Revista Innovación y Software
Vol. 4, No. 2, Mes Septiembre-Febrero, 2023
ISSN: 2708-0935
Pág. 36-51
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 18/03/2023 | Aceptado: 28/06/2023 | Publicado: 30/09/2023

Identificadores persistentes:
DOI: 10.48168/innosoft.s12.a98
ARK: ark:/42411/s12/a98
PURL: 42411/s12/a98

Clasificación de categorías de noticias usando BERT

Classification of news categories using BERT

Bradlhy Luis Machado Medina [1], César Alonso Santillana Quirita [2], Sharmelyn Violeta Bautista Luque [3]

[1] Universidad La Salle, Arequipa, Perú. bmachadom@ulasalle.edu.pe
[2] Universidad La Salle, Arequipa, Perú. csantillanaq@ulasalle.edu.pe
[3] Universidad La Salle, Arequipa, Perú. sbautistal@ulasalle.edu.pe

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

Resumen

El presente proyecto consiste en desarrollar un modelo de Procesamiento del Lenguaje Natural para clasificar noticias utilizando un conjunto de datos o DataSets ya evaluados. El objetivo principal es crear un sistema que pueda identificar y asignar automáticamente las noticias a una de las categorías predefinidas: negocios, entretenimiento, política, deportes o tecnología. Esto implica el preprocesamiento de datos, extracción de características, entrenamiento de un modelo de machine learning y posteriormente su evaluación de rendimiento utilizando métricas como" precisión"," recall 2" F1 − score". Esto permitir 'a determinar que tan bien el modelo puede predecir la categoría correcta para una noticia nueva o no etiquetada. Si el rendimiento del modelo es satisfactorio, se puede utilizar para clasificar noticias no etiquetadas en tiempo real. En resumen, se busca proporcionar una solución eficiente y precisa para organizar y etiquetar el contenido informativo de una noticia con ayuda de la Inteligencia Artificial.

Palabras clave: Clasificación de noticias, procesamiento del lenguaje natural, BERT, machine learning, inteligencia artificial.

Abstract

The present project consists of developing a Natural Language Processing model to classify news using a set of data or DataSets already evaluated. The main objective is to create a system that can automatically identify and assign news to one of the predefined categories: business, entertainment, politics, sports or technology. This involves data preprocessing, feature extraction, training a machinelearning model and then evaluating its performance using metrics such as "accuracy", "recall 2" F1 - score". This will allow to determine how well the model can predict the correct category for a new or unlabeled news item. If the performance of the model is satisfactory, it can be used to classify

unlabeled news in real time. In summary, it seeks to provide an efficient and accurate solution for organizing and labeling the informative content of a news item with the help of Artificial Intelligence.

Keywords: *News classification, natural language processing, BERT, machine learning, artificial intelligence.*

Introducción

En la actualidad, con la globalización, era digital y avance tecnológico, hay una enorme cantidad de información disponible en la nube o físicamente. Esto representa un desafío en términos de organización, clasificación, evaluación y etiquetado. La clasificación de noticias es una tarea fundamental para facilitar el acceso a la información relevante y brindar una experiencia más eficiente a los usuarios, en cuanto a sus preferencias o el tema que desee leer. En este contexto, el presente trabajo de investigación se enfoca en el desarrollo de un modelo de Procesamiento del Lenguaje Natural (*NLP*) para la clasificación automática de noticias.

Se planea diseñar y realizar un sistema capaz de identificar y asignar automáticamente las noticias a una de las siguientes categorías establecidas: negocios, entretenimiento, política, deportes o tecnología. Para lograr dicha clasificación, se emplear ‘a un conjunto de datos ya evaluados (*Dataste*) obtenidos de la plataforma *Tagle* que servirán como base para el entrenamiento del modelo de *machine Lear Ning*.

