Aplicando modelos Transformers sobre la tarea de clasificación en 5 estrellas en las Reviews de Yelp

Alexander Lique Lamas 1, Diego Vásquez Lévano 2, Marcos Alania Vicente 3

Facultad de Ciencias 1*, Universidad Nacional de Ingeniería* 1*, e-mail: alexander.lique.l@uni.pe*
Facultad de Ciencias 2*, Universidad Nacional de Ingeniería* 2*, e-mail: diego.vasquez.l@uni.pe*
Facultad de Ciencias 3*, Universidad Nacional de Ingeniería* 3*, e-mail: malaniav@uni.edu.pe*

Resumen

El campo de la mineria de texto ha tenido una gran acogida en los ultimos años debido al interes de las empresas por conocer la apreciación y perseccion de sus productos por el lado de los clientes, por lo que el análisis de las opiniones de las compradores es de vital importancia, agregando que también es importante para los usuarios al decidir que producto comprar y/o recomendar, por lo que una manera entendible de poder evaluar sus opiniones es en base a la calificación de estrellas. Teniendo presente los avances del Deep Learning en tareas como el análisis de sentimiento, detección de emociones, entre otros, en este trabajo nos enfocamos en la tarea de predicción de estrellas usando modelos de Deep Learning. Se realizó una comparativa entre los modelos transformers más populares, donde todos los modelos fueron entrenados usando la técnica de fine tuning bajo el dataset de YELP y se obtuvo que el modelo con mejor resultado fue Beto bajo las métricas F1-score weight y Accuracy.

Palabras Claves:
Fine-tuning, Clasificación de Estrellas, Lenguaje español , Inteligencia Artificial, modelo Transformer.

Abstract

The field of text mining has been very well received in recent years due to the interest of companies in knowing the appreciation and perception of their products by the customers, so the analysis of the opinions of the buyers is of vital importance, adding that it is also important for users when deciding which product to buy and/or recommend, so an understandable way to evaluate their opinions is based on the star rating. Keeping in mind the advances of Deep Learning in tasks such as sentiment analysis, emotion detection, among others, in this work we focus on the task of star prediction using Deep Learning models. A comparison was made between the most popular transformer models, where all the models were trained using the fine tuning technique under the YELP dataset and it was found that the model with the best result was Beto under the F1-score weight and Accuracy metrics.

Keywords:
Fine-tuning, Start Classification, Spanish language, Artificial Intelligence, Transformer models.

1. INTRODUCCIÓN

Actualmente en internet se generan grandes volumenes de datos de todo tipo de datos, lo que incluye a las opiniones que se pueden extraer de foros, redes sociales, entre otros, que en conjunto tienen un gran valor comercial, todo esto debido a que las empresas

se encuentran interesadas en conocer las opiniones y/o perseccion que tiene los clientes de sus productos o servicios. Bajo esta premisa que muestra una necesidad clara, hace que áreas como el Natural Language Processing (NLP) adquieran una gran importancia y protagonismo.

El análisis de opinión es el área que estudia las emociones y las opiniones. Taxonómicamente podemos estudiarla para nivel de documento, nivel de oraciones o nivel de características, Priya et al [1].

En este trabajo nos centraremos en el nivel de oración donde se lleva a cabo la tarea de clasificar las oraciones en función del numero de estrellas que una review pueda tener, donde a grandes rasgos se puede decir que las reviews con una o dos estrellas corresponden a reviews de connotación negativa, mientras que las de 4 o 5 estrellas corresponden a reviews positivas y teniendo a las 3 estrellas como un nivel intermedio, muy similar a como seria el análisis de sentimiento con una clasificación de 5 cátegorias. Este campo no es fácil de trabajar ya que encontramos problemas como la detección de sarcasmo que en consecuencia genera un peor etiquetado en la clasificación, el alto costo que puede tener crear estas etiquetas y además, la mayoría de los datos están en inglés, Shahnawaz et al [2], a diferencia de otros idiomas como el portugués, Pereira [3].y el español, Navas et al [4].

Para resolver esta tarea se han probado diferentes enfoques, sin embargo, se ha visto que el uso de modelos de aprendizaje profundo ha tenido excelentes resultados, Dang et al [5], Vijayvergia et al [6].

