

MASTER EN INTELIGENCIA ARTIFICIAL



TRABAJO FIN DE MÁSTER

APLICACIÓN DEL NLP AL RESUMEN DE TEXTOS

Alumno: **YANIS NAVARRETE CARMONA**

Director: **SERGIO LUIS NAÑEZ ALONSO**

Convocatoria: Junio

Año: 2021

ABSTRACT

Natural Language Processing (NLP) applications have boomed in recent years. Its use for automatic text summarization (ATS) has been showing surprising results, that are getting closer to those of humans. This paper aims to provide a historical overview to get a good context of the subject, starting from extractive algorithms, which generate summaries by concatenating the most important sentences, to abstractive algorithms, which generate the summary using sentences that do not appear in the original text, in the same way as humans do. The paper shows the performance of some of the SOTA algorithms and offers a solution to one of their major problems: The limited size of text they can support as input. Given this limitation, it has been opted for the combination of extractive and abstractive techniques to achieve a robust algorithm that can deal with any input size, where the extractive algorithm is responsible for selecting the most relevant sentences of the original text and passes this selection to the abstractive model, with an adjusted size to the number of tokens that this model is able to process. In this way, an abstractive summary is obtained. Using this algorithm, an application has been developed so that any user can perform summaries in a simple way. Moreover, the user can choose among the different algorithms and models that are available and adjust the summary to his needs, customizing different parameters such as the desired summary length.

RESUMEN

Las aplicaciones del NLP o procesamiento del lenguaje natural han tenido un auge enorme en los últimos años. Su aplicación al resumen automático de textos ofrece resultados que se acercan cada vez más a los de los humanos. El trabajo pretende realizar un recorrido histórico para obtener un buen contexto de la temática, desde los algoritmos extractivos, que generan resúmenes concatenando las frases más importantes, hasta los abstractivos, los cuales generan el resumen utilizando frases que no aparecen en el texto original, de la misma manera que el ser humano. El trabajo muestra el funcionamiento de algunos de los mejores algoritmos que existen en la actualidad y ofrece una solución a uno de los mayores problemas de estos: El escaso tamaño de texto que pueden soportar como entrada. Ante esta limitación se ha optado por la combinación de técnicas extractivas y abstractivas para lograr un algoritmo robusto ante cualquier tamaño de entrada, donde el algoritmo extractivo se encarga de seleccionar las frases más relevantes del texto original y le pasa esta selección al modelo abstractivo, con un tamaño ajustado al que número de *tokens* que este modelo es capaz de soportar. De esta manera se obtiene un resumen abstractivo. Utilizando este algoritmo, se ha desarrollado una aplicación para que cualquier usuario pueda realizar resúmenes de manera sencilla. Además, el usuario puede elegir entre los diferentes algoritmos y modelos que están a su disposición y ajustar el resumen a su medida, modificando diferentes parámetros como la longitud de resumen deseada.

AGRADECIMIENTOS

A mis padres, a mi hermana y al resto de mi familia, ya que han supuesto un apoyo increíble para mí, desde el día que nací hasta hoy. Me han inculcado valores fundamentales para mi crecimiento personal y académico y me han ayudado a ser quien soy.

A mi pareja, que desde el comienzo de nuestra relación no ha dudado en apoyarme en mis estudios y a progresar en mi vida laboral. Con ella he pasado momentos de desesperación en el tema académico y ha sabido reconducirme y darme fuerza para seguir.

A mis amigos, porque tras tantos años juntos siempre han estado ahí para echarme una mano en lo que me hiciera falta.

A mis profesores, desde mis comienzos en el colegio hasta ahora. Sin ellos mi etapa de aprendizaje hubiera sido totalmente diferente, y algunos me han despertado el interés por ciertos temas que me han llevado a desarrollarme más en estos.

Y por último, a todas esas personas altruistas que comparten sus conocimientos en internet de manera gratuita, lo cual me ha permitido aprender gran parte de las cosas que ahora sé, y que me siguen despertando el interés y la motivación por continuar mi aprendizaje más allá de mi etapa universitaria.

Ahora estoy metido de lleno en el sector de la Inteligencia Artificial, el *Machine Learning* y el *Data Science*, y me considero un apasionado de este mundo tan inquietante.

Por todas estas personas que han dejado su huella en mí estoy donde estoy ahora mismo, y puedo decir orgulloso que me gusta dónde estoy.

