Análisis de sentimientos con BERT y Tensorflow

Miguel Angel Soto Hernandez

Centro de Investigación en Computación, Instituto Politécnico Nacional
Ciudad de México, México
msotoh2021@cic.ipn.mx

Resumen

En el procesamiento del lenguaje natural se ha hecho muy popular la investigación en el análisis de sentimiento, específicamente las reseñas tanto de productos como de servicios. Las redes sociales se han convertido en los lugares más comunes para compartir sus emociones en textos relativamente cortos; estas emociones pueden ser felicidad, tristeza, anisedad, miedo, etc. El análisis de sentimientos en las reseñas de distintos sitios web se logran identificar de una manera más general por los criticos mostrando sentimientos positivos, neutros o negativos. En este proyecto se explorará y propondrá un modelo basado en BERT y Tensorflow el cual pueda clasificar de manera clara estas criticas, en especifico utlizando un dataset de reseñas de usuarios de Google PlayStore.

I Introducción

Las redes sociales, aplicaciones mobiles y de escritorio y los sitios web se han convertido en una fuente muy importante para la recolección de datos, ya que las personas quieren compartir todo lo que pasa en su vida cotidiana, y esto nos proporciona datos de manera masiva en forma de audio, texto y videos como parte de sus opiniones acerca de multiples circunstancias y temas que acontecen en la sociedad día con día. Las opiniones dadas en el formato de texto generalmente suele ser a través de textos cortos como tweets, reseñas o comentarios, los cuales por medio de investigaciones recientes (C. D. Santos y M. Gatti, 2014)[1] podemos inferir que este tipo de textos suelen ser muy efectivos y son una fuente muy importante para generar conocimiento específico para tareas de procesamiento de lenguaje natural, con los cuales se pueden generar distintas investigaciones como lo puede ser la categorización de contenidos, sumarización de documentos, análisis de sentimientos, etc.

Para el proceso de toma de desiciones de la población en general, una de las piezas más importante ha sido la opinión de otras personas es por eso que en este trabajo se busca realizar un modelo el cual a partir de una crítica se pueda determinar si es una critica buena, neutra o mala con la finalidad de hacer esta toma de desiciones aún más rápida, ya que esto nos podría generar en un futuro estadísticas más consisas. El análisis de sentimientos (SA por sus siglas en inglés) ha tenido un rol muy importante en distintos ámbitos sociales que han ocurrido recientemente. Las elecciones del año 2016 en los Estados Unidos (R. Ahmad et al., 2016)[2] ha sido un hecho de investigación, ya que la población compartió sus gustos y sus disgustos hacia un partido político en sitios como Twitter, Facebook e incluso en blogs de internet. Dichos comentarios fueron analizados en su momento y por lo tanto los candidatos pudieron tomar esto a su favor con las estadísticas que les generaban ciertos comentarios. Por lo tanto el análisis de sentimientos ayudó a generar mas popularidad y seguidores a los partidos políticos.

Las reseñas generalmete suelen ser textos cortos que expresan distintas opiniones acerca de productos o servicios que se les ofrece a la población. Estas reseñas como se menciona anteriormente juegan un papel muy importante en si dicho producto o servicio tendrá popularidad o ventas significativas. En las películas, uno de los sitios más visitados para ver reseñas es IMDb, donde se puede conocer al equipo detrás de la película, las reseñas y las calificaciones generales que otorga la comunidad, esto lo podemos observar más detalladamente en la investigación titulada “Classification of sentiment reviews using n-gram machine learning approach“ (A. Tripathy et al., 2016)[3].

En este trabajo de investigación se trabajará con un dataset de más de 15K de reseñas de Google Play Store, que serán clasificadas en 3 categorías: negativas, neutras y positivas. Se presentará un modelo basado en el algoritmo BERT capaz de identificar la categoría de una crítica dada con el fin de responder los siguientes cuestionamientos:

- ¿Qué tan eficaz es el algoritmo BERT para el análisis de sentimientos?
- ¿El modelo es capaz de clasificar los textos en las categorías asignadas con exactitud?

¿Qué es BERT?