Dado que no tenemos un gran corpus de español, entrenar un modelo puede ser realmente costoso e ineficiente, por lo que el uso de Transfer Learning es factible. Transfer Learning consiste en utilizar un modelo que fue entrenado para realizar una tarea y especializarlo en otra. Dentro del campo del NLP podemos encontrar modelos como BERT, RoBERTa, ELECTRA, XLNet y T5, Katikapalli et al [7].donde estos modelos han sido entrenados con un gran corpus de datos en un lenguaje específico para resolver una tarea y también donde se ha visto que se adaptan bien para otras tareas. Otro trabajo adopta un enfoque entrenado en modelos con múltiples lenguajes como BERT multilingüe, Telmo et al[8], donde se busca transformar estos modelos multilingües a un dominio monolingüe para resolver la tarea de analizar el sentimiento, Kuratov et al [9].

En nuestro trabajo nos centramos en 4 modelos basados en la arquitectura transforme: Beto, RoBERTa, ELECTRA y DistilMBert; 4 modelos que fueron entrenados inicialmente bajo diferentes estrateguias.

1.1 Justificación

La creciente necesidad que tienen las empresas de conocer sobre cómo se aceptan sus productos y la falta de un modelo robusto y con resultados aceptables en idioma español para resolver esta tarea.

1.2 Objetivos

Establecer qué modelo es el más adecuado para la tarea de clasificación de reseñas a través de estrellas en el conjunto de datos de reseñas de Yelp.

1.3 Objetivos específicos

- Mostrar las ventajas de usar modelos transformers previamente entrenados para resolver tareas de clasificación de opiniones a través de

estrellas.

- Comparar los modelos transformer Beto, RoBERTa, ELECTRA y DistilMBert usando las métricas de F1-Score weight y Accuracy.

2. CONCEPTOS PREVIOS

2.1 Redes Neuronales

El concepto de redes neuronales se deriva de la estructura de las células nerviosas en el cerebro de humanos y animales. En 1900 se concluyó que estos pequeños componentes físicos del cerebro, las células nerviosas y sus conexiones, son responsables de la conciencia, las asociaciones de pensamientos y la capacidad de aprender, Wolfgang [10].

El primer gran paso hacia las redes neuronales en IA lo dieron en 1943 McCulloch y Pitts en un artículo titulado: "A logical calculus of ideas immanent in nervous activity", Anderson et al [11]. Fueron los primeros en presentar un modelo matemático de la neurona como elemento básico de conmutación del cerebro. Este artículo sentó las bases para la construcción de redes neuronales artificiales y, por lo tanto, para esta rama tan importante de la IA, Bernhard [12].

2.2 Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNN) son de naturaleza secuencial (no paralelismo) y recurrente, en las que la salida se tiene en cuenta para el siguiente cálculo, proporcionando una memoria a corto plazo. Esto representa una ventaja ya que ofrece una solución, en el campo del NLP, pero limitada para textos extensos y con problemas como el Vanishing Gradient. Sin embargo, de ésta surgen otras arquitecturas más sofisticadas. Como las Gated Recurrent Unit (GRU), Cho et al [13] que agregan las puertas de update y reset, o la Long Short-Term Memory (LSTM), Hochreiter et al [14] con las puertas forget, update y exit, que ayudan a resolver el problema del Vanishing Gradient, sin embargo, todavía sufren de problemas de memoria para textos grandes. Es donde aparecen nuevas arquitecturas, como las Transformers, basadas en mecanismos de atención que permiten una comprensión del contexto incluso en texto grande, además de las ventajas de la computabilidad (paralelismo), entre otras.

2.3 Modelos Codificador Decodificador

Estos modelos se basan en el uso de dos módulos. El codificador, que se encargará de analizar la entrada, y el decodificador, que aprovechará la información que analizó el codificador para realizar una predicción. Arquitecturas como PIX2PIX utilizan este enfoque, Isola et al [15]. En el contexto de NLP, las arquitecturas como RNN son ampliamente utilizadas como codificadores y decodificadores, para tareas donde la entrada/salida puede recibir diferentes tipos de secuencias (una o varias entradas/salidas), por lo que podemos clasificarlas en many-to-many, many to one, one to one y one to many. Gracias a este enfoque podemos resolver problemas como la traducción automática neuronal (many to many), descripción de imágenes (one to many), y en nuestro caso la clasificación de estrellas , que es un problema de tipo many-to-one ya que se busca que el codificador del módulo analiza la secuencia de entrada, en este caso las reseñas de Yelp, y teniendo como salida un número que indica la puntuación de estrellas del

1 al 5.

