Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 09/11/2023 | Aceptado: 27/01/2024 | Publicado: 30/03/2024

Identificadores persistentes:
DOI: 10.48168/innosoft.s15.a120
ARK: ark:/42411/s15/a120
PURL: 42411/s15/a120

Clasificación de comentarios de Android usando BERT

Android comment classification using BERT

Susana Rosa Elizabeth Mansilla Ancco [1*], Marcelo Antony Pérez Treviños [2]

[1] Universidad La Salle. Arequipa, Perú. smansillaa@ulasalle.edu.pe
[2] Universidad La Salle. Arequipa, Perú. mperezt@ulasalle.edu.pe

[*] Autor para correspondencia: smansillaa@ulasalle.edu.pe

Resumen

En este proyecto, se centra en el desarrollo de una herramienta de análisis de texto basada en NLP para evaluar comentarios de usuarios de aplicaciones Android, específicamente recopilados de F-Droid. La falta de una solución automatizada para analizar y entender estas opiniones, clasificándolas en tópicos específicos, motiva la investigación. El objetivo es proporcionar a desarrolladores, usuarios y analistas de datos una visión detallada de las preferencias y percepciones de los usuarios. Utilizando conjuntos de datos en inglés entre 2014 y 2017, la propuesta se implementa en Python con la librería Pandas. Se emplea el modelo BERT para la clasificación, con un enfoque específico en la comparación de diferentes modelos, dando como resultado un 97% de exactitud con el uso de BERT. La interfaz gráfica se construye en Visual Studio, permitiendo a los usuarios ingresar comentarios y obtener clasificaciones de tópicos, junto con visualizaciones de nubes de palabras.

Palabras clave: BERT, clasificación de tópicos, clasificación de texto, procesamiento de lenguaje natural.

Abstract

In this project, he focuses on the development of an NLP-based text analysis tool to evaluate user feedback of Android applications, specifically collected from F-Droid. The lack of an automated solution to analyze and understand these opinions, classifying them into specific topics, motivates research. The goal is to provide developers, users, and data analysts with a detailed view of user preferences and perceptions. Using data sets in English between 2014 and 2017, the proposal is implemented in Python with the Pandas library. The BERT model is used for classification, with a specific focus on the comparison of different models, resulting in 97% accuracy with the use of BERT. The graphical interface is built in Visual Studio, allowing users to enter comments and obtain topic rankings, along with word cloud visualizations.

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Keywords: *Topic classification, text classification, natural language processing.*

Introducción

En el contexto del creciente ecosistema de aplicaciones Android, la comprensión de las opiniones de los usuarios se ha vuelto esencial para mejorar la calidad y la experiencia general. Este proyecto se centra en la creación de una herramienta de análisis de texto basada en técnicas de Procesamiento del Lenguaje Natural (NLP) para examinar comentarios de usuarios de aplicaciones Android, específicamente recopilados de la plataforma F-Droid. La motivación subyacente radica en la necesidad de proporcionar a desarrolladores, usuarios y analistas de datos una visión integral de las opiniones expresadas en estos comentarios, clasificándolos en tópicos específicos para identificar patrones y tendencias.

La importancia de este proyecto se destaca en el contexto del constante desarrollo y la evolución del panorama de aplicaciones móviles, donde las opiniones de los usuarios desempeñan un papel crucial en la toma de decisiones para los desarrolladores y las empresas. El análisis de texto a través de técnicas avanzadas de NLP se presenta como una herramienta valiosa para extraer información significativa de grandes conjuntos de datos de comentarios como App, Feature/Functionality, Contents, GUI, Model, Update/Version y Other, permitiendo una comprensión más profunda de las preferencias y percepciones de los usuarios.

El trabajo se centra en abordar preguntas fundamentales relacionadas con las técnicas de clasificación de texto, la identificación de tipos de tópicos discutidos y las herramientas utilizadas en el análisis de texto. Se destaca la comparación de diferentes modelos de clasificación, lo que proporciona una evaluación crítica de sus fortalezas y limitaciones.

En cuanto al entrenamiento del modelo BERT, el trabajo detalla el proceso mediante un pipeline específico, describiendo los datos utilizados en este proceso. Además, se presenta el código de entrenamiento del modelo de manera comprensible, utilizando un pseudocódigo que facilita la comprensión del procedimiento.