*“Los desafíos son los que hacen que la vida sea interesante
y superarlos es lo que hace que la vida tenga sentido.”*

- Joshua J. Marine

ÍNDICE

Abstract	1
Resumen	2
Agradecimientos	3
Índice	5
Índice de Figuras y Tablas.....	8
Abreviaturas	9
Introducción.....	10
1. Justificación.....	13
Objetivos.....	14
Capítulo I. Material y metodología.....	16
Capítulo II. Estado del Arte	18
3. Estudio de la Problemática.....	19
3.1. Deep Learning	19
Redes Neuronales Recurrentes (RNN).....	19
LSTM	20
3.2. Word Embeddings	21
3.3. Modelos de Atención.....	21
3.4. Transformers.....	22
3.5. Evaluación	25
Recall.....	26
Precisión	26
F1-score	27
ROUGE	27
4. Antecedentes	29
4.1. Métodos Extractivos	29
TextRank	29
Luhn.....	29
LexRank	30
Latent Semantic Analysis (LSA).....	30
KL-Sum	30
4.2. Métodos Abstractivos	30
T5.....	31
BERT.....	31
BART	31
PEGASUS	32

Longformer	32
5. Tecnologías para el desarrollo	33
5.1. Librerías Python	33
NumPy	33
NLTK	33
TensorFlow	34
Keras	34
Pytorch	34
Transformers	34
Gensim	35
Sumy	35
Streamlit	35
BeautifulSoup.....	35
5.2. Entorno de Trabajo	36
Jupyter	36
PyCharm.....	36
GitHub.....	36
Capítulo III. Experimentación	38
6. Enfoques	38
6.1. Texto.....	38
Preprocesamiento	39
6.2. Datasets	40
DUC2001	40
CNN / Dailymail	40
6.3. Técnicas Extractivas.....	41
Gensim	41
TextRank	41
Sumy	43
Luhn.....	43
LexRank	44
Latent Semantic Analysis (LSA).....	45
KL-Sum	46
Evaluación.....	47

6.4. Técnicas Abstractivas	48
Transformers	48
T5	49
BART	51
Longformer	51
PEGASUS	53
Evaluación	55
6.5. Limitaciones	56
6.6. Solución	56
Resultados y discusión.....	62
7. Aplicación web.....	62
Conclusiones finales.....	71
8. Trabajo Futuro.....	71
Bibliografía.....	73
Anexo	76

ÍNDICE DE FIGURAS Y TABLAS

Figura 1. Arquitectura de un sistema ATS	12
Figura 2. Estructura básica de una red neuronal	19
Figura 3. Neurona de una RNN.....	20
Figura 4. Secuencia temporal de una neurona de RNN	20
Figura 5. Representación de word embeddings	21
Figura 6. Arquitectura de un transformer	22
Figura 7. Ejemplo de funcionamiento de un transformer en una tarea de traducción.....	23
Figura 8. Descomposición de un transformer en una estructura de encoder-decoder.....	23
Figura 9. Descomposición de encoder y decoder en 6 partes	23
Figura 10. Capas de un encoder	24
Figura 11. Diferencia entre las capas de un encoder y las de un decoder	24
Figura 12. Ejemplo de cálculo del recall (Fuente: Briggs, 2021)	26
Figura 13. Ejemplo de cálculo de la precisión	26
Figura 14. Ejemplo de cálculo del F1-score.....	27
Figura 15. Ejemplo de la obtención de la secuencia común más larga (LCS).....	28
Figura 16. Interfaz de la app.....	63
Figura 17. Opciones de customización	64
Figura 18. Menú desplegable Extractivo/Abstractivo.....	64
Figura 19. Menú desplegable de algoritmos extractivos.....	65
Figura 20. Opciones de customización con el tipo de resumen abstractivo.....	66
Figura 21. Menú desplegable de algoritmos abstractivos	67
Figura 22. Menú desplegable modelos de BART preentrenados.....	67
Figura 23. Menú desplegable modelos de T5 preentrenados.....	67
Figura 24. Menú desplegable modelos de Pegasus preentrenados	68
Figura 25. Resultado de generar un resumen con la app.....	69
 Tabla 1. Puntuaciones ROUGE de algunos algoritmos extractivos sobre el dataset DUC2001 (Fuente: Victor et al., 2019)	 47
Tabla 2. Puntuaciones ROUGE de algunos algoritmos abstractivos sobre el dataset CNN / Dailymail (Fuente: https://paperswithcode.com).	55

ABREVIATURAS

- **NLP:** Natural Language Processing (Procesamiento del Lenguaje Natural).
- **ATS:** Automatic Text Summarization (Resumen Automático de Textos).
- **SOTA:** State-of-the-art (Estado del arte).

