es la razón por la que también se usan en PLN. De hecho, aparte de usarse para clasificar texto, otros modelos del lenguaje las usan como redes de base, como BERT [13]

2.4.5. Modelos de Transformers

Una vez preentrenados, a los modelos de Transformers se les puede hacer *fine-tuning*, entrenarlos para la tarea concreta que se quiere realizar. En el caso de la clasificación de texto, se añade al final del modelo una capa softmax [109] (en el caso de clasificación binaria o multiclase).

Sin embargo, las redes neuronales disponen de muchos hiperparámetros configurables y a veces es difícil elegir los óptimos. Para ello se usan método como el entrenamiento basado en poblaciones (PBT, Population Based Training, en inglés) [110] [111].

Asimismo Sun, Qiu, Xu y col. [112] proponen varias estrategias para mejorar el estado del arte en modelos de clasificación:

- La primera es, antes de hacer el *fine-tuning*, seguir preentrenando el modelo del lenguaje con textos concretos del dominio del lenguaje del problema. A continuación, y opcionalmente, hacer *fine-tuning* del modelo con un *dataset* distinto pero que tenga el mismo objetivo (por ejemplo, clasificación de emociones) y ,finalmente, hacer el proceso de *fine-tuning* con los datos finales.
- Como BERT no acepta entradas de más de 521 tokens [13], los textos largos, como las noticias, no caben enteros en el modelo. La mejor alternativa es elegir los primeros tokens y los últimos de la noticia y obviar los del medio (que suelen tener menos información relevante). En el *paper* también prueban a montar una estructura jerárquica dividiendo los textos, pero no mejora los resultados.
- Para solventar el problema del *catastrophic forgetting* [113] en el *transfer learning* prueban con varios *learning rates* distintos para llegar a la conclusión, que a la hora de hacer *fine-tuning* necesitan ser más pequeños que en el preentrenamiento.

2.5. *Data augmentation*

Pese a que generalmente se dispone de datos de texto, a veces se presenta la necesidad de obtener más datos en algunos datasets de clasificación. Por lo tanto, son relevantes las técnicas para aumentar el número de muestras en los datos en base a modificaciones de los mismos. Estas técnicas suelen ser muy comunes también en procesamiento de imágenes [114] o en aprendizaje automático clásico, en este último campo se conocen como técnicas de *oversampling* o sobremuestreo [115]. Aparte de para la escasez de datos, estas técnicas también son útiles cuando las clases están muy desbalanceadas [116].

A continuación se analizarán las principales técnicas de *data augmentation* para el PLN [117]. También cabe destacar que todos estos métodos se pueden combinar para conseguir variaciones más complejas [118].

2.5.1. Sustitución léxica

La sustitución léxica se centra en cambiar palabras de una frase sin cambiar su significado. Su principal ventaja es que se consiguen muestras distintas y que no cambian de significado, sin embargo para hacer estas sustituciones se necesitan o bien glosarios o modelos del lenguaje.

2.5.1.1. Sustitución basada en tesauros

Esa sustitución cambia una o varias palabras de una frase de forma aleatoria por un sinónimo usando tesauros o bases de datos de sinónimos para crear más muestras

en el *dataset*. Esta técnica es bastante popular dados sus buenos resultados y su relativa facilidad de implementar una vez conseguidos los tesauros [119] [120] [121].

2.5.1.2. Sustitución basada en *embeddings* de palabras

Muy similar a la anterior, pero en vez de usar sinónimos, se usan palabras con vectores muy cercanos en el espacio de embeddings creado por alguno de los métodos explicados en la sección 2.3.1. Ha demostrado ser muy útil [122] [123] y, además, no no se presenta la necesidad de recopilar de forma manual ningún tesauro ya que se confía en un modelo neuronal que ha aprendido las relaciones semánticas del lenguaje.

Otra ventaja es que se pueden crear varias versiones para cada palabra, una por cada una de las n palabras con menos distancia a la original.

2.5.1.3. Sustitución basada en modelos del lenguaje enmascarados

Se pueden usar modelos del lenguaje entrenados con MLM (explicados en 2.3.5.1) para enmascarar palabras aleatorias de una frase y sustituirlas por las predicciones. De la misma forma que con la técnica anterior, se pueden crear varias muestras enmascarando una sola palabra. Esta técnica se ha probado útil [124] ya que el modelo tiene en cuenta el contexto a la hora de hacer su predicción y no solo la semántica de la palabra a sustituir.

2.5.1.4. Reemplazo de palabras basado en TF-IDF

Xie, Dai, Hovy y col. [125] proponen eliminar las palabras que tengan puntuaciones de TF-IDF bajas, ya que no aportan nada al significado de la frase y, por ende, no modifican la etiqueta original de la muestra que se está modificando.

2.5.2. *Backtranslation*

El proceso de *backtranslation* [126] consiste en coger la muestra que se quiere aumentar y traducirla a un tercer idioma. Después se vuelve a traducir al idioma original y, como los traductores automáticos no son perfectos, se obtiene una muestra ligeramente distinta aunque con el mismo significado final, por lo que se puede conservar su etiqueta [127]. Este método se ha probado muy útil en varios escenarios [128] [125].

Esta idea es muy potente ya que se puede repetir el proceso con n idiomas para aumentar drásticamente la cantidad de datos usados para entrenar [128]. También se puede repetir con varios motores de traducción distintos.

El mayor problema logístico de esta técnica es que las APIs de los motores de traducción muchas veces están limitadas para que los usuarios gratuitos no abusen de ellas. Por otra parte se pueden usar modelos disponibles en el Hub de modelos de Huggingface, pero la disponibilidad de idiomas es más limitada.

2.5.3. Expansión de contracciones

Gehring, Auli, Grangier y col. [64] proponen una técnica para expandir o crear contracciones y crear nuevas muestras, por ejemplo: *It's* $\leftrightarrow$ *It is*. Sin embargo esto puede

ser problemático con algunos ejemplos, por ejemplo: *He's* puede ser tanto *He is* como *He has*, por lo tanto no todas las expansiones de las contracciones están permitidas.

Otro problema es que esta técnica depende del idioma que se esté usando, ya que no todos disponen de contracciones ni las tienen con la misma notación.

2.5.4. Inyección de ruido

Estos métodos añaden ruido al texto, de forma que el modelo es más robusto y está preparado para textos reales donde puede haber erratas o errores de codificación.

2.5.4.1. Inyección de faltas de ortografía

Este método se basa en añadir errores ortográficos comunes [118] para generar nuevas muestras. El mayor problema viene a la hora de automatizar la creación de esos errores. Para ese proceso de pueden usar corpus como el propuesto por Grundkiewicz y Junczys-Dowmunt [129] basado en correcciones de errores en la Wikipedia.

2.5.4.2. Inyección de errores tipográficos QWERTY

Esta técnica se basa en introducir errores tipográficos comunes al usar teclados con la disposición QWERTY [117]. Por ejemplo sustituir *cantando* por *cantandi*.

2.5.4.3. Ruido de unigramas

Este método se basa en sustituir palabras usando una tabla de frecuencia de unigramas, creada en base a la frecuencia de aparición de cada palabra en el texto [130]. De esta manera, se sustityen muchas palabras comunes como preposiciones y artículos.

2.5.4.4. Sustitución por token genérico

Xie, Wang, Li y col. [130] proponen también un método en el que se reemplazan palabras aleatorias por un token genérico. De esta manera, se consigue evitar el sobreajuste y se mejoran las puntuaciones en algunos datasets.

2.5.4.5. Inserción aleatoria

Esta técnica selecciona una palabra que no sea *stopword* (pronombres, artículos y preposiciones), busca un sinónimo y se inserta en alguna posición aleatoria de la frase [119].

2.5.4.6. Eliminación aleatoria

Wei y Zou [119] también proponen la eliminación de una o varias palabras aleatorias de una frase para crear una nueva muestra.

2.5.4.7. Intercambio aleatorio

La premisa de este método es elegir aleatoriamente dos palabras de una muestra e intercambiarlas para generar una muestra nueva [119].

2.5.4.8. Desordenado de frases

Para crear más muestras de texto, se desordenan las frases de forma aleatoria [117]. Dependiendo del caso de uso, este método puede no ser efectivo ya que puede llevar a sobreajuste o a ruido en el modelo en el caso de textos donde importa mucho el orden en el que se expresan las ideas.

2.5.5. Manipulación del árbol sintáctico

Este método obtiene el árbol de dependencias de la frase y lo manipula con heurísticas para obtener una nueva frase [131]. Un ejemplo de eso sería transformar una frase de activa a pasiva.

Sin embargo este método es bastante complicado de llevar a la práctica ya que se requiere una gran cantidad de recursos lingüísticos para poder obtener el árbol de dependencias de una frase y reescribirla.

2.5.6. Uso de modelos del lenguaje preentrenados condicionales

Esta técnica se centra en enseñar a modelos generativos del lenguaje a crear más muestras para el *dataset* en base a las muestras que ya existen y unas pocas palabras [132]. Pese a su complejidad y la carga computacional de la misma, se ha demostrado que es una técnica muy útil [133].

2.5.7. Aumento por cruce de muestras

Luque [128] propone separar las muestras en dos partes y usar esas partes para crear muestras nuevas (usando solo partes que tengan la misma clase). Esto, como en la técnica del desordenado de frases, genera muestras que no acaban de ser coherentes en el lenguaje natural, pero que conservan el sentimiento o la idea principal de la clase.

2.5.8. Mixturas de texto

Las mixturas de imágenes son muy comunes en el mundo del análisis de imágenes [134]. Estas mixturas no son más que “mezclar” dos imágenes a nivel de píxel para generar una nueva.

Guo, Mao y Zhang [135] traen ese concepto al procesamiento de lenguaje natural con dos métodos distintos:

- Mixturas de palabras: Se eligen dos frases aleatorias y se combinan sus embeddings de forma ponderada. Después se pasan a la red neuronal de forma normal. La pérdida se calcula también de forma ponderada según los pesos asignados a cada muestra y sus respectivas etiquetas.
- Mixturas de frases: En este caso, primero se pasan las frases por el modelo y después se mezclan las salidas para calcular la pérdida ponderada.

2.6. *Datasets* para clasificación

En esta sección se va a centrar solo en los *datasets* para la clasificación de texto. Un muy buen lugar para encontrar *datasets* de PLN es el Hub de modelos de Huggingfce [11]. En él se disponen más de 800 datasets a fecha de mayo de 2021, 169 de los cuales son *datasets* de clasificación (binaria, multiclase y multietiqueta).

2.6.1. SST

El Standfort Sentiment Treebank [136] es un *dataset* de análisis de sentimiento que consta de 11855 críticas de películas en inglés. Después se analizó con el parser de Stanford y se obtuvieron 215154 frases con sus árboles de dependencia. Esas frases fueron después anotadas manualmente por tres revisores humanos.

Cada frase puede tener una de las siguientes cinco etiquetas: *negativa*, *algo negativa*, *neutral*, *algo positiva* y *positiva*. Este *dataset* con las cinco etiquetas suele llamarse SST-5 o SST de grano fino. También existe el SST-2 o SST binario, donde se clasifica entre las etiquetas negativas y las positivas, descartando las binarias.

Durante años este dataset ha sido un marcador del estado del arte en análisis de sentimiento y forma parte del paquete de pruebas GLUE [137]. Ahora mismo el mejor resultado para el SST-5, con una *accuracy* de 59.1, lo ostenta un modelo de RoBERTa con mejoras para ser explicable [138].

2.6.2. IMDb

El *dataset* de críticas de películas de IMDb [139] es un *dataset* binario con 50000 críticas en inglés publicadas en la *Internet Movie Database (IMDb)*. El *dataset* está equilibrado, lo que significa que contiene el mismo número de muestras positivas que negativas. Las críticas positivas son aquellas con una nota mayor o igual que siete y las negativas las que han recibido una nota menor o igual que cuatro, de esta forma la separación entre las dos clases está bien definida. Asimismo, tampoco se incluyen más de 30 críticas por cada película.

La mayoría de modelos modernos están llegando a puntuaciones bastante altas en los últimos años. El modelo del lenguaje XLNet obtiene una *accuracy* de 96.21.

2.6.3. IbeLEF

IberLEF es un *workshop* anual de PLN con varias tareas en español y otros lenguajes ibéricos [140]. Está organizado por la Sociedad Española de PLN y se fija como objetivo avanzar, mediante la competición, el estado del arte de las técnicas de PLN en los lenguajes mencionados.

Presenta tareas de todo tipo, sobre todo de clasificación de textos y de reconocimiento de entidades (NER, por sus siglas en inglés, *Named Entity Recognition*). Se ha elegido tres de esas tareas para la realización de este trabajo dado su importancia social y su impacto científico.

2.6.3.1. EXIST

EXIST (sEXism Identification in Social neTworks) es una competición cuyas tareas versan sobre clasificar el machismo de los tweets [141].

Se ha creado un *dataset* en base a tweets de 4500 textos en español y otros tantos en inglés. Asimismo se dispone de una parte separada para el test con 2000 textos más en cada idioma. Los textos han sido anotados por 5 anotadores expertos en cuestiones de género. A continuación se han muestreado 6977 tweets para entrenamiento y 3386 para test con el objetivo de que el *dataset* final sea más balanceado. Finalmente, al conjunto de test se le ha añadido una colección de 982 "gabs" de la red social sin censura Gab.com para probar la generalización de los modelos entre redes sociales distintas.

Se plantean dos tareas para este dataset: clasificación binaria entre sexista y no sexista y categorización de tipos de sexismo. La primera tarea se evalúa con la *accuracy* dado que el *dataset* está balanceado y para la segunda tarea se emplea la F1 balanceada.

Al ser un *dataset* nuevo, aun no existen resultados. Estos, se explicarán mejor en el Capítulo de resultados 4.

2.6.3.2. IDPT

IDPT (Irony Detection in Portuguese) es un *dataset* de clasificación binaria para la detección de ironía en tweets y noticias en portugués [142]. Los textos provienen del trabajo previo de Freitas, Vanin, Hogetop y col. [143] Araujo da Silva [144] Schubert y Freitas [145].

El *dataset* se divide en dos subtareas y *sub-datasets*. El primero de detección de la ironía en tweets y el segundo en noticias. El conjunto de test está anotado manualmente por los organizadores. En ambos casos se usa la métrica de la *balanced accuracy* para medir el rendimiento de los modelos.

Este *dataset* es muy interesante para medir el entendimiento del lenguaje por parte de un modelo ya que la detección de la ironía y los sarcasmos es una tarea muy complicada para las máquinas [146]. Esto queda probado por la variedad de datasets de detección de la ironía existente en la literatura [147] [18] [148] [149].

Asimismo, este dataset también es de interés ya que el portugués sufre de los mismos problemas que el español en el campo del PLN pese a ser una de las lenguas más habladas del mundo [14], también.

Las mejores puntuaciones para este *dataset* se discutirán en mayor profundidad en el Capítulo de resultados 4.

2.6.3.3. DETOXIS

DETOXIS es un *dataset* para la detección de toxicidad en comentarios de noticias sobre migración publicadas en webs de noticias españolas [150]. Las tareas sobre el *dataset* son dos; detección de toxicidad (clasificación binaria) y detección del nivel de toxicidad (de 0 a 3, siendo 0 no tóxico y 3 el máximo de toxicidad).

Las tareas de detección de la toxicidad han cobrado mucho interés en la comunidad en los últimos años. Prueba de ello son los siguientes *datasets*: Garibo i Orts [151], Kumar, Ojha, Malmasi y col. [152], Zampieri, Nakov, Rosenthal y col. [153], Struß, Siegel, Ruppenhofer y col. [154]. Todos estos *datasets* están en otros idiomas, DETOXIS es el primero en estar íntegramente en español.

Concretamente, los comentarios proceden de noticias de periódicos online españoles y han sido recopiladas desde agosto de 2017 a julio de 2020 y elegidos tendiendo en cuenta su nivel de controversia y número de comentarios (mínimo 50). En total se han recopilado 4357 comentarios, al rededor del $30\,\%$ es tóxico y se ha reservado el $80\,\%$ de las muestras para entrenamiento. Aparte de anotar la toxicidad, en la parte del entrenamiento se han anotado algunas características lingüísticas del texto que no están disponibles en la parte de *test*. La anotación se ha llevado a cabo por tres anotadores expertos por separado y se ha llevado a cabo un test de acuerdo entre anotadores.

Para evaluar los resultados se ha usado la métrica F1 para la clasificación binaria y el CEM (Closeness Evaluation Measure) [155] para la clasificación multiclase. Esta última métrica es muy útil para problemas de clasifiación ordinal dado que tiene en cuenta el orden de las clases, usando técnicas de la teoría de la medida.

Al ser un *dataset* nuevo, aun no hay resultados. Estos, se explicarán mejor en el Capítulo de resultados 4.

2.6.3.4. HAHA

HAHA (Humor Analysis based on Human Annotation) es una competición de IberLEF con el fin de detectar la presencia o no de humor en tweets escritos en español [156] Esta es ya la tercera edición de dicha competición [157] [17].

Detectar el humor es una tarea muy interesante para el PLN dado que es importante el contexto y poder entender los significados figurados. Prueba de esto es la gran cantidad de trabajos anteriores al respecto [158] [159] [160].

Se han propuesto cuatro tareas distintas para el corpus:

- La primera es la detección binaria del humor, esto es algo subjetivo así que la tarea concretamente consiste en detectar si el autor de un tweet, con ese tweet, tiene fines humorísticos o no. La métrica usada para esta tarea es la F1.
- La segunda tarea consiste en asignar una nota del 1 al 5 al tweet en base a la calidad del humor presente (asumiendo que el tweet tiene intenciones humorísticas). La métrica usada para esta tarea es el RMSE.
- La tercera tarea tiene como fin la clasificación del mecanismo lingüístico usado para conseguir el humor con un tweet de entre los siguientes mecanismos: ironía, juegos de palabras, hipérboles o impacto. La métrica usada para esta tarea es la F1 macro.
- Finalmente, la última tarea tiene como fin asignar los tipos de humor presentes en un tweet como, por ejemplo, humor negro, humor verde, etc. Para esta última tarea, se admiten varias etiquetas para cada muestra. La métrica usada para esta tarea es la F1 macro.

En total se han seleccionado 24000 tweets en la parte de entrenamiento, 6000 para la parte de desarrollo o validación y 6000 más para la parte de test. El procedimiento de anotación consta de un sistema de votos donde al anotador se le pedía que eligiera entre que el tweet no tiene humor o, en el caso de tenerlo, que se le asignara una puntuación del 1 al 5, donde 1 es que no tiene gracia y 5 significa que es desternillante [161] [162].

Para las dos restantes tareas, se ha seleccionado un subconjunto de los datos de la parte de entrenamiento.

2.7. Librerías de PLN

Actualmente se atraviesa un momento en el que la cantitad de recursos aumenta a un ritmo muy rápido, muestra de eso son todas las librerías exclusivamente dedicadas al tratamiento de texto.

2.7.1. Procesado de texto

Existe una gran variedad de librerías que disponen de distintos tokenizadores, analizadores de POS (*part-of-speech*), lematizadores, vectorizadores, etc. Las más relevantes son SpaCy[163], Stanza [164], NLTK [165] y Gensim [166]. Estas librerías son muy útiles para probar los distintos embeddings de palabras existentes.

Por otra parte, todos los métodos de *data augmentation* explicadas en la sección 2.5 están implementados en Python en la librería nlpaug [118].

2.7.2. Redes neuronales

Para la creación de modelos del lenguaje y modelos de clasificación de texto, la librería más relevante actualmente es Huggingface/transformers [11], de la cual ya se ha explicado anteriormente su colección de modelos y *datasets*. Con más de 45000 estrellas y 10000 *forks*, es una librería con gran apoyo de la comunidad que ha conseguido democratizar los grandes y caros modelos del lenguaje actuales.

Implementada en Python y con soporte para los dos grandes frameworks de redes neuronales a bajo nivel, Pytorch [167] y Tensorflow [168], ofrece un gran balance entre simplicidad y posibilidad para modificar los modelos a bajo nivel, siendo así muy atractiva tanto para la academia como para la industria.