Para llevar a cabo la clasificación automatizada de noticias, se realizarán varias etapas. En primera instancia, se lleva a cabo el procesamiento de los datos para limpiar, normalizar el texto y eliminar información innecesaria. Luego, se utilizarán técnicas avanzadas de *NLP* y representaciones de texto, como el modelo *BERT*, para extraer las características relevantes de cada noticia. Posteriormente, se entrenará un modelo de *machine Learning* que aprenderá a asignar las noticias en sus respectivas categorías. Finalmente, se medir ‘a el rendimiento del modelo mediante el uso de métricas como” precisión”, ”$recall^{\iota}$ ”$F1-score$”, para determinar la capacidad del modelo de predecir la categoría correcta para noticias no etiquetadas.

Motivación

Este trabajo planea desarrollar una herramienta de clasificación de tema de textos, específicamente en las categorías ya definidas anteriormente de las noticias.

- ¿En qué dominio del conocimiento está trabajando?

En cuanto al tema de clasificación de temas de noticias, 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.

- ¿Quiénes son los usuarios objetivo?

Los usuarios objetivo de este proyecto puede ser público en general, pero principalmente se enfocan en especialistas en análisis de contenido informativo, es decir periodistas, editores de noticias o investigadores en medios de comunicación. Estos pueden beneficiarse de un sistema automatizado que les permita clasificar y etiquetar rápidamente las noticias en diferentes categorías ten áticas ya preestablecidas.

- ¿Porque es interesante el tema que proponen?

El tema propuesto es interesante puesto que la cantidad de noticias generadas diariamente en el mundo y cada país es enorme, lo que dificulta su procesamiento manual y clasificación. Un sistema automatizado puede ayudar a gestionar esta gran cantidad de información y ahorrar tiempo y esfuerzo a los profesionales encargados de analizar, organizar y clasificar las noticias.

- ¿Cuáles son las preguntas que su proyecto de NLP intenta responder?
    - Q1: ¿Que categoría de noticia es? • Q2: ¿Como se clasificó la noticia?
    - Q3: ¿La clasificación fue correcta?

Problema

El problema en el que se centra el proyecto planteado es encontrar una solución eficiente y precisa que permita la clasificación automática de temas de noticias, superando las limitaciones del procesamiento manual y proporcionando una herramienta útil para los profesionales involucrados en el análisis y la gestión de noticias, reduciendo así el tiempo, esfuerzo y costo de hacerlo manualmente.

Objetivos

Proporcionar una solución automatizada basada en *NLP* que resuelva el problema de la clasificación de temas de noticias, permitiendo una clasificación eficiente y precisa en tiempo real. Esto beneficiará a profesionales de medios de comunicación, analistas de contenido y cualquier persona involucrada en el procesamiento y análisis de grandes volúmenes de noticias.

Datos

Revista Innovación y Software
Vol. 4, No. 2, Mes Septiembre-Febrero, 2023
ISSN: 2708-0935
Pág. 36-51
https://revistas.ulasalle.edu.pe/innosoft

- ¿Qué datos necesitará?

Para el proyecto se necesitarán dos conjuntos principales de datos o *Dataste*, por un lado, los datos de entrenamiento que serán las noticias con su etiqueta ya clasificada en los temas mencionados, y por otro los del *testing*, que serán solamente noticias sin etiqueta alguna. Estos serán usados tanto para definir y entrenar el modelo, como también para probarlo y ver su funcionamiento.

- ¿Como recolectaran los datos?

Se descargar ‘a desde un repositorio en donde se encuentra almacenado.

- ¿Dónde planea obtenerlos?

En *Tagle,* que es una plataforma donde se alojan y almacenan diversos *Dataste* de entrenamiento y *testing* de clasificación de *NLP*, específicamente el *Dataset* denominado News Category Datase.

- ¿Como planea almacenarlos?

En archivos” csv” que son de texto plano para guardar datos tabulares. Cada línea del archivo representa una fila de datos, y los valores de cada fila están separados por comas. Estos organizaran los datos de manera organizada y estructural, además que tendrá poco peso de almacenamiento por el formato que se está usando, para que posteriormente sean consumidos desde el modelo que realizaremos.

- ¿Como accederá a ellos para utilizarlos en su proyecto?

Mediante la lectura de ficheros, abriendo y organizándolos con la librería” Pandas” (*pd*).