Por sus siglas en inglés, BERT significa Pre-training of Deep Bidirectional Transformers for Language Understanding o en español Pre-entrenamiento de Transformadores Bidireccionales Profundos para la Comprensión del Lenguaje.

Modelo presentado por google en 2019, y está diseñado para preentrenar representaciones bidireccionales profundas a partir de texto no etiquetado, condicionando

conjuntamente el contexto izquierdo y derecho en todas las capas. Sus ideas principales son las siguientes:

- Bidireccional: Dada una frase, cada palabra toma contexto de las palabras anteriores y subsecuentes, mirando así hacia atrás y hacia delante.
- Transformadores: En el artículo "Attention is all you need" (Ashish Vaswani et al., 2017) se presenta por primera vez la idea de los transformadores, los cuales tienen como idea principal leer secuencias enteras de fichas, incluyen un mecanismo de atención el cual permite aprender a la máquina las relaciones contextuales entre las palabras.
- Incrustaciones de palabras contextualizadas: El artículo "Extending a Parser to Distant Domains Using a Few Dozen Partially Annotated Examples" (Vidur Joshi et al., 2018) mejor conocido como el algoritmo ELMo introdujo una forma de codificar las palabras en función de su significado o contexto. Es decir, trata la ambigüedad de las distintas palabras que pueden tener multiples significados.

II Trabajos previos

Durante años multiples investigadores han trabajado con el tema abordandolo de distintas maneras, en este trabajo nos enfocaremos en trabajos previos para el análisis de sentimientos basados en reseñas de distintos sitios web.

En el trabajo **Deep Learning Approach for Sentiment Analysis of Short Texts** (A. Hassan y A. Mahmood, 2017)[4] se probaron muchas combinaciones diferentes de hiperparámetros que pueden dar resultados similares. En este casos se dedicó más tiempo a afinar la tasa de aprendizaje de aprendizaje, el abandono y el número de unidades en la capa convolucional de la capa convolucional, ya que estos hiperparámetros tienen un gran impacto en el rendimiento de la predicción. El número de épocas varía entre (5, 10) para ambos conjuntos de datos. En este trabajo se cree que al añadir una capa capa recurrente como sustituto de la capa de agrupación puede reducir eficazmente el número de capas convolucionales necesarias en el modelo para capturar las dependencias a largo plazo. Por lo tanto, se considera la aparición de una capa capas convolucionales y una recurrente en un único modelo ConvLstm, con múltiples ftIters de ancho (3, 4, 5), mapas de características = 256, para las funciones de activación en la capa convolucional utilizamos utilizamos la función lineal rectificada (ReLus) para la no linealidad, el relleno se fijó en cero. Todos los elementos que caen fuera de la matriz se toman como cero. Para reducir la sobrefiguración, se aplica un dropout de 0,5 sólo antes de la capa recurrente. La principal contribución de este trabajo es que explota las capas recurrentes como sustitutas de la capa pooling un aspecto clave de las CNNs son las capas recurrentes, que se que se aplican normalmente después de las capas convolucionales para submuestrear su entrada, una operación máxima es la forma más común de hacer operación de pooling. Sin embargo, en este modelo se eliminó, y la capa recurrente es una capa única de la LSTM vainilla. También se utilizó el recorte de gradiente para la salida de la celda y los gradientes. El LSTM tiene puertas de entrada, olvido y salida, la dimensión del estado oculto La dimensión del estado oculto es de 128.

Por otro lado, en el trabajo de investigación llamado **Sentiment Analysis for Amazon Reviews** (Wanliang Tan et al., 2018)[5] se probó un modelo que se basa en los principales metodos de clasificación como lo son: Naive Bayes, K-nearest Neighbor, Linear Support Vector Machine y Long Short Term Memory dónde probaron dos tipos de características. El primer tipo es un método tradicional. Básicamente, se construyó un diccionario basado en las palabras comunes y se indexó cada palabra. Se fijó el umbral para el diccionario de palabras en 6 ocurrencias y se recogieron 4223 palabras de todo el conjunto de datos. A continuación, se transformó cada reseña en un vector, donde cada valor representa el número de veces que aparece la palabra. Para ello, se probó cambiar el umbral y la longitud del diccionario. Lo que se encontró es que el aumento de la longitud del diccionario no tuvo demasiado efecto en la precisión. Otro tipo de característica que se utilizó es el diccionario 50-d glove2 que fue preentrenado en Wikipedia. Para esta parte básicamente se quizo aprovechar los significados de cada palabra. En este caso, se representó cada reseña mediante el vector medio de los vectores de guante de 50 d de todas las palabras individuales que componen la reseña. Debido a la forma en que se presentó cada reseña, las características se debilitaron y la distancia entre las diferentes reseñas en realidad es no fue tan precisa.