Adicionalmente, se realiza una exploración de trabajos relacionados que han abordado problemas similares. Esta revisión literaria no solo brinda inspiración, sino que también establece un marco de referencia esencial para el diseño y desarrollo de la herramienta propuesta. La información recopilada en la revisión literaria se utilizará como base sólida para la propuesta detallada que sigue en el trabajo.

Motivación

A. Problema

El problema central radica en la ausencia de una herramienta automatizada que permita analizar y comprender las opiniones de los usuarios expresadas en los comentarios de las aplicaciones de Android. Este problema se agrava por la necesidad de realizar la clasificación en un tiempo reducido y con una adecuada categorización del contenido.

B. Objetivo

Este trabajo tiene como objetivo desarrollar una herramienta de análisis de texto para aplicaciones de Android a partir de datos recolectados de F-Droid, para que clasifique los comentarios de los usuarios y según los tópicos pueda tener una referencia para las actualizaciones o mantenimiento del aplicativo.

C. ¿En qué dominio del conocimiento está trabajando?

En cuanto al tema de clasificación de texto en comentarios de usuarios de aplicaciones Android, el dominio del conocimiento en el que se está trabajando es el campo de Procesamiento del Lenguaje Natural (NLP), Inteligencia Artificial (IA) y clasificación de textos.

D. ¿Quiénes son los usuarios objetivos?

Los usuarios objetivos abarcan desarrolladores de aplicaciones, usuarios de Android, empresas de desarrollo, investigadores en experiencia del usuario y analistas de datos. Estos actores desempeñan un papel clave en la mejora continua de los productos, contribuyendo así a una experiencia de usuario mejorada en el uso de aplicaciones móviles.

E. ¿Por qué es interesante el tema que proponen?

El análisis de texto aplicado a los comentarios de usuarios en F-Droid proporciona información valiosa sobre la percepción de los usuarios en diversos tópicos. Esta información se vuelve crucial para mejorar tanto la calidad de las aplicaciones como la experiencia del usuario en el ecosistema Android. El sistema desarrollado tiene la capacidad de gestionar y clasificar estos comentarios en áreas específicas, permitiéndonos enfocarnos en temas particulares y abordar de manera prioritaria aquellos que requieran mantenimiento.

F. ¿Cuáles son las preguntas que su proyecto de NLP intenta responder?

P1: ¿Cuáles son las técnicas utilizadas en la clasificación de texto?

P2: ¿Cuáles son los principales conjuntos de datos?

P3: ¿Qué tipos de tópicos se utilizan?

P4: ¿Se han implementado estrategias específicas en el código para mejorar la precisión del modelo BERT en comparación con otros modelos de clasificación?

P5: ¿Cuáles son las palabras de los comentarios más utilizadas en cada tópico?

Trabajos relacionados

El texto aborda la aplicación del Procesamiento del Lenguaje Natural (PLN) para optimizar la revisión de especificaciones de construcción, con un enfoque especial en la identificación de categorías de riesgo contractuales. Se introducen métodos fundamentales de preprocesamiento de texto, como normalización y tokenización, seguidos de una explicación detallada de técnicas de incorporación de texto, incluyendo el destacado modelo BERT [1].

Trata acerca del uso de Ecco que es una biblioteca de código abierto diseñada para potenciar la transparencia y explicabilidad de los modelos de lenguaje basados en la arquitectura Transformer, como BERT. Su principal enfoque es proporcionar herramientas avanzadas que permitan analizar y visualizar aspectos internos de estos modelos. La implementación con Ecco se realiza de manera eficiente a través de una biblioteca de Python que se integra fácilmente con modelos de lenguaje preentrenados, en particular con BERT. Esta integración permite aprovechar tecnologías web para generar visualizaciones interactivas, mejorando así la comprensión y la interpretabilidad de los modelos. Ecco se fundamenta en diversas bibliotecas de código abierto, incluyendo Scikit-Learn, Matplotlib, NumPy, PyTorch y Transformers, lo que garantiza una base sólida y confiable para sus funcionalidades [2].

En este trabajo de investigación, se analiza el aprendizaje automático supervisado. Admite la máquina de vectores y el algoritmo Naïve Bayes y compara su precisión general, precesión y valor de recuperación. El resultado muestra que, en el caso de las reseñas de aerolíneas, la máquina de vectores de soporte dio mejores resultados que el algoritmo Naïve Bayes [3].

