# Informe Práctica 02 - Reporte de implementación

**Equipo 5 - Integrantes:**

* Stefany Cantero Cardenas
* Juan Manuel Vera Echeverri
* Sebastian Arango Urrea

## Reporte

### **1. Descripción de los pasos realizados para la implementación de las tareas fundamentales**

Se desarrolló un sistema para el procesamiento de textos, abordando tres tareas clave:

* **Clasificación**: Se utilizó un modelo de aprendizaje automático basado en Naïve Bayes para clasificar los documentos en categorías predefinidas.
* **Resumen**: Se implementó un algoritmo extractivo basado en TF-IDF y técnicas de reducción de texto para generar resúmenes automáticos.
* **Asignación de temas con LDA**: Se aplicó Latent Dirichlet Allocation (LDA) para identificar los temas principales en un conjunto de documentos y agruparlos según su similitud semántica.

A continuación una descripción más detallada de los pasos en cada módulo:

#### Clasificación

El proceso comienza con la carga de los datos desde Hugging Face, específicamente un conjunto de noticias en español. Se realiza un preprocesamiento que incluye la limpieza del texto y la normalización de caracteres, además de hacer una exploración de los datos, específicamente en las categorías.

Observamos que los datos de infraestructura son insignificantes y por eso los eliminamos. Luego, se convierten las categorías de las noticias en índices numéricos para que el modelo pueda procesarlas correctamente:

{'gente': 0, 'innovación': 1, 'macroeconomia': 2, 'alianzas': 3, 'reputacion': 4, 'regulaciones': 5, 'otros': 6, 'sostenibilidad': 7}

Posteriormente, los datos se transforman en un formato compatible con los modelos de aprendizaje profundo de la biblioteca Transformers de Hugging Face.

A continuación, se utiliza un tokenizador preentrenado basado en el modelo DistilBERT para convertir el texto en vectores de entrada. Se configura el modelo para clasificación de texto, ajustando parámetros como la tasa de aprendizaje, el tamaño de lote y la cantidad de épocas de entrenamiento. Se emplea un Trainer para facilitar el proceso de ajuste del modelo con los datos de entrenamiento y validación. Durante el entrenamiento, se monitorea el desempeño utilizando métricas como la precisión y el F1-score.

Finalmente, el modelo entrenado se guarda para su posterior uso. Se evalúa su rendimiento en un conjunto de prueba para verificar su efectividad en la clasificación de noticias. Se imprimen las métricas de evaluación para medir la calidad del modelo, asegurando que el entrenamiento haya producido resultados satisfactorios antes de implementarlo en aplicaciones prácticas.

#### Resumen

Se instalaron las bibliotecas necesarias para el procesamiento de lenguaje natural, el uso del modelo, la evaluación de este y la manipulación de datos en general, entre ellos: datasets, transformers, accelerate, optimum, auto-gptq, rouge\_score

Se cargaron datos de noticias en español colombiano desde un archivo **CSV** alojado en **Kaggle**. Para esto, se utilizó la biblioteca kagglehub. Los datos se dividieron en tres conjuntos:

* Conjunto de entrenamiento, Conjunto de validación, Conjunto de prueba

Esta división es importante para entrenar y evaluar el modelo correctamente.

Se cargó un modelo pre entrenado **T5 pequeño** ("google-t5/t5-small") desde **Hugging Face Transformers**, junto con su tokenizador correspondiente. Se verificó la disponibilidad de **GPU** para ejecutar el modelo de manera más eficiente, aprovechando así los recursos de hardware para acelerar los cálculos.

Se definieron funciones específicas para preparar las noticias, y obtener el resumen del modelo **T5** pre entrenado, y en caso deseado, imprimir información pertinente en pantalla.

Se utilizó la métrica ROUGE para evaluar la calidad de los resúmenes generados. Las métricas utilizadas fueron:

* ROUGE-1 (precisión, recall, F1)
* ROUGE-L (longest common subsequence)

Se calcularon las puntuaciones para estas métricas, y se determinó la media de la recuperación ROUGE-1. Además, se generó un gráfico que muestra la evolución de las puntuaciones ROUGE-1 y ROUGE-L, lo que ayuda a visualizar el rendimiento del modelo a lo largo del tiempo.

#### Temas

Inicialmente, se realizó el preprocesamiento de los datos utilizando las librerías pandas, spacy, nltk y collections. El texto a preprocesar fue una combinación de la categoría, el título y el cuerpo de la noticia, con el objetivo de enriquecer el análisis y mejorar la identificación de temas. Dado que para la asignación de temas es crucial preservar el contexto, se optó por lematizar las palabras en lugar de aplicar stemming, ya que la lematización mantiene el significado original de cada palabra. Además, se llevó a cabo una limpieza de stopwords para eliminar términos sin valor informativo. Luego de esto, se filtraron palabras irrelevantes, eliminando aquellas con una frecuencia menor a 10 en el corpus.