Por último, en el trabajo **Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method** (H. M. Keerthi Kumar1 et al., 2018)[6] se plantea un método híbrido, en el que las características se se extraen utilizando tanto métodos estadísticos como léxicos. Además, aplican varios métodos de selección de características como Chi-Square, correlación, ganancia de información e indexación regularizada de preservación de la (RLPI)[5] para las características extraídas por métodos estadísticos. estadísticos. Esto asigna el espacio de entrada de mayor dimensión al espacio de entrada de menor dimensión. de entrada de mayor dimensión al espacio de entrada de menor dimensión. El método de extracción de características basado en el léxico extrae características basadas en los diccionarios del léxico. Las características de ambos métodos se combinan para formar un nuevo conjunto de características que es de dimensión inferior a la dimensión inicial del espacio de entrada. de entrada. El nuevo conjunto de características se clasifica utilizando varios clasificadores como como máquinas de vectores de apoyo (SVM), Naïve Bayes (NB), K- Nearest Neighbor (KNN) y clasificadores de máxima entropía (ME) en el conjunto de datos de reseñas de películas de IMDb.

III Conjunto de datos

Para este trabajo de investigación se estará utilizando un conjunto de datos público de reseñas de aplicaciones de Google Play Store, la cual consta de más de 15 mil datos, el cual será etiquetado en tres categorías: positivas, neutras y negativas. Dicha categorización se realizará como parte del preprocesamiento del conjunto de datos con la finalidad de normalizarlo, para esto se tomará en cuenta las calificaciones de los usuarios dadas en su reseña, y quedará de la siguiente manera:

