

PROYECTO FINAL

MARTA GORRAIZ
IRENE GONZÁLEZ
IKER MARCELO

ÍNDICE

1. INTRODUCCIÓN	2
2. ESTADO DEL ARTE.....	2
3. DATASET Y SCRAPING.....	4
3.1 SCRAPING.....	4
4. ESTRUCTURA DEL PROYECTO.....	6
4.1 EXTRACTIVO	6
4.2 ABSTRACTIVO.....	8
5. COMPARATIVA	10
6. APORTACIONES	24
7. REPOSITORIO	25
8. BIBLIOGRAFIA.....	26

1. INTRODUCCIÓN

El objetivo principal de este proyecto es crear un sistema que sea capaz de generar resúmenes automáticos de textos en español, combinando dos enfoques que se complementan entre sí: por un lado, el resumen extractivo, que consiste en seleccionar las frases más importantes del texto original; y por otro, el resumen abstractivo, que genera frases nuevas usando modelos de lenguaje modernos, como los basados en Transformers.

Para entrenar y probar estos modelos, se ha trabajado con el dataset MLSUM, un conjunto de artículos de noticias en español que ya vienen con resúmenes hechos por humanos. Este dataset es muy útil porque cada entrada incluye el texto completo, el resumen, y también otros datos como el título, la fecha y la fuente del artículo.

Además de usar este dataset, se ha añadido un proceso de scraping sobre artículos científicos de Wikipedia y otro sobre periódicos en línea. Esto se hizo para ampliar el corpus con contenido más actual y variado, y así poder comprobar el comportamiento de los modelos con otros textos no provenientes de dataset preparados para ello.

Durante el proyecto se han probado tanto métodos clásicos de resumen extractivo, como TF-IDF y TextRank, como también modelos más avanzados, como T5-small, que es un modelo encoder-decoder capaz de generar resúmenes más naturales, fluidos y coherentes.

2. ESTADO DEL ARTE

La generación automática de resúmenes es una tarea fundamental dentro del procesamiento del lenguaje natural, tiene como objetivo condensar la información más relevante de un documento de forma coherente y comprensible. Existen dos enfoques en esta tarea: el extractivo, que selecciona fragmentos literales del texto original, y el abstractivo, que genera nuevas frases a partir del contenido, lo que implica una mayor comprensión semántica.

A lo largo de los últimos años, el desarrollo de técnicas de resumen automático ha avanzado de manera significativa, especialmente con la aparición de modelos basados en Transformers [1]. Sin embargo, gran parte de estos avances se han centrado en el idioma inglés, dejando al español en una posición rezagada debido a la menor disponibilidad de datos y recursos lingüísticos. Esto ha comenzado a cambiar con la aparición de datasets multilingües como MLSUM [2] y la adaptación de modelos como T5 [3], BART [4] o mBART en español [5].

A continuación, se revisan las principales técnicas extractivas y abstractivas, así como los modelos y datasets más relevantes. Además, se presenta una breve comparativa entre enfoques y se contextualiza la solución propuesta en este proyecto.

Por un lado, los modelos de resumen extractivo seleccionan fragmentos, ya sean palabras u oraciones, del texto original que se consideran más relevantes, concatenándolos para formar un resumen. Tradicionalmente, esto se ha logrado mediante heurísticas sencillas como la frecuencia de palabras o la posición de las frases en el documento [6]. En español, estos enfoques son aplicables sin cambios drásticos, requiriendo solo recursos como listas de stop-words en español. Los algoritmos basados en grafos, como TextRank y LexRank, son otra técnica destacada basada en PageRank [7]. En estas técnicas, se construye un grafo donde cada oración es un nodo y las aristas ponderan la similitud entre oraciones, para identificar las oraciones centrales se aplican algoritmos de ranking. Estos métodos funcionan independientemente del idioma, obteniendo buenos resultados con ajustes mínimos, como eliminar tildes o usar stop-words en español.

Las soluciones extractivas tienden a conservar la redacción y detalle del texto original, evitando introducir información no existente y asegurando fidelidad al documento. Además, suelen ser eficientes computacionalmente y sencillas de implementar [8]. Sin embargo, el resultado puede ser menos fluido, ya que las oraciones seleccionadas pueden no conectar perfectamente. También es común encontrar que los resúmenes extractivos son más largos de lo esperado o incluyen redundancias al no parafrasear o combinar la información.

Por otro lado, en los métodos abstractivos el sistema genera oraciones nuevas en las que condensa el contenido del texto, pudiendo parafrasear y reorganizar la información. Este tipo de métodos implican un proceso más complejo puesto que requiere comprender el texto para luego sintetizarlo usando lenguaje natural correcto. Tradicionalmente, los modelos abstractivos se basan en arquitecturas de seq2seq entrenadas con pares documento-resumen. Sin embargo, en años recientes los transformers han revolucionado esta tarea [1]. Entre los modelos más destacados se encuentran T5 de Google [3] y BART de Facebook [4]. El modelo de Google es una familia de modelos encoder-decoder pre-entrenados en múltiples tareas de lenguaje (traducción, resumen, preguntas, etc.) bajo un mismo formato de entrada-salida de texto. Por su parte, BART es un transformer autoencoder pre-entrenado mediante denoising, corrompiendo texto y obligando al modelo a reconstruirlo. Muchos de estos modelos inicialmente se desarrollaron para inglés, pero ya existen variantes multilingües o adaptaciones al castellano. Por ejemplo, Facebook desarrolló mBART-25 y posteriormente mBART-50, versiones de BART entrenadas en datos de decenas de idiomas para permitir su fine-tuning multilingüe [5].

En general, los resúmenes abstractivos suelen ser más compactos y fluidos, pudiendo reestructurar la información de manera más natural que un extractivo. Sin embargo, presentan desafíos: requieren modelos más complejos y gran cantidad de datos anotados para entrenamiento. Además, pueden cometer errores de generación, como las alucinaciones, especialmente si el modelo no ha sido bien entrenado o la entrada se sale del dominio conocido [9].

En este proyecto, se combina un modelo ligero (t5-small) con fine-tuning en español, con datos obtenidos de varias fuentes, acercando la investigación a una aplicación real.