Para la representación del texto, se utilizó Gensim para construir tanto el diccionario como el corpus. En la etapa de selección del número óptimo de temas, se evaluó la coherencia del modelo utilizando la métrica basada en la similitud del coseno (c\_v), probando valores entre 2 y 15. El valor más alto de coherencia se obtuvo cuando se seleccionaron 15 temas, con una puntuación de 46.44%. Una vez definido este número, se procedió con el entrenamiento del modelo LDA con 15 temas por artículo. Finalmente, se validó el modelo sobre datos de prueba, obteniendo una coherencia de 46.47%, lo que indica que la asignación de temas fue apropiada en la mayoría de los casos.

Tanto en entrenamiento como en validación, el modelo mostró una coherencia elevada, evidenciando que los temas fueron interpretados correctamente y se logró una segmentación adecuada del contenido de los artículos.

### **2. Descripción de las decisiones técnicas tomadas para cada módulo y su interfaz**

#### Clasificación

Se decidió por usar un modelo destilado de bert multilingual específicamente “distilbert-multilingual-model” ya que es un modelo más ligero para el entrenamiento y se optó también por un entrenamiento robusto guardando este modelo para que sea utilizado por la interfaz en donde el usuario solo debe poner el título y el contenido de la noticia para observar que tipo de noticia es.

#### Resumen

Se decidió poner el mismo modelo t5 en el `main.ipynb` y refactorizar el código de forma que pueda funcionar dentro de una función, los valores de los parámetros para el resumen como max\_length (256), min\_length (24), length\_penalty (4.0), num\_beams (4) fueron seleccionados en base a prueba y error con las métricas, dando un buen rendimiento y manteniendo equilibrado el desempeño, se hace un pequeño arreglo para la estructura del resumen para retornarlo lo mejor posible.

#### Temas

Se hizo uso del modelo LDA exportado en formato .model (Gensim), la implementación de este se realizó por medio de la función get\_topics() cuya entrada son la categoría, el título y el cuerpo de la noticia. Dentro de esta función se lleva a cabo el respectivo preprocesamiento del texto y seguidamente se obtiene el corpus que el modelo recibe como entrada para realizar la identificación de los temas. La salida de la función son los 15 temas más relevantes de la noticia.

#### Interfaz

Implementamos una interfaz simple en el mismo notebook con widgets, para crear elementos dinámicos con los cuales interactuar dentro del archivo con el código, y algunas funciones para manejar el comportamiento de estos widgets, agregando una pequeña interfaz, más amigable con el usuario promedio, para poder usar dentro de este archivo, esta consta de elementos tales como un botón para seleccionar el archivo csv, el cual tendrá texto y título, y se le procederá sacar por cada noticia en el csv, el resumen, la categoría, y los subtemas si asi es deseado, de manera automática.

### **3. Descripción de los resultados obtenidos**

#### Clasificación

En el entrenamiento, obtuvimos un accuracy de 80% y un f1-score de 80,16%. A continuación se muestra la matriz de confusión que se obtuvo:

Los resultados fueron los esperados y muy buenos llevando a una clasificación acertada de las noticias.