- 1 ó 2 estrellas: negativa
- 3 estrellas: neutra
- 4 ó 5 estrellas: positiva

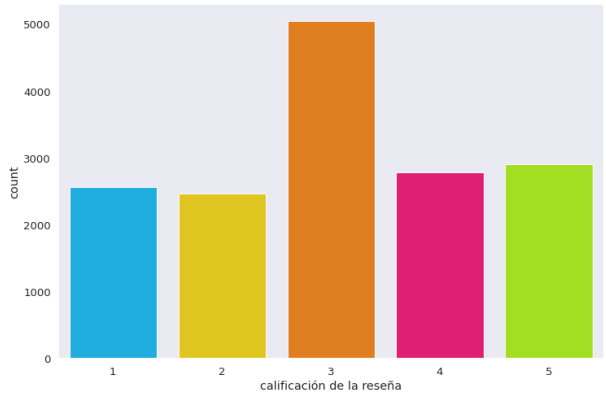

Figura 1: Conjunto de datos disperso

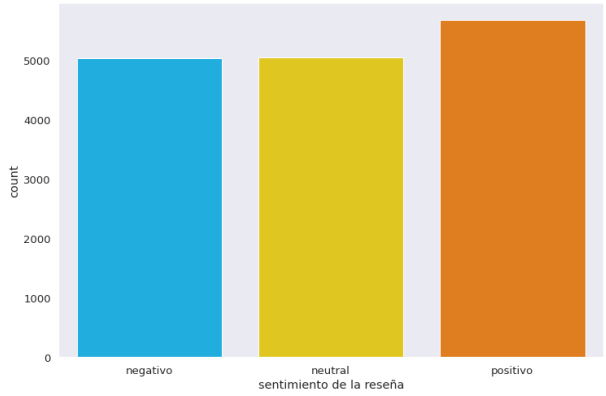

Figura 2: Conjunto de datos normalizado

IV Modelo

El modelo que se presenta en este trabajo esta basado en el modelo preentrenado BERT básico (Jacob Devlin et al., 2019)[7]. A pesar de que hay muchos ayudantes que facilitan el uso de BERT con la biblioteca Transformers. Dependiendo de la tarea, es posible que pueda utilizar BertForSequenceClassification, BertForQuestionAnswering o algún otro. Para este caso en aprticular, utilizaremos el BertModel básico y construiremos nuestro clasificador de sentimiento sobre él . Este modelo de análisis de sentimientos delega la mayor parte del trabajo a BERT, sin embargo, se usó una capa de abandono (dropout) y una capa totalmente conectada (fully-connected). Las capas del nuevo modelo quedarán ordenadas de la siguiente manera:

- **Modelo BERT preentrenado**: con el seremos capaces de obtener todo el algoritmo previamente entrenado.
- **Capa de abandono (dropout)**: para ciertas regularizaciones.
- **Capa totalmente conectada (fully-connected)**: para la salida de nuestro modelo.

Por último, para obtener las probabilidades predichas de nuestro modelo entrenado, aplicaremos la función softmax a las salidas.

V Metodología

La metodología que se seguirá para crear este modelo de análisis de sentimiento basado en el algoritmo BERT se dividirá en 4 secciones, las cuales son: Exploración del conjunto de datos, preprocesamiento del conjunto de datos, análisis de sentimiento con BERT y por último la evalución. Dichas secciones se describen de manera detallada más adelante.

1. Exploración del conjunto de datos

En esta sección se siguen una serie de pasos para conocer a fondo nuestro conjunto de datos y darnos una mayor idea de con que clase de datos estamos trabajando. Primero se descarga el dataset que contiene reseñas de Google Play Store. Una vez descargado se guarda en datos de entrenamiento y de prueba y se puede comenzar a inspeccionar. Nuestro conjunto de datos contiene la siguiente información:

- **userName**: corresponde al nombre del usuario, el cuál realizó la reseña. Por fines de seguridad y protección personal este no se tomará en cuenta para entrenar el modelo.
- **userImage**: corresponde a la imagen del usuario que escribió la reseña, de la misma manera, esta no será utilizada para el entrenamiento del modelo.
- **content**: corresponde a la reseña que escribió el usuario. Se utilizará para el entrenamiento del modelo.
- **score**: corresponde a la calificación o estrellas que otorgó el usuario. Se utilizará para el entrenamiento del modelo.
- **thumbsUpCount**: son las interacciones que tuvieron los demás usuarios con el comentario del usuario, también es conocido como “likes“. No se utilizará para el entrenamiento del modelo.

- **reviewCreatedVersion**: versión de la aplicación a la que se le realizó la reseña. No se utilizará para el entrenamiento del modelo.
- **at**: fecha en la que se realizó la reseña. No se utilizará para el entrenamiento del modelo.
- **replyContent**: comentario a la reseña dada por los demás usuarios. No se utilizará para el entrenamiento del modelo.
- **repliedAt**: fecha en la que se le comentó al usuario por su reseña. No se utilizará para el entrenamiento del modelo.
- **sortAt**: forma en la que fue ordenado el dataset a la hora de la extracción. No se utilizará para el entrenamiento del modelo.
- **appId**: id de la aplicación que fue reseñada. No se utilizará para el entrenamiento del modelo.

2. Preprocesamiento del conjunto de datos

Los modelos de aprendizaje automático no funcionan con texto sin procesar. Es por tal motivo que es necesario convertir el texto en números. El algoritmo BERT requiere aún más atención y alguno de los requisitos para dicho algoritmo son los siguientes:

- Añadir tokens especiales para separar frases y hacer la clasificación, a continuación se muestran algunos ejemplos:
  - [**SEP**]: marcador de final de frase.
  - [**CLS**]: este token se debe añadir al principio de cada frase, para que BERT sepa que estamos haciendo la tarea de clasificación.
  - [**PAD**]: este token se utiliza para los rellenos extra que puedan existir en el texo.
  - [**UNK**]: BERT entiende los tokens que estaban en el conjunto de entrenamiento. Todo lo demás puede codificarse utilizando este token.
- Pasar secuencias de longitud constante (introducir relleno)
- Crear una matriz de ceros (token de relleno) y unos (token real) llamada máscara de atención.