3. DATASET Y SCRAPING

Para este proyecto se ha utilizado como dataset principal MLSUM, disponible en la plataforma Hugging Face. Este conjunto incluye miles de artículos de prensa en varios idiomas, entre ellos el español. Cada artículo viene con el texto completo y un resumen hecho por humanos, lo que lo hace muy útil para entrenar y evaluar modelos de resumen automático, tanto extractivos como abstractivos.

Se eligió MLSUM principalmente por:

- Disponibilidad en español.
- La calidad de los resúmenes, que sirven como referencia real.
- Formato ya preparado para tareas de resumen.

Cada entrada en el dataset tiene:

- text: el cuerpo completo del artículo.
- summary: el resumen hecho por humanos.
- title: el título del artículo.
- date y url: información adicional como fecha y fuente.

3.1 SCRAPING

Además del dataset principal, se implementó un pequeño proceso de scraping para obtener contenido actualizado en español directamente desde la sección nacional del diario “La Razón”. El script se conecta a la página principal de la sección “España” del diario y descarga el contenido HTML. A través de la biblioteca BeautifulSoup, se identifican las secciones de la portada donde se encuentran incrustados los artículos, y se extraen sus enlaces internos. Para evitar duplicados y garantizar eficiencia, se filtran los enlaces únicos y se limita la selección a los primeros artículos disponibles.

Una vez obtenidos los enlaces, el script accede a cada uno de ellos de forma individual. Para cada artículo, recupera el título principal y el subtítulo, y los concatena en una única cadena que representa el resumen del texto. Luego, analiza el contenido principal del artículo y extrae todos los párrafos del cuerpo del texto y los unifica en una sola variable, representando el texto completo.

Cada entrada, compuesta por el resumen (titular + subtítulo) y el texto principal, se almacena en una estructura de datos. Al finalizar la recolección, toda la información se vuelca en un archivo en formato CSV con codificación UTF-8. Este archivo contiene dos columnas, el resumen y el contenido de los artículos. Este proceso permite ampliar el dataset MLSUM con noticias actuales.

Además, como no queríamos que el dataset se limitase a resúmenes de noticias, realizamos un scraping adicional de Wikipedia. El objetivo era complementar el conjunto de datos original con información real y reciente sobre temas científicos y tecnológicos, y aplicar un modelo generativo para crear resúmenes automáticos.

El script accede automáticamente a diez artículos de Wikipedia en español relacionados con temas como “Inteligencia artificial”, “Red neuronal artificial”, “Entropía”, entre otros. Para ello, se define previamente una lista con los nombres de los artículos a consultar. A través de la librería wikipedia, el script se conecta a la versión en español de Wikipedia, busca cada tema y descarga el contenido correspondiente. Para evitar exceder el límite de entrada del modelo generativo, solo se toman los primeros 4000 caracteres de cada artículo.

```
import wikipedia
wikipedia.set_lang("es")

TEMAS = [ "Inteligencia artificial", "Red neuronal artificial", "Algoritmo
genético", "Teoría cuántica de campos", "Entropía", "Teoría del
caos", "Célula madre", "Fotosíntesis", "Relatividad
general", "Nanotecnología"]

LIMITE_TEXTO = 4000

for tema in TEMAS:
    print(f"\nProcesando tema: {tema}")
    try:
        texto = wikipedia.page(tema).content[:LIMITE_TEXTO]

    except Exception as e:
        print(f"Error con {tema}: {e}")
}
```

Una vez descargado el texto, se realiza una limpieza básica para eliminar elementos innecesarios como encabezados, referencias, y saltos de línea sobrantes. Esta limpieza se hace mediante expresiones regulares, a través de una función específica.

Con el texto ya limpio, se envía un prompt al modelo de Gemini, que genera un resumen breve compuesto por dos frases. Este modelo permite sintetizar la información principal del artículo de forma automática, sin intervención humana. Cada resultado, es decir, el texto limpio junto a su resumen, se guarda en un archivo .tsv separado por tabulaciones.

```
def resumir(texto):
    prompt = f"Resume el siguiente texto en exactamente 2 frases claras y
concisas:\n\n{texto}"
    try:
        respuesta = modelo.generate_content(prompt)
        return respuesta.text.strip()
    except Exception as e:
        return f"[ERROR] {e}"
```

Como resultado final es un conjunto de artículos reales con sus resúmenes generados por inteligencia artificial. Este archivo sirve para complementar el dataset ya seleccionado.

4. ESTRUCTURA DEL PROYECTO

En esta parte se explican los dos enfoques principales que se han usado en el proyecto: el extractivo y el abstractivo. Cada uno sigue una estrategia diferente para hacer resúmenes, y gracias a ellos se ha podido probar tanto métodos más clásicos como modelos más avanzados.

A continuación se cuenta cómo se ha implementado cada uno, qué herramientas y modelos se han utilizado, y qué decisiones técnicas se han tomado en cada caso.

4.1 EXTRACTIVO

A continuación, se explica cada uno de los enfoques extractivos.

Lead-n

Esta técnica consiste en seleccionar las “n” primeras frases del texto original como resumen. Esta estrategia parte de la observación de que, en muchos géneros textuales, especialmente en el periodismo y en géneros informativos, las ideas principales suelen aparecer al comienzo del documento. Por tanto, extraer las primeras frases proporciona un resumen razonablemente representativo sin necesidad de cálculos lingüísticos o estadísticos complejos.

La implementación es directa, primero se segmenta el texto en frases utilizando el tokenizador de oraciones de NLTK con soporte para español, luego se seleccionan las primeras n frases y se concatenan para formar el resumen final. En caso de que el texto tenga menos de n frases, se devuelve el texto completo.

Esta técnica destaca por su eficiencia computacional y su simplicidad de uso, ya que no requiere modelos preentrenados, vectorización ni análisis semántico. Si bien no captura relaciones de importancia contextual ni semántica entre las frases, su rendimiento es sorprendentemente competitivo en corpus periodísticos, donde la estructura narrativa está diseñada precisamente para presentar la información más relevante al principio.

La principal limitación del enfoque Lead-n es su falta de adaptabilidad a textos no periodísticos. Al depender únicamente del orden de aparición de las frases, puede fallar en géneros donde la información clave no se encuentra al inicio, como ensayos, informes técnicos o narrativa académica. En esos contextos, el resumen generado puede ser superficial o irrelevante.