2.8. Conclusiones

Como se ha mostrado a lo largo de este estudio, son muchas las nuevas técnicas de procesamiento del lenguaje natural que han aparecido en los últimos dos años, se han creado muchos modelos del lenguaje, se han explotado muchas técnicas de entrenamiento y se han creado muchos *datasets* para probar todos esos modelos y se han creado numerosas librerías de código abierto para hacer accesible todos esos recursos a todo el mundo.

Sin embargo, existe cierto desinterés de la industria y la academia hacia los modelos neuronales del lenguaje en castellano, siendo este un gran nicho y un idioma muy utilizado a nivel mundial.

Por lo tanto, hay mucho lugar para la mejora, aunque sea difícil en cuestión de recursos computacionales [31], dada la situación actual de la ciencia en España.

Capítulo 3

Desarrollo

En este capítulo se describirán las tareas realizadas durante este trabajo. Primero se describirán los *datasets* elegidos para los experimentos. A continuación se describirá brevemente el software usado y su implementación. Finalmente, se describirán detalladamente las soluciones propuestas para cada *dataset*.

3.1. Descripción del problema

Para la realización de este trabajo, se han elegido *datasets* de clasificación de texto, tanto multiclase como clasificación binaria. Asimismo, nos centraremos en problemas que sean únicamente de clasificación de texto, esto significa que no habrá ningún tipo de datos categóricos y las muestras serás solo entradas en lenguaje natural.

También se ha usado como criterio el idioma de los textos, poniendo énfasis en los textos en español, en línea con los objetivos de este trabajo.

Todos los *datasets* utilizados pertenecen a la edición de 2021 de IberLef. De esta forma podemos comparar los resultados directamente con los mejores resultados del estado del arte más actual. Concretamente, se han elegido tres *datasets*; EXIST, IDPT y DETOXIS, explicados brevemente durante el resumen del estado del arte 2.6.

3.1.1. EXIST

Se ha elegido este problema ya que trata problemas de relevancia social muy presentes en las redes sociales actualmente. Asimismo, ha habido una alta participación en la competición, lo que asegura que los mejores resultados estén en el estado del arte. Finalmente, el hecho de que sea un corpus plurilingüe hace el problema más desafiante, lo cual es perfecto para mostrar el potencial de los métodos que se presentarán más adelante.

En la tabla 3.1 podemos ver la distribución de las muestras para cada una de las tareas en este *dataset*, no están desagregadas por idiomas ya que su distribución es similar. Se observa un claro balanceo entre las clases *machista* y *no machista* aunque ese balanceo desaparece para la tarea 2, donde existen clases que son casi el doble de grandes que otras y la clase *no machista* es más grande que la suma de todas las otras clases.

Clase		**muestras Tarea 1**	**muestras Tarea 2**
no machista		3600	3600
desigualdad ideológica	machista	3377	866
dominación, estereotipado			809
misoginia violencia no sexual			685
violencia sexual			517
cosificación			500

Cuadro 3.1: Distribución de las muestras de EXIST

De este análisis preliminar se concluye que habrá que tomar medidas para el desbalanceo de la tarea 2 y para prevenir el sobreajuste ya que no se dispone de muchos datos.

La siguiente tabla es útil para ver, de una forma general, qué tipo de muestras hay en el *dataset*:

I love poetry books, so I'm reading the one i have on this plane flight and one of the flight attendants (black women) goes "it's good to see a brotha reading something that's is so deep"	no machista
me quiero follar al osito bimbo, sus deliciosas curvas me excitan demasiado, tanto que me hacen llegar al orgasmo de solo verlas, me gustaría darle unas fuertes nalgadas a sus blancas y suaves pompas joder me tienes demente ni puedo verte en el supermercado porque se me pone dura	no machista
Can the fellas participate or is this just for the ladies/Non binary people because I don't wanna get clowned.	desigualdad ideológica
ive been sooo interesting my whole life and i just want to be a boring trophy wife now	dominación, estereotipado
Cuando un hombre gana dinero , sueña con darle lo mejor a su esposa , cuando una mujer gana dinero , se siente apoderada ,que no necesita ningún hombre a su lado #LaVida	dominación, estereotipado
Fucking skank	misoginia violencia no sexual
Bitches be begging me to fw them just to give me a reason not to fw them. Lol	violencia sexual
[mención a usuario borrada por privacidad] Cuando gustes puta te hago que sientas eso tu dime pea ir a mamar te esa pinche vagina zorra o ser tu piloto para que mames como una pinche perra	violencia sexual
some women just don't deserve onlyfans, bitches be UGLY as fuck and ask you to pay $20 to see their UGLY FAT BLOTCHED TITTIES, BITCH!	cosificación
todas putas	cosificación

Cuadro 3.2: Ejemplos de las distintas clases en EXIST

3.1.2. IDPT

Se ha elegido este problema dado el gran interés académico en solventar una tarea tan complicada como es la detección de ironía [169]. El hecho de poder detectar la ironía sin más contexto que el propio texto, implicaría que el modelo, aparte de entender la semántica y la sintaxis, entiende lo implícito que no viene escrito y la diferencia entre el discurso normal y el hiperbólico. Asimismo, de la misma forma que pasa con el español, el corpus presenta un problema más desafiante por estar en portugués, un idioma muy hablado pero con pocos recursos de PLN.

Respecto al tamaño del *dataset*, la parte de entrenamiento de los tweets tiene 15212 muestras, y la parte de entrenamiento de las noticias, tiene 18494. Para la parte del test, se han etiquetado 300 muestras para cada tarea de forma independiente. De esta manera, los modelos que tengan buen resultado en test, demostrarán haber generalizado correctamente.

En la tabla 3.3 podemos ver la gran diferencia y desbalanceo entre clases. Destaca que la mayoría de tweets son irónicos y, en cambio, la mayoría de noticias, no lo son.

Tweets		**Noticias**	
Clase	**Nº de muestras**	**Clase**	**Nº de muestras**
irónico	12736	irónico	7222
no irónico	2476	no irónico	11272

Cuadro 3.3: Distribución de las muestras de IDPT

En la siguiente tabla podemos ver varios ejemplos de muestras de ambos *datasets*. Es bastante relevante la diferencia en el registro de la escritura y la extensión de los textos. Esto prueba que las dos tareas de la competición son distintas aunque tengan un objetivo distinto.

Quando cheguei a Montemor, gozavam comigo por montar a cavalo, agora toda a gente anda na equitação	tweet irónico
Para TIM, mudanças regulatórias devem acontecer em conjunto - / telecom telefonia mercado regulação economia	tweet no irónico
O fim do mundo não chegou, mas o fim do emprego sim. O office boy Antonio Nascimento acabou demitido depois de mandar seu chefe se f*** ontem. Ele deixava a empresa em São Paulo após um dia de expediente quando, ao se despedir, virou-se para o chefe e o xingou. O chefe, incrédulo, perguntou o que o rapaz dissera e ele repetiu o xingamento. No ponto de ônibus, o boy disse aos colegas que não se importava porque o mundo ia acabar hoje. O boy disse ainda que passaria a noite toda na farra e não voltaria para a casa, onde sua mulher o aguardava. Hoje, Antonio perdeu o emprego. E a mulher. “O pior de tudo é ficar ouvindo piadinhas de gente que diz que isso não é o fim do mundo”, disse ele.	noticia irónica

Foi sancionada com vetos a Lei Complementar 173, publicada dia 28 de maio de 2020 no Diário Oficial da União, que estabelece o Programa Federativo de Enfrentamento ao Coronavírus para Estados, Distrito Federal e Municípios. O plano prevê a negociação de empréstimos, a suspensão dos pagamentos de dívidas contratadas com a União (estimadas em R$ 65 bilhões) e a entrega de R$ 60 bilhões para os governos locais aplicarem em ações de enfrentamento à pandemia.	noticia no irónica

Cuadro 3.4: Ejemplos de las distintas clases en IDPT

3.1.3. DETOXIS

Este problema también es muy relevante socialmente en la actualidad data la alta polarización del discurso en las redes sociales. Actualmente hay mecanismos manuales para reportar comentarios tóxicos, sin embargo, al ser manuales hace que el proceso sea muy lento e ineficaz. De esta manera, poder automatizar (hasta cierto punto) este proceso podría conseguir simplificar dichos trámites, resultando en un ahorro para las redes sociales y menos toxicidad en general.

Por otra parte, es un corpus perfecto para poder experimentar y mejorar los actuales modelos del lenguaje para el español que, como se ha estudiado en el Capítulo 2, están muy atrasados en comparación a otros idiomas como el inglés, pese a la cantidad de hablantes que existen para la lengua.

Aparte de las etiquetas para cada clase, el *dataset* tiene una serie de etiquetas lingüísticas que hacen de indicadores del tipo de discurso. Estas etiquetas solo están disponibles en los datos de entrenamiento y tampoco van a ser usados dado a que el foco de este estudio es obtener los mejores resultados usando solo métodos basados en modelos del lenguaje. Finalmente, las muestras también tienen información sobre el hilo de comentarios al que pertenecen y si son un comentario nuevo o respuesta a otro. Por las mismas razones que con las características lingüísticas, esta información no será usada en este trabajo.

Clase	**Nº de muestras**
no tóxico	2317
levemente tóxico	808
tóxico	269
muy tóxico	69

Cuadro 3.5: Distribución de las muestras de DETOXIS

En la tabla de arriba se puede ver la distribución de las muestras en la partición de entrenamiento de los datos. El hecho más notable es que el $67\,\%$ de las muestras no son tóxicas, esto significa que el *dataset* está muy desbalanceado. Para la segunda tarea, se puede observar también un desbalanceo muy notable.

La siguiente tabla es un ejemplo ilustrativo de los tipos de textos presentes en los datos:

Los detuvieron en ronda malaga, un saludo	no tóxico
Tremenda pole si señor	no tóxico
Son sus costumbres y hay que respetarlas...	levemente tóxico
Loss mas valientes, los que mejor cortan nuestras cabezas, Para vosotros, socialistas, izquierdistas, y no racistas,	levemente tóxico
Esto es lo que importas cuando los rescatas en lugar de hundirlos.	tóxico
Con lo difícil que es cortar un filete con nervios, no sé como cortan las cabezas tan bien.	tóxico
Está claro que vienen los mejores. Haced que pase putos rojos de mierda.	muy tóxico
Vosotros sois la mayoría subnormales. Inhumanos e hijos de puta pero es que además no sabéis leer. Retrasados, está diciendo la noticia que el que llevaba la daga y le cortó la cabeza fue el puto mafioso que los metió en un barco de mierda a cambio de 2.500€.	muy tóxico

Cuadro 3.6: Ejemplos de las distintas clases en DETOXIS

3.2. Implementación

Aquí se explicará la implementación del proceso de entrenamiento de los modelos. Dado a que todos son modelos de clasificación, se ha podido reutilizar casi todo el código para todos las soluciones propuestas.

Se ha elegido Python como lenguaje de programación dada la gran cantidad de recursos existentes tanto para PLN como para aprendizaje automático y redes neuronales. Concretamente se ha usado la librería de *huggingface/transformers* con el *backend* de *Pytorch* para los modelos, *huggingface/datasets* [170] para la lectura y procesamiento de los datos y para la aumentación de los datos se ha usado la librería *nlpaug*, todas de código abierto.

Para la lectura, análisis y procesado de los datos se ha utilizado *huggingface/datasets* ya que ofrece las mismas capacidades que *Pandas* [171] [172] y le suma una muy buena compatibilidad con la librería usada para el entrenamiento de modelos, *huggingface/transformers*.

Para el entrenamiento de los modelos, se ha usado la clase *Trainer* de *huggingface/transformers*. Esta clase permite añadir una capa de abstracción encima del código de *Pytorch*, facilitando así la creación de experimentos con diferentes tipos de modelos sin tener que hacer demasiados cambios al código.

3.3. Soluciones propuestas

En esta sección se explicarán los modelos que se han creado para cada una de las distintas tareas, teniendo en cuento todo lo expuesto anteriormente y con el foco en usar modelos preentrenados de código abierto y bajo coste computacional. Para simplificar los modelos, se ha decidido no usar datos externos al problema (salvo los que están implícitos en los modelos preentrenados) para enriquecer el modelo.

Primero se explicarán las elecciones comunes a todas las tareas y, a continuación, se expondrán los modelos concretos para cada una de las tareas.

3.3.1. Preprocesado de los datos

Se ha llevado a cabo un preprocesado simple donde se han sustituido algunas expresiones comunes en las redes sociales por sus formas más normalizadas. Esto es especialmente útil cuando se usan modelos del lenguaje preentrenados con corpus que no tienen textos de redes sociales.

- Cada URL ha sido reemplazada por el *string* "[URL]" de forma que no haya tokens extraños cuando el tokenizador intente procesar las URL. Asimismo, tampoco se puede extraer demasiada información semántica sobre los temas a tratar (machismo, ironía y toxicidad) a partir de una URL. La única información relevante de cara al modelo es el hecho de que en el texto hay una URL.
- Los caracteres almohadilla ("#") han sido eliminado ("#ejemplo" $\rightarrow$ "ejemplo") porque los modelos del lenguaje usados para el *fine-tuning* tampoco han sido entrenados viendo este tipo de escritura. De hecho, normalmente las almohadillas acaban siendo usadas como palabras normales en redes sociales como Twitter.
- Se ha sustituido cada nombre de usuario por el *string* "[USER]" ya que el nombre exacto de un usuario no aporta ninguna información al modelo. La única característica relevante un nombre de usuario en el texto, es el hecho de saber que se está mencionando a una persona, pero no qué persona se está mencionando.
- Finalmente se han normalizado todas las risas ("jasjajajajj" $\rightarrow$ "haha") ya que se suelen escribir de muchas formas y lo único que hacen es introducir ruido en el modelo.

3.3.2. Optimización de parámetros

Para el proceso de *fine-tuning* se ha llevado a cabo una optimización de los parámetros de los modelos mediante una búsqueda en malla. La búsqueda se ha llevado a cabo con una validación cruzada estratificada de cinco hojas.

A continuación se exponen los parámetros que se han optimizado y los espacios de búsqueda:

- *Learning rate*: $(1e-6, 1e-5, 3e-5, 5e-5, 1e-4)$
- Tamaño de *batch*: $(8, 16, 32)$
- Ratio de *dropout*: $(0.08, 0.1, 0.12)$

En todos los modelos, la mejor combinación ha sido la siguiente:

- *Learning rate*: $1e-5$
- Tamaño de *batch*: 16
- Ratio de *dropout*: 0.1

3.3.3. *Baselines*

Se ha creado un modelo *baseline* para poder tener una comparativa simple y rápida a todos los modelos que se han generado. Este modelo usa el método de extracción de características del *Hashing Trick* (*HashingVectorizer* en scikit-learn) y un clasificador *Random Forest* [173].

Este método es muy similar al método de bolsa de palabras, sin embargo, tiene la posibilidad de proyectar los vectores en una esfera euclídea unitaria, en vez de dejar los vectores solo con las frecuencias de los términos.

Aparte, esos vectores son hasheados con la función Murmurhash [174], esto tiene algunas ventajas computacionales, sobre todo cuando se está tratando con una gran cantidad de datos.

3.3.4. EXIST

Como el *dataset* está en varios idiomas, se pueden seguir varias estrategias. Una de ellas sería usar un modelo plurilingüe, como mBERT [13] y otra sería usar un modelo para cada idioma. En este caso se ha elegido hacer pruebas tanto con mBERT como con dos modelos (BETO [22] para el español y BERT [13] para el inglés).

Para la segunda tarea, dado que las clases están desbalanceadas, se ha probado, aparte de hacer un simple entrenamiento multiclase, a hacer un modelo jerárquico. De esta forma, el modelo de la primer tarea, clasifica entre machista y no machista y otro modelo solo ve los que el primer modelo ha etiquetado de machistas.

Como se ha visto anteriormente, el tamaño del *dataset* es pequeño y esto puede conllevar problemas de entrenamiento si estamos usando redes neuronales. Por lo tanto se han seguido dos estrategias de *Data Augmentation* para aumentar el tamaño de los datos; la *Backtranslation* [126] y, aprovechando que es un *dataset* plurilingüe, podemos usar los datos de esos idiomas para aumentar el tamaño del *dataset* usando traducción simple.

3.3.4.1. *Backtranslation*

Para esta prueba se han usado los modelos de Helsinki NLP [175], basados en el modelo de traducción Marian [176] y, para los lenguajes no disponibles, también se ha usado la API de Google Translate.

Se ha hecho pruebas con 5, 10, 20 y 30 idiomas para ver si siempre se mejora al añadir más muestras o se acaba obteniendo sobreajuste. Dichos idiomas (expresados en el ISO 639-1 son los siguientes): *eu*, *la*, *zh-cn*, *hi*, *bn*, *pt*, *ru*, *ja*, *pa*, *mr*, *te*, *tr*, *ko*, *fr*, *de*, *vi*, *ta*, *ur*, *it*, *ar*, *fa*, *ha*, *kn*, *id*, *pl*, *uk*, *ro*, *eo*, *sv* y *el*. También se usaron *es* y *en* para los datos en inglés y en español, respectivamente.

Para cada muestra del corpus, se ha elegido de manera aleatoria un idioma de los listados anteriormente como pivote para llevar a cabo el proceso de *Backtranslation*. De esta forma se ha doblado el tamaño de los datos.

3.3.4.2. Traducción plurilingüe

Siguiendo el razonamiento de la *backtranslation*, también se podría usar más datos etiquetados (siempre y cuando siguieran el mismo *gold standard*) en otros idiomas. Esta hipótesis se puede probar en *datasets* plurilingües.

De esta forma, se han ampliado los datos traduciendo todo el texto inglés a español y *viceversa*. Con esto se debería obtener un entrenamiento más robusto, evitando el sobreajuste ya que las muestras generadas son completamente nuevas para los modelos (en el caso de usar un modelo para cada idioma), justo al revés que con la *backtranslation*. La mayor desventaja de este método es que no permite aumentar los datos tanto como la *backtranslation* u otros métodos de *Data Augmentation*.

3.3.5. IDPT

Como esta tarea es en portugués, en vez de eso ha usado el modelo BERTimbau [90], un BERT entrenado en portugués que es mejor que mBERT y ha alcanzado el estado del arte. Sin embargo, también se ha probado con mBERT para ver cuánto mejora.

Como no hay demasiados datos en los dos *datasets* que componen esta tarea, se usarán también técnicas de *Data Augmentation*. Concretamente se usará la sustitución de palabras mediante un MLM, explicada en la sección 2.5.1.3.

De esta forma, para cada muestra del *dataset*, se enmascaran aleatoriamente el $15\,\%$ de los tokens y se usa el modelo BERTimbau para predecirlos, creando una muestra modificada. Con este método, se obtiene el un dataset el doble de grande. Esto se podría repetir con varias pasadas y generar más muestras, sin embargo, llevaría a sobreajuste.

3.3.6. DETOXIS

En este caso, al ser la tarea en español, se ha usado el modelo BETO [22], explicado en la Sección 2.3.6.1. De la misma forma que en la tarea anterior, la cantidad de datos es limitada y se ha usado la misma técnica de *Data Augmentation*.

Aparte, al ser un *dataset* de clasificación multiclase ordinal (donde la etiqueta 1 se parece más a la 0 y a la 2 que a la 3, se han probado las siguientes técnicas para la segunda tarea de la competición:

- La aproximación más sencilla probada ha sido tratar las dos tareas como si fueran un único problema de cuatro clases. En la primera clase, todo lo que fuera distinto a *no tóxico*, se identificaría como tóxico.
- En la segunda aproximación, primero se ha entrenado un modelo de clasificación binaria para distinguir entre *no tóxico* y el resto de las clases (todas con algún nivel de toxicidad). Después, se usa otro modelo especializado solo en encontrar el nivel de toxicidad. Esta estrategia se podría asimilar a una estrategia jerárquica.
- De una forma similar a la anterior, en esta estrategia se ha dividido la tarea en tres problemas de clasificación binaria distintos; clasificación entre *no tóxico* y el resto de las clases, *levemente tóxico* y *tóxico* o *muy tóxico* y entre *tóxico* y *muy tóxico*. Con esto, se busca tener modelos muy específicos que puedan diferenciar

leves cambios en la toxicidad. Sin embargo, al haber tan pocas muestras en algunos casos, estos podría desembocar en sobreajuste.