2.4 Transformers

Los modelos Transformers son una de las arquitecturas más utilizadas en la actualidad, siendo el campo del procesamiento del lenguaje natural (NLP) donde se están realizando grandes aportaciones como GPT-3, Brown et al [16], BERT y más.

Una de las ventajas de estos modelos es que pueden paralelizarse, a diferencia de otras arquitecturas predecesoras, como las RNN, que no podían debido a su naturaleza secuencial. Los autores de la arquitectura de transformadores propusieron un modelo basado en modelos de codificador-decodificador, apoyándose en mecanismos de atención.

En la figura podemos ver que se le agrega un positional encoding al codificador y decodificador, esto se debe a que ya no estamos trabajando con un RNN en el cual teníamos una posición establecida para cada secuencia de palabras, en consecuencia se le agrega un signal que depende de la ubicación de la incrustación para que el modelo tenga una noción del orden de esta palabra en la oración.

También el codificador y decodificador están compuestos por varios módulos que se pueden apilar, los cuales se ven en la Figura 1 como Nx, lo que indica que se está apilando N veces, y en el paper original se apilan 6 de ellos. Dentro del codificador tenemos una multi-head attention layer, seguida de una norm layer y un residual block para luego pasar a una forward layer que también tiene una residual y norm layer. El decodificador también tiene 6 capas, sin embargo, se agrega una capa adicional de multi-head attention entre la forward layer y la first multi-head attention. Adicionalmente se le agrega una masking layer a la first multi-head attention, esto debido a que para el entrenamiento oculta los tokens a predecir.

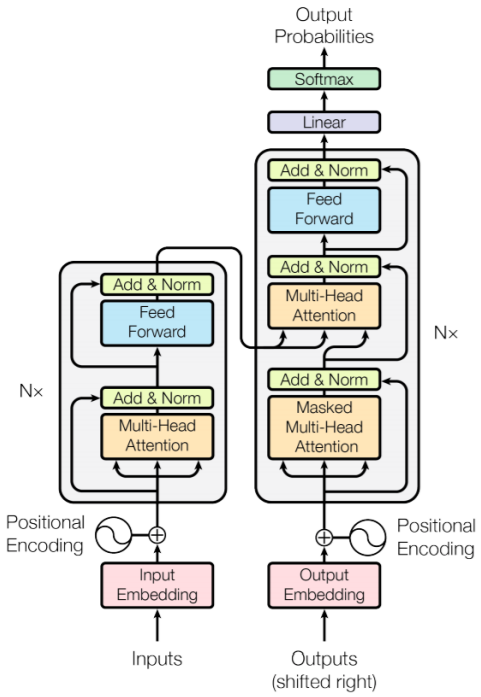

Figura 1: Transformer Model Architecture, Vaswani et al [17]

2.5 Aprendizaje Auto Supervisado

A diferencia del enfoque supervisado hace uso de etiquetas para realizar el entrenamiento teniendo como consecuencia buenos resultados en la tarea especifica. El hecho de depender de etiquetas hace que el proceso de recolectarlas sea costoso, consuma mucho tiempo y en algunos casos muy dificil encontrar en ciertos dominios como en la medicina, finanzas, etc. El aprendizaje auto supervisado, Katikapalli et al[18] es un nuevo paradigma el cual ha traido el interés a los investigadores en los ultimos años debido a su capacidad usar datos no etiqueta-

dos para inyectar conocimiento universal acerca de imágenes, lenguaje, sonido atravez de modelos pre entrenados. La idea consta de darle una de entrada, donde una fracción de la data esta etiquetda, y este tiene que aprender a predecir esta nueva etiqueta.

2.6 Estrategias de Pre-Entrenamiento

estándar debido lo costoso que puede resultar entrenar un modelo desde cero. Dentro del Aprendizaje auto supervisado, Chaudhary [19], encontramos un marco llamado tareas de pretexto las cuales hacen uso de la datos, que son usados en el entrenamiento, el cual nos permitiran generar etiquetas y resolver problemas no supervisados haciendo uso de métodos supervisados. Esto se le conoce como pre-entrenamiento. Las representaciones aprendidas se pueden usar como punto de partida para especializarlos en una tarea supervisada.