Luhn Algorithm

Esta implementación es una versión moderna del algoritmo de resumen propuesto por Hans Peter Luhn en 1958. Este enfoque se basa en la hipótesis de que las frases más relevantes de un texto contienen una alta concentración de palabras clave, (frecuentes pero no comunes). El proceso comienza dividiendo el texto en oraciones y tokenizando las palabras, eliminando las stopwords. Luego, se calcula la frecuencia de las palabras no triviales mediante un vectorizador CountVectorizer, y se identifican las más importantes dentro del documento.

A continuación, el algoritmo analiza cada frase buscando bloques de palabras clave consecutivas, cuanto más largas y densas son estas secuencias, mayor puntuación recibe la frase. La puntuación se calcula con una fórmula heurística que premia la densidad semántica: $(\text{longitud del bloque})^2 / (\text{longitud del bloque} + 1)$. Finalmente, se seleccionan las frases con mayor puntuación para formar el resumen.

Esta técnica combina eficiencia con una cierta capacidad para capturar la estructura del contenido. Sin embargo, su principal limitación es que no considera el significado contextual de las palabras ni relaciones semánticas entre frases, por lo que puede fallar en textos donde la relevancia no se correlaciona directamente con la densidad léxica.

Frequent Words

Esta implementación se basa en una técnica extractiva sencilla y efectiva, fundamentada en el principio de que las palabras más frecuentes dentro de un texto suelen ser indicativas de sus temas principales. Al igual que en el enfoque anterior, el procedimiento comienza segmentando el texto en frases y tokenizando las palabras, tras lo cual se eliminan las stopwords y los tokens no alfabéticos. Con las palabras restantes, se construye un diccionario de frecuencias mediante la clase Counter de Python.

Cada frase se puntúa sumando las frecuencias de las palabras que contiene, asumiendo que aquellas con más repeticiones están más relacionadas con el contenido central del texto. Las frases con mayor puntuación se consideran las más representativas y se seleccionan para formar el resumen.

Aunque este enfoque comparte con el método de Luhn el uso de la frecuencia léxica como indicador de importancia, se diferencia en que no tiene en cuenta la agrupación ni la densidad local de las palabras clave. Es decir, no importa si las palabras relevantes están dispersas o juntas dentro de la frase. Esto lo convierte en un método más simple pero también más propenso a seleccionar frases largas o fuera de contexto si contienen muchas palabras frecuentes.

TextRank

Este enfoque implementa un resumen extractivo basado en el algoritmo TextRank, una técnica inspirada en el algoritmo PageRank desarrollado por Google para la clasificación de páginas web. En el contexto del procesamiento de lenguaje natural, TextRank construye un grafo de frases, donde cada nodo representa una oración y los enlaces entre ellas reflejan su similitud léxica. A través de un proceso iterativo, se calcula la importancia relativa de cada frase dentro del documento en función de su conectividad con otras oraciones relevantes.

En esta implementación se utiliza la librería sumy, que simplifica el procesamiento. El texto de entrada se convierte en un objeto de tipo documento (PlaintextParser) y se tokeniza en español. A continuación, el algoritmo TextRank se aplica con soporte para stemming y eliminación de stopwords mediante las herramientas internas de sumy.

El resultado es una selección de las n frases más centrales según la estructura semántica del texto, concatenadas como resumen final. Esta técnica destaca por su robustez y

adaptabilidad a diferentes dominios, ya que no depende de un corpus de entrenamiento ni de la posición de las frases, sino de las relaciones de similitud entre ellas.

Como toda técnica basada en grafos, su principal limitación radica en la calidad de la medida de similitud utilizada, si el texto es muy corto, muy ruidoso o con frases poco conectadas entre sí, el grafo puede resultar disperso y las puntuaciones poco informativas.

TextRank con matriz de similitud TF-IDF

Esta implementación es una versión personalizada del algoritmo TextRank que utiliza similitud por coseno entre vectores TF-IDF para construir un grafo de frases. A diferencia de la versión estándar basada en coincidencia de palabras, esta variante emplea una representación vectorial más precisa y semánticamente informativa para evaluar la similitud entre oraciones.

El proceso comienza segmentando el texto en frases. Cada frase se transforma en un vector TF-IDF mediante `TfidfVectorizer`, lo que permite capturar la importancia relativa de cada término dentro del contexto del documento. A partir de estos vectores, se construye una matriz de similitud por coseno, la cual representa el grado de relación entre cada par de frases.

Esta matriz sirve como base para construir un grafo no dirigido, donde los nodos representan frases y los pesos de las aristas corresponden a su similitud. El algoritmo de PageRank, implementado mediante `networkx.pagerank`, se aplica para determinar la importancia relativa de cada nodo dentro del grafo. Finalmente, se seleccionan las frases con mayor puntuación y se ordenan por su posición original para formar el resumen.

Esta versión de TextRank ofrece una mejor sensibilidad semántica que las implementaciones basadas únicamente en coincidencia de palabras, especialmente útil en textos más largos o con vocabulario más variado. Sin embargo, su rendimiento depende directamente de la calidad de la vectorización y puede verse afectado en textos muy breves o poco estructurados.

4.2 ABSTRACTIVO

El modelo empleado para el resumen abstractivo es T5-small, una versión reducida de la familia T5, creada por Google Research. T5 propone un enfoque unificado para todas las tareas de procesamiento de lenguaje natural, formulándolas como problemas seq2seq.

En este proyecto se emplea la variante t5-small, una versión reducida del modelo original, con aproximadamente 60 millones de parámetros. Escogimos esta versión ya que es adecuada para entornos con recursos computacionales limitados y permite realizar un proceso completo de fine-tuning sobre un conjunto de datos de tamaño realista como MLSum sin comprometer la viabilidad técnica.

La arquitectura de T5-small se basa en encoder-decoder. El codificador recibe el texto completo de entrada, en este caso, el cuerpo de una noticia, y genera representaciones latentes. Estas representaciones son utilizadas por el decodificador, que produce el resumen de manera autoregresiva, generando un token a la vez en función del contexto anterior y de la información extraída por el encoder.

La siguiente tabla resume las especificaciones arquitectónicas principales del modelo t5-small.