Los modelos tradicionales generalmente necesitan obtener buenas características de muestra mediante métodos artificiales y luego clasificarlas con algoritmos clásicos de aprendizaje automático. Por lo tanto, la eficacia del método está restringida en gran medida por la extracción de características. Sin embargo, a diferencia de los modelos tradicionales, el aprendizaje profundo integra la ingeniería de características en el proceso de ajuste del modelo mediante el aprendizaje de un conjunto de transformaciones no lineales que sirven para asignar características directamente a los resultados [4].

Trata acerca del sistema de clasificación de texto corto para descripciones de transacciones bancarias, que consta de tres etapas principales: preprocesamiento, análisis de aprendizaje automático (ML) y clasificación. En la etapa de

preprocesamiento, se recopilan datos de transacciones bancarias, se tokenizan y eliminan palabras sin significado. En el análisis de ML, se extrae conocimiento lingüístico mediante la creación de léxicos basados en categorías de interés, utilizando diversas características como datos léxicos, cantidad de transacción, fecha y n-gramos de palabras y caracteres. La clasificación se realiza mediante un clasificador de Máquinas de Soporte Vectorial (SVM), abordando el desafío de clasificar texto breve. La evaluación del sistema se realiza en conjuntos de datos de descripciones de transacciones bancarias españolas, comparando resultados con enfoques competidores y utilizando métricas como precisión, recall y F-measure [5].

Aborda el desafío de la desigualdad de clases en conjuntos de datos, resaltando que muchos algoritmos funcionan mejor cuando las clases están representadas de manera equitativa. Para superar este problema, implementa la funcionalidad class_weight de sklearn.utils, lo que resulta en mejoras significativas para su conjunto de datos desequilibrado. El código proporcionado muestra la implementación práctica, desde el preprocesamiento de datos hasta el manejo del desequilibrio de clases mediante pesos calculados, el entrenamiento del modelo y las fases de validación y prueba de rendimiento. Demuestra la eficacia de abordar la desigualdad de clases y destaca la rapidez en la que se puede lograr el ciclo completo de desarrollo del modelo [6].

Propuesta

A. Datos

Se eligió utilizar un conjunto de datos en inglés que se centra en comentarios de aplicaciones de Android, seleccionándolo debido a su disponibilidad y a que es el conjunto de datos más extenso identificado, abarcando el periodo entre 2014 al 2017. Estos conjuntos de datos se obtuvieron del repositorio de bases de datos en GitHub y se almacenaron en formato estructurado CSV.

El análisis y procesamiento de los datos se llevarán a cabo en el lenguaje de programación Python, haciendo uso de la librería Pandas para la lectura eficiente de los archivos CSV.

En el proyecto, se emplearán dos conjuntos de datos relacionados con comentarios de aplicaciones de Android obtenidos a través de F-Droid. El primer conjunto, presentado en la Tabla 1, contiene información detallada sobre los usuarios. Por otro lado, el segundo conjunto, presentado en la Tabla 2, abordará la clasificación de los tópicos según el contenido de los comentarios.

Tabla 1. Descripción de atributos de la primera base de datos

Atributos	**Descripción**
Id	Identificador del usuario
Package	Nombre del paquete
Review	Comentario del usuario
Date	Fecha
Star	Clasificación por estrella
Versión_id	Versión

Tabla 2. Descripción de atributos de la segunda base de datos

Atributos	Descripción
Id	Identificador del usuario
Review	Titulo del comentario
Intention	Intención del comentario
Topic	Clasificación del comentario

Con el objetivo de entrenar el modelo propuesto, se generará un nuevo archivo en formato CSV a partir de los dos conjuntos de datos mencionados. En este archivo, presentado en la Tabla 3, se utilizará el comentario del usuario del primer conjunto de datos, y se incorporará el tópico del segundo conjunto de datos asociado a dicho comentario. Este nuevo archivo servirá como base para el desarrollo del modelo propuesto en el proyecto donde se trabajara con 7 tópicos como App, Feature/Functionality, Contents, GUI, Model, Update/Version y Other.

Tabla 3. Descripción de atributos de la tercera base de datos

Atributos	Descripción
Review	Comentario del usuario
Topic	Clasificación del comentario

El tercer dataset se genera en formato CSV, con un total de 90801 registros, y con un tamaño aproximado de 6.04 MB en el disco duro.

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

B. Diseño

Para este proyecto, se emplearon dos bases de datos con el objetivo de mejorar la información disponible. Para lograr esto, se creó una tercera base de datos que contiene comentarios de la primera base, vinculando mediante un identificador compartido con la segunda base para clasificar al tópico. Este procedimiento permitió establecer la correspondencia correcta entre los comentarios y los respectivos tópicos.