En el Procesamiento Natural del Lenguaje el uso de modelos Pre entrenados (tales como BERT, ROBERTA, GPT entre otros que fueron pre-entrenados en diversas estrátegias) que han aprendido a extraer las características del lenguaje han tenido buenos resultados al momento de especializarlos en una tarea, como por ejemplo Análisis de sentimiento, Preguntas y respuestas, etc.

La ventaja de realizar estas estrategias sobre el conjunto de datos es que se realizan tareas sencillas de programar de tal manera de que se evita depender de etiquetas y asi poder tener modelos robustos. Algunas estrategias aplicadas en el pre-entrenamiento de los modelos transformers son:

Modelamiento del lenguaje enmascarado: Esta estrategia consta de tomar la oración de entrada y ocultar una palabra por un token especial llamados [Mask], la idea es que el modelo sea pre-entrando en la tarea de predecir que palabra debe estar en vez el token especial.

Me [**MASK**] la Inteligencia Artificial ⇒ Me **gusta** la Inteligencia Artificial

Modelado del lenguaje autoregresivo: Se trata de tomar la data cruda y tratar de pre entrenar al modelo en base a predecir la siguiente palabra en la oración dada una secuencia de palabras.

Todo es _____ ⇒ Todo es posible.

Predicción del siguiente oración: De la data se toma dos oraciones consecutivas o una oración y otra aleatoria. La tarea Es clasificar dada dos oraciones si estas son consecutivas o no.

(1) Voy de viaje a Japon. **Traere recuerdos para todos del viaje.**

(2) Voy de viaje a Japon. **Llama a tu abuela por su cumpleaños**.

Como se ve en los ejemplos en (1) las dos oraciones son consecutivas a diferencias de (2) que no parecen serlo.

Predicción del Token por reemplazo. Se trata de dar reemplazar ciertos tokens de la oración por otros de tal manera de que El modelo aprenda a predecir si la oración posee tokens reemplazados.

El chef **cocina** la cena ⇒ El chef **come** la cena.

2.7 BERT

BERT es un modelo basado en arquitectura Transformer, que ha sido entrenado con un enorme corpus

de datos, lo que era Wikipedia y Google Books, y esto solo se conformó con un codificador. Este modelo es el más utilizado para tareas precisas debido a su solidez al realizar diferentes tareas, por ejemplo, preguntas y respuestas, análisis de sentimientos, etc.

A diferencia de otros modelos como GPT, que analizan la entrada de izquierda y derecha, o ELMO, Peters et al [20], que hace uso de dos LSTM para poder analizar las entradas en ambas direcciones; BERT también analiza las entradas en ambas direcciones, sin embargo, a diferencia de ELMO, BERT es un modelo único.

Los autores de BERT muestran este modelo en dos variantes que son BERT base y BERT large. Ambos modelos cuentan con un gran número de capas de codificador, siendo 12 y 24 respectivamente. También tiene 768 y 1024 capas delanteras y también 12 y 16 capas de atención, respectivamente. Al recibir una entrada, el modelo agrega un token [CLS], que por problemas de clasificación, BERT condensa toda la información analizada bidireccionalmente en este token especial, por lo que para este tipo de tareas solo es necesario analizar este token. Si recibe dos pares de oraciones de entrada, usa un token [SEP] que separa estos dos pares y luego agrega la incrustación aprendida a cada token que indica si pertenece a la oración A o a la oración B.

Con respecto al entrenamiento de este modelo, se dividió en dos partes, la primera pre-entrenamiento que ocultó algunos tokens en la entrada y obligó al modelo a predecir qué token falta en la oración y un segundo pre-entrenamiento que quería que el modelo predecir, es decir, dada una oración A, predecir si la oración dada B es la oración que sigue o es una oración aleatoria. Para ello se utilizó un corpus de datos donde el 50 % de las frases B eran las siguientes y el otro porcentaje aleatorias.

2.8 ROBERTA

RoBERTa Yinhan Liu et al[20], es un modelo introducido por FACEBOOK, el cual consiste en realizar un enmascaramiento de la data en el mismo proceso de entrenamiento, al que se llama enmascaramiento dinamico, lo que brinda que para una epoca determinada el enamscaramiento de una oracion sera diferente al de las siguientes epocas. A su vez, el entrenamiento de RoBERTa se basa solo en el enmascaramiento, eliminandose la prediccion de la siguiente oración. Este modelo a diferencia del de BERT fue entrando con un corpus de datos más grande , 160GB de texto.