- Para prevenir dicho sobreajuste y obtener una mejor generalización, también se ha probado con una aproximación basada en *transfer-learning*, similar a la presentada por Sun et al. [112]. En vez de usar siempre el mismo BETO preentrenado para cada modelo que se vaya a entrenar, se usa como modelo base el modelo del paso anterior. Por ejemplo, el modelo que clasifica entre *levemente tóxico* y *tóxico* se ha entrenado usando como modelo base, el que clasifica entre *no tóxico* y las demás clases.

Capítulo 4

Resultados

A continuación se explicarán los resultados obtenidos en los experimentos descritos en la sección anterior y se compararán con los resultados de otros participantes en la competiciones para poder tener una visión más clara sobre su distancia al estado del arte pese a las restricciones autoimpuestas en este trabajo (software libre, modelos computacionalmente simples y ausencia de análisis lingüísticos).

Asimismo, para probar la calidad individual de las propuestas, se realizarán estudios de ablación, donde al mejor modelo se le quita una característica para ver cuánto empeora. Esto es posible ya que, una vez terminadas las competiciones, los datos de test etiquetados son publicados.

4.1. Configuración Experimental

Todos los modelos han sido entrenados con una tarjeta gráfica NVIDIA Tesla P100-PCIE-16GB y un procesador Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz con 500GB de memoria RAM.

Se ha usado el siguiente software: Python3.8, transformers 4.5.1 [11], pytorch 1.8.1 [167], scikit-learn 0.24.1 [177] y nlpaug 1.1.3 [118].

4.2. EXIST

En la Tabla 4.1 se pueden observar los resultados de los modelos propuestos para la primera tarea de EXIST. Los resultados del *baseline* Tfidf+SVM y de AI-UPV_team (los ganadores de la task) proceden del resumen de la tarea [141]. Asimismo, *HV+RF* hace referencia a HashingVectorizer + RandomForestClassifier.

El resto de modelos de la tabla (*2BERTs+Backtranslation*, *2BERTs* and *2BERTs+ Traducción plurilingüe* donde 2BERTs se refiere a los diferentes modelos del lenguaje usados dependiendo del idioma) son los explicados en la Sección 3.3.4 y la *baseline* propuesta.

A simple vista se puede apreciar que todos los modelos de BERT superan con creces a los modelos de clásicos y que los modelos plurilingües (mBERT), pese a ser muy versátiles, tienen un rendimiento que se ve eclipsado por modelos más específicos para cada idioma.

Los modelos de *Backtranslation* tuvieron buenos resultados en los primeros experimentos de entrenamiento, sin embargo acabaron sobreajustando mucho con el conjunto de test, obteniendo 0.7479 de *accuracy*, solo un poco mejor que el BERT multilingüe (0.734), pero peor que los dos modelos de BERT (uno para cada idioma). Esto prueba que las técnicas de *Data Augmentation* no son siempre útiles.

Cabe destacar que los modelos que hacen uso de la Traducción plurilingüe, han obtenido una accuracy de 0.7683 en el conjunto de test. Esto demuestra una mejor generalización que la del modelo sin ninguna técnica de incremento de datos (0.7603). Con esto, el mejor modelo propuesto, se posiciona muy cerca del estado del arte en esta competición, que solo es un $1.58\,\%$ mejor. Con este resultado, se ha obtenido la séptima posición de un total de 31 equipos formados por los mejores investigadores en NLP a nivel iberoamericano.

Modelo	***Accuracy***
HV+RF	0.6830
Tfidf+SVM	0.6845
mBERT	0.7341
2BERTs+*Backtranslation*	0.7479
2BERTs	0.7603
2BERTs+Traducción plurilingüe	0.7683
AI-UPV_team	0.7804

Cuadro 4.1: Resultados para la tarea 1 de EXIST

Para la segunda tarea, los resultados son similares a los ya vistos. En la Tabla 4.2 se pueden observar con más detalle. Como antes, la *baseline* Tfidf+SVM y el resultado de AI-UPV_team proceden del resumen de la competición [141]. Se puede apreciar que los modelos han tenido un comportamiento consistente a lo obtenido en la tarea anterior. De esta forma, los resultados siguen estando muy cerca del estado del arte.

Modelo	**F1**
Tfidf+SVM	0.3950
HV+RF	0.4131
mBERT	0.4961
2BERTs+*Backtranslation*	0.5174
2BERTs	0.5218
2BERTs+Traducción plurilingüe	0.5295
AI-UPV_team	0.5787

Cuadro 4.2: Resultados para la tarea 2 de EXIST

Para resumir y cuantificar las mejoras de los modelos propuestos, podemos ver el estudio de ablación para el mejor modelo obtenido (*2BERTs+Traducción plurilingüe*) en la tarea 1. Cada entrada de la siguiente tabla tiene un elemento eliminado del modelo.

Modelo	***Accuracy***
Mejor modelo	0.7683
Modelo por defecto (sin Grid-Search)	0.7451
Todas las letras en minúsculas	0.7599
Sin *augmentation*	0.7603
Sin preprocesado	0.7678

Cuadro 4.3: Estudio de ablación para los modelos de la tarea 1 de EXIST

Esto demuestra que la mayoría de las ideas presentadas producen algún tipo de mejora en el sistema. La primera observación que se puede hacer sobre estos resultados es que la mejora más significativa ha sido la selección de unos buenos hiperparámetros para el modelo. Este resultado ha sido consistente en todos los experimentos realizados (ver Tablas 4.6 y 4.9), lo que significa que la elección óptima de hiperparámetros es crucial para conseguir un buen *fine-tuning* en los modelos del lenguaje.

En cambio, hay mejoras que son muy pequeñas, como la del preprocesado de los datos, y que podrían considerarse dentro del margen de error de la red. Sin embargo, esto no es así ya que, de forma consistente, en todos los experimentos llevados a cabo en este trabajo, dicho preprocesado mejora algo los resultados. Esto se puede observar en las los estudios de ablación de IDPT y de DETOXIS (Tablas 4.6 y 4.9 respectivamente).

Cabe destacar también el efecto de usar el texto tal cual sin pasar a minúsculas (técnica común en el PLN para que el vocabulario sea más pequeño). Esto se debe a que para los textos presentes en redes sociales, el hecho de que se escriba completamente en mayúsculas puede contener mucha información semántica, sobre todo en el contexto de identificar comportamientos machistas o violentos. Se podrá observar un resultado similar en las otras dos tareas dado a que el contexto y el registro del lenguaje es muy similar (ver Tablas 4.6 y 4.9).

Finalmente, también es muy notable la gran mejora de la Traducción plurilingüe que prueba la hipótesis sobre la capacidad de esta técnica de *Data Augmentation* para mejorar la generalización del modelo en corpus plurilingües.

4.3. IDPT

En la Tabla 4.4 se pueden observar los resultados de los modelos en el conjunto de test para la primera tarea de IDPT. Los resultados de TeamPiLN (ganadores de la tarea) son los presentados en el resumen de la competición. Los modelos presentados en esta tarea son *BERTimbau* y *BERTimbau-aug*, sin *Data Augmentation* y con *Data Augmentation*, respectivamente tal como se ha explicado en la Sección 3.3.5. Se ha probado con las versiones *base* y *large* de BERTimbau también ya que no suponía un gran aumento de tamaño del modelo.

Se puede apreciar, como en los resultados anteriores, que los modelos del lenguaje, quedan por delante de los métodos clásicos por un gran margen. Cabe destacar que, pese a no ser una gran mejora de rendimiento, las técnicas de incremento de datos propuestas también consiguen mejorar el resultado del modelo.

También se comprueba un hecho que viene siendo la tónica habitual en los mode-

los del lenguaje y es que a mayor tamaño del modelo, mejores resultados. En este caso la mejora no es demasiado sustancial, hecho que se puede deber al preentrenamiento o la pequeña diferencia de tamaño entre los modelos. Se obersva, aun así, que *BERTimbau-base* obtiene una *Balanced Accuracy* de 0.4831 mientras que *BERTimbau-large* consigue 0.4912.

En resumen, los modelos propuestos obtienen grandes resultados dada su simplicidad, demostrando que encontrar los mejores hiperparámetros puede tener un impacto crucial a la hora de optimizar el rendimiento del modelo.

Estos modelos han logrado el cuarto puesto entre todos los participantes, probando otra vez, que esta aproximación abierta y simple tanto de implementar como computacionalmente, es una elección muy sólida en una gran variedad de casos.

Modelo	**bacc**
HV+RF	0.3316
BERTimbau-base	0.4831
BERTimbau-large	0.4912
BERTimbau-large-aug	0.5000
TeamPiLN	0.5239

Cuadro 4.4: Resultados para la tarea 1 de IDPT

Para la segunda tarea, los resultados no fueron tan buenos como los obtenidos en la primera. En la Tabla 4.5 se pueden apreciar más en detalle. En esta ocasión, TeamBERT4EVER obtuvo los mejores resultados, lo que significa que nadie obtuvo un modelo perfecto para las dos tareas, probando que la diferencia en el registro escrito existente entre las redes sociales y las noticias, es bastante amplia.

En el caso del modelo presentado, las causas parecen radicar en que BERTimbau no puede tratar tan bien los textos largos y formales que suelen tener las noticias donde, si existe la ironía, suele ser bastante menos obvia que la que se puede presentar en otros contextos más espontáneos. Salvo eso, podemos comprobar que los resultados siguen la misma línea que en la tarea anterior.

Modelo	**bacc**
HV+RF	0.5423
BERTimbau-base	0.7692
BERTimbau-large	0.7804
BERTimbau-large-aug	0.7858
TeamBERT4EVER	0.9215

Cuadro 4.5: Resultados para la tarea 2 de IDPT

En la siguiente tabla se presenta el estudio de ablación para el mejor modelo la primera tarea de IDPT (*BERTimbau-large-aug*):

Modelo	***bacc***
Mejor modelo	0.5000
Modelo por defecto (sin Grid-Search)	0.4721
Todas las letras en minúsculas	0.4846
Sin *augmentation*	0.4912
Sin preprocesado	0.4975

Cuadro 4.6: Estudio de ablación para los modelos de la tarea 1 de IDPT

En este estudio se observan resultados similares a los obtenidos en EXIST, donde la búsqueda de hiperparámetros resulta ser la optimización más relevante para el modelo. Las demás propuestas, todas resultan en alguna mejora para el modelo, ya sea más o menos relevante.

4.4. DETOXIS

En la Tabla 4.7 se pueden observar los resultados en el conjunto de test para la primera tarea. Nótese que los resultados de las *baselines* ChainBOW, Word2VecSpacy y el resultado de SINAI_team (ganadores de la tarea) provienen del resumen de la tarea [150]. Los modelos propuestos para esta tarea son *BETO-multiclase* y *BETO-binario* ambos con y sin *Data Augmentation*, tal como se ha explicado en la Sección 3.3.6.

Cabe destacar que los modelos clásicos que actúan como *baselines* obtienen resultados considerablemente peores a los obtenidos por los modelos del lenguaje.

Se puede observar que a penas hay diferencia entre el modelo multiclase (0.5721) y el binario (0.5777). Esto puede parecer sorprendente, pero se explica dada la gran cantidad de muestras no tóxicas existentes en comparación al resto de las muestras, lo que las hace muy fáciles de identificar en ambos casos.

Finalmente, se puede observar que la estrategia de *Data Augmentation* obtiene consistentemente alrededor de 0.02 puntos más que los modelos simples. Con este resultado, se ha obtenido la segunda posición en la competición, probando otra vez que la simplicidad de usar BETO con una buena selección de parámetros puede generar resultados muy buenos.

Modelo	**F1**
Word2VecSpacy	0.1523
ChainBOW	0.3747
HV+RF	0.4159
BETO-multiclase	0.5721
BETO-binario	0.5777
BETO-multiclase-aug	0.5981
BETO-binario-aug	0.6000
SINAI_team	0.6461

Cuadro 4.7: Resultados para la tarea 1 de DETOXIS

Para la segunda tarea, los resultados fueron similares a los obtenidos en la primera.

En la Tabla 4.8 se pueden apreciar con más detalle. Como antes, los *baselines* ChainBOW Word2VecSpacy y el resultado de SINAI_team han sido obtenidos del resumen de la tarea [150].

Siguiendo la tendencia observada durante todo el estudio, los resultados de los modelos clásicos no están a la altura de los obtenidos por los modelos del lenguaje basados en BERT.

Es destacable que la aproximación de *transfer learning* (*BETO-transfer*) obtiene mejores resultados que otras aproximaciones. Pese a que los últimos modelos de la cadena entrenan con muy pocos datos, el hecho de tener el "conocimiento" aportado por los otros modelos hace que los resultados provenientes de este modelo sea muy buenos.

Por otra parte, se puede observar que casi no existe diferencia entre el modelo simple multiclase y el modelo que primero detecta si un comentario es tóxico o no y después clasifica la toxicidad, en caso de haberla (*BETO-2models-aug*). Estos resultados son coherentes con los obtenidos en la primera tarea, mostrando que los modelos no salen perjudicados de tener que clasificar una clase extra que esté tan bien definida como la de *no tóxico*.

Estos resultados obtuvieron la quinta posición entre todos los equipos participantes (24), lo cual continua probando que el planteamiento presentado es muy prometedor dada su simplicidad y la carencia de complejos análisis lingüísticos.

Modelo	**CEM**
Word2VecSpacy	0.6116
HV+RF	0.6214
ChainBOW	0.6535
BETO-2_modelos-aug	0.6891
BETO-multiclase-aug	0.6913
BETO-3_modelos-aug	0.704
BETO-transfer	0.7172
BETO-transfer-aug	0.7189
SINAI_team	0.7495

Cuadro 4.8: Resultados para la tarea 2 de DETOXIS

En la siguiente tabla se presenta el estudio de ablación para el mejor modelo la primera tarea de DETOXIS (*BETO-binario-aug*):

Modelo	***F1***
Mejor modelo	0.6000
Modelo por defecto (sin Grid-Search)	0.5526
Sin *augmentation*	0.5777
Todas las letras en minúsculas	0.5814
Sin preprocesado	0.5943

Cuadro 4.9: Estudio de ablación para los modelos de la tarea 1 de DETOXIS

Los resultados siguen en la línea de lo ya expuesto, con mejoras por cada técnica propuesta.

Cabe destacar en este caso que el proceso de *Data Augmentation* ha supuesto una mejora bastante considerable para el modelo, con una diferencia de casi 0.03. Por otro lado, se observa que, en este caso, el preprocesado tiene un papel ligeramente más relevante que en los otros corpus.

Capítulo 5

Conclusiones y trabajo futuro

En esta sección se expondrán las conclusiones a las que se ha llegado a lo largo de este trabajo y, finalmente, se enumerarán una serie de propuestas planteadas como trabajo futuro.

5.1. Conclusiones

A lo largo de este trabajo, se ha demostrado que el PLN puede ser de mucha ayuda detectando y clasificando ciertos tipos de comportamientos de interés en el texto escrito, ya sea en registros formales como las noticias o más informales como las redes sociales. Asimismo, también se ha comprobado que existe un gran margen de mejora, sobre todo para la lengua española.

Los resultados obtenidos por los sistemas presentados en este trabajo son muy prometedores dado su rendimiento y su simplicidad. Fruto de este trabajo, se han publicado tres papers en solitario y otro como co-autor principal en *Proceedings of the Iberian Languages Evaluation Forum*. Asimismo, se ha obtenido la segunda posición en dos de esas competiciones y en las restantes se han obtenido resultados muy prometedores. Estas publicaciones y estos resultados prueban que los modelos presentados están en el estado del arte al haber competido con los mejores investigadores de PLN de toda iberoamérica.

Por otra parte, como resultados técnicos se han conseguido los siguientes hitos relevantes:

- Se ha probado, una vez más, la crucial importancia de unos buenos hiperparámetros en las redes neuronales.
- Se ha propuesto un novedoso método de *Data Augmentation* en corpus plurilingües
- Se ha observado que las técnicas de *Data Augmentation* pueden llevar al sobreajuste o pueden mejorar la generalización, dependiendo del problema.
- Se ha demostrado que existen técnicas abiertas que no requieren excesivos recursos computacionales y que pueden alcanzar el estado del arte en problemas de clasificación de texto, ya sea binaria o multiclase

Finalmente, se ha demostrado que, pese a la falta de recursos preexistentes, existen las herramientas suficientes para crear modelos muy competentes en español y otras lenguas que sean de interés, ya no solo para la academia, sino también para la industria.

5.2. Trabajo futuro

A lo largo de este trabajo se ha podido ir observando que existen muchos caminos por los que seguir expandiendo esta investigación. De todos estos caminos, se han elegido cinco líneas de trabajo relevantes para ser exploradas en el futuro.

La primera sería obtener unos parámetros aun mejores para el modelo. Esto se podría hacer con una búsqueda del tipo *Population Based Training* [110], donde se eligen las muestras que mejor resultado han obtenido con una distribución de datos aleatoria y se usan esos parámetros con ligeras modificaciones aleatorias en la parte de la población con peores resultados. Esta búsqueda consigue explorar muchas opciones en muy poco tiempo y suele ser bastante mejor que búsquedas bayesianas ya que requieren hacer búsquedas mucho más exhaustivas [178].

La segunda sería hacer una mayor exploración de todos los métodos de *Data Augmentation* existentes y hacer pruebas combinándolos. Asimismo, la creación de un criterio para concretar si estos métodos van a ser útiles en función de los datos podría ahorrar mucho tiempo de experimentación.

Como los modelos del lenguaje están en continuo cambio y cada poco tiempo aparecen nuevas versiones, sería muy interesante hacer pruebas con los últimos modelos del estado del arte. Concretamente, ByT5 [179] se salta el paso de la tokenización y tokeniza a nivel de byte (ni siquiera de caracter). Esto se ha probado útil en muchos escenarios, concretamente en escenarios donde los usuarios cometen muchas faltas de ortografía, como podrían ser las redes sociales. Por otra parte, modelos como el RoFormer [180] han probado ser mucho más robustos en la clasificación de textos largos, como podrían ser noticias.

Por otra, están empezando a aparecer modelos del lenguaje entrenados con corpus de ciertos contextos específicos, por ejemplo de Twitter. El hecho de que un modelo esté preentrenado con datos en un registro muy parecido al del que tendrán los datos del *fine-tuning* puede desembocar en grandes mejoras en los resultados [112]. Ejemplo de estos modelos son TWilBert [23] para el español y BERTweet [181] para el inglés.

Finalmente, dado que se ha probado que los mejores modelos del lenguaje son aquellos entrenados en una gran cantidad de datos y con arquitecturas muy complejas computacionalmente, sería muy interesante explorar técnicas de entrenamiento colaborativo. Diskin et al. [182] demuestran que la comunidad puede unirse para entrenar modelos del lenguaje de forma distribuida para idiomas con pocos recursos preexistentes. De esta forma, se podría conseguir un modelo del lenguaje preentrenado en español con alguna arquitectura más moderna que la de BERT.

En resumen, el campo del PLN está experimentando una época de grandes avances constantes y existe una gran cantidad de recursos disponibles. Sin embargo, aun hay un gran trabajo por hacer por parte de la ciencia abierta y con presupuestos más bajos para poder llegar al nivel de las grandes multinacionales. De la misma manera,

la mayoría de estos avances se han dado para el inglés, sin embargo el español sigue muy atrás, de momento.

Bibliografía

[1] W. Mwangi, W. Cheruiyot y col., "A Survey of Information Retrieval Techniques," *Advances in Networks*, vol. 5, n.º 2, pág. 40, 2017.

[2] H. Lin y V. Ng, "Abstractive Summarization: A Survey of the State of the Art," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, págs. 9815-9822, jul. de 2019. DOI: 10.1609/aaai.v33i01.33019815.

[3] S. Yang, Y. Wang y X. Chu, *A Survey of Deep Learning Techniques for Neural Machine Translation*, 2020. arXiv: 2002.07526 [cs.CL].

[4] M. Malik, M. K. Malik, K. Mehmood e I. Makhdoom, "Automatic speech recognition: a survey," *Multimedia Tools and Applications*, vol. 80, n.º 6, págs. 9411-9457, mar. de 2021, ISSN: 1573-7721. DOI: 10.1007/s11042-020-10073-7. dirección: https://doi.org/10.1007/s11042-020-10073-7.