Componente	Especificación
Número de capas	6 encoder y 6 decoder
Dimensión de embeddings	512
Submódulos de atención	8
Dimensión feed-forward	2048
Vocabulario	32000 tokens
Total de parámetros	~60 millones

En el contexto de esta práctica, el modelo fue ajustado para realizar resumen abstractivo sin modificar su arquitectura interna. Aunque el modelo original utiliza una plantilla textual explícita en las tareas, en este caso no fue necesario incluirla, ya que el conjunto de datos ya está alineado semánticamente con la tarea de resumen. Cada ejemplo consta de un texto completo y su correspondiente resumen de referencia, lo que permite al modelo aprender directamente la tarea sin depender de indicaciones explícitas en la entrada.

Aunque es cierto que modelos de mayor tamaño como t5-base o mBART podrían ofrecer una mayor capacidad de generalización y generación de lenguaje más natural, el uso de t5-small permite realizar experimentos controlados, reproducibles y adaptados a entornos computacionales académicos. Además, al ser un modelo encoder-decoder puro, resulta más adecuado para tareas generativas que modelos tipo decoder-only como GPT.

Durante la fase de preprocesamiento, se tokenizaron tanto los textos como los resúmenes utilizando el tokenizer correspondiente al modelo T5. Se definieron longitudes máximas de entrada y salida 512 tokens para el texto y 128 para el resumen en la primera versión, y para la segunda versión se duplicó a 1024 y 256. También se activaron las opciones de truncamiento y padding para asegurar que los datos se ajustaran al tamaño esperado por la arquitectura del modelo.

Una vez procesado el dataset, se configuró el entrenamiento mediante la clase Seq2SeqTrainer de Hugging Face, especificando los hiperparámetros clave:

Parámetro	Versión 1	Versión 2
Modelo base	t5-small	t5-small
Longitud máxima de entrada	512 tokens	1024 tokens
Longitud máxima de resumen	128 tokens	256 tokens
Épocas de entrenamiento	3	5
Tamaño de batch por dispositivo	4	4
gradient_accumulation_steps	8	8
Learning rate	5e-5	3e-5
Guardado de checkpoints	Máximo 2	Máximo 4
Métrica para seleccionar mejor modelo	Ninguna	eval_loss
load_best_model_at_end	No	Sí

5. COMPARATIVA

Para analizar el comportamiento de los modelos de resumen utilizados, se realizó una comparación entre modelos abstractivos desarrollados y modelos extractivos ya existentes. Se evaluaron ocho enfoques distintos:

Modelos abstractivos:

- Modelo 1: T5-small con una configuración básica.
- Modelo 2: T5-small con configuración mejorada.
- Modelo 3: mt5_multilingual_XLSum, un modelo preentrenado para resumen multilingüe, sin ajuste adicional.

Modelos extractivos:

- Lead-3: selecciona automáticamente las tres primeras oraciones del texto.
- Luhn: se basa en la frecuencia de palabras significativas.
- TF-IDF: pondera términos poco frecuentes pero relevantes.
- TextRank: modelo basado en grafos inspirado en PageRank.
- TextRank + TF-IDF: combinación de grafos y ponderación estadística.

Cada modelo procesó el mismo texto fuente, el cual relataba el accidente del buque Cuauhtémoc en el puente de Brooklyn, Nueva York. Luego se evaluaron los resúmenes generados mediante métricas automáticas y una valoración manual individual de cada integrante del grupo, para finalmente unificar las evaluaciones.

Para evaluar el rendimiento de los modelos, se usaron tres métricas automáticas muy comunes en tareas de resumen:

- ROUGE: mide cuántas coincidencias hay entre el resumen generado y el de referencia.
 - ROUGE-1 se centra en las palabras individuales (unigramas).
 - ROUGE-2 analiza pares de palabras seguidas (bigramas), y da una idea de si el resumen tiene buena cohesión.
 - ROUGE-L mide la secuencia más larga de palabras en común, lo cual ayuda a ver si la estructura general del texto se mantiene.
- BERTScore: a diferencia de ROUGE, no se basa en coincidencias exactas, sino en la similitud del significado. Compara el resumen generado con el texto original usando vectores de BERT, y calcula precisión, recall y F1.
- Tasa de compresión: mide cuánto se ha reducido el texto original al hacer el resumen. Una tasa más alta indica que el resumen es más corto, pero eso no significa que sea mejor si pierde información clave.

En cuanto al rendimiento de los modelos abstractivos, el Modelo 1 fue el que peores resultados obtuvo. El resumen que generó fue muy corto debido a que los resúmenes de los datasets son muy cortos y delimitamos el número máximo de tokens. Sin embargo, a pesar de la salida limitada, se centró en detalles poco relevantes, como las dimensiones del barco, dejando fuera partes importantes del suceso. Esto se reflejó en sus puntuaciones más bajas, sobre todo en ROUGE-2 y BERTScore, lo que indica que el texto no tenía buena cohesión ni reflejaba bien el significado original. Su configuración más limitada (menos tokens, menos épocas y un learning rate más alto) influyó claramente en este resultado.

El Modelo 2, con una configuración más completa (más tokens, más épocas de entrenamiento y una tasa de aprendizaje más baja), consiguió un resumen mucho más equilibrado y coherente. Captó mejor el contenido esencial y presentó el texto de forma clara. Igualó incluso la puntuación F1 de BERTScore del modelo XLSum, alcanzando un valor de 0.893, lo que demuestra que fue capaz de mantener muy bien el significado del texto original.

El Modelo 3 (XLSum), como era de esperar por ser un modelo ya entrenado con muchos datos multilingües, obtuvo las mejores puntuaciones en ROUGE. Su resumen fue bueno a nivel estructural, pero se notó cierta repetición de frases que le restó naturalidad. Aun así, fue el modelo con mayor compresión (0.20), lo que demuestra que tiene una buena capacidad de síntesis.

En cuanto a los métodos extractivos, el enfoque Lead-3 fue el que mejores métricas obtuvo en general, sin embargo, hay que recordar que este método no es siempre fiable ya que simplemente escoge las 3 primeras frases del texto original. En esta ocasión, al ser una noticia y comenzar con lo importante para enganchar al lector, la estrategia funcionó

Los métodos Luhn y TextRank también dieron buenos resultados, especialmente en fidelidad semántica, con un F1 de 0.943, lo que muestra que captaron bien el sentido del texto. Igual que Lead-3, generaron resúmenes bastante extensos, con compresiones bajas (alrededor de 0.45), lo cual puede no ser ideal si se necesita algo más resumido.