2.9 DistilBERT

Modelos como RoBERTa o Bert dan muy buenos resultados para las tareas de Analisis de sentimientos, preguntas y respuestas, entre otros. Sin embargo, el costo computacional que requiere para entrenar estos modelos es muy alto, requiriendo hasta dias para poder poder realizar un entrenamiento completo. Afortunadamente existen soluciones a esta problematica, como la destilacion de modelos, Geoffrey Hinton et al [21], que aplicado al modelo de BERT obtenemos DistillBERT, Victor Sanh [22] et al, su version destilada.

Este proceso de destilación consta de trabajar con dos modelos, uno maestro que enseña al otro modelo, llamado modelo alumno. Para el caso de distilBert, se tomo al modelo Bert como modelo maestro, mientras que el modelo alumno fue el modelo destilado con la mitad de capas que Bert. Como resultado se obtuvo un modelo mas ligero pero con un menor rendimiento.

Para este proceso se usa una pérdida triple (triplet loss) que combina el modelado del lenguaje, destilación y pérdida de incrustación de coseno.

2.10 Electra

Electra , es un modelo pre entrenado en la tarea de la predicción de token de reemplazo. Se vio experimentalmente, que da mejores resultados que la estrategia de enmascaramiento.

Este modelo fue entrenado en base a dos redes neuronales, un generador y un discriminador. El generador, en este caso, un modelo de BERT entrenado en la tarea del modelado del lenguaje el cual se encargara de crear una secuencia corrupta y se lo pasara al discriminador quien tratara de distinguir que tokens fueron reemplazados. Tomando el discriminador como base para especialiazarlo en alguna tarea específica.

3. ANÁLISIS

En esta sección tiene como objetivo pesentar el análisis del conjuntos de datos usado asi como tambien presentar el hardware y métodologia realizada para el entrenamiento.

3.1 Conjunto de datos

El conjunto de datos usado se trata de las "reviews" de Yelp. Este sitio web ofrece a los usuarios opiniones de más de 100 millones de reseñas. Asi como también ofrece estas reviews para el ámbito académico.

	Start	Review
0	5	dr. goldberg offers everything i look for in a...
1	2	Unfortunately, the frustration of being Dr. Go...
2	4	Been going to Dr. Goldberg for over 10 years. ...
3	4	Got a letter in the mail last week that said D...
4	1	I don't know what Dr. Goldberg was like before...

Figura 2: Conjunto de datos de Yelp

3.2 Procesamiento del Conjunto de datos

Para esta etapa primero teniendo la data en ingles se removió los links, emoticones, signos de puntuación menos los puntos y las comas. Posteriormente, se usó la api de google translate y se tradujo al español del total de reviews 1911813 y en test 147129 en el entrenamiento y test, inmediamtene despues se separó el conjunto de test en validación y test. Finalmente se hizo un análsis del dataset.

Se vio en la media de los datos se encuentra en 137 palabras y que la longitud de las reviews en su mayoria rondan entre las 1 y 200 palabras como se ve en la Fig 4 de la data de entrenamiento .

```
count    637271.000000
mean        137.656760
std         121.575576
min           1.000000
25%          56.000000
50%         102.000000
75%         179.000000
max        1049.000000
Name: Text, dtype: float64
```

Figura 3: Distribución del conjunto de datos de entrenamiento

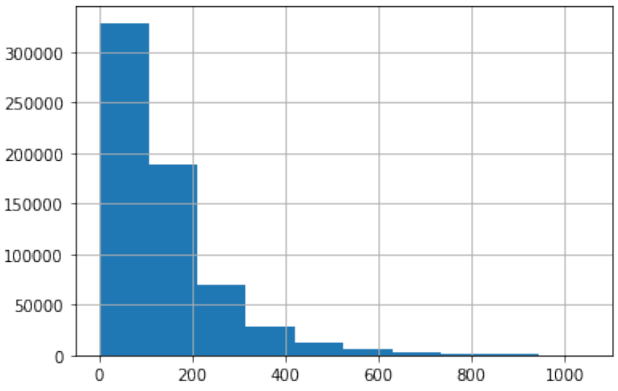

Figura 4: Números vs Tamaño de las reviews

También se vió la distribución de las reviews don-

de se pudo notar que esta bien balanceado el conjunto de datos.