Diseño

Para utilizar la herramienta propuesta y ayudar a los usuarios a realizar las tareas que responden a las preguntas mencionadas en la motivación, se deber ‘a tener en cuenta las siguientes funcionalidades:

- Categoría de noticia (Q1): Se debe permitir a los usuarios ingresar una noticia y utilizar el modelo de procesamiento del lenguaje natural desarrollado para clasificarla en una de las categorías predefinidas.
    - Clasificación de la noticia (Q2): Después de clasificar la noticia en una categoría específica, la herramienta debe mostrar al usuario la etiqueta o categoría asignada.
- Evaluación de la clasificación (Q3): Se debe proporcionar una función de evaluación de la precisión de la clasificación realizada por el modelo. Esto incluye las métricas ya mencionadas, que permitirán al usuario determinar la calidad de la clasificación realizada y evaluar si fue correcta, además que, si tiene buena precisión en los resultados, generaremos confianza en nuestro diseño.

El código, debería ser una implementación eficiente del modelo de procesamiento del lenguaje natural y de las funciones de clasificación. Se usar ‘a *BERT* con la librería *Pythorch* y *transformers* para el procesamiento en una plataforma como Colab, Además también se controlar ‘a el versiona miento con ayuda de dicha herramienta y tendrá comentarios en los códigos, además de definición importante en el archivo si es necesario para una mayor comprensión

Trabajos relacionados

En cuanto a los trabajos relacionados, consultamos diversos sitios o páginas web que realizaban la implementación de modelos similares, como *ScalerTopic* [1], *NewsClasifficationusingBERT* [2] o *AnalyticsV idhya* [3], que sirvieron como guía para elegir y trabajar el tema, además que nos pareció interesante porque es un tema de actualidad y con el constante crecimiento de día a día de información se va haciendo ineficiente hacerlo de forma manual.

Revisión Literaria

En esta sección, presentaremos algunos de los trabajos relacionados que ya han clasificado diferentes noticias de diferentes idiomas utilizando diferentes modelos. Hay muchos trabajos donde la clasificación se centra en patrones lingüísticos, estructura de las oraciones, significado semántico de las oraciones, y se extraen algunas palabras clave para detectar alguna categoría específica.

Las Redes Neuronales Recurrentes y las Redes Neuronales Convolucionales son los enfoques más utilizados para realizar tareas de procesamiento del lenguaje natural en la clasificación de noticias.

Hayes *et al* [2], categorizaron noticias en amplias categorías ten áticas utilizando métodos de NLP con técnicas de coincidencias de patrones, basados en algoritmos de *machine Lear Ning* para identificar palabras y frases clave en un contexto adecuado. Su estudio no realiza un análisis semántico o sintáctico completo de las historias, sino que depende de reconocimiento fragmentario. Se divide en dos fases: sintáctico y confirmación. La sintáctico busca identificar todas las categorías posibles en función de las palabras y frases presentes en la historia, mientras que la confirmación busca evidencia adicional para respaldar una hipótesis o determinar si el lenguaje utilizado puede haber llevado a una hipótesis incorrecta. Cabe destacar que el sistema logró una precisión promedio del 93%.

[1] https://www.scaler.com/topics/machine-learning/bbc-news-classification/
[2] https://www.kaggle.com/code/foolofatook/news-classification-using-bert/notebook
[3] https://www.analyticsvidhya.com/blog/2021/12/text-classification-of-news-articles/

Wu *et al* [4], Realizan un webcrawler para obtener noticias de Internet y se procesa la segmentación de oraciones mediante el algoritmo Jieba. Luego, se marca la clasificación para cada noticia procesada y se ordenan las palabras por número de ocurrencias.

Chy *et al* [1], propusieron un modelo para clasificar noticias en bengalí utilizando el clasificador Naive Bayes para categorizar 34 tipos de noticias. Utilizaron su propio *web crawler* para recopilar un conjunto de datos de varias páginas noticias en línea y aplicaron el enfoque estadístico de *Naive Bayes* para clasificar diferentes noticias. Aplican técnicas de estadístico, stemming y eliminación de palabras para mejorar la precisión de noticias y evalúa el sistema utilizando medidas de precisión y recall. Sin embargo, su grafico de Recall-Precision mostró una precisión inferior al 80% y no proporciono correctamente los nombres de las categorías de noticias.