Una vez obtenido el tercer conjunto de datos, se procedió con el preprocesamiento, que incluyó la eliminación de duplicados, espacios en blanco, signos, emojis y caracteres no ASCII. Después de esta fase de limpieza, se llevó a cabo un balanceo de datos para garantizar que los siete tópicos tuvieran una cantidad equitativa de muestras. Dado que algunos tópicos presentaban una falta de datos, se calcularon los pesos de tópico inversamente proporcionales a la frecuencia de cada uno, abordando así el desbalance.

Posteriormente, se dividió el conjunto de datos en conjuntos de entrenamiento y prueba. Se empleó el tokenizador BERT para procesar y codificar los comentarios en ambos conjuntos. A continuación, se definió un modelo BERT específico para la clasificación en función de los siete tópicos.

El entrenamiento del modelo se llevó a cabo a lo largo de varias épocas, utilizando una función de pérdida ponderada por los pesos de tópico para contrarrestar el desbalance existente. Tras completar el entrenamiento, se evaluó el modelo en el conjunto de prueba y se calcularon métricas de precisión, recall y F1.

Finalmente, el modelo entrenado y el codificador de etiquetas se guardaron en archivos para su posterior utilización. Estos procesos que se explicó con anterioridad se ve reflejado en la Figura 1.

Los usuarios de la herramienta pueden interactuar con la visualización generada para llevar a cabo un análisis de los tópicos. La interfaz proporciona un cuadro de texto que permite a los usuarios ingresar un comentario, y el sistema entrenado por el modelo responderá mostrando el tópico al que pertenece dicho comentario. Además, la herramienta presenta una Nube de Palabras para cada tópico que incluye App, Feature/Functionality, Contents, GUI, Model, Update/Version y Other.

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

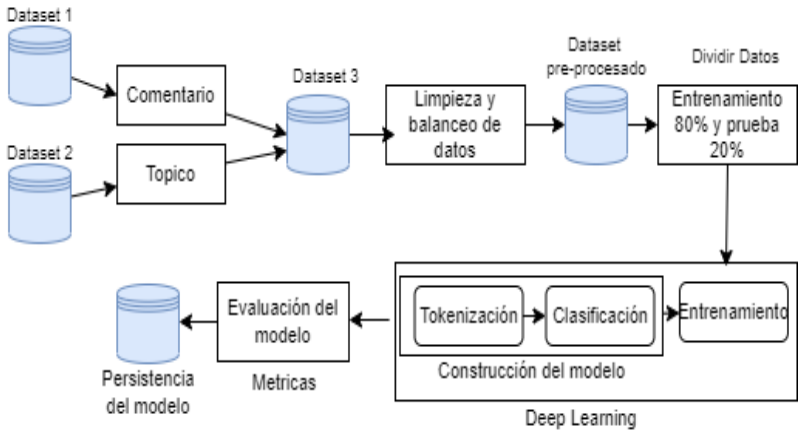

Figura 1. Diagrama para la clasificación de comentarios según tópico con Deep Learning.

C. Pseudocódigo

En la realización de este proyecto, seguimos meticulosamente los pasos delineados en el algoritmo de la Tabla 4, implementado en Python. A continuación, se detallan las fases clave del proceso:

- Obtención del Conjunto de Datos.
- Procedemos a identificar y obtener las etiquetas únicas de los diversos tópicos presentes en el conjunto de datos.
- Inicializamos el codificador de etiquetas para convertir estas en valores numéricos con label_encoder,y se almacena en nueva columna label.
- Se determinaron los pesos de clase, siendo inversamente proporcionales a la frecuencia de cada tópico de los valores almacenados en label.
- Comenzamos a dividir los datos en conjuntos de entrenamiento y prueba.
- Se procedió a la inicialización del tokenizador, configurando el procesamiento en bert base uncased.
- Se tokeniza y codifica los datos de entrenamiento y prueba, se llevó a cabo la conversión de los textos de los comentarios en vectores numéricos.
- Iniciamos la configuración de un modelo de clasificación de secuencias utilizando bert base uncased.
- Configuración del Optimizador en base al clasificador.