La librería Transformers proporciona una amplia variedad de modelos Transformer (incluyendo BERT). Por lo tanto podemos realizar tareas de tokenización juntando así los demás tokens especiales. Una vez hecho esto, podemos obtener un conteo de tokens(palabras) por cada reseña como se puede observar en la figura 3.

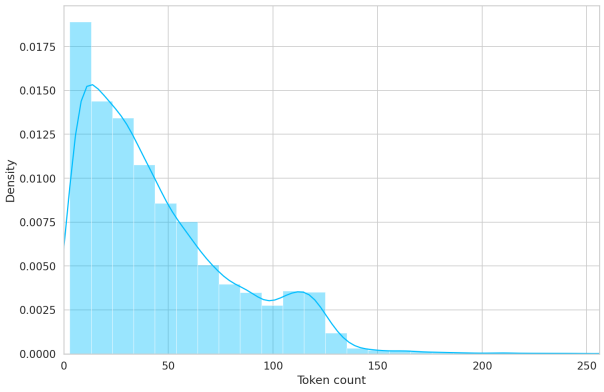

Figura 3: Conteo de palabras (tokens) por reseña

3. Análisis de sentimientos con BERT

En esta sección se realizará el modelo previamente descrito en el partado de **Modelo** incluyendo todas las capas una por una. Después de esto se realiza el reentreanmiento del modelo con las capas añadidas con el optimizador AdamW proveido por Hugging Face.

Los autores de BERT dan algunas recomendaciones para el ajuste fino (fine-tunning), de las cuales algunas serán utilizadas en el entrenamiento de este modelo. Estas recomendaciones son:

- **Tamaño de lote**: 16, 32
- **Tasa de aprendizaje (Adam)**: $5e^{-5}$, $3e^{-5}$, $2e^{-5}$
- **Número de épocas**: 2, 3, 4

Ignoraremos la recomendación del numero de épocas con la finalidad de obtener un mejor score, y para esto la cambiaremos a 10.

4. Evaluación

En esta sección obtendremos nuestros resultados del entrenamiento de nuestro modelo, realizando las siguientes pruebas:

- **Precision**: mide que tan bien se realizó la predicción de un texto.
- **Recall**: la información no recuperada de un texto que debería recuperar.
- **F1 Score**: es la media armónica entre la presición y el recall.

De esta manera obtendremos la matriz de confusión. Una vez obtenidos todos los datos anteriores, se realizará una predicción dandole a nuestro modelo un texto de nuestro conjunto de datos de prueba.

VI Resultados y discusión

Esta sección muestra los resultados del experimento de análisis de sentimientos usando el algoritmo BERT

como base con las capas de abandono (dropout) y la capa completamente conectada (fully-connected).

Después de 10 epocas de entrenamiento, se obtuvieron los siguientes resultados:

- **Tiempo total de ejecución**: 43 minutos, 23 segundos.
- **Pérdida de entrenamiento**: 0.03768671146314381
- **Exactitud**: 0.8932655654383737

En la figura 4 tenemos un gráfico que nos muestra el historial de exacitud tanto de el conjunto de datos de entrenamiento como los de validación en el entrenamiento de nuestro modelo a través de las 10 épocas por las cual fue entrenado. Se puede observar que con los datos de prueba el entrenamiento a través de las épocas van en aumento hasta que llega un punto en el que se estabiliza, es por esto que no se proponen más épocas, ya que si esto sucede puede caer en un sobreajuste y esto no permitiría que nuestro modelo se pueda probar con datos ajenos al conjunto de datos con el que se probó.