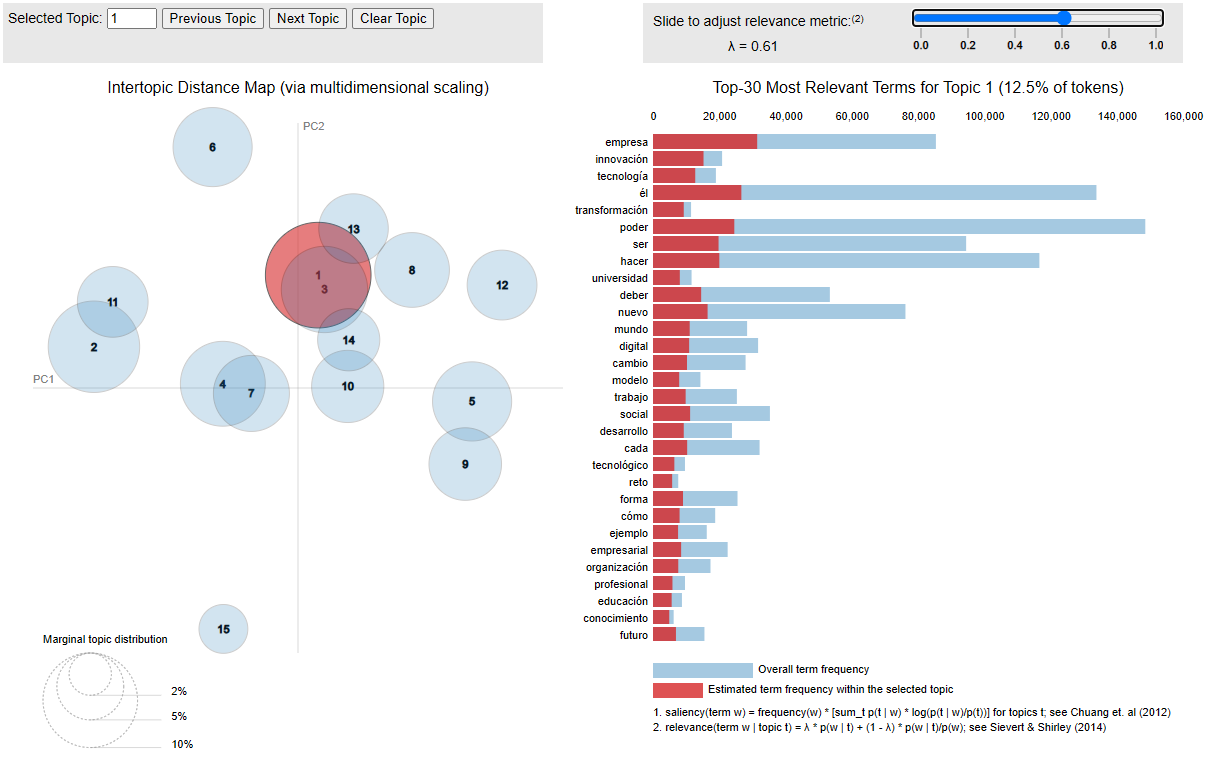
#### Resumen

El análisis de las métricas ROUGE-1 y ROUGE-L revela un rendimiento del modelo de resumen con variabilidad significativa. Si bien el recall de ROUGE-1 muestra una capacidad moderada para identificar palabras clave en los resúmenes de referencia, la precisión es más baja, sugiriendo posibles resúmenes demasiado extensos o imprecisos. Los puntajes de ROUGE-L, que evalúan la coincidencia de subsecuencias más largas, son considerablemente más bajos, indicando que el modelo no captura eficazmente las estructuras y la coherencia del texto original. Esta disparidad entre las métricas indica la necesidad de ajustes en el modelo para mejorar la precisión y la coherencia de los resultados generados.

La inconsistencia observada en las métricas a través de diferentes entradas sugiere una falta de generalización del modelo. El modelo parece funcionar adecuadamente en algunos casos, pero no logra un rendimiento consistente. Las bajas puntuaciones en ROUGE-L apuntan a debilidades en la capacidad del modelo para mantener la cohesión y la estructura del texto resumido. Para mejorar el modelo, se recomienda ajustar los hiper parámetros, explorar arquitecturas alternativas, y aumentar la cantidad y la diversidad de los datos de entrenamiento. Adicionalmente, la evaluación con otras métricas de calidad textual podría ofrecer una perspectiva más completa del rendimiento del modelo.

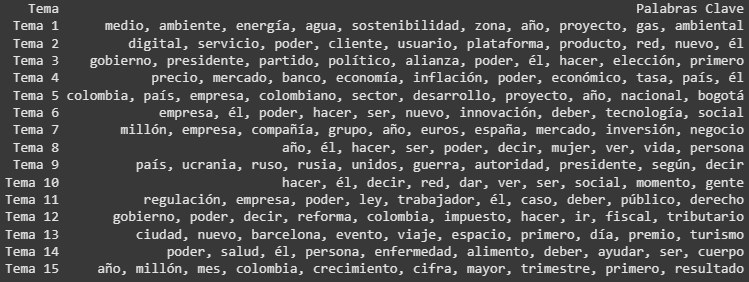
#### Temas

En entrenamiento el modelo obtuvo una coherencia de 46.44% y en validación este valor fue de 46.47%. Lo cual indica que el modelo es capaz de identificar los temas apropiadamente para la mayoría de los casos sin caer en una mala interpretación ni el sobreajuste. Los resultados se pueden visualizar en un dashboard generado por la librería PyLDAvis en la cual se tienen dos divisiones: gráfico de distancias con burbujas en el cual los temas más cercanos son aquellos que tienen más palabras en común y las burbujas más grandes son las de mayor predominancia en el corpus, y una lista de palabras clave que se puede ajustar para ver las palabras con mayor o menor frecuencia usando el deslizador.



*Figura 4. Visualización de resultados modelo LDA.*

Las palabras representativas de cada tema son:



*Figura 5. Tabla de temas y palabras clave.*