[5] Y. Wang, Y. Wang, J. Liu y Z. Liu, *A Comprehensive Survey of Grammar Error Correction*, 2020. arXiv: 2005.06600 [cs.CL].

[6] A. Clementeena y P. Sripriya, "A literature survey on question answering system in Natural Language Processing," *International Journal of Engineering and Technology(UAE)*, vol. 7, págs. 452-455, jun. de 2018. DOI: 10.14419/ijet.v7i2.33.14209.

[7] V. Yadav y S. Bethard, *A Survey on Recent Advances in Named Entity Recognition from Deep Learning models*, 2019. arXiv: 1910.11470 [cs.CL].

[8] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu y L. He, *A Survey on Text Classification: From Shallow to Deep Learning*, 2020. arXiv: 2008.00364 [cs.CL].

[9] A. Zouaq y R. Nkambou, "A Survey of Domain Ontology Engineering: Methods and Tools," en *Advances in Intelligent Tutoring Systems*, R. Nkambou, J. Bourdeau y R. Mizoguchi, eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, págs. 103-119, ISBN: 978-3-642-14363-2. DOI: 10.1007/978-3-642-14363-2_6. dirección: https://doi.org/10.1007/978-3-642-14363-2_6.

[10] J. Cambronero, H. Li, S. Kim, K. Sen y S. Chandra, *When Deep Learning Met Code Search*, 2019. arXiv: 1905.03813 [cs.SE].

[11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest y A. M. Rush, "Transformers: State-of-the-Art Natural Language Processing," en *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, oct. de 2020, págs. 38-45. dirección: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser e I. Polosukhin, *Attention Is All You Need*, 2017. eprint: arXiv:1706.03762.

[13] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2018. eprint: arXiv:1810.04805.

[14] *The most spoken languages*, https://blog.esl-languages.com/blog/learn-languages/most-spoken-languages-world/, Último acceso: 22-06-2021.

[15] G. G. Subies, *Estudio teórico sobre modelos de secuencias con redes neuronales recurrentes para la generación de texto*, 2019. dirección: https://github.com/GuillemGSubies/TFG.

[16] ——, *Estudio práctico sobre modelos de secuencias con redes neuronales recurrentes para la generación de texto*, 2019. dirección: https://github.com/GuillemGSubies/TFG.

[17] L. Chiruzzo, S. Castro, M. Etcheverry, D. Garat, J. Prada y A. Rosa, "Overview of HAHA at IberLEF 2019: Humor analysis based on human annotation," sep. de 2019.

[18] R. O. Bueno, F. M. R. Pardo, D. I. H. Farías, P. Rosso, M. Montes-y-Gómez y J. Medina-Pagola, "Overview of the Task on Irony Detection in Spanish Variants," en *IberLEF@SEPLN*, 2019.

[19] M. Aragon, M. A. Carmona, M. Montes, H. J. Escalante, L. Villaseñor-Pineda y D. Moctezuma, "Overview of MEX-A3T at IberLEF 2019: Authorship and aggressiveness analysis in Mexican Spanish tweets," ago. de 2019.

[20] M. C. Díaz-Galiano, M. Vega, E. Casasola, L. Chiruzzo, M. Á. G. Cumbreras, E. Martínez-Cámara, D. Moctezuma, A. M. Ráez, M. A. S. Cabezudo, E. S. Tellez, M. Graff y S. Miranda-Jiménez, "Overview of TASS 2019: One More Further for the Global Spanish Sentiment Analysis Corpus," en *IberLEF@SEPLN*, 2019.

[21] M. García-Vega, M. Díaz-Galiano, M. García-Cumbreras, F. Plaza-Del-Arco, A. Montejo-Ráez, S. M. Zafra, E. Martínez-Cámara, C. Aguilar, M. Antonio, S. Cabezudo, L. Chiruzzo y D. Moctezuma, "Overview of TASS 2020: Introducing Emotion Detection," sep. de 2020.

[22] J. Cañete, G. Chaperon, R. Fuentes y J. Pérez, "Spanish Pre-Trained BERT Model and Evaluation Data," en *to appear in PML4DC at ICLR 2020*, 2020.

[23] J. Á. González, L.-F. Hurtado y F. Pla, "TWilBert: Pre-trained Deep Bidirectional Transformers for Spanish Twitter," *Neurocomputing*, 2020, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2020.09.078. dirección: http://www.sciencedirect.com/science/article/pii/S0925231220316180.

[24] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei e I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019.

[25] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, Y.-L. Boureau y J. Weston, *Recipes for building an open-domain chatbot*, 2020. arXiv: 2004.13637 [cs.CL].

[26] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu y A. Fan, *Multilingual Translation with Extensible Multilingual Pretraining and Finetuning*, 2020. arXiv: 2008.00401 [cs.CL].

[27] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang y A. Ahmed, *Big Bird: Transformers for Longer Sequences*, 2021. arXiv: 2007.14062 [cs.LG].

[28] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua y C. Raffel, *mT5: A massively multilingual pre-trained text-to-text transformer*, 2021. arXiv: 2010.11934 [cs.CL].

[29] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang y M. Zhou, *ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training*, 2020. arXiv: 2001.04063 [cs.CL].

[30] K. Song, X. Tan, T. Qin, J. Lu y T.-Y. Liu, *MPNet: Masked and Permuted Pre-training for Language Understanding*, 2020. arXiv: 2004.09297 [cs.CL].

[31] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever y D. Amodei, "Language Models are Few-Shot Learners," 2020. arXiv: 2005.14165 [cs.CL].

[32] H. He y J. Lin, "Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement," en *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California: Association for Computational Linguistics, jun. de 2016, págs. 937-948. DOI: 10.18653/v1/N16-1108. dirección: https://www.aclweb.org/anthology/N16-1108.

[33] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman y E. Ruppin, "Placing search in context: The concept revisited," vol. 20, ene. de 2001, págs. 406-414. DOI: 10.1145/503104.503110.

[34] P. J. Ortiz Suárez, B. Sagot y L. Romary, "Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures," en *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*, P. Bański, A. Barbaresi, H. Biber, E. Breiteneder, S. Clematide, M. Kupietz, H. Lüngen y C. Iliadi, eds., Cardiff, United Kingdom: Leibniz-Institut für Deutsche Sprache, jul. de 2019. DOI: 10.14618/IDS-PUB-9021. dirección: https://hal.inria.fr/hal-02148693.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li y P. J. Liu, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, 2020. arXiv: 1910.10683 [cs.LG].

[36] Y. Bengio, R. Ducharme, P. Vincent y C. Janvin, "A Neural Probabilistic Language Model," *J. Mach. Learn. Res.*, vol. 3, n.º null, págs. 1137-1155, mar. de 2003, ISSN: 1532-4435.

[37] R. Bellman, R. Bellman y R. Corporation, *Dynamic Programming*, ép. Rand Corporation research study. Princeton University Press, 1957. dirección: https://books.google.es/books?id=rZW4ugAACAAJ.

[38] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*. 1961.

[39] G. V. Trunk, "A Problem of Dimensionality: A Simple Example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, n.º 3, págs. 306-307, 1979. DOI: 10.1109/TPAMI.1979.4766926.

[40] M. Verleysen y D. François, "The Curse of Dimensionality in Data Mining and Time Series Prediction," en *Proceedings of the 8th International Conference on Artificial Neural Networks: Computational Intelligence and Bioinspired Systems*, ép. IWANN'05, Barcelona, Spain: Springer-Verlag, 2005, págs. 758-770. DOI: 10.1007/11494669_93.

[41] M. Kusner, Y. Sun, N. Kolkin y K. Weinberger, "From Word Embeddings To Document Distances," en *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach y D. Blei, eds., ép. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, jul. de 2015, págs. 957-966. dirección: http://proceedings.mlr.press/v37/kusnerb15.html.

[42] I. Moya, M. Chica, J. L. Sáez-Lozano y Ó. Cordón, "An agent-based model for understanding the influence of the 11-M terrorist attacks on the 2004 Spanish elections," *Knowledge-Based Systems*, vol. 123, págs. 200-216, 2017, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2017.02.015. dirección: https://www.sciencedirect.com/science/article/pii/S0950705117300825.

[43] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss y J. Han, *Automated Phrase Mining from Massive Text Corpora*, 2017. arXiv: 1702.04457 [cs.CL].

[44] T. Mikolov, K. Chen, G. Corrado y J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[45] O. Levy e Y. Goldberg, "Dependency-Based Word Embeddings," en *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, Maryland: Association for Computational Linguistics, jun. de 2014, págs. 302-308. DOI: 10.3115/v1/P14-2050. dirección: https://www.aclweb.org/anthology/P14-2050.

[46] Y. Liu, Z. Liu, T.-S. Chua y M. Sun, "Topical Word Embeddings," en *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ép. AAAI'15, Austin, Texas: AAAI Press, 2015, págs. 2418-2424, ISBN: 0262511290.

[47] P. Bojanowski, E. Grave, A. Joulin y T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, págs. 135-146, 2017.

[48] J. J. Lastra-Díaz, J. Goikoetxea, M. A. Hadj Taieb, A. García-Serrano, M. Ben Aouicha y E. Agirre, "A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art," *Engineering Applications of Artificial Intelligence*, vol. 85, págs. 645-665, 2019, ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2019.07.010. dirección: https://www.sciencedirect.com/science/article/pii/S0952197619301745.

[49] Z. S. Harris, "Distributional structure," *Word*, vol. 10, n.º 2-3, págs. 146-162, 1954.

[50] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, 1972.

[51] J. Camacho-Collados y M. T. Pilevar, "From Word To Sense Embeddings: A Survey on Vector Representations of Meaning," *Journal of Artificial Intelligence Research*, vol. 63, págs. 743-788, dic. de 2018. DOI: 10.1613/jair.1.11259.

[52] J. Pennington, R. Socher y C. D. Manning, "GloVe: Global Vectors for Word Representation," en *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, págs. 1532-1543. dirección: http://www.aclweb.org/anthology/D14-1162.

[53] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado y J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," en *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani y K. Q. Weinberger, eds., Curran Associates, Inc., 2013, págs. 3111-3119. dirección: http://papers.nips.cc/paper/5021-

distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[54] A. Joulin, E. Grave, P. Bojanowski y T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[55] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou y T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[56] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun y S. Fidler, *Skip-Thought Vectors*, 2015. arXiv: 1506.06726 [cs.CL].

[57] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis y K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Systems with Applications*, vol. 69, págs. 214-224, 2017, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2016.10.043. dirección: https://www.sciencedirect.com/science/article/pii/S095741741630584X.

[58] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee y L. Zettlemoyer, *Deep contextualized word representations*, 2018. arXiv: 1802.05365 [cs.CL].

[59] I. Sutskever, O. Vinyals y Q. V. Le, *Sequence to Sequence Learning with Neural Networks*, 2014. arXiv: 1409.3215 [cs.CL].

[60] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk e Y. Bengio, *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, 2014. arXiv: 1406.1078 [cs.CL].

[61] M.-T. Luong, H. Pham y C. D. Manning, *Effective Approaches to Attention-based Neural Machine Translation*, 2015. arXiv: 1508.04025 [cs.CL].

[62] D. Bahdanau, K. Cho e Y. Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate*, 2016. arXiv: 1409.0473 [cs.CL].

[63] H. Levesque, E. Davis y L. Morgenstern, "The winograd schema challenge," en *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, Citeseer, 2012.

[64] J. Gehring, M. Auli, D. Grangier, D. Yarats e Y. N. Dauphin, *Convolutional Sequence to Sequence Learning*, 2017. arXiv: 1705.03122 [cs.CL].

[65] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves y K. Kavukcuoglu, *Neural Machine Translation in Linear Time*, 2017. arXiv: 1610.10099 [cs.CL].

[66] A. Radford e I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018.

[67] K. Sakaguchi, R. L. Bras, C. Bhagavatula e Y. Choi, *WinoGrande: An Adversarial Winograd Schema Challenge at Scale*, 2019. arXiv: 1907.10641 [cs.CL].

[68] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma y R. Soricut, *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*, 2020. arXiv: 1909.11942 [cs.CL].

[69] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov y L. Zettlemoyer, *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*, 2019. eprint: arXiv:1910.13461.

[70] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer y O. Levy, *SpanBERT: Improving Pre-training by Representing and Predicting Spans*, 2020. arXiv: 1907.10529 [cs.CL].

[71] J. Zhang, Y. Zhao, M. Saleh y P. J. Liu, *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*, 2020. arXiv: 1912.08777 [cs.CL].

[72] P. He, X. Liu, J. Gao y W. Chen, *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*, 2021. arXiv: 2006.03654 [cs.CL].

[73] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy y S. R. Bowman, *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*, 2020. arXiv: 1905.00537 [cs.CL].

[74] G. Lample y A. Conneau, *Cross-lingual Language Model Pretraining*, 2019. eprint: arXiv:1901.07291.

[75] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov y Q. V. Le, *XLNet: Generalized Autoregressive Pretraining for Language Understanding*, 2019. eprint: arXiv:1906.08237.

[76] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer y V. Stoyanov, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, 2019. eprint: arXiv:1907.11692.

[77] V. Sanh, L. Debut, J. Chaumond y T. Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, 2019. eprint: arXiv:1910.01108.

[78] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer y V. Stoyanov, *Unsupervised Cross-lingual Representation Learning at Scale*, 2019. eprint: arXiv:1911.02116.

[79] K. Clark, M.-T. Luong, Q. V. Le y C. D. Manning, *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*, 2020. arXiv: 2003.10555 [cs.CL].

[80] J. Lu, D. Batra, D. Parikh y S. Lee, *ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks*, 2019. arXiv: 1908.02265 [cs.CV].

[81] C. Sun, A. Myers, C. Vondrick, K. Murphy y C. Schmid, *VideoBERT: A Joint Model for Video and Language Representation Learning*, 2019. arXiv: 1904.01766 [cs.CV].

[82] A. Baevski, H. Zhou, A. Mohamed y M. Auli, *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, 2020. arXiv: 2006.11477 [cs.CL].

[83] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah y B. Sagot, "CamemBERT: a Tasty French Language Model," 2019. DOI: 10.18653/v1/2020.acl-main.645. eprint: arXiv:1911.03894.

[84] H. Le, L. Vial, J. Frej, V. Segonne, M. Coavoux, B. Lecouteux, A. Allauzen, B. Crabbé, L. Besacier y D. Schwab, *FlauBERT: Unsupervised Language Model Pre-training for French*, 2019. eprint: arXiv:1912.05372.

[85] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord y M. Nissim, *BERTje: A Dutch BERT Model*, 2019. eprint: arXiv:1912.09582.

[86] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter y S. Pyysalo, *Multilingual is not enough: BERT for Finnish*, 2019. eprint: arXiv:1912.07076.

[87] R. Agerri, I. S. Vicente, J. A. Campos, A. Barrena, X. Saralegi, A. Soroa y E. Agirre, *Give your Text Representation Models some Love: the Case for Basque*, 2020. eprint: arXiv:2004.00033.

[88] B. Chan, S. Schweter y T. Möller, *German's Next Language Model*, 2020. arXiv: 2010.10906 [cs.CL].

[89] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang y G. Hu, "Revisiting Pre-Trained Models for Chinese Natural Language Processing," en *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, Online: Association for Computational Linguistics, nov. de 2020, págs. 657-668. dirección: https://www.aclweb.org/anthology/2020.findings-emnlp.58.

[90] F. Souza, R. Nogueira y R. Lotufo, "BERTimbau: pretrained BERT models for Brazilian Portuguese," en *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020.

[91] Y. Kuratov y M. Arkhipov, *Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language*, 2019. arXiv: 1905.07213 [cs.CL].

[92] A. Safaya, M. Abdullatif y D. Yuret, "KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media," en *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona (online): International Committee for Computational Linguistics, dic. de 2020, págs. 2054-2059. dirección: https://www.aclweb.org/anthology/2020.semeval-1.271.

[93] M. Farahani, M. Gharachorloo, M. Farahani y M. Manthouri, *ParsBERT: Transformer-based Model for Persian Language Understanding*, 2020. arXiv: 2005.12515 [cs.CL].

[94] J. Cañete, *Compilation of Large Spanish Unannotated Corpora*, Zenodo, mayo de 2019. DOI: 10.5281/zenodo.3247731. dirección: https://doi.org/10.5281/zenodo.3247731.

[95] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain," *Psychological Review*, págs. 65-386, 1958.

[96] S. S. Haykin, *Neural networks and learning machines*. Pearson Education, 2009, pág. 124.

[97] Y. Bengio, P. Simard y P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, n.º 2, págs. 157-166, mar. de 1994, ISSN: 1045-9227. DOI: 10.1109/72.279181.

[98] S. Hochreiter, Y. Bengio y P. Frasconi, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," J. Kolen y S. Kremer, eds., 2001.

[99] D. E. Rumelhart, G. E. Hinton y R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, págs. 533-, oct. de 1986. dirección: http://dx.doi.org/10.1038/323533a0.

[100] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, n.º 8, págs. 2554-2558, 1982, ISSN: 0027-8424. dirección: http://view.ncbi.nlm.nih.gov/pubmed/6953413.

[101] R. Pascanu, T. Mikolov e Y. Bengio, "On the Difficulty of Training Recurrent Neural Networks," en *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ép. ICML'13, Atlanta, GA, USA: JMLR.org, 2013, págs. III-1310-III-1318. dirección: http://dl.acm.org/citation.cfm?id=3042817.3043083.

[102] S. Hochreiter y J. Schmidhuber, "Long Short-term Memory," *Neural computation*, vol. 9, págs. 1735-80, dic. de 1997. DOI: 10.1162/neco.1997.9.8.1735.

[103] A. Rosá, L. A. Alemany, I. Castellón, L. Chiruzzo, H. Curell, A. F. Montraveta, S. Góngora, M. Malcuori, G. Vázquez y D. Wonsever, "Overview of FACT at IberLEF 2020: Events Detection and Classification," en *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, M. ángel García Cumbreras, J. Gonzalo, E. M. Cámara, R. Martínez-Unanue, P. Rosso, S. M. J. Zafra, J. A. O. Zambrano, A. Miranda, J. P. Zamorano, Y. Gutiérrez, A. Rosá, M. Montes-y-Gómez y M. G. Vega, eds., ép. CEUR Workshop Proceedings, vol. 2664, CEUR-WS.org, 2020, págs. 197-205. dirección: http://ceur-ws.org/Vol-2664/fact%5C_overview.pdf.

[104] M. E. Aragón, H. J. Jarquín-Vásquez, M. Montes-y-Gómez, H. J. Escalante, L. V. Pineda, H. Gómez-Adorno, J. P. Posadas-Durán y G. Bel-Enguix, "Overview of MEX-A3T at IberLEF 2020: Fake News and Aggressiveness Analysis in Mexican Spanish," en *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020*, M. ángel García Cumbreras, J. Gonzalo, E. M. Cámara, R. Martínez-Unanue, P. Rosso, S. M. J. Zafra, J. A. O. Zambrano, A. Miranda, J. P. Zamorano, Y. Gutiérrez, A. Rosá, M. Montes-y-Gómez y M. G. Vega, eds., ép. CEUR Workshop Proceedings, vol. 2664, CEUR-WS.org, 2020, págs. 222-235. dirección: http://ceur-ws.org/Vol-2664/mex-a3t%5C_overview.pdf.

[105] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk e Y. Bengio, *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, 2014. eprint: arXiv:1406.1078.

[106] J. Chung, C. Gulcehre, K. Cho e Y. Bengio, *Gated Feedback Recurrent Neural Networks*, 2015. eprint: arXiv:1502.02367.

[107] ——, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014. dirección: https://arxiv.org/abs/1412.3555.

[108] M. Schuster y K. Paliwal, "Bidirectional Recurrent Neural Networks," *Trans. Sig. Proc.*, vol. 45, n.º 11, págs. 2673-2681, nov. de 1997, ISSN: 1053-587X. DOI: 10.1109/78.650093.

[109] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," en *Neurocomputing*, F. F. Soulié y J. Hérault, eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, págs. 227-236, ISBN: 978-3-642-76153-9.