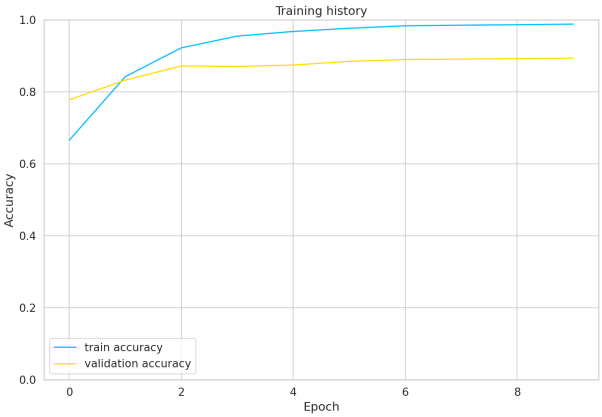

Figura 4: Historial de entrenamiento

En la figura numero 5 se muestran los resultados obtenidos al llevar a cabo las pruebas de presicion, recall, y f1 para las diferentes etiquetas de nuestro modelo.

	precision	recall	f1-score	support
negative	0.89	0.87	0.88	245
neutral	0.83	0.85	0.84	254
positive	0.92	0.93	0.92	289
accuracy			0.88	788
macro avg	0.88	0.88	0.88	788
weighted avg	0.88	0.88	0.88	788

Figura 5: Reporte de resultados

En lo que respecta a la matriz de confusión dada por el modelo, se puede observar que el desempeño del algoritmo es lo suficientemente bueno, ya que el número de predicciones correctas que hace el modelo para cada una de las diferentes etiquetas es por mucho mayor a las malas predicciones o falsos positivos que da. Esto se puede observar de manera visual en la figura 6.

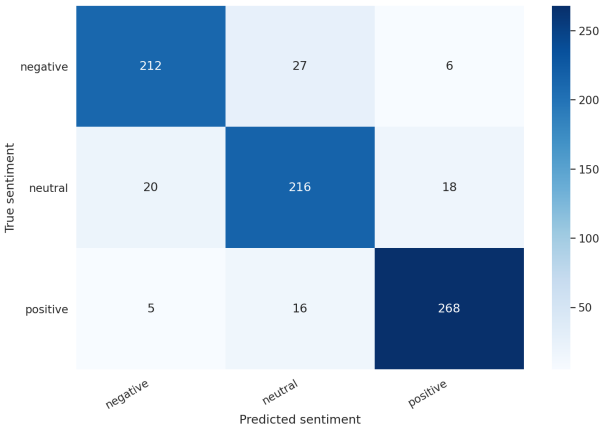

Figura 6: Matriz de confusión

Por último, se llevaron a cabo pruebas introduciendo texto que no conocía nuestro modelo con el conjunto de datos de prueba, del cual nosotros si sabemos el verdadero resultado pero el modelo no. El texto introducido y la etiqueta correspondiente se muestran a continuación:

- **Texto**: "I used to use Habitica, and I must say this is a great step up. I'd like to see more social features, such as sharing tasks - only one person has to perform said task for it to be checked off, but only giving that person the experience and gold. Otherwise, the price for subscription is too steep, thus resulting in a sub-perfect score. I could easily justify $0.99/month or eternal subscription for $15. If that price could be met, as well as fine tuning, this would be easily worth 5 stars."
- **Etiqueta de sentimiento**: neutral.

La figura 7 nos muestra el resultado que se obtuvo al predecir el texto con el.

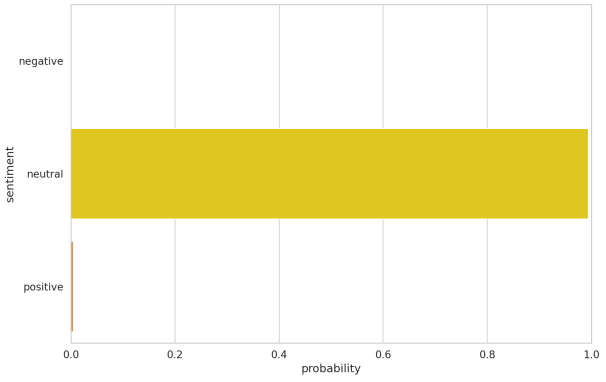

Figura 7: Resultado de predicción de texto no conocido por el modelo

Al tener una exactitud de predicción de poco mas de 89 %, el modelo es capaz de analizar sentimientos por medio de un clasificador basado en el algoritmo BERT en conjunto con las capas de abandono (dropout) y la capa completamente conectada (fully-connected).

VII Trabajo futuro