INTRODUCCIÓN

Desde hace más de 5000 años, el lenguaje escrito ha sido el medio más importante para documentar y transmitir el conocimiento. Esto ha cobrado aún más importancia en la actual era de la digitalización. A causa del descenso en los costes de producir, almacenar y reproducir textos digitales, la cantidad de texto disponible de manera digital, ya sea de manera local u online, está creciendo de manera vertiginosa. Resumir estos textos se convierte en una necesidad a la hora de ayudar a los usuarios a manejar esta información y percibir mejor la información. El resumen de textos, en una primera clasificación, se puede dividir en dos tipos: resumen manual y resumen automático. El resumen manual consiste en la creación de un resumen de manera manual por un experto humano. Esta tarea consume mucho tiempo, esfuerzo y coste. En la práctica, condensar una cantidad grande de texto es una tarea complicada. Es necesario, por tanto, automatizar esta tarea de manera que sea más rápida, barata y no requiera intervención humana.

El resumen automático de textos o ATS (Automatic Text Summarization, por sus siglas en inglés) es el proceso de acortar el contenido de una fuente textual, de manera que retenga la información más importante. Los ATS están ganando importancia en el terreno del NLP por las ingentes cantidades de texto que se encuentran en internet. Este tipo de contenido aumenta diariamente de manera exponencial, y se publica en diferentes formatos como artículos, noticias, papers científicos, documentos legales, e-mails, etc. Es una tarea complicada para el usuario obtener información y conocimientos a partir de volúmenes tan grandes de datos. Además, una gran parte de estos datos es redundante o no contiene información valiosa. La manera más eficiente de acceder a las partes más importantes de los textos, sin tener que pasar por las partes redundantes, es condensar éstos de manera que solamente incluyan información útil. El objetivo principal de un sistema ATS consiste en crear un resumen que incluya las ideas principales del documento, en menos espacio y minimizando las repeticiones (Moratanch & Chitrakala, 2017). Un buen resumen debe ser consistente y fluido y contener todos los temas importantes, sin ser repetitivo. Los sistemas ATS ayudan al usuario a extraer los principales puntos del documento sin necesidad leerlo completo. Con esto, el usuario se ahorrará mucho tiempo. Esto puede resultar útil en muchos ámbitos como generar informes en una empresa, ayudar a estudiantes y científicos a buscar las ideas más importantes que sean relevantes para su investigación, resumir a un médico la información médica de un paciente o recopilar las noticias principales que puedan resultar útiles a un periodista o usuario.

Para entender el trabajo propuesto conviene entender los principales conceptos. En Radev et al. (2002), según Radev: “Un resumen puede definirse como un texto que se produce a partir de uno o más textos, que transmite información importante del texto o los textos originales y que no supera la mitad del texto o los textos originales y que, por lo general, es significativamente menor que este.”

En Maybury (1995), Maybury definió el resumen automatizado de la siguiente manera: “Un resumen eficaz extrae la información más importante de una fuente (o fuentes) para producir una versión abreviada de la información original para un usuario o tarea en particular”.

Según la cantidad de documentos utilizados para hacer el resumen, los sistemas ATS se pueden clasificar en sistemas de resumen mono-documento o multi-documento. El primero, produce un resumen a partir de un único documento mientras que, el segundo, lo genera a partir de un conjunto de documentos.

Los sistemas ATS se pueden aplicar siguiendo el método extractivo, abstractivo o híbrido. El método extractivo consiste en seleccionar las frases más relevantes del texto original y generar el resumen utilizando estas mismas frases. Es el método más básico y sencillo, sin embargo, dista mucho de la forma en la que los humanos resumen la información. Los humanos utilizan el método abstractivo, el cual consiste en resumir el texto usando frases y palabras que difieren del texto original. Durante muchos años, las técnicas extractivas fueron el principal objeto de investigación. Sin embargo, en los últimos años han sido pocos los avances en el resumen extractivo de textos. Las investigaciones más recientes de este método centran sus propuestas en mejorar los enfoques ya existentes o en combinar varios de los métodos extractivos ya conocidos. El enfoque alternativo, del resumen abstractivo, ha ganado mucha relevancia recientemente, especialmente con la revolución del Deep Learning y los modelos de atención que se aplican en los *transformers*. Aunque los dos métodos anteriores son los principales, también existe el método híbrido, donde para el resumen final se utilizan ambos métodos, extractivo y abstractivo.