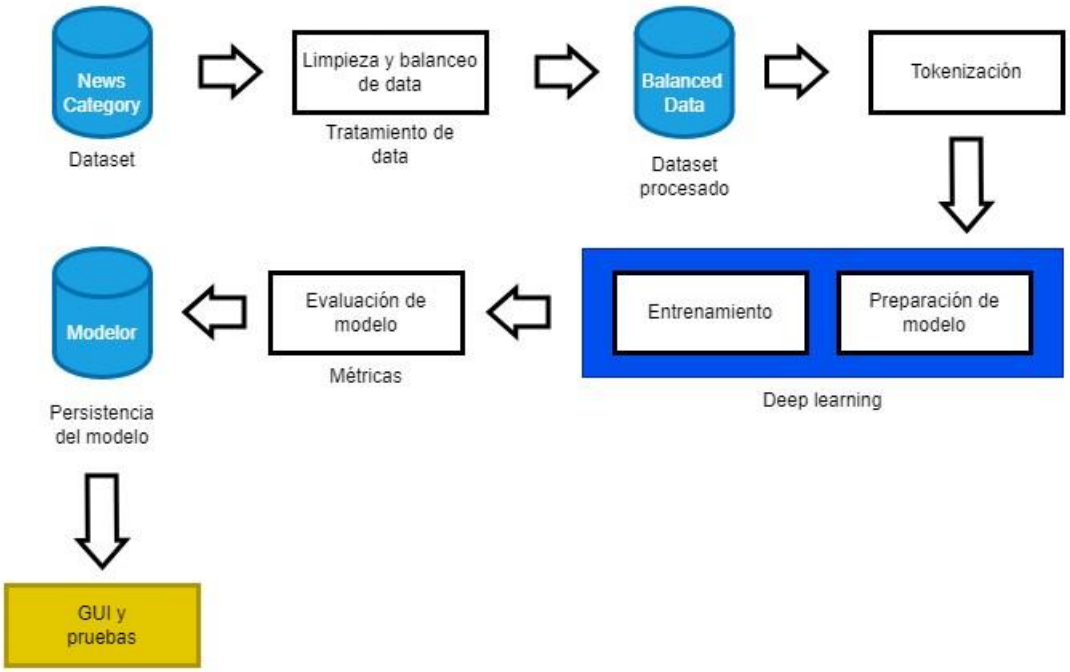
Rahman *et al* [3], implementan un enfoque a nivel de caracteres que categoriza noticias bengalís. Utilizan dos modelos de *deep Lear Ning* CNN y LSTM, respectivamente, para la clasificación, usando la ley de Paretto. La mayor precisión y puntuación F1 se obtienen para el conjunto de las bases de datos utilizando el modelo LSTM del 94% aproximadamente.

Wang *et al* [5], presentan un paper académico que alberga un nuevo conjunto de datos, N24News, que contiene información tanto textual como visual de cada noticia y que tiene 24 categorías. El paper también muestra cómo usar un método multimodal para mejorar la clasificación de noticias.

En base a dicha información implementaremos un modelo usando *BERT* para categorizar diferentes tipos de noticias.

Diseño

Revista Innovación y Software
Vol. 4, No. 2, Mes Septiembre-Febrero, 2023
ISSN: 2708-0935
Pág. 36-51
https://revistas.ulasalle.edu.pe/innosoft



Para la clasificación de noticias seguiremos los pasos mostrados en la Fig. 1.

- *DataSet* News Category Dataset v3: Se obtienen los datos a partir de un *dataset* de noticias clasificado en formato *csv*.
- Tratamiento de datos: En esta fase vamos a limpiar y balancear el conjunto de datos.
- Tokenización: Hace referencia al proceso de dividir un texto en unidades más pequeñas para la mejor comprensión por parte del modelo.
- *Deep Learning*: En esta face prepararemos el modelo a usar" bert-base-uncased" para el posterior entrenamiento de nuestro modelo con *BERT*.
- Evaluación del modelo: En esta etapa se realizar 'a la evaluación del modelo usando las métricas que se estableció.
- Persistencia: En esta face se almacenar 'a el modelo ya entrenado para su posterior consumo.
- GUI y pruebas: Finalmente desplegaremos nuestro modelo mediante una interfaz que permita ingresar *inputs* de noticias y que se nos dé a que categoría pertenece.