El modelo TF-IDF, por su parte, sí logró reducir más el texto (compresión de 0.31), pero a cambio obtuvo un rendimiento más bajo en las métricas de calidad, con un F1 de 0.920, siendo el que peor puntuación sacó entre los métodos extractivos.

Finalmente, la combinación de TextRank + TF-IDF fue bastante equilibrada. Obtuvo un F1 de 0.929 y una compresión de 0.44, por lo que logró un buen balance entre reducir el texto y mantener su significado. Por eso, puede considerarse una opción eficiente cuando se busca un término medio entre calidad y síntesis.

A continuación, se muestran las imágenes correspondientes a una de las simulaciones realizadas para comparar los tres modelos de resumen abstractivo. El texto original utilizado en la prueba hace referencia a un accidente ocurrido en el puente de Brooklyn (Nueva York).

Cada uno de los modelos procesó el mismo texto original y generó un resumen automático. En las capturas siguientes se puede observar, para cada modelo:

- El resumen generado.
- Las métricas automáticas obtenidas: ROUGE-1, ROUGE-2, ROUGE-L, BERTScore (Precisión, Recall y F1), y la tasa de compresión.

Estas imágenes reflejan los resultados expuestos anteriormente en el análisis comparativo, y permiten valorar de forma directa las diferencias entre los modelos en cuanto a fidelidad, capacidad de síntesis y calidad de generación textual.

Introduce un texto en español y compara cómo resumen dos modelos distintos fine-tuneados.

texto

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se disponía a abandonar la ciudad. La que iba a ser una gran fiesta de despedida, después de una escala de cuatro días, acabó en una tragedia difícil de explicar. Al menos dos miembros de la tripulación murieron y otras 20 personas resultaron heridas al partirse los tres mástiles del buque, que fueron chocando uno a uno contra la base del puente más famoso de la ciudad.

Los cadetes encaramados sobre los palos -el mayor tiene una altura de 45 metros, mientras que la pasarela del puente alcanza alrededor de 39- fueron los últimos en darse cuenta de lo que iba a suceder. Pasadas las ocho de la tarde, el barco iniciaba su maniobra de zarpe. Los cadetes, dando la espalda al puente de Brooklyn, decían adiós a la multitud agolpada en el Pier 17, del lado de Manhattan. Segundos después, el buque pareció desequilibrarse, según algunos testigos, y en reversa, empujado por la corriente, se fue directo contra el puente. Algunos de los muchachos cayeron al vacío mientras otros quedaron suspendidos de los arneses en el aire.

Limpiar

Enviar

Resumen Modelo 1

los cadetes encaramados sobre los palos tiene una altura de 45 metros, mientras la pasarela del puente alcanza alrededor de 39 . el barco iniciaba su maniobra de zarpe

ROUGE Modelo 1

rouge1: 0.23
rouge2: 0.198
rougeL: 0.23

BERTScore Modelo 1

Precision: 0.941
Recall: 0.820
F1: 0.876

Compresión Modelo 1

Tasa de compresión: 0.14

Resumen Modelo 2

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se dispona a abandonar la ciudad

ROUGE Modelo 2

rouge1: 0.25
rouge2: 0.236
rougeL: 0.25

BERTScore Modelo 2

Precision: 0.961
Recall: 0.833
F1: 0.893

Compresión Modelo 2

Tasa de compresión: 0.15

Resumen Modelo XLSum

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se disponía a abandonar la ciudad. Cuando se estaba preparando para salir, se preparaba para zarpear.

ROUGE Modelo 3

rouge1: 0.263
rouge2: 0.249
rougeL: 0.263

BERTScore Modelo 3

Precision: 0.953
Recall: 0.840
F1: 0.893

Compresión Modelo 3

Tasa de compresión: 0.20

Resumen Extractivo LEAD-3

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se disponía a abandonar la ciudad. Al menos dos miembros de la tripulación murieron y otras 20 personas resultaron heridas a partir de los tres mástiles del buque, que fueron chocando uno a uno contra la base del puente más famoso de la ciudad. Los cadetes encaramados sobre los palos -el mayor tiene una altura de 45 metros, mientras que la pasarela del puente alcanza alrededor de 39- fueron los últimos en darse cuenta de lo que iba a suceder.

ROUGE LEAD-3

rouge1: 0.673
rouge2: 0.665
rougeL: 0.673

BERTScore LEAD-3

Precision: 0.983
Recall: 0.907
F1: 0.943

Compresión LEAD-3

Tasa de compresión: 0.51

Resumen Extractivo Luhn

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se disponía a abandonar la ciudad. Lo que iba a ser una gran fiesta de despedida, después de una escala de cuatro días, acabó en una tragedia difícil de explicar. Al menos dos miembros de la tripulación murieron y otras 20 personas resultaron heridas a partir de los tres mástiles del buque, que fueron chocando uno a uno contra la base del puente más famoso de la ciudad.

ROUGE Luhn

rouge1: 0.635
rouge2: 0.633
rougeL: 0.635

BERTScore Luhn

Precision: 0.992
Recall: 0.898
F1: 0.943

Compresión Luhn

Tasa de compresión: 0.45

Resumen Extractivo TextRank

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se disponía a abandonar la ciudad. Lo que iba a ser una gran fiesta de despedida, después de una escala de cuatro días, acabó en una tragedia difícil de explicar. Al menos dos miembros de la tripulación murieron y otras 20 personas resultaron heridas a partir de los tres mástiles del buque, que fueron chocando uno a uno contra la base del puente más famoso de la ciudad.

ROUGE TextRank

rouge1: 0.635
rouge2: 0.633
rougeL: 0.635

BERTScore TextRank

Precision: 0.992
Recall: 0.898
F1: 0.943

Compresión TextRank

Tasa de compresión: 0.45

Resumen Extractivo TF-IDF

Los cadetes, dando la espalda al puente de Brooklyn, decían adiós a la multitud agrupada en el Pier 17, del lado de Manhattan. Segundos después, el buque pareció desestabilizarse, según algunos testigos, y en reversa, empujado por la corriente, se fue directo contra el puente. Algunos de los muchachos cayeron al vacío mientras otros quedaron suspendidos de los arneses en el aire.