La arquitectura general de un sistema ATS, como se muestra en la

Figura 1, consiste en los siguientes pasos:

1. Preprocesado: En esta fase se produce una representación estructurada del texto original usando técnicas lingüísticas como la segmentar de frases, tokenizar de palabras, borrar los *stopwords*, etiquetado *part-of-speech* (POS), truncación (*stemming*), lematización, etc.
2. Procesado: Tras el preprocesado, se aplica uno de los métodos ATS para convertir el documento o documentos en un resumen. En el apartado Antecedentes se enumerarán las diferentes técnicas de resumen.
3. Postprocesado: Finalmente, se resolverán algunos problemas que hayan podido surgir al generar el resumen.

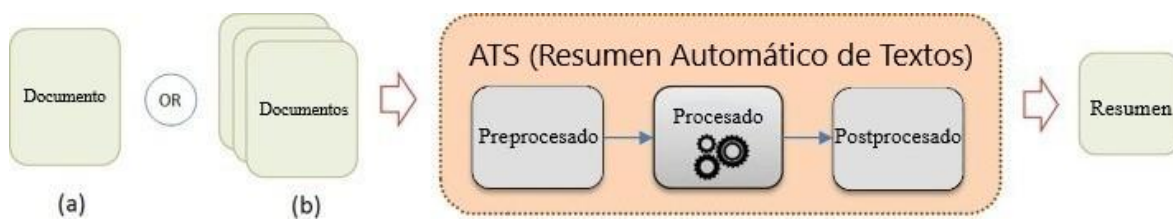


Figura 1. Arquitectura de un sistema ATS

1. JUSTIFICACIÓN

Hay varias razones por las que la aplicación del NLP al resumen de textos ha sido objeto de este trabajo.

El procesamiento del lenguaje natural está en auge, y cada vez son más las investigaciones y aplicaciones desarrollan en este sector. En concreto, los sistemas de resumen automático de textos pueden resultar de gran utilidad en muchos sectores. Como se ha nombrado anteriormente, algunos de los sectores que se pueden beneficiar de estos son: usuarios que quieran optimizar su tiempo a la hora de informarse sobre alguna noticia o algún tema de interés, empresas que deseen generar informes de manera rápida y sin recurrir a sus empleados, científicos y estudiantes para ayudarse en sus investigaciones, médicos con sus informes médicos o periodistas para una búsqueda más optimizada de noticias.

Puede resultar agotador cuando uno se tiene que leer un texto muy largo para sacar información relevante del mismo. Es fácil que, incluso después de haber leído gran parte del texto, uno se olvide de información importante del principio. Por ello, conseguir un resumen de calidad de un texto puede ahorrar mucho tiempo y esfuerzo, por lo que resulta de gran valor práctico.

Por otra parte, el presente trabajo aporta un recorrido histórico por los principales métodos y técnicas utilizados en el campo del ATS, así como una explicación detallada de todos los conceptos y temas a tener en cuenta para entender en profundidad el asunto.

Finalmente, el resultado de este trabajo puede ser aplicado y utilizado por cualquier usuario de manera sencilla y sin necesidad de poseer conocimientos técnicos.

OBJETIVOS

El objetivo de esta investigación es analizar las técnicas actuales de resumen automático de textos e implementar un sistema ATS que utilice métodos y técnicas que se encuentren en el estado del arte actual.

El primer objetivo consistirá en hacer un recorrido histórico por los trabajos e investigaciones más relevantes en el marco de los ATS, centrándose en los más recientes, que conforman el estado del arte del resumen automático de textos. Para ello, se analizarán los métodos y técnicas usados en los trabajos más exitosos.

A continuación, tras haber explicado cuales son estas técnicas SOTA (*state-of-the-art*) y cómo funciona el mecanismo de generación de resúmenes, se aplicará estos algoritmos haciendo uso de modelos previamente entrenados y ajustados para la tarea de resumen. Así se podrá observar los resultados de estos y analizarlos.

Como ya se ha nombrado en puntos anteriores, existen dos principales métodos diferenciados a la hora de crear un resumen: el método extractivo y el método abstractivo. Ambos son formados de manera totalmente distinta, ya que uno utiliza frases y oraciones directamente del texto original, y el otro crea un resumen generando texto nuevo. De esta forma, se analizará una de las principales limitaciones de los modelos abstractivos, los cuales sufren de problemas a la hora de resumir textos largos. Se propondrá una solución para solventar este problema y se enseñará resultados de la misma.