[110] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando y K. Kavukcuoglu, *Population Based Training of Neural Networks*, 2017. arXiv: 1711.09846 [cs.LG].

[111] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez e I. Stoica, "Tune: A Research Platform for Distributed Model Selection and Training," *arXiv preprint arXiv:1807.05118*, 2018.

[112] C. Sun, X. Qiu, Y. Xu y X. Huang, *How to Fine-Tune BERT for Text Classification?* 2020. arXiv: 1905.05583 [cs.CL].

[113] M. McCloskey y N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," en, ép. Psychology of Learning and Motivation, G. H. Bower, ed., vol. 24, Academic Press, 1989, págs. 109-165.

DOI: https://doi.org/10.1016/S0079-7421(08)60536-8. dirección: https://www.sciencedirect.com/science/article/pii/S0079742108605368.

[114] X. Wang, K. Wang y S. Lian, "A survey on face data augmentation for the training of deep neural networks," *Neural Computing and Applications*, vol. 32, n.º 19, págs. 15 503-15 531, mar. de 2020, ISSN: 1433-3058. DOI: 10.1007/s00521-020-04748-3. dirección: http://dx.doi.org/10.1007/s00521-020-04748-3.

[115] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah y A. Hussain, "Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study," *IEEE Access*, vol. 4, págs. 7940-7957, 2016. DOI: 10.1109/ACCESS.2016.2619719.

[116] N. Rout, "Handling Imbalanced Data: A Survey," ene. de 2018.

[117] A. Chaudhary, *A Visual Survey of Data Augmentation in NLP*, https://amitness.com/2020/05/data-augmentation-for-nlp, 2020.

[118] E. Ma, *NLP Augmentation*, https://github.com/makcedward/nlpaug, 2019.

[119] J. Wei y K. Zou, *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*, 2019. arXiv: 1901.11196 [cs.CL].

[120] X. Zhang, J. Zhao e Y. LeCun, *Character-level Convolutional Networks for Text Classification*, 2016. arXiv: 1509.01626 [cs.LG].

[121] A. Thyagarajan y J. Mueller, "Siamese Recurrent Architectures for Learning Sentence Similarity," nov. de 2015.

[122] W. Y. Wang y D. Yang, "That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using #petpeeve Tweets," en *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal: Association for Computational Linguistics, sep. de 2015, págs. 2557-2563. DOI: 10.18653/v1/D15-1306. dirección: https://www.aclweb.org/anthology/D15-1306.

[123] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang y Q. Liu, *TinyBERT: Distilling BERT for Natural Language Understanding*, 2020. arXiv: 1909.10351 [cs.CL].

[124] S. Garg y G. Ramakrishnan, *BAE: BERT-based Adversarial Examples for Text Classification*, 2020. arXiv: 2004.01970 [cs.CL].

[125] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong y Q. V. Le, *Unsupervised Data Augmentation for Consistency Training*, 2020. arXiv: 1904.12848 [cs.LG].

[126] R. Sennrich, B. Haddow y A. Birch, "Improving Neural Machine Translation Models with Monolingual Data," en *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, ago. de 2016, págs. 86-96. DOI: 10.18653/v1/P16-1009. dirección: https://www.aclweb.org/anthology/P16-1009.

[127] S. Edunov, M. Ott, M. Auli y D. Grangier, *Understanding Back-Translation at Scale*, 2018. arXiv: 1808.09381 [cs.CL].

[128] F. M. Luque, *Atalaya at TASS 2019: Data Augmentation and Robust Embeddings for Sentiment Analysis*, 2019. arXiv: 1909.11241 [cs.CL].

[129] R. Grundkiewicz y M. Junczys-Dowmunt, "The WikEd Error Corpus: A Corpus of Corrective Wikipedia Edits and Its Application to Grammatical Error Correction," en *PolTAL*, 2014.

[130] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky y A. Y. Ng, *Data Noising as Smoothing in Neural Network Language Models*, 2017. arXiv: 1703.02573 [cs.LG].

[131] C. Coulombe, *Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs*, 2018. arXiv: 1812.04718 [cs.CL].

[132] A. Anaby-Tavor, B. Carmeli, E. Goldbraich, A. Kantor, G. Kour, S. Shlomov, N. Tepper y N. Zwerdling, *Not Enough Data? Deep Learning to the Rescue!* 2019. arXiv: 1911.03118 [cs.CL].

[133] V. Kumar, A. Choudhary y E. Cho, *Data Augmentation using Pre-trained Transformer Models*, 2021. arXiv: 2003.02245 [cs.CL].

[134] H. Zhang, M. Cisse, Y. N. Dauphin y D. Lopez-Paz, *mixup: Beyond Empirical Risk Minimization*, 2018. arXiv: 1710.09412 [cs.LG].

[135] H. Guo, Y. Mao y R. Zhang, *Augmenting Data with Mixup for Sentence Classification: An Empirical Study*, 2019. arXiv: 1905.08941 [cs.CL].

[136] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng y C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," en *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Computational Linguistics, oct. de 2013, págs. 1631-1642. dirección: https://www.aclweb.org/anthology/D13-1170.

[137] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy y S. R. Bowman, *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*, 2019. arXiv: 1804.07461 [cs.CL].

[138] Z. Sun, C. Fan, Q. Han, X. Sun, Y. Meng, F. Wu y J. Li, *Self-Explaining Structures Improve NLP Models*, 2020. arXiv: 2012.01786 [cs.CL].

[139] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng y C. Potts, "Learning Word Vectors for Sentiment Analysis," en *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, jun. de 2011, págs. 142-150. dirección: https://www.aclweb.org/anthology/P11-1015.

[140] M. Montes, P. Rosso, J. Gonzalo, E. Aragón, R. Agerri, M. Á. Álvarez-Carmona, E. Á. Mellado, J. Carrillo-de-Albornoz, L. Chiruzzo, L. Freitas, H. G. Adorno, Y. Gutiérrez, S. M. J. Zafra, S. Lima, F. M. Plaza-de-Arco y M. Taulé, "Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021)," en, en *CEUR Workshop Proceedings*, 2021.

[141] F. Rodríguez-Sánchez, J. Carrillo-de-Albornoz, L. Plaza, J. Gonzalo, P. Rosso, M. Comet y T. Donoso, "Overview of EXIST 2021: sEXism Identification in Social neTworks," *Procesamiento del Lenguaje Natural*, vol. 67, n.º 0, 2021, ISSN: 1989-7553.

[142] U. Brisolara Corrêa, L. Pereira dos Santos, L. Coelho y L. A. de Freitas, "IDPT2021 at IberLEF: Overview of the Task on Irony Detection in Portuguese," *Proceedings of the Iberian Languages Evaluation Fórum (IberLEF 2021), co-located with the 37th Conference of the Spanish Society for Natural Language Processing (SEPLN 2021). CEUR Workshop Proceedings, 2021*, 2021.

[143] L. Freitas, A. Vanin, D. Hogetop, M. Bochernitsan y R. Vieira, "Pathways for irony detection in tweets," mar. de 2014, ISBN: 978-1-4503-2469-4. DOI: 10.1145/2554850.2555048.

[144] F. Araujo da Silva, "Detecção de Ironia e Sarcasmo em Língua Portuguesa: uma abordagem utilizando Deep Learning," Tesis doct., feb. de 2018. DOI: 10.13140/RG.2.2.18896.81924.

[145] G. Schubert y L. Freitas, *The Construction of a Corpus for Detecting Irony and Sarcasm in Portuguese*, oct. de 2020.

[146] R. González-Ibáñez, S. Muresan y N. Wacholder, "Identifying Sarcasm in Twitter: A Closer Look," en *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, jun. de 2011, págs. 581-586. dirección: https://www.aclweb.org/anthology/P11-2102.

[147] B. Ghanem, J. Karoui, F. Benamara, V. Moriceau y P. Rosso, "IDAT at FIRE2019: Overview of the Track on Irony Detection in Arabic Tweets," dic. de 2019, págs. 10-13. DOI: 10.1145/3368567.3368585.

[148] R. K. Gupta e Y. Yang, "CrystalNest at SemEval-2017 Task 4: Using Sarcasm Detection for Enhancing Sentiment Classification and Quantification," en *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada: Association for Computational Linguistics, ago. de 2017, págs. 626-633. DOI: 10.18653/v1/S17-2103. dirección: https://www.aclweb.org/anthology/S17-2103.

[149] S. Frenda, A. Cignarella, V. Basile, C. Bosco, V. Patti y P. Rosso, "IronITA @ EVALITA 2018 Irony Detection in Italian Tweets Task Guidelines," sep. de 2018.

[150] M. Taulé, A. Ariza, M. Nofre, E. Amigó y P. Rosso, "Overview of the DETOXIS Task at IberLEF-2021: DEtection of TOXicity in comments In Spanish," *Procesamiento del Lenguaje Natural*, vol. 67, 2021.

[151] Ó. Garibo i Orts, "Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter at SemEval-2019 Task 5: Frequency Analysis Interpolation for Hate in Speech Detection," en *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, jun. de 2019, págs. 460-463. DOI: 10.18653/v1/S19-2081. dirección: https://www.aclweb.org/anthology/S19-2081.

[152] R. Kumar, A. K. Ojha, S. Malmasi y M. Zampieri, "Evaluating Aggression Identification in Social Media," English, en *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, Marseille, France: European Language Resources Association (ELRA), mayo de 2020, págs. 1-5, ISBN: 979-10-95546-56-6. dirección: https://www.aclweb.org/anthology/2020.trac-1.1.

[153] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis y Ç. Çöltekin, *SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)*, 2020. arXiv: 2006.07235 [cs.CL].

[154] J. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand y M. Klenner, "Overview of GermEval Task 2, 2019 Shared Task on the Identification of Offensive Language," oct. de 2019.

[155] E. Amigó, J. Gonzalo, S. Mizzaro y J. Carrillo-de-Albornoz, *An Effectiveness Metric for Ordinal Classification: Formal Properties and Experimental Results*, 2020. arXiv: 2006.01245 [cs.CL].

[156] L. Chiruzzo, S. Castro, S. Góngora, A. Rosá, J. A. Meaney y R. Mihalcea, "Overview of HAHA at IberLEF 2021: Detecting, Rating and Analyzing Humor

in Spanish," *Procesamiento del Lenguaje Natural*, vol. 67, n.º 0, 2021, ISSN: 1989-7553.

[157] S. Castro, L. Chiruzzo y A. Rosa, "Overview of the HAHA Task: Humor Analysis based on Human Annotation at IberEval 2018," sep. de 2018.

[158] R. Mihalcea y C. Strapparava, "Making Computers Laugh: Investigations in Automatic Humor Recognition," en *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada: Association for Computational Linguistics, oct. de 2005, págs. 531-538. dirección: https://www.aclweb.org/anthology/H05-1067.

[159] J. Sjöbergh y K. Araki, "Recognizing Humor Without Recognizing Meaning," vol. 4578, jul. de 2007, págs. 469-476, ISBN: 978-3-540-73399-7. DOI: 10.1007/978-3-540-73400-0_59.

[160] S. Castro, M. Cubero, D. Garat y G. Moncecchi, "Is This a Joke? Detecting Humor in Spanish Tweets," *Advances in Artificial Intelligence - IBERAMIA 2016*, págs. 139-150, 2016, ISSN: 1611-3349. DOI: 10.1007/978-3-319-47955-2_12. dirección: http://dx.doi.org/10.1007/978-3-319-47955-2_12.

[161] S. Castro, L. Chiruzzo, A. Rosá, D. Garat y G. Moncecchi, "A Crowd-Annotated Spanish Corpus for Humor Analysis," en *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, Melbourne, Australia: Association for Computational Linguistics, jul. de 2018, págs. 7-11. DOI: 10.18653/v1/W18-3502. dirección: https://www.aclweb.org/anthology/W18-3502.

[162] L. Chiruzzo, S. Castro y A. Rosá, "HAHA 2019 Dataset: A Corpus for Humor Analysis in Spanish," English, en *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France: European Language Resources Association, mayo de 2020, págs. 5106-5112, ISBN: 979-10-95546-34-4. dirección: https://www.aclweb.org/anthology/2020.lrec-1.628.

[163] M. Honnibal, I. Montani, S. Van Landeghem y A. Boyd, *spaCy: Industrial-strength Natural Language Processing in Python*, 2020. DOI: 10.5281/zenodo.1212303. dirección: https://doi.org/10.5281/zenodo.1212303.

[164] P. Qi, Y. Zhang, Y. Zhang, J. Bolton y C. D. Manning, "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages," en *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020. dirección: https://nlp.stanford.edu/pubs/qi2020stanza.pdf.

[165] E. Loper y S. Bird, "NLTK: The Natural Language Toolkit," *CoRR*, vol. cs.CL/0205028, 2002. dirección: http://dblp.uni-trier.de/db/journals/corr/corr0205.html#cs-CL-0205028.

[166] R. Rehurek y P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, n.º 2, 2011.

[167] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai y S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," en *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox y R. Garnett, eds., Curran Associates, Inc., 2019, págs. 8024-8035. dirección: http://papers.neurips.cc/

paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[168] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu y X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. dirección: https://www.tensorflow.org/.

[169] D. Davidov, O. Tsur y A. Rappoport, "Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon," en *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, ép. CoNLL '10, Uppsala, Sweden: Association for Computational Linguistics, 2010, págs. 107-116, ISBN: 9781932432831.

[170] T. Wolf, Q. Lhoest, P. von Platen, Y. Jernite, M. Drame, J. Plu, J. Chaumond, C. Delangue, C. Ma, A. Thakur, S. Patil, J. Davison, T. L. Scao, V. Sanh, C. Xu, N. Patry, A. McMillan-Major, S. Brandeis, S. Gugger, F. Lagunas, L. Debut, M. Funtowicz, A. Moi, S. Rush, P. Schmidd, P. Cistac, V. Muštar, J. Boudier y A. Tordjmann, "Datasets," *GitHub. Note: https://github.com/huggingface/datasets*, vol. 1, 2020.

[171] T. pandas development team, *pandas-dev/pandas: Pandas*, ver. latest, feb. de 2020. DOI: 10.5281/zenodo.3509134. dirección: https://doi.org/10.5281/zenodo.3509134.

[172] W. McKinney, "Data Structures for Statistical Computing in Python," en *Proceedings of the 9th Python in Science Conference*, S. van der Walt y J. Millman, eds., 2010, págs. 56-61. DOI: 10.25080/Majora-92bf1922-00a.

[173] T. K. Ho, "Random decision forests," en *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, vol. 1, 1995, págs. 278-282.

[174] A. Appleby, *MurmurHash*, 2008.

[175] J. Tiedemann y S. Thottingal, "OPUS-MT — Building open translation services for the World," en *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

[176] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins y A. Birch, "Marian: Fast Neural Machine Translation in C++," en *Proceedings of ACL 2018, System Demonstrations*, Melbourne, Australia: Association for Computational Linguistics, jul. de 2018, págs. 116-121. DOI: 10.18653/v1/P18-4020. dirección: https://www.aclweb.org/anthology/P18-4020.

[177] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, págs. 2825-2830, 2011.

[178] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe e I. Shcherbatyi, *scikit-optimize/scikit-optimize*, ver. v0.8.1, sep. de 2020. DOI: 10.5281/zenodo.4014775. dirección: https://doi.org/10.5281/zenodo.4014775.

[179] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts y C. Raffel, *ByT5: Towards a token-free future with pre-trained byte-to-byte models*, 2021. arXiv: 2105.13626 [cs.CL].

[180] J. Su, Y. Lu, S. Pan, B. Wen e Y. Liu, *RoFormer: Enhanced Transformer with Rotary Position Embedding*, 2021. arXiv: 2104.09864 [cs.CL].

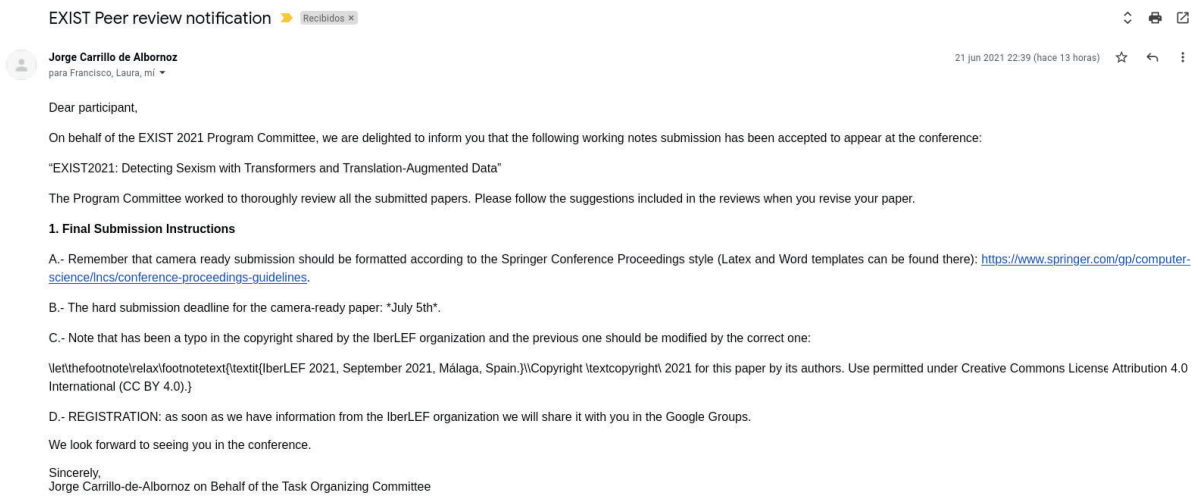
[181] D. Q. Nguyen, T. Vu y A. T. Nguyen, "BERTweet: A pre-trained language model for English Tweets," en *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.

[182] M. Diskin, A. Bukhtiyarov, M. Ryabinin, L. Saulnier, Q. Lhoest, A. Sinitsin, D. Popov, D. Pyrkin, M. Kashirin, A. Borzunov, A. V. del Moral, D. Mazur, I. Kobelev, Y. Jernite, T. Wolf y G. Pekhimenko, *Distributed Deep Learning in Open Collaborations*, 2021. arXiv: 2106.10207 [cs.LG].

Apéndice A

Paper EXIST en Proceedings of the Iberian Languages Evaluation Forum

A.1. Aceptación



EXIST Peer review notification Recibidos ×

Jorge Carrillo de Albornoz
para Francisco, Laura, mí

21 jun 2021 22:39 (hace 13 horas)

Dear participant,

On behalf of the EXIST 2021 Program Committee, we are delighted to inform you that the following working notes submission has been accepted to appear at the conference:

"EXIST2021: Detecting Sexism with Transformers and Translation-Augmented Data"

The Program Committee worked to thoroughly review all the submitted papers. Please follow the suggestions included in the reviews when you revise your paper.

1. Final Submission Instructions

A.- Remember that camera ready submission should be formatted according to the Springer Conference Proceedings style (Latex and Word templates can be found there): https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines.

B.- The hard submission deadline for the camera-ready paper: *July 5th*.

C.- Note that has been a typo in the copyright shared by the IberLEF organization and the previous one should be modified by the correct one:

\let\thefootnote\relax\footnotetext{\textit{IberLEF 2021, September 2021, Málaga, Spain.}\\Copyright \textcopyright\ 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).}

D.- REGISTRATION: as soon as we have information from the IberLEF organization we will share it with you in the Google Groups.

We look forward to seeing you in the conference.

Sincerely,
Jorge Carrillo-de-Albornoz on Behalf of the Task Organizing Committee

A.2. *Paper*

A continuación se adjunta el *paper* enviado después de las *peer reviews* dado que aun no se ha publicado la revista.

EXIST2021: Detecting Sexism with Transformers and Translation-Augmented Data

Guillem García Subies[1]

Instituto de Ingeniería del Conocimiento, Francisco Tomás y Valiente st., 11 EPS, B Building, 5th florr UAM Cantoblanco. 28049 Madrid, Spain
guillem.garcia@iic.uam.es

Abstract. This paper describes a system created for the EXIST 2021 shared task, framed within the IberLEF 2021 workshop. We present an approach mainly based in fine-tuned BERT models and Data Augmentation with translation and backtranslation. We show an approach to face multilingual problems augmenting the data without the overfitting that an aggressive backtranslation can generate. Our models far outperform the baselines and achieve results close to to the state-of-the-art.