- En la función evaluar_modelo se evalúa el modelo en el conjunto de prueba y calcular métricas como la precisión, la precisión por clase, recall y la puntuación F1 para luego retornar estas métricas.
- Se define el número de épocas a evaluar en este caso es 9.
- Se realiza el entrenamiento del modelo 9 veces, y en cada entrenamiento donde se utiliza una función de pérdida ponderada para mejorar el aprendizaje del modelo, se retorna el total de pérdidas.
- Evaluación Final y Métricas, evaluamos el modelo en el conjunto de prueba y obtenemos la precisión del modelo. También imprimimos las métricas obtenidas durante el proceso en la función de evaluación.

Tabla 4. Algoritmo de entrenamiento BERT

Algoritmo 1: Proceso Tópicos Android
Entrada: CD => conjunto de datos label=>Datos de los tópicos en valores numéricos entrenamiento_df =>Datos de Entrenamiento prueba_df =>Datos de Prueba num_labels =>Cantidad de tópicos test_dataloader =>Carga de lotes de datos durante la prueba. train_dataloader => Carga de lotes de datos durante la entrenamiento. **Salida**: Metrica_Calculado 1: df= cargar datos(CD) 2: labels = obtener_etiquetas_unicas (df[topico]) 3:label= label encoder.codificador_etiquetas (df[topico]) 4: calcular_pesos_topico(df['label']) 5: dividir_datos(df,prueba size=02,entrenamiento size=0.8) 6: tokenizador=tokenizador.preenternado(bert base uncased) 7: tokenizar_codificar(entrenamiento_df, tokenizador) 8: tokenizar_codificar(prueba_ df, tokenizador) 9:clasificador= clasificador_preentrenado(bert base uncased) 10: optimizador(clasificador) 11: funcion evaluar_modelo(clasificador, test_dataloader): 12: retorna accuracy, precision, recall, f1

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

```
13: num_epocas = 9
14: para epocas =1 .. num_epocas(train_dataloader)
15:        retorna total_perdidas
16: Imprimir Metrica_Calculado
```

Resultado

A. Entrenamiento del modelo BERT

El entrenamiento del modelo BERT se llevó a cabo en Colab debido a la necesidad de un proceso más eficiente utilizando la GPU. Se obtuvieron los siguientes resultados de efectividad durante el entrenamiento de BERT. El entrenamiento duró 3 horas con 15 minutos, con un total de 90,801 comentarios clasificados.

Los resultados proporcionados por el entrenamiento del modelo BERT se muestran en las métricas de precisión, recall, F1 y para calcular la exactitud.

Se tiene dos modelos adicionales como Máquinas de Vectores de Soporte (SVM) y Naive Bayes(NB) que también sirven para la clasificación de texto, y con el modelo que BERT se quiere medir el nivel de precisión y eficiencia que tiene en comparación con los demás.

Primero, se aplica la fórmula de precisión para cada tópico (1), midiendo la exactitud de las predicciones positivas, esto se muestra en la Tabla 5.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

Tabla 5. Comparación de modelos métrica – precision

	Precision		
	SVM	**NB**	**BERT**
App	0.93	0.60	0.98
Feature/Functionality	0.92	0.73	0.97
Other	0.96	0.79	0.99
Contents	0.87	0.99	0.84
GUI	0.84	0.99	0.86
Model	0.85	0.99	0.84
Update/Version	0.87	0.99	0.84

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Segundo, se aplica la fórmula de recall para cada tópico (2), midiendo la capacidad del modelo para capturar todas las instancias positivas, esto se muestra en la Tabla 6.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

Tabla 6. Comparación de modelos métrica – recall

	Recall		
	SVM	**NB**	**BERT**
App	0.94	0.79	0.98
Feature/Functionality	0.93	0.89	0.97
Other	0.97	0.56	0.96
Contents	0.76	0.03	0.97
GUI	0.67	0.06	0.95
Model	0.79	0.04	0.99
Update/Version	0.88	0.08	0.98

Tercero, se aplica la fórmula de F1 para cada tópico (3), que es la media ponderada de precisión y recall. Esto se muestra en la Tabla 7.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Tabla 7. Comparación de modelos métrica – F1

	F1		
	SVM	**NB**	**BERT**
App	0.93	0.68	0.98
Feature/Functionality	0.92	0.80	0.97
Other	0.97	0.65	0.97
Contents	0.81	0.05	0.90
GUI	0.74	0.11	0.90
Model	0.82	0.08	0.91

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Update/Version	0.88	0.16	0.91

Finalmente, se aplica la fórmula de exactitud (4), que mide la proporción general de predicciones correctas, que se visualiza en la Tabla 8.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Examples} \quad (4)$$

Tabla 8. Comparación de modelos métrica – accuracy

	SVM	**NB**	**BERT**
Accuracy	0.93	0.69	0.97

Como se visualiza en la exactitud, el modelo propuesto utilizando BERT tiene un mejor rendimiento en la clasificación de los tópicos, alcanzando un 97% de instancias correctas.