ROUGE TF-IDF

rouge1: 0.472
rouge2: 0.468
rougeL: 0.472

BERTScore TF-IDF

Precision: 0.965
Recall: 0.879
F1: 0.920

Compresión TF-IDF

Tasa de compresión: 0.31

Resumen Extractivo TextRank-TFIDF

El buque escuela Cuauhtémoc de la Armada mexicana ha chocado la tarde de este sábado contra el puente de Brooklyn, en Nueva York, cuando se disponía a abandonar la ciudad. Los cadetes encaramados sobre los palos -el mayor tiene una altura de 45 metros, mientras que la pasarela del puente alcanza alrededor de 39- fueron los últimos en darse cuenta de lo que iba a suceder. Los cadetes, dando la espalda al puente de Brooklyn, decían adiós a la multitud agrupada en el Pier 17, del lado de Manhattan.

ROUGE TextRank-TFIDF

rouge1: 0.605
rouge2: 0.595
rougeL: 0.605

BERTScore TextRank-TFIDF

Precision: 0.968
Recall: 0.893
F1: 0.929

Compresión TextRank-TFIDF

Tasa de compresión: 0.44

Modelo	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore F1	Compresión
Modelo 1 (T5 v1)	0.23	0.198	0.23	0.876	0.14
Modelo 2 (T5 v2)	0.25	0.236	0.25	0.893	0.15
Modelo 3 (XLSum)	0.263	0.249	0.263	0.893	0.2
Lead-3	0.673	0.665	0.673	0.943	0.51
Luhn	0.635	0.633	0.635	0.943	0.45
TF-IDF	0.472	0.468	0.472	0.92	0.31
TextRank	0.635	0.633	0.635	0.943	0.45
TextRank + TF-IDF	0.605	0.595	0.605	0.929	0.44

Además de las pruebas realizadas con textos de noticias, también se hicieron evaluaciones con otro tipo de contenido, en este caso un texto informativo más general. Esto nos permitió ver cómo se comportaban los modelos con temáticas distintas y no tan narrativas como las noticias. A continuación, se muestran los resultados obtenidos al aplicar los modelos abstractivos y extractivos sobre un texto relacionado con el concepto de “agricultura”.

En el caso de los modelos abstractivos, el modelo T5-small con configuración básica (Modelo 1) generó un resumen breve, centrado en la definición general de la agricultura. Aunque su tasa de compresión fue la más alta entre los modelos abstractivos, logró una buena fidelidad semántica, como muestra su BERTScore. Sin embargo, no logró cubrir adecuadamente todo el contenido relevante del texto, lo que se reflejó en puntuaciones ROUGE moderadas.

El modelo 2, que corresponde a T5-small con una configuración mejorada, ofreció mejores resultados en todas las métricas ROUGE. Esto indica una mayor cohesión y cobertura del contenido. Aunque su BERTScore bajó ligeramente, el resumen fue más complejo y equilibrado lo que lo convierte en el modelo abstractivo más eficaz del experimento.

En contraste, el modelo multilingüe preentrenado presentó una compresión más alta, pero con una caída considerable en las métricas ROUGE y BERTScore. Este modelo generó un resumen más generalista y global, esto sugiere una menor fidelidad. Esto puede deberse a su entrenamiento en múltiples idiomas y dominios, lo que dio lugar a una alucinación del contenido.

Por otro lado, los modelos extractivos obtuvieron en general puntuaciones más altas en ROUGE y BERTScore, dado que trabajan con fragmentos reales del texto original. No

obstante, su principal desventaja fue la baja compresión, lo que indica que tienen una baja capacidad de síntesis.

El enfoque de Luhn logró equilibrar precisión y síntesis, con ROUGE-2 de 0.435 y BERTScore F1 de 0.932, acompañado de una compresión de 0.28, siendo más eficiente que TextRank sin perder demasiada fidelidad. El modelo basado en TF-IDF, por su parte, mostró resultados notables en ROUGE y BERTScore F1, y una compresión intermedia, lo que demuestra una buena capacidad de síntesis con sensibilidad semántica.

Finalmente, la variante que combina TextRank con TF-IDF ofreció un balance con una compresión de 0.51 y un BERTScore F1 de 0.935, con ROUGE-2 de 0.667. Esta combinación permite mantener la coherencia y relevancia del contenido sin depender de fragmentos consecutivos ni simplemente las primeras frases del texto.

En conjunto, los modelos extractivos sobresalen en fidelidad y exactitud gracias a que extraen frases originales del texto, pero a costa de generar resúmenes más extensos. Por el contrario, los modelos abstractivos ofrecen resúmenes más compactos y naturales, pero requieren un entrenamiento cuidadoso para no perder información relevante o introducir datos erróneos. Entre ellos, el modelo T5-small ajustado (versión 2) demostró ser el más equilibrado en términos de calidad, cohesión y eficiencia.

texto

La agricultura es el conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y el cultivo de la tierra para la producción de alimentos. Comprende todo un conjunto de acciones humanas que transforma el medio ambiente natural.

Las acciones relacionadas son las que integran el llamado sector agrícola. Todas las actividades económicas del sector tienen su fundamento en la explotación de los recursos que la tierra origina, favorecida por la acción del ser humano: alimentos vegetales como cereales, frutas, hortalizas, pastos cultivados y forrajes; fibras utilizadas por la industria textil; cultivos energéticos, etc.

La agricultura[1] también incluye una demanda global del ramo y el servicio de la alimentación mundial depende en gran medida del clima y de las técnicas para poder hacer la tierra fértil, conserva su origen en la propiedad privada y en la explotación de la tierra entregada a familias para poder establecerse. Es una actividad de gran importancia estratégica como base fundamental para el desarrollo autosuficiente y dinero de las naciones. La ciencia que estudia la práctica de la agricultura es la agronomía.