Keywords: Sexism Detection · BERT · Transformers · Data Augmentation · Backtranslation · Multilingual Corpora

1 Introduction

With the crescent trends in social rights and equal rights demands, NLP can help in detecting harmful and sexist behaviors. The EXIST (sEXism Identification in Social neTworks) [15] shared task proposes, during this third edition of the IberLEF [11] workshop, a dataset to detect sexism in it's most broad definition and also kinds of sexism.

This article summarizes our participation in all the EXIST tasks. Given the success of Transformer-inspired language models [19], both in academia and industry [20], we decided to use already pre-trained BERT [4] models. Specifically, we face the multilingual problem using different models for every language. We also conjecture that a good way to augment the data in multilingual problems is translating the data into the other languages of the dataset, so instead of having n_i for every language, we have $\sum n_i$ samples for every language. As the dataset is not too big, we also explore the Data Augmentation with Backtranslation [16].

In the next section, we will briefly see some previous work related to this topic. In Section 3 we will go through a brief description of the tasks and the corpora. Then, in Section 4, we will explain the main ideas behind the proposed models. In Section 5 we will present a summary of the experiments we carried out and the results we got. Finally, in Section 6 we will expose the main conclusions of our work and results and we will also propose some ideas for future work.

2 Related Work

There is an extensive bibliography on Sentiment Analysis and text classification in social networks, however not that much work has been done about identifying and classifying sexist behaviors in different languages.

For instance, Anzovino et al. [1] propose a sexism classification dataset also in Spanish and in English and proposed some solutions based on n-grams and classic machine learning models like SVMs. Following that line, Frenda et al. [5] use the previous dataset (only the English part) in combination with others to detect both misogyny and sexism following a similar approach of classic NLP.

More recent work by Grosz et al. [7], focuses on the sexism in the workplace and they obtain stat-of-the art results with GloVe embeddings and modified LSTM so they have attention mechanisms.

3 Tasks Description

The main corpus consists of 6977 tweets both in English and Spanish for the train split and 3368 tweets and 982 "gabs" (from gab.com) also in both languages for the test one.

The first task, Sexism Identification, consists of classifying tweets between **sexist** and **non-sexist**. There are 3600 **non-sexist** tweets and 3377 **sexist tweets**, so we can consider that the problem is well-balanced. The metric used for this dataset is the Accuracy.

For the second task, Sexism Categorization, there are six classes: **non-sexist**, **ideological-inequality**, **stereotyping-dominance**, **misogyny-non-sexual-violence**, **sexual-violence** and **objectification**. As we can see in Table 1, the dataset is unbalanced, so the F-measure is used as the ranking metric. We can see a similar distribution of the classes if we split the corpora into Spanish and English.

Class		**Nº Samples Task1**	**Nº Samples Task2**
non-sexist		3600	3600
ideological-inequality	sexist	3377	866
stereotyping-dominance			809
misogyny-non-sexual-violence			685
sexual-violence			517
objectification			500

Table 1. Distribution of Samples

In the table below, we can see some illustrative examples of the data and their labels:

I love poetry books, so I'm reading the one i have on this plane flight and one of the flight attendants (black women) goes "it's good to see a brotha reading something that's is so deep"	non-sexist
Can the fellas participate or is this just for the ladies/Non binary people because I don't wanna get clowned.	ideological-inequality
ive been sooo interesting my whole life and i just want to be a boring trophy wife now	stereotyping-dominance
Fucking skank	misogyny-non-sexual-violence
Bitches be begging me to fw them just to give me a reason not to fw them. Lol	sexual-violence
some women just don't deserve onlyfans, bitches be UGLY as fuck and ask you to pay $20 to see their UGLY FAT BLOTCHED TITTIES, BITCH!	objectification

Table 2. Examples of the different classes

4 Models

4.1 Data Preprocessing

We performed a simple preprocessing where we substituted some expressions with a more normalized form:

- Every URL was replaced with the token "[URL]" so we don't get strange tokens when the tokenizer tries to process and URL. Furthermore, no semantic information about sexism can be inferred from a URL, the only information relevant for the model is that there is a URL in that token.
- The hashtag characters ("#") were deleted ("#example" → "example") because the base language models we will use, are trained in generic text and might not understand their meaning. Furthermore, most of hashtags are used the same way as normal words.
- We replaced every username with the generic token "[USER]" because the exact name of a user does not really add any information about the sexism. The only relevant feature is knowing if someone was mentioned or not, but not who.
- Finally we normalized every laugh ("jasjajajajj" → "haha") so we minimize the noise of the misspellings, common in social networks.

4.2 Baselines

We created some baselines so we can compare our models properly. We selected a HashingVectorizer + RandomForest and a multilingual BERT (mBERT). This way, we can compare our models to a classic feature extraction model and a simple BERT-based one.

4.3 Language Models

We decided to fine-tune one language model for every language. For the Spanish language, we selected BETO [3], a BERT model trained with the Spanish Unannotated Corpora (SUC) [2] that has proven to be much better than the multilingual BERT model. For the English part of the dataset we used the original BERT model [4].

For the second task, given the imbalance in the classes, we performed a hierarchical classification, where the model from Task 1 classifies between **sexist** and **non-sexist** and another model is trained to detect specific kinds of sexism.

In addition, for the fine-tuning process, we carried out a Grid-search optimization over the main parameters of the neural network: learning rate, batch size and dropout rate. The search was performed with a 5-fold stratified cross-validation with the following grid: Learning rate, $(1e-6, 1e-5, 3e-5, 5e-5, 1e-4)$; batch size, $(8, 16, 32)$ and dropout rate, $(0.08, 0.1, 0.12)$. The best parameters for both models were: learning rate, $1e-5$; batch size, 16 and dropout rate, 0.1.

4.4 Data Augmentation

As the dataset is relatively small, we decided to run Data Augmentation techniques. We followed two different strategies to increase the amount of data in the corpora; Backtranslation [16] and translation of the different languages in the dataset.

Backtranslation This method consists of translating the samples into a pivot language and then translating them back into the original language. Given that the existing translation methods are not perfect, we get samples that are written in a slightly different way, but keep the original meaning. In particular, this technique has been proven useful for sentiment analysis and with twitter corpora before [10]. In this case, we used 30 pivot languages. For the translations we used the translation models of Helsinki NLP [18] based on the Marian model [9] and the Google Translate API for the ones that were not available in the Helsinki NLP models.

The selected languages are the following (expressed in ISO 639-1): *eu*, *la*, *zh-cn*, *hi*, *bn*, *pt*, *ru*, *ja*, *pa*, *mr*, *te*, *tr*, *ko*, *fr*, *de*, *vi*, *ta*, *ur*, *it*, *ar*, *fa*, *ha*, *kn*, *id*, *pl*, *uk*, *ro*, *eo*, *sv* and *el*. Also *es* and *en* were used for English and Spanish datasets respectively.

Multilingual Translation Following the above reasoning, we can also use labeled data (with the same gold standard) in other languages. So we translated every English sample into Spanish and *vice-versa*. This way, we should have a more robust training and avoid overfitting because the "new" samples are completely new for that language's model, opposed to the slightly modified samples from Backtranslation.

5 Experiments and Results

5.1 Experimental Setup

We trained all the models with a NVIDIA Tesla P100-PCIE-16GB GPU and a Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz CPU with 500GB of RAM memory.

The software we used was Python3.8, transformers 4.5.1 [20], pytorch 1.8.1 [13] and scikit-learn 0.24.1 [14].

5.2 Results

In the Table 3 we can see the results for our models in the test set of the first task. Note that the Tfid+SVM baseline and the AI-UPV_team (winner of the task) are taken from the task Overview [15]. The runs we presented for the contest were *2BERTs+Backtranslation*, *2BERTs* and *2BERTs+Multilingual translation* where 2BERTs refer to the different models used for each language, explained in Section 4.3.

The Backtranslation models had good results in our first training experiments, however they proved to overfit a lot for this task with an accuracy of 0.7479, just a bit better than the multilingual BERT baseline (0.734). This shows that Data Augmentation techniques are not always useful. Next, we can see that the Multilingual Translation models obtained an accuracy of 0.7683, which proves a better generalization than the model without any augmentation (0.7603). With this, our model is positioned very close to the best result in the competition, that is only 1.58% better.

Model	**Accuracy**
HV+RF	0.6830
Tfidf+SVM	0.6845
mBERT	0.7341
2BERTs+Backtranslation	0.7479
2BERTs	0.7603
2BERTs+Multilingual translation	0.7683
AI-UPV_team	0.7804

Table 3. Results for task1

For the second task, the results were similar to the ones obtained in the first task. In the Table 4 we can look at them in more detail. Again, the Tfidf+SVM baseline and the AI-UPV_team results come from the task Overview [15]. We can see that our models behaved consistently like in the first task, but the results were not that good. Despite that, the results are still very close to the best.

Model	F-measure
Tfidf+SVM	0.3950
HV+RF	0.4131
mBERT	0.4961
2BERTs+Backtranslation	0.5174
2BERTs	0.5218
2BERTs+Multilingual translation	0.5295
AI-UPV_team	0.5787

Table 4. Results for task2

To sum up the improvements of our models, we can see an ablation study for our best model (*2BERTs+Multilingual translation*) in the task 1 where each entry has a feature removed from the best model. This proves that most of the ideas introduced, produced some kind of improvement to the system. The most significant improvement was the selection of good hyperparameters for the model. Finally, it is also very remarkable that we get a large improvement by Multilingual Translation, proving our hypothesis about the ability of this Data Augmentation technique to generalize in Multilingual corpora.

Model	Accuracy
Best model	0.7683
Default model (no Grid-Search)	0.7451
Uncased	0.7599
No augmentation	0.7603
No preprocessing	0.7678

Table 5. Ablation study for the task1 models

6 Conclusions and Future Work

Through this shared task, we have seen that NLP can be of great help in detecting and classifying unwanted toxic and sexist behavior in social networks and there is still a long way to go.

The results obtained by our systems are very promising given their great performance and their simplicity. Furthermore, we proposed a new way of facing multilingual problems that provides better results. All this is very significant and could lead to much better results when combined with other improvements from the state-of-the-art.

We believe that our results could improve a lot using specific language models trained with corpora from social networks like TWilBert [6] for Spanish and BERTweet [12] for English. Another interesting approach would be to use a general language model and further pre-train it with corpora from the same domain [17] as the final task. These corpora would be easy to obtain given that

the authors of the EXIST2021 shared task, gathered it from a list of keywords [15]. Finally, we have proven that good hyperparameters are also key for a good neural network so a better search, like the Population Based Training [8], would further improve the model.

Acknowledgments

This work has been partially funded by the Instituto de Ingeniería del Conocimiento (IIC) and the hardware used was also provided by the IIC.

References

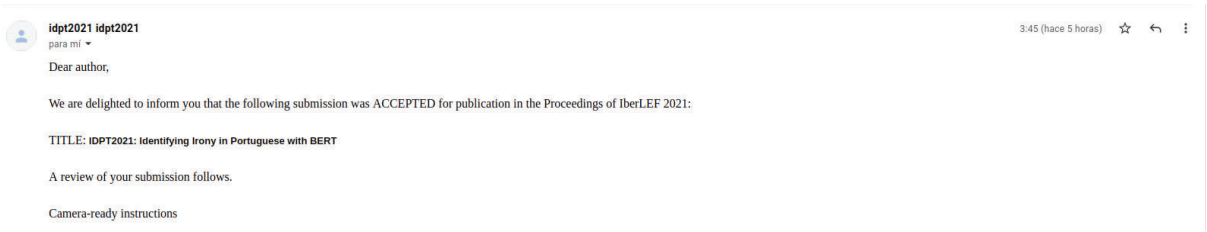
1. Anzovino, M., Fersini, E., Rosso, P.: Automatic identification and classification of misogynistic language on twitter. In: NLDB (2018)
2. Cañete, J.: Compilation of large spanish unannotated corpora (May 2019). https://doi.org/10.5281/zenodo.3247731, https://doi.org/10.5281/zenodo.3247731
3. Cañete, J., Chaperon, G., Fuentes, R., Pérez, J.: Spanish pre-trained bert model and evaluation data. In: to appear in PML4DC at ICLR 2020 (2020)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
5. Frenda, S., Ghanem, B., Montes-y Gómez, M., Rosso, P.: Online hate speech against women: Automatic identification of misogyny and sexism on twitter. Journal of Intelligent & Fuzzy Systems **36**(5), 4743–4752 (2019)
6. Ángel González, J., Hurtado, L.F., Pla, F.: Twilbert: Pre-trained deep bidirectional transformers for spanish twitter. Neurocomputing (2020). https://doi.org/https://doi.org/10.1016/j.neucom.2020.09.078, http://www.sciencedirect.com/science/article/pii/S0925231220316180
7. Grosz, D., Conde-Cespedes, P.: Automatic detection of sexist statements commonly used at the workplace (2020)
8. Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K.: Population based training of neural networks (2017)
9. Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A.F., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in C++. In: Proceedings of ACL 2018, System Demonstrations. pp. 116–121. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-4020, https://www.aclweb.org/anthology/P18-4020
10. Luque, F.M.: Atalaya at tass 2019: Data augmentation and robust embeddings for sentiment analysis (2019)
11. Montes, M., Rosso, P., Gonzalo, J., Aragón, E., Agerri, R., Ángel Álvarez Carmona, M., Álvarez Mellado, E., de Albornoz, J.C., Chiruzzo, L., Freitas, L., Adorno, H.G., Gutiérrez, Y., Zafra, S.M.J., Lima, S., de Arco, F.M.P., Taulé, M.: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021). In: CEUR Workshop Proceedings (2021)

12. Nguyen, D.Q., Vu, T., Nguyen, A.T.: BERTweet: A pre-trained language model for English Tweets. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (2020)
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
14. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
15. Rodríguez-Sánchez, F., de Albornoz, J.C., Plaza, L., Gonzalo, J., Rosso, P., Comet, M., Donoso, T.: Overview of exist 2021: sexism identification in social networks. Procesamiento del Lenguaje Natural **67**(0) (2021)
16. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 86–96. Association for Computational Linguistics, Berlin, Germany (Aug 2016). https://doi.org/10.18653/v1/P16-1009, https://www.aclweb.org/anthology/P16-1009
17. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? (2020)
18. Tiedemann, J., Thottingal, S.: OPUS-MT — Building open translation services for the World. In: Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT). Lisbon, Portugal (2020)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
20. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6

Apéndice B

Paper IDPT en Proceedings of the Iberian Languages Evaluation Forum

B.1. Aceptación



idpt2021 idpt2021
para mí
3:45 (hace 5 horas)

Dear author,

We are delighted to inform you that the following submission was ACCEPTED for publication in the Proceedings of IberLEF 2021:

TITLE: **IDPT2021: Identifying Irony in Portuguese with BERT**

A review of your submission follows.

Camera-ready instructions

B.2. *Paper*

A continuación se adjunta el *paper* enviado después de las *peer reviews* dado que aun no se ha publicado la revista.

GuillemGSubies at IDPT2021: Identifying Irony in Portuguese with BERT

Guillem García Subies[1]

Instituto de Ingeniería del Conocimiento, Francisco Tomás y Valiente st., 11 EPS, B Building, 5th floor UAM Cantoblanco. 28049 Madrid, Spain
guillem.garcia@iic.uam.es

Abstract. This paper describes a system created for the IDPT 2021 shared task, framed within the IberLEF 2021 workshop. We present an approach mainly based on fine-tuned BERT models using a Grid-Search and Data Augmentation with MLM substitution. Our models far outperform the baselines and achieve results close to to the state-of-the-art.

Keywords: Irony Detection · BERT · Transformers · Data Augmentation · BERTimbau

1 Introduction

Although irony can be relatively easy to identify for humans, it is not so easy to detect for NLP models [5], mainly because the information can be implicit and usually doesn't use the literal meaning of the words used. This makes the task of irony detection prefect to evaluate the evolution of NLP systems.

The IDPT (Irony Detection in Portuguese) shared task proposes, during this third edition of the IberLEF [15] workshop, a corpus to detect irony in tweets and news written in Portuguese [1]. This article summarizes our participation in all the IDPT tasks.

Given the success of Transformer-inspired language models [23], both in academia and industry [24], we decided to use already pre-trained BERT [6] models. Furthermore, their ability to understand contextual information can be very useful for the irony detection task. Specifically, we will use BERTimbau [21] with hyperparameters Grid-Search. To address the problem of small data, we will use Data Augmentation techniques.

1.1 Task Description

There are two corpora, one for tweets and one for news (task1 and task2 respectively). For both of them, the problem is binary classification, where the sample can be ironic or not.

The tweets corpus has 15212 tweets and the news one has 18494 news for their train splits. The data is collected from various preexisting sources [8] [4] [14]. Then, the test splits are composed of 300 tweets and 300 news gathered and annotated by the organizers of the tasks. This will help to create models that generalize very well.

The metric used to evaluate the results is the balanced accuracy. This is mainly because both datasets are very unbalanced as we can see in the table 1. It is also notable the difference between tasks; most of the tweets are ironic while most of the news are not ironic.

Tweets		**News**	
Class	**Nº of samples**	**Class**	**Nº of samples**
ironic	12736	ironic	7222
not ironic	2476	not ironic	11272

Table 1. Distribution of Samples

1.2 Goals

This work is focused on proving that it is possible to use open source resources and relatively small language models (compared to the newest models like GPT-3 [2]) to obtain state-of-the-art results. Specifically, the main goal is to obtain a Portuguese language model that can detect the irony in the text and meet the requirements explained before.

1.3 Summary of the proposal

To achieve the goals explained above we will fine-tune a BERTimbau model [21] with Grid-search optimized parameters. Along with this model, the data is first preprocessed with simple heuristics and then, augmented with Masked Language Model word masking.

In the next section, we will briefly see some previous work related to this topic. Then, in Section 3, we will explain the main ideas behind the proposed models and the experiments we did. In Section 4 we will present a summary of the results we got. Finally, in Section 5 we will expose the main conclusions of our work and results, and we will also propose some ideas for future work.

2 Related Work

There is an extensive bibliography on Sentiment Analysis and irony detection in social media given the high scientific interest in solving such a difficult problem. Some early attempts to create corpora in this field were for sarcasm detection, for instance Davidov et al. used Amazon Mechanical Turk to create a corpus with 5.9 million tweets [5] and Riloff et al. explore the identification of sarcastic

tweets that have a positive word or comment followed by an undesirable situation [20].

There have also been some attempts to create irony detection corpora in other languages than English. For example, Ptacek et al. [19] created a Czech sarcasm binary classification dataset for tweets and also propose a n-gram and heuristics based embeddings that are feed intro classic machine learning models. Liebrecht et al. [12] collect a Dutch sarcasm dataset from tweets that included the hashtag *#sarcasm* and hypothesize about that hashtag being the digital equivalent of non-verbal expressions in live interactions. Bilal et al. [10] collect irony detection datasets in different languages in order to show that good models can be trained even when the data for some language is scarce.

Following this trend, there have been a lot of irony detection competitions these last years. For instance, IDAT [9] proposed a binary classification problem to detect irony in tweets written in Arabic. The best model was a feature based one with classic machine learning models, outperforming even BERT models. IroSvA, [16], proposed a binary irony classification problem for Spanish tweets in different Spanish dialects. This time, the best model fed a Word2Vec embeddings into a Transformer model as a weights initialization.

For the Portuguese language, Carvalho et al. [3] detect irony in newspaper comments using simple glossaries, proving that complex linguistic features do not work for irony. Following the same trend Freitas et al. [7] create a list of relevant patterns to detect irony in Portuguese tweets.

It is notable that some of the models used in these works still use linguistic features and heuristics to detect the irony. However, we will focus on the potential of language models to solve this task without any linguistic features.