Bibliotecas utilizadas:

- *Transformers*: Proporciona métodos para descargar y entrenar fácilmente modelos pre entrenados.

- *Torch*: Es un *framework* para diseñar y entrenar redes neuronales.
- *Pandas*: Pandas es una librería de Python especializada en la manipulación y el análisis de datos. *Sklearn*: Para evaluar el modelo con las métricas establecidas

Resultados:

Se decidió trabajar colaborativamente en Colab para utilizar los recursos de la GPU. Además, utilizamos transformers. Al utilizar Transformers en nuestro clasificador de noticias, podemos aprovechar su potencial para procesar y comprender el lenguaje humano. Los modelos reentrenados disponibles en la biblioteca nos permitieron extraer características relevantes de los textos, capturar la semántica y contextualizar la información, lo que resultó fundamental para la clasificación de las noticias en diferentes categorías o temas. Las categorías y el tamaño del conjunto de datos se muestran a continuación:

- Las categorías de las noticias son las siguientes: *ENTERTAINMENT, POLITICS, STY LE&BEAUTY, TRAVEL, WELLNESS*
- El tamaño del conjunto de datos a usar es de 49070
- La data se encuentra balanceada, donde cada categoría tiene un total de 9814 datos.

Proceso:

1. **Configuración de GPU:** Dado que entrenaremos una red neuronal usamos Google Colab porque proporciona acceso a GPUs gratuitas. Al asociar Colab con **CUDA**, se aprovechó mejor el poder de la GPU para acelerar las operaciones computacionales necesarias durante el entrenamiento del clasificador de noticias. Para verificar la disponibilidad de una GPU se utilizó la biblioteca PyTorch y se configuro el dispositivo de entrenamiento para usar la GPU disponible. Más tarde, en nuestro ciclo de entrenamiento, cargaremos datos en el dispositivo.
2. **Selección de categorías y balanceo de data:** Utilizaremos la biblioteca *panda* para leer un archivo JSON que contiene el conjunto de datos de noticias. Luego, filtramos las categorías deseadas y combinamos las columnas relevantes en un nuevo DataFrame. A continuación, se realizar ‘a el balanceo de datos utilizando *RandomUnderSampler* para igualar el número de muestras en todas las categorías, ¿como se muestra en la Fig.? Finalmente, el *DataFrame* balanceado se guarda en un archivo JSON en Google Drive. De esta manera realizamos el proceso de preparación de datos para el entrenamiento de un clasificador de noticias.

```
//Imprimiendo la cantidad total de categorías seleccionadas print ("Total data to be
used:" ), df.groupby ("category".size().sum())
ENTERTAINMENT9814
POLITICS          9814
STYLE & BEAUTY 9814
TRAVEL            9814
HELLNESS          9814

Total data be used: 49070
```

3. **Tratamiento de datos:** Se emplea *sklearn* para dividir el conjunto de datos de noticias en entrenamiento y prueba. Donde el 20% de los datos se destinará a las pruebas, mientras que el 80% se utilizará para el entrenamiento, lo que es fundamental para evaluar la precisión y el rendimiento del clasificador.
4. **Tokenización:** Se crea un tokenizador *BERT* utilizando el modelo pre-entrenado de *Transformers* de *Hugging Face* para cargar el tokenizador *BERT* pre-entrenado. De esta manera se preparan los datos para el entrenamiento y la evaluación del modelo de clasificación utilizando BERT. Convirtiendo los textos en secuencias numéricas para obtener las etiquetas correspondientes y definir los parámetros necesarios para el modelo de clasificación de noticias
5. **Modelo de clasificación:** En el siguiente seudocódigo se puede ver que se crea un modelo de clasificación de secuencias utilizando el modelo pre-entrenado de *BERT*, además de un optimizador utilizando el algoritmo de optimización *AdamW* y se pasan los parámetros del modelo para ser optimizados. Además, se definen los parámetros para el entrenamiento, como el número de épocas, que fueron diez ´épocas, y la tasa de aprendizaje.

```
model = BertForSequenceClassification.from_pretrained(
'bert-base-uncased',
num_labels=num_classes,
output_attentions=False,
output_hidden_states=False

batch_size = 32 # Tama~no del lote epochs
= 10 # N´umero de ´épocas learning_rate =
2e-5 # Tasa de aprendizaje

optimizer = AdamW(model.parameters(),
```

```
    lr=learning_rate, eps=1e-8)
    )
```