Finalmente, el último objetivo de este trabajo pasa por desarrollar una herramienta que aplique estas técnicas haciendo uso de un algoritmo de resumen de textos con el fin de poner en práctica los conceptos y métodos que se desarrollan en el presente trabajo. Posteriormente, será interesante analizar su funcionamiento y los resultados del mismo dentro del marco actual.

Esta herramienta estará, por supuesto, accesible a cualquier usuario que tenga interés en probarla y utilizarla para generar sus resúmenes de manera automática.

Capítulo I. Material y Metodología

CAPÍTULO I. MATERIAL Y METODOLOGÍA

Para llevar a cabo esta revisión sistemática, se ha llevado a cabo una búsqueda e investigación extensa acerca de la tecnología que trata el trabajo. Se ha revisado los artículos de investigación más exitosos, los cuales han ayudado a dotar a este trabajo de una consistencia y rigor bibliográfico. Para ello, se ha citado todos estos artículos y demás fuentes a las que se ha recurrido y, finalmente, se han enumerado todos y cada uno de ellos en el apartado Bibliografía. Esta bibliografía constará tanto de papers antiguos, del siglo pasado, como del presente siglo, para contener información relevante desde los orígenes de los sistemas de resumen automático hasta los últimos que forman el estado del arte actual. La principal fuente de conocimiento serán los artículos de investigación, sin embargo, también se listarán algunas páginas de internet, donde también reside una gran cantidad de información valiosa.

La estrategia de búsqueda utilizada para obtener los estudios científicos que han sido utilizados ha consistido en la búsqueda en Google de palabras y frases clave. Algunos ejemplos son “Text Summarization NLP”, “Automatic Text Summarization”, “Extractive Text Summarization”, “Abstractive Text Summarization”, etc. La búsqueda ha sido realizada casi completamente en inglés debido a que este idioma ocupa el mayor número de páginas y artículos científicos. Además de Google, la mayor parte de los artículos citados han sido encontrados navegando de artículo en artículo.

Para decidir qué artículos incluir en este trabajo, se ha seguido un criterio de relevancia, donde han sido utilizados los artículos más importantes para la tecnología. Para conocer esta relevancia se puede ver el número de veces que un artículo ha sido citado, así se puede conocer cuales han constituido las bases para las siguientes investigaciones. Algunos artículos han sido descartados debido a la escasa importancia que han supuesto para llegar al actual estado del arte.

Para la parte de la experimentación y posterior desarrollo de la herramienta, se hará uso de diferentes librerías del lenguaje de programación Python. Estas librerías se enumerarán en el apartado Tecnologías para el desarrollo del capítulo II Estado del Arte.

Capítulo II. Estado del Arte

CAPÍTULO II. ESTADO DEL ARTE

El resumen automático de textos o ATS es uno de los desafíos más estudiados a lo largo de la historia en el campo del Procesamiento del Lenguaje Natural (NLP). Las primeras investigaciones se remontan al año 1958 con el trabajo de Luhn (Luhn, 1958), en el cual extrae, de manera automática, resúmenes de artículos de revistas y artículos científicos.

Los sistemas ATS han creado numerosos retos en la comunidad científica:

1. Identificar los segmentos más informativos en el texto de entrada para ser incluidos en el resumen de salida (Radev et al., 2002).
2. Resumir documentos extensos, como los libros.
3. Resumir múltiples documentos (Hahn & Mani, 2000).
4. Generar un resumen abtractivo similar a uno producido por un humano.
5. Evaluar resúmenes generados por un ordenador sin la necesidad del resumen generado por un humano, para compararlos.

Hoy en día, los investigadores todavía sueñan con generar resúmenes con la suficiente precisión para:

- Cubrir los temas más relevantes
- Ser legible y cohesivo
- No incluir datos redundantes o repetidos

Desde los comienzos en los años 50, los investigadores han intentado mejorar las técnicas y los métodos para generar resúmenes de forma automática, de forma que estos se equiparen a los que hacen los humanos. Recientemente han sido publicados muchos estudios acerca de los sistemas ATS. La mayoría se centran en técnicas y métodos de resumen extractivo, como Nazari & Mahdavi (2019), ya que el resumen abtractivo requiere de técnicas de NLP complejas y está lleno de retos. Algunos métodos indirectos de resumen abtractivo emplean técnicas como la fusión de oraciones (Barzilay & McKeown, 2005) o la fusión de frases (Bing et al., 2015). Sin embargo, estas estrategias indirectas de fusión llevan a una construcción pobre de las oraciones.