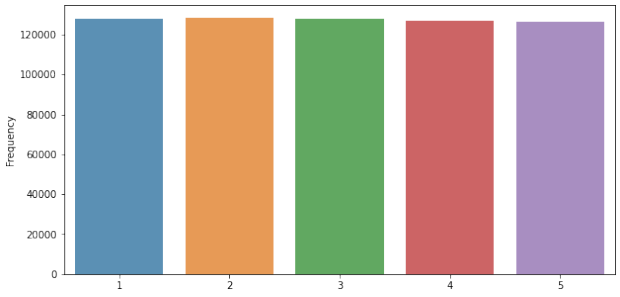

Figura 5: Cantidad de reviews vs estrella

Por último se intentó ver si la cantidad de palabras por review seria una carácteristica útil para el entrenamiento sin embargo se vió que no hay una correlación lineal entre la clasficación de reviews y la cantidad de palabras, por ended se descarto esta característica, (no hay correlación entre dicha carácteristica y el target).

Figura 6: Cantidad de reviews vs estrella

3.3 Ajuste Fino sobre los modelos

Debido a que entrenar un modelo desde cero será computacionalmente costoso, optamos por partir de modelos pre-entrenados en diferentes estrategias lo que les permite estar capacitado para entender el contexto del lengauje como lo son BERT, RoBERTa. Electra y DistilMBERT. como estamos trabajando con datos en español, entonces usamos sus Variantes como lo es Beto y DisitillMBert, Robertin . Para aplicar el proceso de ajuste fino tomamos su vector cls (token de clasificación),. que es el que contiene de forma resumida todos los información de la frase de entrada y luego agregamos un Dropout capa para apagar ciertas neuronas y hacer el modelo no overfit, finalmente, agregamos una capa simple donde volveremos la predicción, Liu et al [23].

3.4 Métricas

Para medir los resultados de los dos modelos, nos basamos en las métricas más comunes en el problema de clasificación, las cuales son accuracy y F1 score, métricas que podemos extraer de la mátriz de confusión.

- Accuracy: Consiste en una división entre el número de aciertos y el total de ejemplos.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Sin embargo, esta métrica puede no reflejar un resultado real ya que el modelo puede estar clasificando bien en una clase pero mal en otras y aun así obtener un porcentaje alto. Es por eso que también elegimos la métrica f1-score.

- F1 score: Esta métrica hace uso de otras dos métricas que son la precisión y la recuperación y de las cuales se toma la media armónica.