Resumen Modelo 1

la agricultura es un conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y un cultivo de la tierra para la producción de alimentos

ROUGE Modelo 1

rouge1: 0.258
rouge2: 0.233
rougeL: 0.249

BERTScore Modelo 1

Precision: 0.967
Recall: 0.846
F1: 0.903

Compresión Modelo 1

Tasa de compresión: 0.15

Resumen Modelo 2

La agricultura también incluye una demanda global del ramo y el servicio de la alimentación mundial depende en gran medida de clima y las técnicas para poder hacer la tierra fértil

ROUGE Modelo 2

rouge1: 0.304
rouge2: 0.272
rougeL: 0.296

BERTScore Modelo 2

Precision: 0.943
Recall: 0.844
F1: 0.891

Compresión Modelo 2

Tasa de compresión: 0.17

Resumen Modelo 3 (XLSum)

La agricultura es una de las actividades más importantes de la economía mundial y es la principal fuente de atención global para el desarrollo de los países del hemisferio norte, el Pacífico, la Unión Europea, Estados Unidos, Japón, China, Corea del Sur, África Occidental, Sudamérica, América Latina.

ROUGE Modelo 3

rouge1: 0.223
rouge2: 0.08
rougeL: 0.135

BERTScore Modelo 3

Precision: 0.861
Recall: 0.836
F1: 0.848

Compresión Modelo 3

Tasa de compresión: 0.26

La agricultura es el conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y el cultivo de la tierra para la producción de alimentos. Todas las actividades económicas del sector tienen su fundamento en la explotación de los recursos que la tierra origina, favorecida por la acción del ser humano: alimentos vegetales como cereales, frutas, hortalizas, pastos cultivados y forrajes; fibras utilizadas por la industria textil; cultivos energéticos, etc. La agricultura también incluye una demanda global del ramo y el servicio de la alimentación mundial depende en gran medida del clima y de las técnicas para poder hacer la tierra fértil, conserva su origen en la propiedad privada y en la explotación de la tierra entregada a familias para poder establecerse.

ROUGE LEAD-3

rouge1: 0.822
rouge2: 0.815
rougeL: 0.822

BERTScore LEAD-3

Precision: 0.982
Recall: 0.943
F1: 0.962

Compresión LEAD-3

Tasa de compresión: 0.69

Resumen Extractivo Luhn

La agricultura es el conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y el cultivo de la tierra para la producción de alimentos. Comprende todo un conjunto de acciones humanas que transforma el medio ambiente natural. Las acciones relacionadas son las que integran el llamado sector agrícola.

ROUGE Luhn

rouge1: 0.44
rouge2: 0.435
rougeL: 0.44

BERTScore Luhn

Precision: 0.991
Recall: 0.879
F1: 0.932

Compresión Luhn

Tasa de compresión: 0.28

Resumen Extractivo TF-IDF

La agricultura es el conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y el cultivo de la tierra para la producción de alimentos. Las acciones relacionadas son las que integran el llamado sector agrícola. Todas las actividades económicas del sector tienen su fundamento en la explotación de los recursos que la tierra origina, favorecida por la acción del ser humano: alimentos vegetales como cereales, frutas, hortalizas, pastos cultivados y forrajes; fibras utilizadas por la industria textil; cultivos energéticos, etc.

ROUGE TF-IDF

rouge1: 0.636
rouge2: 0.627
rougeL: 0.636

BERTScore TF-IDF

Precision: 0.982
Recall: 0.914
F1: 0.947

Compresión TF-IDF

Tasa de compresión: 0.46

La agricultura es el conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y el cultivo de la tierra para la producción de alimentos. Todas las actividades económicas del sector tienen su fundamento en la explotación de los recursos que la tierra origina, favorecida por la acción del ser humano: alimentos vegetales como cereales, frutas, hortalizas, pastos cultivados y forrajes; fibras utilizadas por la industria textil; cultivos energéticos, etc. La agricultura también incluye una demanda global del ramo y el servicio de la alimentación mundial depende en gran medida del clima y de las técnicas para poder hacer la tierra fértil, conserva su origen en la propiedad privada y en la explotación de la tierra entregada a familias para poder establecerse.

ROUGE TextRank

rouge1: 0.822
rouge2: 0.815
rougeL: 0.822

BERTScore TextRank

Precision: 0.982
Recall: 0.943
F1: 0.962

Compresión TextRank

Tasa de compresión: 0.69

Resumen Extractivo TextRank-TFIDF

La agricultura es el conjunto de actividades económicas y técnicas relacionadas con el tratamiento del suelo y el cultivo de la tierra para la producción de alimentos. La agricultura también incluye una demanda global del ramo y el servicio de la alimentación mundial depende en gran medida del clima y de las técnicas para poder hacer la tierra fértil, conserva su origen en la propiedad privada y en la explotación de la tierra entregada a familias para poder establecerse. La ciencia que estudia la práctica de la agricultura es la agronomía.

ROUGE TextRank-TFIDF

rouge1: 0.682
rouge2: 0.667
rougeL: 0.682

BERTScore TextRank-TFIDF

Precision: 0.969
Recall: 0.903
F1: 0.935

Compresión TextRank-TFIDF

Tasa de compresión: 0.51

Modelo	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore F1	Compresión
Modelo 1 (T5 v1)	0.258	0.233	0.249	0.903	0.15
Modelo 2 (T5 v2)	0.304	0.272	0.296	0.891	0.17
Modelo 3 (XLSum)	0.223	0.080	0.135	0.848	0.26
Lead-3	0.822	0.815	0.822	0.962	0.69
Luhn	0.440	0.435	0.440	0.932	0.28
TF-IDF	0.636	0.627	0.636	0.947	0.46
TextRank	0.822	0.815	0.822	0.962	0.69
TextRank + TF-IDF	0.682	0.667	0.682	0.935	0.51

Además de las métricas automáticas, también realizamos una evaluación manual subjetiva. Cada integrante del grupo valoró individualmente los resúmenes generados por los distintos modelos, teniendo en cuenta tres criterios: coherencia (fluidez del texto), claridad (facilidad de comprensión) y fidelidad (precisión respecto al contenido original). Las puntuaciones se asignaron en una escala del 1 al 5, donde 5 representa una valoración muy positiva y 1 una muy negativa. Posteriormente, se discutieron las valoraciones en grupo y se calculó la media para cada modelo.