2. ESTUDIO DE LA PROBLEMÁTICA

Para poner al lector en situación, es necesario hablar de los conceptos que resultan de especial interés en el presente trabajo. Es valioso entender estos conceptos, que son usados en el campo del NLP y en los resúmenes automáticos de textos, y que forman lo que es el actual estado del arte.

3.1. DEEP LEARNING

El Deep Learning (aprendizaje profundo, en español) es un tipo de algoritmo de Machine Learning cuya estructura está formada por redes neuronales artificiales (ANN). Las redes neuronales son un modelo computacional que imita al cerebro humano y que consiste en un conjunto de unidades mínimas, llamadas neuronas artificiales, conectadas entre sí para transmitir información entre ellas (Figura 2). De esta forma, la red neuronal recibe información por sus entradas, la somete a diversas operaciones y produce unos valores de salida («Red neuronal artificial», 2021).

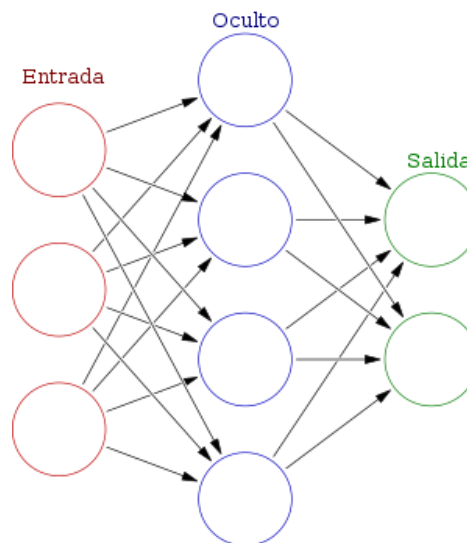


Figura 2. Estructura básica de una red neuronal

REDES NEURONALES RECURRENTE (RNN)

Las redes neuronales recurrentes, o Recurrent Neural Networks (RNN) en inglés, son un tipo de redes neuronales que permiten tratar la dimensión de “tiempo”. Esto es así porque estas redes ya no actúan solo en una dirección, hacia delante, sino que cada neurona incluye una conexión que retroalimenta a la misma, permitiéndole mantener la información a lo largo del tiempo (Torres, 2019).

En la Figura 3 se puede apreciar una neurona de este tipo, donde la neurona recibe una entrada, produce una salida y además envía esa salida a sí misma.



Figura 3. Neurona de una RNN

En cada instante de tiempo (como se puede observar en la Figura 4), la neurona recurrente recibe la entrada de la capa anterior, así como su propia salida del instante de tiempo anterior y, con esto, genera su salida.

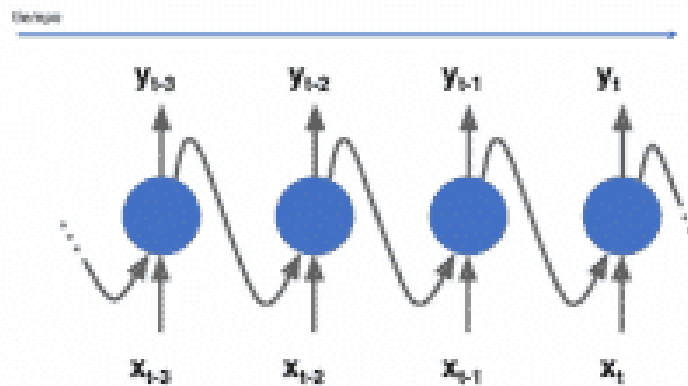


Figura 4. Secuencia temporal de una neurona de RNN

Siguiendo con esta idea, en una capa de neuronas recurrentes, en cada instante de tiempo, cada una de las neuronas recibe dos entradas, la de la capa anterior y la del instante anterior de su misma capa.

LSTM

LSTM (*Long short-term memory*) es una arquitectura de red neuronal recurrente (RNN) que amplía su memoria para aprender experiencias relevantes de hace mucho tiempo. Esta arquitectura está diseñada de manera que es capaz de recordar información durante un largo periodo de tiempo. Además, puede determinar qué palabra de entrada puede ser olvidada y qué palabras deben mantenerse en la memoria, siendo así útil para la tarea de resumen de textos.

2.2. WORD EMBEDDINGS

En el procesamiento del lenguaje natural, la unidad mínima de una frase son las palabras y símbolos que forman las frases. Estas palabras se denominan *tokens*, por lo que cada palabra o signo de puntuación es un *token*.