$$F1 = \frac{2 \cdot prescicion \cdot recall}{presicion + recall}$$

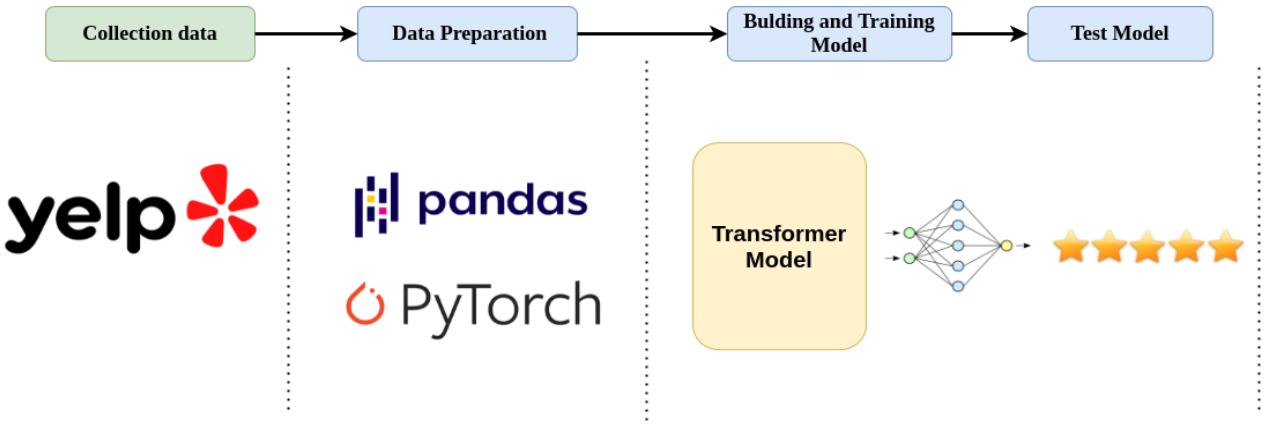

Figura 7: Proceso del trabajo

3.5 Entrenamiento

Para el entrenamiento su uso una GPU Nvidia GeForce RTX 2070 SUPER 8GB, RAM instalada 32.0 GB (31.8 GB usable), sistema operativo de 64 bits Windows 11 Pro, procesador Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz).
El número de palabras a recibir de entrada fue de 128 palabras, debido a las limitaciones con respecto al poder computacional, junto con un truncanmiento HEAD TAIL, que coge los 64 primeros y últimos tokens de la oración. Para los hiperparámetros tomamos los valores recomendados para tareas de ajuste fino de BERT. Se tomó radio de aprendizaje de 2e-5. Para el optmizador se usó Adam y la función de perdida llamada entropía cruzada, con entrenamiento de 1 época. Además, se utilizó el mini lote con un tamaño de lote de 32, ya que como trabajamos con un gran número de Reviews", puede ser computacionalmente costoso usar todos los ejemplos como se haría utilizando el descenso de gradiente.

4. OBSERVACIONES

En la tabla mostramos los resultados de los modelos en función las métricas establecidas como lo son accuracy y f1 weight.

- Se descarto el usar una caracteristica extra, como la longitud de la review debido a que no hay una correlación entre el valor a predecir y esta caracteristica.

- Se trabajo con una época ya que se vio en el experimento que en la siguientes solo hubo mejora en el error de entrenamiento y no en el de validación lo cual muestra que esta entrando en un sobreajuste.

- Se optó por usar un Max Len (longitud máxima del modelo) de 128 tokens debido a que con este limite se los resultados obtenidos son cercanos a los resultados que se pueden obtener con una mayor Max Len, pero con un tiempo de entrenamiento menor.

- El mejor modelo fue Beto usando la métrica de F1-Score weight con, el cual no superó por mucho a los demás modelos. Estos resultados se obtuvieron usando los hiperparametros

Modelo	Accuracy(%)	F1 Weight(%)
Beto	67.53	67.74
Roberta	66.93	67.02
Electra	66.65	66.73
DistillMBert	63.97	64.13

Cuadro 1: Resultados de los módelos

de LR:2e-5, Batch Size: 32, Max Lenght: 128, usando un truncamiento Head and Tail.

- Si bien es cierto que se han dado resultados interesantes, resulta complicado interpretar porque el modelo elige una determinada estrella para una u otra review.
- Como era de esperarse el modelo con menor resultado y el más ligero fue DistilMBERT.

5. CONCLUSIONES

Ya que el poder analizar las opiniones de las personas en base a las reviews es algo de interes para las empresas, en este trabajo se realizó una comparación entre modelos con arquitecturas transformers pre-entrenadas usando diferentes estrategias para la tarea de la clasificación de reviews a través de estrellas, de esto se obtuvo que Beto obtiene un mejor resultado en comparación a los demás modelos bajo las métricas de F1-Score weight y accuracy. Sin embargo, todos los modelos obtuvieron resultados cercanos, exceptuando al modelo DistilMBERT, el más ligero de todos y que se diferencia alrededor de un 3 y 4 porciento.

Por otro lado, la interpretabilidad del modelo resulta complicada de entender en cuanto a la eleccion de una estrella u otra para una review dada.

Para trabajos futuros nos centraremos en analizar la interpretabilidad de los modelos, así como probar en otros corpus de datos de diferentes dominio mas especificos como el de peliculas, comida, entre otros. Adicionalmente, se probarán diferentes hiperparámetros y con otros tipos de truncamiento que se pueden encontrar en la literatura.

Referencias

[1] Priya Chakriswaran y col. "Emotion AI-Driven Sentiment Analysis: A Survey, Future Research Directions, and Open Issues". En: *Applied Sciences* 9 (2019), pág. 5462.

[2] Parmanand Astya y col. "Sentiment analysis: approaches and open issues". En: *2017 International Conference on computing, Communication and automation (ICCCA)*. IEEE. 2017, págs. 154-158.