3 Models

3.1 Data Preprocessing

We performed a simple preprocessing where we substituted some expressions with a more normalized form:

- Every URL was replaced with the token "[URL]", so we don't get strange tokens when the tokenizer tries to process a URL. Furthermore, no semantic information about irony can be inferred from a URL, the only information relevant for the model is that there is a URL in that token.
- The hashtag characters ("#") were deleted ("#example" → "example") because the base language models we will use, are trained in generic text and might not understand their meaning. Furthermore, most of the hashtags are used the same way as normal words.
- We replaced every username with the generic token "[USER]" because the exact name of a user does not really add any information about the irony. The only relevant feature is knowing if someone was mentioned or not, but not who.
- Finally, we normalized every laugh ("jasjajajajj" → "haha"), so we minimize the noise of the misspellings, common in social networks.

3.2 Baselines

We created some baselines, so we can compare our models properly. We selected a HashingVectorizer + RandomForest. This way, we can compare our models to a classic feature extraction model.

3.3 Language Models

We used BERTimbau [21], a Portuguese BERT model that outperforms mBERT and the previous state-of-the-art. Specifically, we used the large model, *bert-large-portuguese-cased*. For the fine-tuning process, we carried out a Grid-search optimization over the main parameters of the neural network: learning rate, batch size and dropout rate. The search was performed with a 5-fold stratified cross-validation with the following grid: Learning rate, $(1e-6, 1e-5, 3e-5, 5e-5, 1e-4)$; batch size, $(8, 16, 32)$ and dropout rate, $(0.08, 0.1, 0.12)$. The best parameters for both models were: learning rate, $1e-5$; batch size, 16 and dropout rate, 0.1.

3.4 Data Augmentation

As the dataset is relatively small, we decided to run Data Augmentation techniques. The selected strategy was the Data Augmentation through the masking of words with a Masked Language Model, BERTimbau. For every sample in the dataset, we randomly masked 15% of the tokens and used BERTimbau to predict them, creating a modified sample. With this method, we obtained double the amount of the original samples.

4 Experiments and Results

4.1 Experimental Setup

The software we used was Python3.8, transformers 4.5.1 [24], pytorch 1.8.1 [17], scikit-learn 0.24.1 [18] and nlpaug 1.1.3 [13].

4.2 Results

In the Table 2 we can see the results for our models in the test set of the first task. Our runs for this task are *BERTimbau* and *BERTimbau-aug*, without data augmentation and with data augmentation, respectively as explained in Section 3.3.

We can see that the language model far outperforms classic methods like hashing tricks and a random forest. We can also see that, although the Data Augmentation does not provide a great performance boost, it is still useful in order to have better models. All in all, our models obtain great results given their simplicity, proving that finding the right parameters for the model is crucial for optimizing the performance. These results are placed fourth among all the

participating teams, which proves that our approach, given it's simplicity and the lack any linguistic analysis, is very good.

Model	**bacc**
HV+RF	0.3316
BERTimbau	0.4912
BERTimbau-aug	0.5000

Table 2. Results for task1

For the second task, the results were not as good as the ones obtained in the first task. In the Table 3 we can look at them in more detail. It looks like BERTimbau did not behave so well with the news dataset.

Model	**bacc**
HV+RF	0.5423
BERTimbau	0.7804
BERTimbau-aug	0.7858

Table 3. Results for task2

5 Conclusions and Future Work

Through this shared task, we have seen that NLP can be of great help in detecting irony from natural language in social networks and there is still a long way to go. The results obtained by our systems are very promising given their great performance and their simplicity. This compilation of methods is very significant because it could lead to much better results when combined with other improvements from the state-of-the-art. Particularly, the Data Augmentation approach with the Grid-search have proven to work really well in this context. We therefore consider that we have achieved our goals for this shared task.

However, we believe that our results could improve a lot using specific language models trained only with corpora from social networks. Another interesting approach would be to use a general language model and further pre-train it with corpora from the same domain [22] as the final task. Finally, we have proven that good hyperparameters are also key for a good neural network, so a better search, like the Population Based Training [11], would further improve the model.

Acknowledgments

This work has been partially funded by the Instituto de Ingeniería del Conocimiento (IIC) and the hardware used was also provided by the IIC.

References

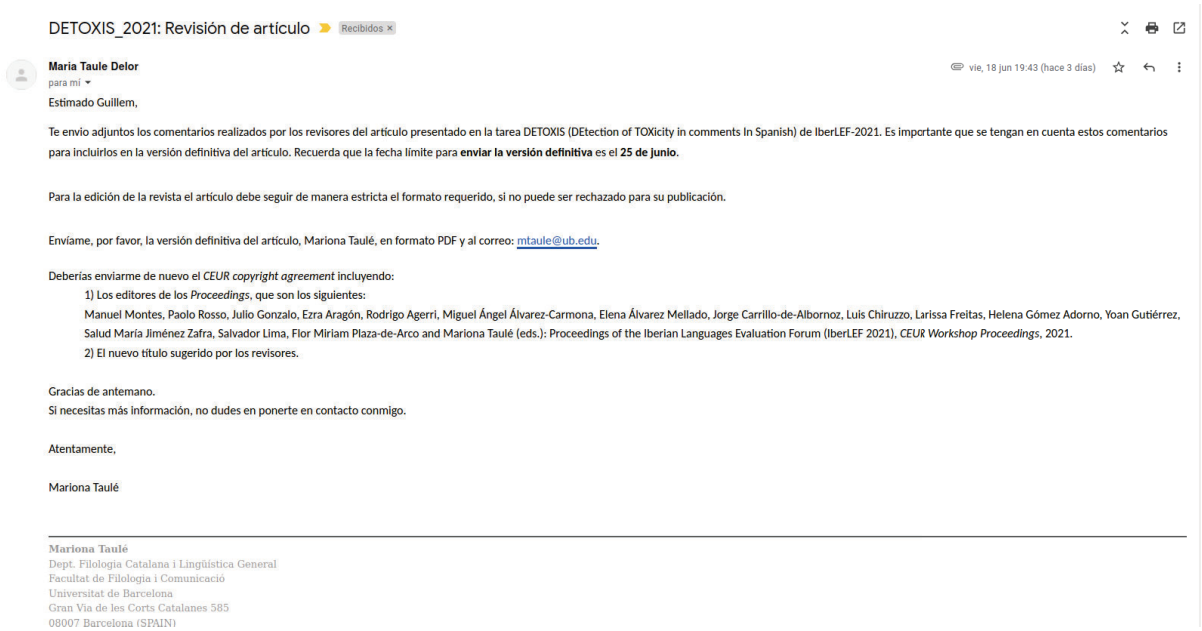
1. Brisolara Corrêa, U., Pereira dos Santos, L., Coelho, L., A. de Freitas, L.: Idpt2021 at iberlef: Overview of the task on irony detection in portuguese. Proceedings of the Iberian Languages Evaluation Fórum (IberLEF 2021), co-located with the 37th Conference of the Spanish Society for Natural Language Processing (SEPLN 2021). CEUR Workshop Proceedings, 2021 (2021)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020)
3. Carvalho, P., Sarmento, L., Silva, M.J., De Oliveira, E.: Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-. In: Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion. pp. 53–56 (2009)
4. DA SILVA, F.R.A.: Detecção de ironia e sarcasmo em língua portuguesa: Uma abordagem utilizando deep learning (2018)
5. Davidov, D., Tsur, O., Rappoport, A.: Semi-supervised recognition of sarcastic sentences in twitter and amazon. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning. p. 107–116. CoNLL '10, Association for Computational Linguistics, USA (2010)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
7. Freitas, L., Vanin, A., Vieira, R., Bochernitsan, M.: Some clues on irony detection in tweets. In: WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web (05 2013). https://doi.org/10.1145/2487788.2488012
8. de Freitas, L.A., Vanin, A.A., Hogetop, D.N., Bochernitsan, M.N., Vieira, R.: Pathways for irony detection in tweets. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing. pp. 628–633 (2014)
9. Ghanem, B., Karoui, J., Benamara, F., Moriceau, V., Rosso, P.: Idat at fire2019: Overview of the track on irony detection in arabic tweets. In: Proceedings of the 11th Forum for Information Retrieval Evaluation. pp. 10–13 (2019)
10. Ghanem, B., Karoui, J., Benamara, F., Rosso, P., Moriceau, V.: Irony detection in a multilingual context. In: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (eds.) Advances in Information Retrieval. pp. 141–149. Springer International Publishing, Cham (2020)
11. Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K.: Population based training of neural networks (2017)
12. Liebrecht, C., Kunneman, F., van den Bosch, A.: The perfect solution for detecting sarcasm in tweets #not. In: Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. pp. 29–37. Association for Computational Linguistics, Atlanta, Georgia (Jun 2013), https://www.aclweb.org/anthology/W13-1605
13. Ma, E.: Nlp augmentation. https://github.com/makcedward/nlpaug (2019)
14. Marten, G.S., de Freitas, L.A.: The construction of a corpus for detecting irony and sarcasm in portuguese. Brazilian Journal of Development **7**(5), 47973–47984 (2021)

15. Montes, M., Rosso, P., Gonzalo, J., Aragón, E., Agerri, R., Ángel Álvarez Carmona, M., Álvarez Mellado, E., de Albornoz, J.C., Chiruzzo, L., Freitas, L., Adorno, H.G., Gutiérrez, Y., Zafra, S.M.J., Lima, S., de Arco, F.M.P., Taulé, M.: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021). In: CEUR Workshop Proceedings (2021)
16. Ortega-Bueno, R., Rangel, F., Hernández Farıas, D., Rosso, P., Montes-y Gómez, M., Medina Pagola, J.E.: Overview of the task on irony detection in spanish variants. In: Proceedings of the Iberian languages evaluation forum (IberLEF 2019), co-located with 34th conference of the Spanish Society for natural language processing (SEPLN 2019). CEUR-WS. org. vol. 2421, pp. 229–256 (2019)
17. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
19. Ptáček, T., Habernal, I., Hong, J.: Sarcasm detection on Czech and English Twitter. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 213–223. Dublin City University and Association for Computational Linguistics, Dublin, Ireland (Aug 2014), https://www.aclweb.org/anthology/C14-1022
20. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., Huang, R.: Sarcasm as contrast between a positive sentiment and negative situation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 704–714. Association for Computational Linguistics, Seattle, Washington, USA (Oct 2013), https://www.aclweb.org/anthology/D13-1066
21. Souza, F., Nogueira, R., Lotufo, R.: BERTimbau: pretrained BERT models for Brazilian Portuguese. In: 9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear) (2020)
22. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? (2020)
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
24. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6

Apéndice C

Paper DETOXIS en Proceedings of the Iberian Languages Evaluation Forum

C.1. Aceptación



DETOXIS_2021: Revisión de artículo Recibidos ×

María Taule Delor
para mí
vie, 18 jun 19:43 (hace 3 días)

Estimado Guillem,

Te envio adjuntos los comentarios realizados por los revisores del artículo presentado en la tarea DETOXIS (DEtection of TOXicity in comments In Spanish) de IberLEF-2021. Es importante que se tengan en cuenta estos comentarios para incluirlos en la versión definitiva del artículo. Recuerda que la fecha límite para **enviar la versión definitiva** es el **25 de junio**.

Para la edición de la revista el artículo debe seguir de manera estricta el formato requerido, si no puede ser rechazado para su publicación.

Envíame, por favor, la versión definitiva del artículo, Mariona Taulé, en formato PDF y al correo: mtaule@ub.edu.

Deberías enviarme de nuevo el *CEUR copyright agreement* incluyendo:

1) Los editores de los *Proceedings*, que son los siguientes:
Manuel Montes, Paolo Rosso, Julio Gonzalo, Ezra Aragón, Rodrigo Agerri, Miguel Ángel Álvarez-Carmona, Elena Álvarez Mellado, Jorge Carrillo-de-Albornoz, Luis Chiruzzo, Larissa Freitas, Helena Gómez Adorno, Yoan Gutiérrez, Salud María Jiménez Zafra, Salvador Lima, Flor Miriam Plaza-de-Arco and Mariona Taulé (eds.): Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021), *CEUR Workshop Proceedings*, 2021.
2) El nuevo título sugerido por los revisores.

Gracias de antemano.
Si necesitas más información, no dudes en ponerte en contacto conmigo.

Atentamente,

Mariona Taulé

Mariona Taulé
Dept. Filologia Catalana i Lingüística General
Facultat de Filologia i Comunicació
Universitat de Barcelona
Gran Via de les Corts Catalanes 585
08007 Barcelona (SPAIN)

C.2. *Paper*

A continuación se adjunta el *paper* enviado después de las *peer reviews* dado que aun no se ha publicado la revista.

GuillemGSubies at IberLEF-2021 DETOXIS task: Detecting Toxicity with Spanish BERT

Guillem García Subies[1]

Instituto de Ingeniería del Conocimiento, Francisco Tomás y Valiente st., 11 EPS, B Building, 5th floor UAM Cantoblanco. 28049 Madrid, Spain
guillem.garcia@iic.uam.es

Abstract. This paper describes a system created for the DETOXIS 2021 shared task, framed within the IberLEF 2021 workshop. We present an approach mainly based in fine-tuned BERT models using a Grid-Search and Data Augmentation with MLM substitution. This approach only takes into account the textual data from the dataset to prove the power of language models. Our models far outperform the baselines and achieve results close to the state-of-the-art.

Keywords: Toxicity Detection · BERT · Transformers · Data Augmentation · BETO

1 Introduction

Polarization can be a very problematic issue in society, especially on social media. There are manual mechanisms to report these behaviors, however they can be slow and inefficient. To address this, we can use NLP to detect automatically these undesirable toxic behaviors. The DETOXIS (DEtection of TOxicity in comments In Spanish) [15] shared task proposes, during this third edition of the IberLEF [10] workshop, a corpus to detect toxicity level in comments on internet forums and newspapers discussions.

This article summarizes our participation in all the DETOXIS tasks. Given the success of Transformer-inspired language models [16], both in academia and industry [17], we decided to use already pre-trained BERT [5] models. Specifically, we will use BETO [4] with some extra transfer learning techniques for ordinal classification problems and a hyperparameters Grid-Search. To address the problem of small data, we will use Data Augmentation techniques.

In the next section, we will briefly see some previous work related to this topic. In Section 3 we will go through a brief description of the tasks and the corpus. Then, in Section 4, we will explain the main ideas behind the proposed models. In Section 5 we will present a summary of the experiments we carried out and the results we got. Finally, in Section 6 we will expose the main conclusions of our work and results and we will also propose some ideas for future work.

IberLEF 2021, September 2021, Málaga, Spain.

2 Related Work

There is an extensive bibliography on Sentiment Analysis and text classification in social networks, however not that much work has been done about identifying and classifying toxic behaviors until 2019.

Most of the toxicity detection datasets are focused on classifying what kind of toxicity is present in the text, instead of the level of toxicity. Basile et al. [2] propose a task for the identification of toxicity against women and migrant people in Spanish and English Tweets. Struß et al., [13] also focus on Twitter, but now with German tweets. Other corpora focus in a more multilingual emphasis like Kumar et al. [8] and Zampieri et al. [18].

However, all have a common denominator. The best results have been obtained with some kind of Transformers or BERT-based model. For instance, the best models in Struß et al., [13] are fine-tuned BERTs with German pretraining on general data or fine-tuned BERTs pre-trained with specific German tweets. The best results in Kumar et al. [8] also point at BERT and Transformers, specifically a bootstrap aggregation of BERT models. Finally, the best results in Zampieri et al. [18] are also some kind of Transformer based models and ensembles of them.

This is a clear indicator of the trends in the state-of-the-art for this topic.

3 Tasks Description

The main corpus consists of 3463 comments posted in Spanish online newspapers and forums for the train split and 890 for the test one. They were collected from August 2017 to July 2020. Furthermore, the articles were selected taking into account their potential toxicity and the number of comments in them (more than 50 comments).

For the first task, the comments are annotated into two categories; **toxic** and *not toxic*. The second task consists of further classifying that toxicity into four levels of toxicity; **toxicity_level_0=not toxic**, **toxicity_level_1=mildly toxic**, **toxicity_level_2=toxic** and **toxicity_level_3=very toxic**.

In addition to the classification labels, for every sample there are also annotation about the argumentation, constructiveness, stance, target, stereotype, sarcasm, mockery, insult, improper language, aggressiveness and intolerance. However none of these are public in the test set, so they will be ignored in this study. Finally, for every comment, there is also a label indicating if the comment is a response to another comment or not. This information will not be used either because we will only focus on the textual data.

The metrics used to evaluate the results are the F-measure for the task1 and the Closeness Evaluation Metric (CEM) [1] for the second task. This last metric is very useful for ordinal classification problems given that it takes into account the order of the classes using concepts from Measurement Theory.

Class	Nº of samples
not toxic	2317
mildly toxic	808
toxic	269
very toxic	69

Table 1. Distribution of Samples

In the table above we can see the distribution of the samples in the train split. The most notable fact is that 67% of the samples are not toxic, so the dataset is unbalanced. For the second task, we can also see a very notable unbalance.

In the table below, we can see some illustrative examples of the data and their labels:

Los detuvieron en ronda malaga, un saludo	not toxic
Loss mas valientes, los que mejor cortan nuestras cabezas, Para vosotros, socialistas, izquierdistas, y no racistas,	mildly toxic
Esto es lo que importas cuando los rescatas en lugar de hundirlos.	toxic
Está claro que vienen los mejores. Haced que pase putos rojos de mierda.	very toxic

Table 2. Examples of the different classes

4 Models

4.1 Data Preprocessing

We performed a simple preprocessing where we substituted some expressions with a more normalized form:

- Every URL was replaced with the token "[URL]" so we don't get strange tokens when the tokenizer tries to process and URL. Furthermore, no semantic information about toxicity can be inferred from a URL, the only information relevant for the model is that there is a URL in that token.
- Finally we normalized every laugh ("jasjajajajj" → "haha") so we minimize the noise of the misspellings, common in social networks.

4.2 Baselines

We created some baselines so we can compare our models properly. We selected a HashingVectorizer + RandomForest. This way, we can compare our models to a classic feature extraction model.

4.3 Language Models

We used BETO [4], a BERT model trained with the Spanish Unannotated Corpora (SUC) [3] that has proven to be much better than the multilingual BERT model.

We tried different training strategies given that the classes are related to each other:

- The simplest approach we tried is treating both tasks as the same one, with a multiclass classification model. For the first task, everything different from **not toxic** would be considered **toxic**.
- For the second approach, we first trained a binary classification model to distinguish between **not toxic** and **toxic** for the first task. Then we trained a multiclass model to classify between the three levels of toxicity.
- Similarly to the last approach, we then tried to transform the second task into three different binary classification problems; classifying between **not toxic** and the rest of the classes, **mildly toxic** and **toxic** or **very toxic**, and between **toxic** and **very toxic**. With this, we tried to have very specific models that can differentiate slight changes in toxicity.
- As there are not too many samples for the last models in the previous approach to learn correctly, we also tried with a transfer learning approach, similar to the one presented by Sun et al. [14]. Instead of using always the same BETO pretrained model for every finetuned model, we used the finetuned model from the step before, i.e. the model that classifies **mildly toxic** and **toxic** has as base model the one that classifies **not toxic** and **mildly toxic**.

In addition, for the fine-tuning process, we carried out a Grid-search optimization over the main parameters of the neural network: learning rate, batch size and dropout rate. The search was performed with a 5-fold stratified cross-validation with the following grid: Learning rate, $(1e-6, 1e-5, 3e-5, 5e-5, 1e-4)$; batch size, $(8, 16, 32)$ and dropout rate, $(0.08, 0.1, 0.12)$. The best parameters for both models were: learning rate, $1e-5$; batch size, 16 and dropout rate, 0.1.

4.4 Data Augmentation

As the dataset is relatively small, we decided to run Data Augmentation techniques. The selected strategy was the Data Augmentation through the masking of words with a Masked Language Model, BETO.

For every sample in the dataset, we randomly masked 15% of the tokens and used BETO to predict them, creating a modified sample. With this method, we obtained double the amount of the original samples.

5 Experiments and Results

5.1 Experimental Setup

We trained all the models with a NVIDIA Tesla P100-PCIE-16GB GPU and a Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz CPU with 500GB of RAM memory.

The software we used was Python3.8, transformers 4.5.1 [17], pytorch 1.8.1 [11], scikit-learn 0.24.1 [12] and nlpaug [9] 1.1.3.