Los *word embeddings* son un enfoque del NLP que representa las palabras como vectores de números reales. Esta representación de la distribución de las palabras en un espacio vectorial ayuda a los algoritmos de Machine Learning a obtener mejores resultados en tareas de NLP, ya que agrupa las palabras sintáctica y semánticamente similares (Mikolov et al., 2013). Aunque las representaciones son complejas, la Figura 5 se puede observar una representación simple de las relaciones entre palabras similares.

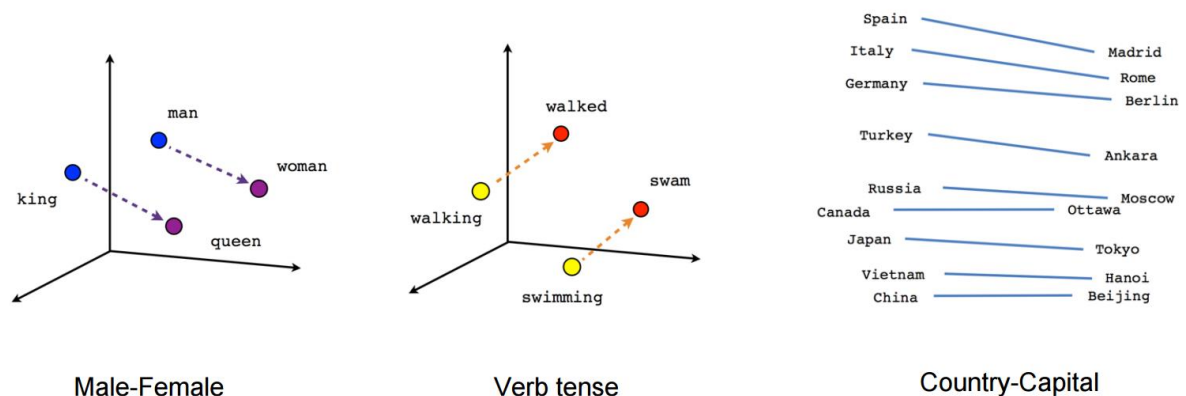


Figura 5. Representación de word embeddings

2.3. MODELOS DE ATENCIÓN

Ante la persistencia de los problemas de interdependencia de palabras alejadas entre sí, surgieron los modelos de atención. En la representación del lenguaje, técnicas como LSTM no son capaces de comprender las relaciones entre palabras distanciadas.

La técnica de los modelos de atención consiste en almacenar los contextos que se producen en cada iteración del codificador o *encoder* para, seguidamente, transferir estos contextos al decodificador o *decoder*. Así, se consigue que el modelo aprenda a prestar “atención” a los tramos de la secuencia de entrada que más le convenga.

2.4. TRANSFORMERS

Con el crecimiento del Deep Learning en la última década, así como la irrupción de los *word embeddings*, las técnicas de NLP no han hecho más que avanzar. Así, hasta finales del año 2017 los modelos basados en RNN que hacían uso de vectores o *embeddings* se convirtieron en el estándar ya que eran capaces de tener en cuenta tanto el significado general de cada palabra como la posición de las mismas en la frase (Vaca, 2020).

A finales del año 2017, Google presentó la arquitectura del Transformer, un modelo que sustituía las capas recurrentes, como las LSTM, por las denominadas capas de atención (Vaswani et al., 2017). Estas capas de atención codifican cada palabra de una frase teniendo en cuenta el resto de la frase, por lo que se introdujo el contexto en la representación matemática del texto. Por este motivo, los *transformers* también son denominados Embeddings Contextuales.

La arquitectura de un *transformer* se puede ver en la Figura 6.