NV muestra un rendimiento más bajo, especialmente en recall y F1-score. SVM tiene un rendimiento general sólido pero ligeramente inferior a BERT, y BERT muestra un rendimiento superior en todas las métricas y categorías en comparación con SVM y NV. BERT es especialmente fuerte en las categorías con más datos (App, Feature/Functionality, Other), donde supera significativamente a SVM y NV.

Según las comparaciones entre los 3 modelos, BERT ofrece el rendimiento más consistente y superior en todas las métricas evaluadas.

B. Word Cloud

Se utilizó la librería Word Cloud (nube de palabras) en Python para identificar qué palabras son más recurrentes en cada tópico. Esto se visualiza en palabras, cuanto más presente esté una palabra en el texto, más grande aparecerá en la nube de palabras. Esto se presentó desde las Figura 2 hasta la Figura 8.

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Figura 2. Nube de palabras del tópico App.

Figura 3. Nube de palabras del tópico Feature/Functionality.

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Figura 4. Nube de palabras del tópico Contents.

Figura 5. Nube de palabras del tópico GUI.

Figura 6. Nube de palabras del tópico Model.

Revista Innovación y Software
Vol. 5, No. 1, Mes Marzo-Agosto, 2024
ISSN: 2708-0935
Pág. 94-110
https://revistas.ulasalle.edu.pe/innosoft

Figura 7. Nube de palabras del tópico Update/Version.

Figura 8. Nube de palabras del tópico Other.

C. Interfaz gráfica

Este modelo de entrenamiento se utilizó para clasificar comentarios en una página web con respecto a los temas para los cuales fue entrenado, se visualiza la interfaz gráfica para el uso de este modelo en la Figura 9.

Figura 9. Interfaz gráfica de la página web.

Conclusión

Este proyecto ha logrado desarrollar con éxito una herramienta de análisis de texto basada en NLP para evaluar comentarios de usuarios de aplicaciones Android. La implementación del modelo BERT, respaldada por una cuidadosa metodología, ha demostrado un rendimiento superior, alcanzando una precisión del 97% en la clasificación de tópicos. La interfaz gráfica proporciona una experiencia interactiva para los usuarios, permitiéndoles ingresar comentarios y obtener clasificaciones junto con visualizaciones de nubes de palabras. Este proyecto no solo aborda la falta de soluciones automatizadas en la comprensión de opiniones de usuarios, sino que también destaca la importancia de las técnicas avanzadas de NLP en la extracción de información significativa de grandes conjuntos de datos.

Contribución de Autoría

Susana Rosa Elizabeth Mansilla Ancco: Conceptualización, Investigación, Metodología, Software, Validación, Redacción - borrador original. **Marcelo Antony Pérez Treviños:** Conceptualización, Investigación, Metodología, Análisis formal, Recursos, Visualización, Supervisión, Administración de proyectos, Adquisición de fondos, Curación de datos, Escritura, revisión y edición.

Referencias

[1]. S. Moon, S. Chi, and S.-B. Im, “Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from Transformers (Bert),” Automation in Construction, October, 2022. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0926580522003387.

[2]. Alammar, J. "Ecco: An open source library for the explainability of transformer language models", Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing: System demonstrations, pp.249-257,2021.

[3]. Rahat, A. Mohaimin, A. Kahir, and A. Mohammad. "Comparison of Naive Bayes and SVM Algorithm based on sentiment analysis using review dataset", 8th International Conference System Modeling and Advancement in Research Trends (SMART). IEEE, 2019.

[4]. Li, Q., et al., “A survey on text classification: From traditional to deep learning”, ACM Transactions on Intelligent Systems and Technology (TIST), vol. 13, no.2, pp. 1-41,2022.

[5]. G.Mendez, S., et al. “Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus”, IEEE Access, vol. 8, pp. 61642-61655,2020.

[6]. Nikhil, “Bert: Handling class imbalance in text classification,” Medium, December, 2023. [Online]. Available: https://medium.com/@nikviz/bert-handling-class-imbalance-in-language-models-7fe9ccc62cb6.