6. **Preparación del modelo:** En la Fig.**??** se prepara los conjuntos de datos y los cargadores de datos para el entrenamiento y la evaluación del modelo. Además, se mueve el modelo a la GPU si está disponible y se verifica la ubicación de los parámetros del modelo.

```
#Crear objetos DataLoader para los conjuntos de entrenamiento y prueba

train_dataset = NewsDataset(train_input_ids, train_attention_masks, train_labels)
test_dataset = NewsDataset(test_input_ids, test_attention_masks, test_labels)

train_dataloader = DataLoader(train_dataset, batch_size=batch_size)
test_dataloader = DataLoader(test_dataset, batch_size=batch_size) )
```

7. **Métricas:** Para calcular diferentes métricas de evaluación del modelo se importaron algunas funciones de la biblioteca *sklearn.metrics* y de *torch.nn.functional*. Las métricas que se usarán serán: precisión, Recall y F1-score. Estas funciones de métricas serán utilizadas después de realizar predicciones con el modelo entrenado y las etiquetas de prueba. De esta manera se proporciona una comprensión más completa del desempeño del clasificador de noticias.

8. **Entrenamiento y prueba:** NO encontramos en el proceso de entrenamiento y evaluación del modelo de clasificación de noticias utilizando el algoritmo *BERT*. En cada época, se itera sobre los lotes de datos de entrenamiento y se calcula la pérdida, las predicciones y las métricas de desempeño: **precisión, F1, recall y precisión**, para el conjunto de entrenamiento. Luego, se realiza una evaluación similar en el conjunto de prueba. Al final de cada ´época, se imprimen y muestran las métricas promedio de entrenamiento y evaluación. Esto permite monitorear el desempeño del modelo a lo largo del entrenamiento y evaluar su capacidad para clasificar noticias en las 5 categorías indicadas.

```
For epoch en rango(epochs)
#Inicializar variables de entrenamiento a 0.0
#para el entrenamiento y evaluación
```

```
Model.train()
For batch en train_dataloader
#get inputs y labels del batch
#Realizar pasos de entrenamiento
#Actualizar métricas de entrenamiento

Calcular promedios de métricas de entrenamiento
Print métricas de entrenamiento

Model.eval()

Print métricas de validación
```

9. **Persistencia del modelo:** En la función guardar modelo se toma como entrada el modelo. Luego, se guarda el *state _dict* del modelo en el archivo especificado utilizando la función *torch.save*. El *state dict* contiene los parámetros entrenados del modelo. Finalmente, se imprime un mensaje para indicar la ubicación donde se ha almacenado el modelo.

```
def guardar_modelo(modelo, nombre_archivo):
# Guardar el state_dict del modelo en un archivo path =
'/content/drive/MyDrive/NLP_NewsClas/' + nombre_archivo
torch.save(modelo.state_dict(), path)
# Imprimir un mensaje
print('Model stored in: ', nombre_archivo)
```

La función *cargar _modelo* se encarga de cargar el estado de un modelo previamente guardado.

```
def cargar_modelo(nombre_archivo):
path = '/content/drive/MyDrive/NLP_NewsClas/' + nombre_archivo # Crear una
configuraci´on de Bert usando los parámetros por defecto config =
BertConfig(num_labels=5)
# Crear una instancia del modelo de clasificación de secuencias
#usando la configuraci´on modelo =
BertForSequenceClassification(config) # Cargar el estado
del modelo desde el archivo
modelo.load_state_dict(torch.load(path)) # Devolver el
modelo cargado return modelo
```