[3] Denilson Alves Pereira. "A survey of sentiment analysis in the Portuguese language". En: *Artificial Intelligence Review* (2020), págs. 1-29.

[4] María Navas-Loro y Víctor Rodríguez-Doncel. "Spanish corpora for sentiment

analysis: a survey". En: *Language Resources and Evaluation* 54 (jun. de 2020). DOI: 10.1007/s10579-019-09470-8.

[5] Nhan Cach Dang, Maria N Moreno-Garcia y Fernando De la Prieta. "Sentiment analysis based on deep learning: A comparative study". En: *Electronics* 9.3 (2020), pág. 483.

[6] Aditya Vijayvergia y Krishan Kumar. "STAR: rating of reviewS by exploiting variation in emoTions using trAnsfer leaRning framework". En: *2018 Conference on Information and Communication Technology (CICT)*. 2018, págs. 1-6. DOI: 10.1109/INFOCOMTECH.2018.8722356.

[7] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan y Sivanesan Sangeetha. "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing". En: *CoRR* abs/2108.05542 (2021). arXiv: 2108.05542. URL: https://arxiv.org/abs/2108.05542.

[8] Telmo Pires, Eva Schlinger y Dan Garrette. "How multilingual is Multilingual BERT?" En: *CoRR* abs/1906.01502 (2019). arXiv: 1906.01502. URL: http://arxiv.org/abs/1906.01502.

[9] Yuri Kuratov y Mikhail Y. Arkhipov. "Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language". En: *CoRR* abs/1905.07213 (2019). arXiv: 1905.07213. URL: http://arxiv.org/abs/1905.07213.

[10] Wolfgang. Ertel. *Introduction to Artificial Intelligence [electronic resource] / by Wolfgang Ertel.* eng. 1st ed. 2011. Undergraduate Topics in Computer Science. London: Springer London, 2011. ISBN: 0-85729-299-4.

[11] James A Anderson y Edward Rosenfeld. "Neurocomputing: Foundations of Research MIT Press". En: *Cambridge, MA* (1988).

[12] Bernhard Mehlig. *Machine Learning with Neural Networks: An Introduction for Scientists and Engineers*. Cambridge University Press, 2021.

[13] KyungHyun Cho y col. "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches". En: *CoRR* abs/1409.1259 (2014). arXiv: 1409.1259. URL: http://arxiv.org/abs/1409.1259.

[14] Sepp Hochreiter y Jürgen Schmidhuber. "Long Short-Term Memory". En: *Neural Comput.* 9.8 (nov. de 1997), págs. 1735-1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[15] Phillip Isola y col. "Image-to-Image Translation with Conditional Adversarial Networks". En: *CoRR* abs/1611.07004 (2016). arXiv: 1611.07004. URL: http://arxiv.org/abs/1611.07004.

[16] Tom B. Brown y col. "Language Models are Few-Shot Learners". En: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[17] Ashish Vaswani y col. "Attention Is All You Need". En: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[18] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan y Sivanesan Sangeetha. "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing". En: *CoRR* abs/2108.05542 (2021). arXiv: 2108.05542. URL: https://arxiv.org/abs/2108.05542.

[19] Amit Chaudhary. *Self supervised representation learning in NLP*. Sep. de 2020. URL: https://amitness.com/2020/05/self-supervised-learning-nlp/?fbclid=IwAR1H-2IrCuNf7X%20VbfaqF-ByieWPDHH8RC0aer9sXiCdUbLmSzNDP4miNLfo.

[20] Matthew E. Peters y col. "Deep contextualized word representations". En: *CoRR* abs/1802.05365 (2018). arXiv: 1802.05365. URL: http://arxiv.org/abs/1802.05365.

[21] Geoffrey Hinton, Oriol Vinyals y Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. DOI: 10.48550/ARXIV.1503.02531. URL: https://arxiv.org/abs/1503.02531.

[22] Victor Sanh y col. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". En: *CoRR* abs/1910.01108 (2019). arXiv: 1910.01108. URL: http://arxiv.org/abs/1910.01108.

[23] Zefang Liu. *Yelp Review Rating Prediction: Machine Learning and Deep Learning Models*. 2020. DOI: 10.48550/ARXIV.2012.06690. URL: https://arxiv.org/abs/2012.06690.