Los resultados fueron los siguientes:

- Modelo 1 (T5-small, configuración básica): Obtuvo puntuaciones de 2, 2 y 3, con una media de 2.3. El resumen resultó demasiado corto, poco representativo del evento central y excesivamente centrado en detalles irrelevantes.
- Modelo 2 (T5-small, configuración mejorada): Recibió puntuaciones de 4, 5 y 5, alcanzando una media de 4.7. Fue el modelo mejor valorado por su capacidad de síntesis, claridad expositiva y detección adecuada de los aspectos clave del texto.
- Modelo 3 (XLSum preentrenado): Fue valorado con 4, 4 y 3, lo que dio una media de 3.7. Aunque se consideró correcto a nivel estructural, se detectó cierta repetición de ideas y menor precisión en algunos detalles, lo que restó puntos en fidelidad.
- Lead-3: Obtuvo puntuaciones de 4, 4 y 4, con una media de 4.0. Si bien no hubo generación activa de contenido, su estrategia de tomar las primeras frases fue efectiva. Los resúmenes fueron claros y directos, aunque con falta de síntesis.
- Luhn: Recibió puntuaciones de 4, 3 y 4, con una media de 3.7. Su selección de frases clave fue generalmente acertada, aunque la estructura resultó algo densa y menos fluida que otros modelos. Aun así, la fidelidad fue alta.

- TF-IDF: Las puntuaciones fueron 3, 3 y 4, obteniendo una media de 3.3. Fue valorado como suficientemente fiel pero con cierta falta de claridad. Algunas frases parecían descontextualizadas, lo que afectó la coherencia global del resumen.
- TextRank: Se valoró con 3, 4 y 4, dando una media de 3.7. Generó un resumen más legible que TF-IDF, aunque todavía algo largo. Se valoró positivamente su fidelidad, aunque se mencionó cierta rigidez en la transición entre frases.
- TextRank + TF-IDF: Recibió valoraciones de 4, 4 y 5, alcanzando una media de 4.3. Este modelo fue uno de los mejor equilibrados según la evaluación manual. Destacó por conservar bien el contenido relevante sin perder claridad, y por ofrecer una estructura más natural que otros modelos extractivos.

En conjunto, la evaluación subjetiva permitió complementar los resultados automáticos, ofreciendo una perspectiva más centrada en la experiencia del lector. Se evidenció que los modelos abstractivos bien ajustados (especialmente el Modelo 2) pueden generar resúmenes más naturales y comprensibles, mientras que algunos enfoques extractivos (como Lead-3 o la combinación TextRank+TF-IDF) ofrecieron una muy buena fidelidad al texto original, a costa de ser menos sintéticos

6. APORTACIONES

El desarrollo de este proyecto ha sido realizado de forma colaborativa, procurando que cada uno participara en en las diferentes etapas del sistema de generación de resúmenes. El trabajo se ha dividido de la siguiente manera:

- Marta: Implementó el scraping de Wikipedia y desarrolló los modelos extractivos basados en Luhn y Frequent Words. Participó en el análisis de resultados y en la evaluación de comprensión y métricas automáticas, además de contribuir a la redacción del informe final.
- Irene: Desarrolló el scraping de noticias desde medios digitales, entrenó la primera versión del modelo abstractivo T5-small y aplicó las técnicas de preprocesamiento y tokenización. También participó en la documentación y en la integración de los resúmenes generados.
- Iker: Implementó los métodos de resumen basados en TextRank y su variante con TF-IDF, entrenó y ajustó la segunda versión del modelo T5-small, además de coordinar la comparación entre modelos extractivos y abstractivos y colaborar en la elaboración del informe final.

7. REPOSITORIO

El código fuente y los recursos utilizados durante el desarrollo del proyecto se encuentran disponibles en el repositorio de GitHub. Este repositorio está estructurado en varias secciones que agrupan los diferentes componentes del trabajo.

Colab: Contiene notebooks desarrollados en Google Colab:

- **abstractivo.ipynb:** Entrenamiento del modelo T5-small mediante fine-tuning sobre los datasets.
- **comparativa_extractivo_abstractivo.ipynb:** Implementación de métodos extractivos y análisis comparativo entre enfoques extractivos y modelos abstractivos.

Scraper: Incluye scripts en Python diseñados para la obtención de datos reales desde fuentes online:

- **scraping.py:** Extracción de noticias recientes desde el diario La Razón.
- **scraping_Gemini.py:** Recolección de artículos científicos desde Wikipedia.

Dataset: Agrupa los conjuntos de datos empleados para el entrenamiento y evaluación de los modelos:

- **noticias_10.csv:** Noticias reales recopiladas mediante scraping de medios digitales.
- **wikipedia_resumenes.tsv:** Artículos técnicos extraídos de Wikipedia junto con sus resúmenes generados.

Documentación: Contiene la documentación desarrollada:

- **documentacion_final.pdf:** Memoria detallada que la metodología utilizada, comparativas entre modelos y el análisis de los resultados obtenidos.

Enlace: <https://github.com/IreeneGG/NLP>

8. BIBLIOGRAFIA

- [1] Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., & Staiano, J. (2020). MLSUM: The Multilingual Summarization Corpus. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). <https://doi.org/10.18653/v1/2020.emnlp-main.647>
- [2] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research, 21(140), 1–67. <https://jmlr.org/papers/v21/20-074.html>
- [3] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Proceedings of ACL 2020. <https://doi.org/10.18653/v1/2020.acl-main.703>
- [4] Liu, Y., & Lapata, M. (2019). Text Summarization with Pretrained Encoders. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP). <https://doi.org/10.18653/v1/D19-1387>
- [5] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. Proceedings of ICML 2020. <http://proceedings.mlr.press/v119/zhang20ae.html>
- [6] Hashemi, H., Narayan, S., & McDonald, R. (2021). XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. Findings of ACL 2021. <https://doi.org/10.18653/v1/2021.findings-acl.413>
- [7] Ahuir, A., Martos, R., & Ortiz, A. (2021). NASes: Entrenamiento de un modelo BART para generación de resúmenes en español. Procesamiento del Lenguaje Natural, 67, 83–90. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6403>
- [8] Vogel-Fernández, A., Espinosa-Anke, L., & Saggion, H. (2022). esT5: Neural Text Generation for Spanish. Procesamiento del Lenguaje Natural, 69, 65–76. <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6481>
- [9] Bernoldi, M., & Tolosa, I. (2022). Resumen automático de noticias en español con BERTSum y BETO. IberLEF 2022 - SEPLN. http://ceur-ws.org/Vol-3202/spanishNLP_paper6.pdf