Revista Innovación y Software
Vol. 4, No. 2, Mes Septiembre-Febrero, 2023
ISSN: 2708-0935
Pág. 36-51
https://revistas.ulasalle.edu.pe/innosoft

10. **Prueba del modelo:** Finalmente se prueba el modelo entrenado utilizando una instancia del tokenizador BERT, el modelo previamente cargado y un texto de entrada. El texto se codifica utilizando el tokenizador y se pasa al modelo para realizar la inferencia. Se aplica una función *softmax* a la salida del modelo para obtener las probabilidades de cada clase, así como se muestra en la Fig. **??**. Estas probabilidades se imprimen para mostrar la clasificación del texto de entrada. Además, se muestran las categorías y los códigos asignados a esas categorías en el conjunto de entrenamiento, lo que ayuda a verificar las etiquetas utilizadas durante el entrenamiento.

```
# Crear una instancia del tokenizador de Bert
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

# Definir una entrada de texto para probar el modelo texto =
"cuerpo noticia"

# Codificar el texto usando el tokenizador entrada =
tokenizer(texto, return_tensors='pt') # Obtener la
salida del modelo usando el método __call__ salida =
modelo(**entrada)

# Aplicar una función softmax para obtener las probabilidades probabilidades =
F.softmax(salida.logits, dim=1)

# Mostrar las probabilidades del modelo print(probabilidades)
```

Comparación:

Comparando el desempeño que tuvo BERT con Naive Bayes y el antiguo modelo de Bert en la clasificación de categorías de noticias, se tomaron en cuenta algunos aspectos importantes como lo son:

Preprocesamiento de datos: Se aplicaron técnicas de estadístico utilizando medidas de precisión y recall.

Tamaño del conjunto de datos: Para un gran conjunto de datos, por ejemplo, más de 45k Naive Bayes es menos efectivo. Para un conjunto un conjunto de datos grande, fue mejor entrenar con un modelo más complejo como lo es BERT. No obstante, si el conjunto de datos es relativamente pequeño, Naive Bayes puede ser más efectivo debido a su menor requerimiento computacional.

Revista Innovación y Software
Vol. 4, No. 2, Mes Septiembre-Febrero, 2023
ISSN: 2708-0935
Pág. 36-51
https://revistas.ulasalle.edu.pe/innosoft

En Naive Bayes

- **Representación de características:** Naive Bayes utiliza una representación de características basada en la frecuencia de palabras o n-gramas. BERT captura mejor el contexto y las relaciones semánticas entre las palabras, lo que condujo a un mejor rendimiento en tareas de clasificación de texto.
- **Eficiencia computacional:** Naive Bayes es un modelo más rápido en términos de entrenamiento y predicción en comparación con BERT, que es un modelo más complejo y requiere más recursos computacionales.
- **Evaluación del rendimiento:** En cuanto a las métricas de evaluación adecuadas para la clasificación de texto se usaron: precisión, recall y F1-score.

En la **Tabla 1** se muestra la cantidad de datos, número de categorías y las métricas de precisión para comparar el rendimiento de los modelos predecesores. Además, se agregó una comparación con un modelo de Bert, denominado Bert antiguo que usa la misma Base de Datos. Para el entrenamiento se usó la ley de Paretto.

Métricas	BERT mejorado	Bert antiguo	Naive Bayes	Roberta	VIT
Categor´ıas	5	40	12	24	32
accuracy	94%	70%	85%	91%	92%
F1	94%	no se ubica	75%	87%	94%
Recall	94%	no se ubica	78%	no se ubica	no se ubica

Cuadro 1: Cuadro comparativo

En comparación con el modelo Bert antiguo y Naive Bayes, el modelo BERT mejorado muestra una mejora significativa en todas las métricas de evaluación. El accuracy aumentó del 0.7% al 94%, lo que indica un rendimiento mucho mejor en la clasificación de procesos. Además, el modelo BERT mejorado supera al Naive Bayes en términos de precisión, con un 94% frente al 85%. El F1 score también muestra una mejora considerable, alcanzando un valor de 94% para BERT mejorado y 75% para Naive Bayes.