5.2 Results

In the Table 3 we can see the results for our models in the test set of the first task. Note that the ChainBOW baseline, Word2VecSpacy baseline and the SINAI_team (winner of the task) results are taken from the task Overview [15]. Our runs for this task are *BETO-multiclass* and *BETO-binary* both with and without data augmentation as explained in Section 4.3. Note that some of the results presented here were obtained after the labeled test set was published so we could analyze in depth our models.

We can see that there is almost no difference between the binary and multiclass models. This might happen because there is a great difference in the amount of samples and toxicity between the not toxic comments and the rest, which makes them easy to indentify in every situation. Finally we can see the the Data Augmentation strategy obtained around 0.02 points more than the models without augmentation. Our result was the second best in the competition, which proves that the simplicity of using BETO with some Grid-Search can yield really good results.

Model	**F-measure**
Word2VecSpacy	0.1523
ChainBOW	0.3747
HV+RF	0.4159
BETO-multiclass	0.5721
BETO-binary	0.5777
BETO-multiclass-aug	0.5981
BETO-binary-aug	0.6000
SINAI_team	0.6461

Table 3. Results for task1

For the second task, the results were similar to the ones obtained in the first task. In the Table 4 we can look at them in more detail. Again, ChainBOW baseline, Word2VecSpacy baseline and the SINAI_team results are taken from the task Overview [15]. We can see that our transfer learning approach (*BETO-transfer*) obtains better results than the other approaches and that there is

almost no difference between the simple multiclass approach and the one that first detects the not toxic comments (*BETO-2models-aug*). These results are in line with the ones in the first task, showing that adding the not toxic class to the models, will not make them worse.

This results are placed fifth among all the participating teams (24), which proves that our approach, given it's simplicity and the lack of any linguistic analysis, is very good.

Model	**CEM**
Word2VecSpacy	0.6116
HV+RF	0.6214
ChainBOW	0.6535
BETO-2_models-aug	0.6891
BETO-multiclass-aug	0.6913
BETO-3_models-aug	0.704
BETO-transfer	0.7172
BETO-transfer-aug	0.7189
SINAI_team	0.7495

Table 4. Results for task2

6 Conclusions and Future Work

Through this shared task, we have seen that NLP can be of great help in detecting and classifying unwanted toxic behavior in social networks and there is still a long way to go.

The results obtained by our systems are very promising given their great performance and their simplicity. This compilation of methods is very significant because it could lead to much better results when combined with other improvements from the state-of-the-art.

We believe that our results could improve a lot using specific language models trained with corpora from social networks like TWilBert [6]. Another interesting approach would be to use a general language model and further pre-train it with corpora from the same domain [14] as the final task. Finally, we have proven that good hyperparameters are also key for a good neural network so a better search, like the Population Based Training [7], would further improve the model.

Acknowledgments

This work has been partially funded by the Instituto de Ingeniería del Conocimiento (IIC) and the hardware used was also provided by the IIC.

References

1. Amigó, E., Gonzalo, J., Mizzaro, S., de Albornoz, J.C.: An effectiveness metric for ordinal classification: Formal properties and experimental results (2020)
2. Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F.M., Rosso, P., Sanguinetti, M.: SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 54–63. Association for Computational Linguistics, Minneapolis, Minnesota, USA (Jun 2019). https://doi.org/10.18653/v1/S19-2007, https://www.aclweb.org/anthology/S19-2007
3. Cañete, J.: Compilation of large spanish unannotated corpora (May 2019). https://doi.org/10.5281/zenodo.3247731, https://doi.org/10.5281/zenodo.3247731
4. Cañete, J., Chaperon, G., Fuentes, R., Pérez, J.: Spanish pre-trained bert model and evaluation data. In: to appear in PML4DC at ICLR 2020 (2020)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
6. Ángel González, J., Hurtado, L.F., Pla, F.: Twilbert: Pre-trained deep bidirectional transformers for spanish twitter. Neurocomputing (2020). https://doi.org/https://doi.org/10.1016/j.neucom.2020.09.078, http://www.sciencedirect.com/science/article/pii/S0925231220316180
7. Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K.: Population based training of neural networks (2017)
8. Kumar, R., Ojha, A.K., Malmasi, S., Zampieri, M.: Evaluating aggression identification in social media. In: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying. pp. 1–5. European Language Resources Association (ELRA), Marseille, France (May 2020), https://www.aclweb.org/anthology/2020.trac-1.1
9. Ma, E.: Nlp augmentation. https://github.com/makcedward/nlpaug (2019)
10. Montes, M., Rosso, P., Gonzalo, J., Aragón, E., Agerri, R., Ángel Álvarez Carmona, M., Álvarez Mellado, E., de Albornoz, J.C., Chiruzzo, L., Freitas, L., Adorno, H.G., Gutiérrez, Y., Zafra, S.M.J., Lima, S., de Arco, F.M.P., Taulé, M.: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021). In: CEUR Workshop Proceedings (2021)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
13. Struß, J.M., Siegel, M., Ruppenhofer, J., Wiegand, M., Klenner, M.: Overview of germeval task 2, 2019 shared task on the identification of offensive language.

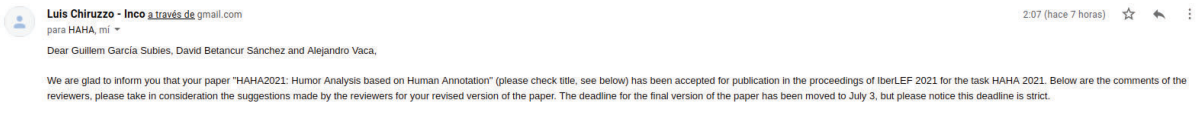
In: Preliminary proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019), October 9 – 11, 2019 at Friedrich-Alexander-Universität Erlangen-Nürnberg. pp. 352 – 363. German Society for Computational Linguistics & Language Technology und Friedrich-Alexander-Universität Erlangen-Nürnberg, München [u.a.] (2019), https://nbn-resolving.org/urn:nbn:de:bsz:mh39-93197
14. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? (2020)
15. Taulé, M., Ariza, A., Nofre, M., Amigó, E., Rosso, P.: Overview of the detoxis task at iberlef-2021: Detection of toxicity in comments in spanish. Procesamiento del Lenguaje Natural **67** (2021)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
17. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6
18. Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., Çöltekin, Ç.: SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In: Proceedings of the Fourteenth Workshop on Semantic Evaluation. pp. 1425–1447. International Committee for Computational Linguistics, Barcelona (online) (Dec 2020), https://www.aclweb.org/anthology/2020.semeval-1.188

Apéndice D

Paper HAHA en Proceedings of the Iberian Languages Evaluation Forum

Se ha decidido no incluir este *paper* en el desarrollo del TFM ya que la autoría es compartida y las técnicas usadas aportan poco a lo ya expuesto. Sin embargo, se incluye en el anexo ya que es un resultado relevante fruto de este trabajo.

D.1. Aceptación


Luis Chiruzzo - Inco a través de gmail.com
para HAHA, mí
2:07 (hace 7 horas)

Dear Guillem García Subies, David Betancur Sánchez and Alejandro Vaca,

We are glad to inform you that your paper "HAHA2021: Humor Analysis based on Human Annotation" (please check title, see below) has been accepted for publication in the proceedings of IberLEF 2021 for the task HAHA 2021. Below are the comments of the reviewers, please take in consideration the suggestions made by the reviewers for your revised version of the paper. The deadline for the final version of the paper has been moved to July 3, but please notice this deadline is strict.

D.2. *Paper*

A continuación se adjunta la primera versión del *paper* enviada al workshop.

HAHA2021: Humor Analysis based on Human Annotation

Guillem García Subies[1], David Betancur Sánchez[1], Alejandro Vaca[1]

Instituto de Ingeniería del Conocimiento, Francisco Tomás y Valiente st., 11 EPS, B Building, 5th florr UAM Cantoblanco. 28049 Madrid, Spain
guillem.garcia@iic.uam.es david.betancur@iic.uam.es
alejandro.vaca@iic.uam.es

Abstract. This paper describes a system created for the HAHA 2021 shared task, framed within the IberLEF 2021 workshop. We present an approach mainly based in fine-tuned and hyperparameter-optimized BERT models for binary, multi class and multi label classification. Our models far outperform the baselines and achieve results close to to the state-of-the-art. We also present a shap-values based model to explain predictions on what is humorous and what's not.

Keywords: humour Detection · BERT · Transformers · multiclass · multilabel · hyperparameter optimization

1 Introduction

Humor research has been done historically from different domains such as linguistics, history, literature and psychology. Machine learning and computational linguistics are some tools that have been implemented on certain studies [2,9,12] but there's still a lot to tackle.

This article shows the process of using a BERT model in Spanish to predict some of the present tasks such as humor prediction, humor mechanism and humor target. For all the tasks, a fine-tuning was performed for binary, multiclass and multilabel problems. Additionally, for the binary task, a hyperparameter optimization was performed.

After predictions were performed, some explicability models were used to show the true portions of the text that was giving the humor to give us some insights on why some of them produce laughter.

2 Related Work

Most of the work in humor detection is dated before the appearance of Transformer [14] models. After this milestone, the state-of-the-art models started to be based in Transformers.

In a generic way, Sun et al. [13] propose very interesting techniques for text classification. However these are more focused in longer text.

Specifically, for short and humorous texts, ColBERT [1] a novel approach based on sentence embeddings that achieves the best state-of-the-art results for English data.

For the Spanish language, we can underline the results of the 2019 HAHA shared task [5] where the best results were BERT-based models.

3 Tasks Description

The main corpus consists of 24000 texts in Spanish for training and 6000 for evaluation. For each text we predicted if it was humorous or not, the mechanism of the joke and the target or targets the joke had.

The first task, Humor Detection consisted on determining if a tweet is a joke or not (intended humor by the author or not). The performance of this task was measured using the F1 score of the 'humorous' class.

The second task consisted on Funniness Score Prediction. Here we didn't made a competitive model.

The third task was Humor Mechanism Classification, which consisted on predicting the mechanism by which the tweet conveys humor from a set of classes such as irony, wordplay or exaggeration. In this task, only one class per tweet was allowed. The performance of this task was be measured using the Macro-F1 score.

The fourth and last task was Humor Target Classification, which consisted on predicting the target of the joke (what it is making fun of) from a set of classes such as racist jokes, sexist jokes, etc. The performance of this task was measured using the Macro-F1 score.

As we can see in Table 1, the dataset is unbalanced. That is the main reason that the F-measure is used as the ranking metric.

Class		**Nº Samples Task1**	**Nº Samples Task3**
non-humor		14747	14747
wordplay	humor	9253	701
reference			578
exaggeration			476
unmasking			441
misunderstanding			416
absurd			566
irony			371
analogy			319
embarrassment			301
parody			255
stereotype			230
insults			146

Table 1. Distribution of Samples

In the table below, we can see some illustrative examples of the data and their labels:

La realidad es dura pero se tiene que afrontar.	non-humor
#20CosasQueHacerAntesDeMorir: Enseñarles la diferencia entre: -Hay de haber -Ahí de lugar -Ay de exclamar - Ai se eu te pego.	reference
Te quiero pero #YoTan Twitter y tú tan Facebook.	analogy
Cambié mi contraseña de Twitter por "incorrecta", si se me olvida, twitter me la recordará: Su contraseña es "incorrecta". Soy una genio	irony
WhatsApp cayó varias veces en 2015 y vos todavía no caes que nadie te soporta.	insults
Soy virgen, lo juro por mis dos hijos!	absurd
—Bienvenido a los X-Men, ¿cuál es tu poder? —Creo regresaré con mi Ex —Muy bien, te llamaremos "Bestia".	parody
–¿Tiene pastillas para la diarrea? –No. –Ok, deme un rollo de papel higiénico :(	embarrassment
—Hola linda, ¿Por qué tan sola? —Es que me vine a tirar un pedo.	unmasking
—¿Y Thomas? —No, yo no tomo. —No, ¿Que si Thomas vino? —No me gusta el vino. —¡No! ¡¿Que si llegó Thomas?! —No, no tomaré ni aunque llegues.	misunderstanding
Teníamos una farmacia pero la cerramos porque no teníamos mas remedio. #fb	wordplay
Si yo fuera presidente haria pintar la casa de gobierno de celeste , porque soy varon #chistes #humor	stereotype
Doctor, ¿cuanto me queda de vida? - Diez... - Diez qué? - Diez, nueve, ocho, siete... #humortico	exaggeration

Table 2. Examples of the different classes

4 Models

4.1 Data Preprocessing

We performed a simple preprocessing where we substituted some expressions with a more normalized form:

- Every URL was replaced with the token "[URL]" so we don't get strange tokens when the tokenizer tries to process and URL. Furthermore, no semantic information about humor can be inferred from a URL, the only information relevant for the model is that there is a URL in that token.
- The hashtag characters ("#") were deleted ("#example" → "example") because the base language models we will use, are trained in generic text and might not understand their meaning. Furthermore, most of hashtags are used the same way as normal words.

– We replaced every username with the generic token "[USER]" because the exact name of a user does not really add any information about the humor. The only relevant feature is knowing if someone was mentioned or not, but not who.
– Finally we normalized every laugh ("jasjajajajj" → "haha") so we minimize the noise of the misspellings, common in social networks.

4.2 Baselines

The competition owners provided some baselines to compare our models with. The baselines consisted on the following models:

– task 1: Naive Bayes with tfidf features. (0.6493 F1 over the dev corpus)
– task 2: SVM regression with tfidf features (0.6532 RMSE over the dev corpus)
– task 3: Naive Bayes with tfidf features (0.1038 macro-F1 over the dev corpus)
– task 4: Assign label X if the tweet contains one of the "top" words for label X on the training corpus (top words were selected as the 50th to 60th most frequent words for the label) (0.0595 F1 over the dev corpus)

4.3 Language Models

For our main language models we selected BETO [4], a BERT model trained with the Spanish Unannotated Corpora (SUC) [3] that has proven to be much better than the multilingual BERT model. The fine-tuning was performed distinctly for each task, varying the last layer of the model architecture to make binary (task1), multiclass (task3) and multilabel (task4) predictions. For Task1 and Task3, the default loss was used. For Task4 a custom loss was included in the Trainer class to handle multilabel data. BCEWithLogitsLoss from pytorch was used as the custom loss for this task.

In addition, for the fine-tuning process, on Task1 we carried out a Grid-search optimization over the main parameters of the neural network: learning rate, batch size and dropout rate. The search was performed with a 5-fold stratified cross-validation with the following grid: Learning rate, $(1e-6, 1e-5, 3e-5, 5e-5, 1e-4)$; batch size, $(8, 16, 32)$ and dropout rate, $(0.08, 0.1, 0.12)$. The best parameters for both models were: learning rate, $1e-5$; batch size, 16 and dropout rate, 0.1.

On task4 for final predictions we computed the f1 metric through different thresholds on the validation set in order to convert the logits to classes. The threshold values evaluated were 0.2, 0.3 and 0.4. The best one on validation was 0.2 but on test the best result was on a 0.4 threshold.

For explainability, shap values were calculated for each token of the sentences. Random sentences were evaluated for insights look-up.

5 Experiments and Results

5.1 Experimental Setup

We trained all the models with a NVIDIA Tesla P100-PCIE-16GB GPU and a Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz CPU with 500GB of RAM memory.

The software we used was Python3.8, transformers 4.5.1 [15], pytorch 1.8.1 [10], shap 0.39.0 [8] and scikit-learn 0.24.1 [11].

5.2 Results

In the Table 3 we can see the results for our models in the test set of the first task, where we obtained the second place.

Model	**Accuracy**
BETO finetuned	0.8716
Naive+TFIDF	0.6619
Winner	0.8850

Table 3. Results for task1

For the third task, the results dropped in terms of F1, this for the difficulty of a multiclass model. In the Table 4 we can look at them in more depth.

Model	**Accuracy**
BETO	0.2522
Naive+TFIDF	0.1001
Winner	0.3396

Table 4. Results for task3

Finally, for the task 4 results varied a lot. The baseline was very poor and the winner was very far on top of the other competitors. Our model did good compared to the baseline, but there is a long way to reach the winner.

Model	**Accuracy**
BETO	0.3110
Top-words	0.0527
Winner	0.4228

Table 5. Results for task3

For the explicability shap model we visualized the sentences, highlighting the important parts that led the model to make a "humor" prediction. We found 3 main cases:

- case 1: Some jokes have a very specific format, such as dialogues between characters. For example on 1 the "–" that characterize a dialog, is very important on a prediction for a humor sentence.
- case 2: Some jokes are not exactly on the train set, but some are very similar, so the model overfits under this jokes and gives high values for predictions. For example in 2 seems like there is an overfitting. We searched on the train set and found this tweet: —Mi amor ¿me compras un teléfono? —¿Y el otro? —El otro me va a comprar un iPad —¡ME REFERÍA AL OTRO TELÉFONO! — :decepcionado: AY!! Both texts are very similar, so the model overfits.
- case 3: Finally there is the case that we think represent the best kind of prediction. Where the model understands relations between words and set the context as a humorous one. One example can be seen on 3

Fig. 1. Case for format joke

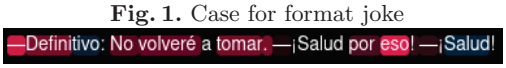


Fig. 2. Case for repeated joke

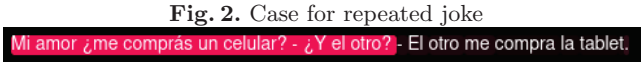
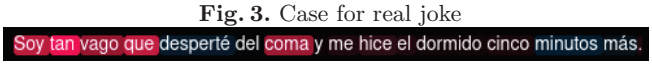


Fig. 3. Case for real joke



6 Conclusions and Future Work

Through this shared task, we have seen that NLP can be of great help in detecting and classifying humorous and non-humorous texts and there is still a long way to go.

The results obtained by our systems are very promising given their great performance and their simplicity. Furthermore, the use of explicability models

can really help get some insight on models behaviour for this kind of data. All this is very significant and could lead to much better results when combined with other improvements from the state-of-the-art.

We believe that our results could improve a lot using specific language models trained with corpora from social networks like TWilBert [6] for Spanish tweets. Finally, we have proven that good hyperparameters are also key for a good neural network so a better search, like the Population Based Training [7], would further improve the model.

Acknowledgments

This work has been partially funded by the Instituto de Ingeniería del Conocimiento (IIC) and the hardware used was also provided by the IIC.

References

1. Annamoradnejad, I., Zoghi, G.: Colbert: Using bert sentence embedding for humor detection (2021)
2. Castro, S., Cubero, M., Garat, D., Moncecchi, G.: Is this a joke? detecting humor in spanish tweets (11 2016). https://doi.org/10.1007/978-3-319-47955-2_12
3. Cañete, J.: Compilation of large spanish unannotated corpora (May 2019). https://doi.org/10.5281/zenodo.3247731, https://doi.org/10.5281/zenodo.3247731
4. Cañete, J., Chaperon, G., Fuentes, R., Pérez, J.: Spanish pre-trained bert model and evaluation data. In: to appear in PML4DC at ICLR 2020 (2020)
5. Chiruzzo, L., Castro, S., Etcheverry, M., Garat, D., Prada, J.J., Rosá, A.: Overview of haha at iberlef 2019: Humor analysis based on human annotation. In: IberLEF@ SEPLN. pp. 132–144 (2019)
6. Ángel González, J., Hurtado, L.F., Pla, F.: Twilbert: Pre-trained deep bidirectional transformers for spanish twitter. Neurocomputing (2020). https://doi.org/https://doi.org/10.1016/j.neucom.2020.09.078, http://www.sciencedirect.com/science/article/pii/S0925231220316180
7. Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K.: Population based training of neural networks (2017)
8. Lundberg, S.M., Lee, S.I.: Shap: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf
9. Mihalcea, R., Strapparava, C.: Learning to laugh (automatically): Computational models for humor recognition. Computational Intelligence **22**, 126–142 (05 2006). https://doi.org/10.1111/j.1467-8640.2006.00278.x
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In:

Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
11. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
12. Sjöbergh, J., Araki, K.: Recognizing humor without recognizing meaning (07 2007). https://doi.org/10.1007/978-3-540-73400-0_59
13. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? (2020)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
15. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020), https://www.aclweb.org/anthology/2020.emnlp-demos.6