En cuanto a la comparación con RoBERTa esta vendría a ser la versión optimizada de BERT que fue entrenada con textos mucho más grande y durante más tiempo, lo que le permite capturar aún más información contextual y semántica

de una noticia, además de imágenes. Sin embargo, la calidad de las noticias de entrenamiento es menores cuando se le pasa poco texto, en este caso habría que enviar una foto para mejorar el F1 y el recall. Tal como se muestra en la Tabla 1. EL accuracy y el F1 son menores a los obtenidos por el entrenamiento realizado con Bert.

Se usa VIt, junto con el algoritmo Jieba para tokenizar el texto de noticias. Uso de machine Lear Ning para entrenar a la computadora y registrar las palabras clave para cada categoría de noticias. Esto implica que se utiliza el procesamiento de lenguaje natural y el machine Lear Ning en el proceso de clasificación de noticias.

Cabe destacar que la cantidad de datos y el número de categorías son diferentes en todas las comparaciones.

Es importante tener en cuenta que el rendimiento de los modelos puede variar dependiendo de diversos factores, como el tamaño y calidad de los datos de entrenamiento, la selección de características y los selección utilizados en el entrenamiento. También se destaca que el uso de la ley de Paretto, el balanceo de datos, el uso de GPU con la librería PyTorch y la limpieza de datos contribuyeron a la mejora considerable de las evaluaciones tomadas en el modelo BERT mejorado.

Conclusiones:

En conclusión, la calidad y la preparación adecuada de los conjuntos de datos utilizados para entrenar modelos, como BERT, son factores críticos que influyen en los resultados de la evaluación y las métricas obtenidas. El uso de un dataset balanceado y una limpieza exhaustiva de los datos, centrándose en los campos relevantes para el entrenamiento, se revela como un aspecto crucial para obtener resultados óptimos.

Un dataset balanceado garantiza que el modelo se entrene con una representación equitativa de las distintas clases o categorías presentes en los datos. Esto evita sesgos y permite que el modelo aprenda de manera equilibrada, mejorando así su capacidad para clasificar correctamente las muestras de prueba.

La limpieza de datos también desempeña papel fundamental. Al eliminar ruido, datos irrelevantes o redundantes, y garantizar la integridad de los campos necesarios para el entrenamiento, se mejora la calidad y la coherencia del conjunto de datos. Esto se traduce en un aprendizaje más preciso por parte del modelo, lo que se reflejará en una evaluación más confiable y en métricas más sólidas.

Contribución de Autoría

Bradlhy Luis Machado Medina: Conceptualización, Análisis formal, Investigación, Visualización, Metodología, Software, Validación, Redacción - borrador original, Curación de datos, Escritura, revisión y edición. **César Alonso Santillana Quirita:** Conceptualización, Análisis formal, Investigación, Visualización, Metodología, Software,

Validación, Redacción - borrador original, Curación de datos, Escritura, revisión y edición. **Sharmelyn Violeta Bautista Luque:** Conceptualización, Análisis formal, Investigación, Visualización, Metodología, Software, Validación, Redacción - borrador original, Curación de datos, Escritura, revisión y edición.

Referencias

[1] Abu Nowshed Chy, Md Hanif Seddiqui, and Sowmitra Das. Bangla news classification using naive bayes classifier. In *16th Int'l Conf. Computer and Information Technology*, pages 366–371. IEEE, 2014.

[2] Philip J Hayes, Laura E Knecht, and Monica J Cellio. A news story categorization system. In *Second Conference on Applied Natural Language Processing*, pages 9–17, 2000.

[3] Md Mahbubur Rahman, Rifat Sadik, and Al Amin Biswas. Bangla document classification using character level deep Lear Ning. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6. IEEE, 2020.

[4] Meng-Jin Wu, Tzu-Yuan Fu, Yao-Chung Chang, and Chia-Wei Lee. A study on natural language processing classified news. In *2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, pages 244–247. IEEE, 2020.

[5] Zhen Wang, Xu Shan, Xiangxie Zhang, and Jie Yang. N24news: A new dataset for multimodal news classification, 2022.