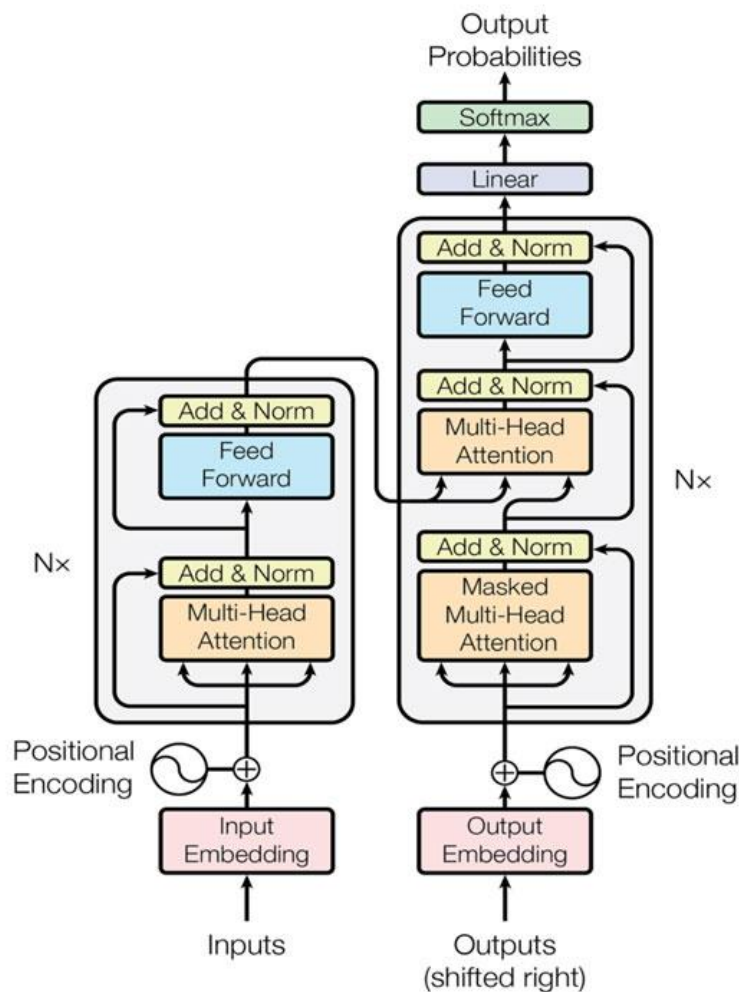


Figura 6. Arquitectura de un transformer

A más alto nivel, el funcionamiento de un *transformer*, considerando este como una caja negra, se puede representar como en la Figura 7 (Alammar, 2018).



Figura 7. Ejemplo de funcionamiento de un transformer en una tarea de traducción

El flujo por el *encoder-decoder* se muestra en la Figura 8.

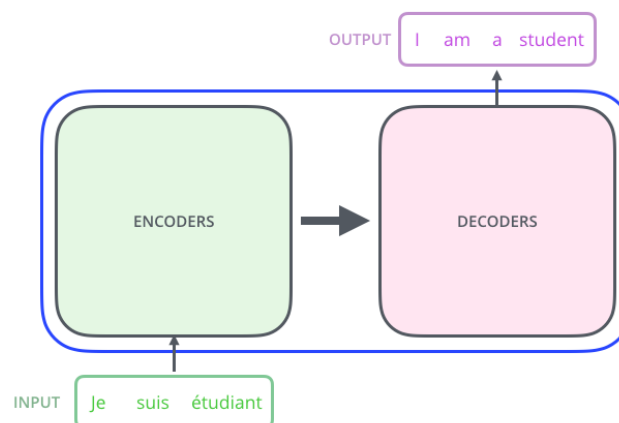


Figura 8. Descomposición de un transformer en una estructura de encoder-decoder

Realmente, el *encoder* es una pila de seis *encoders*, igual que el *decoder* tiene seis *decoders* (ver Figura 9).

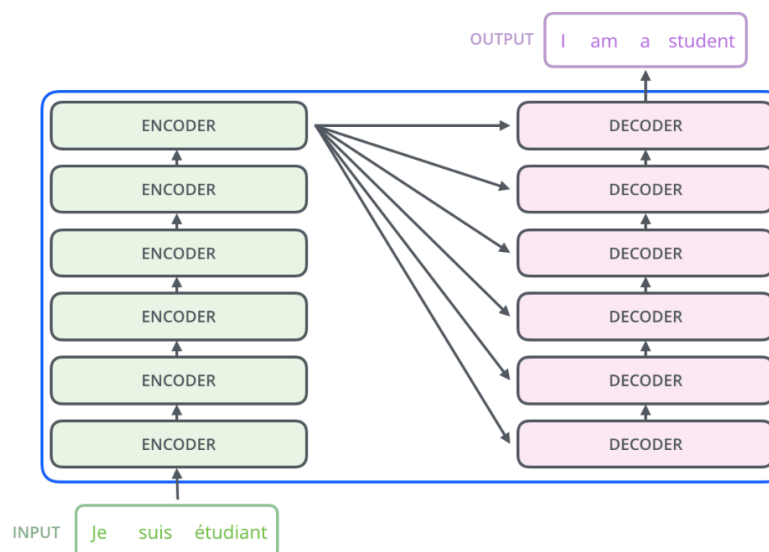


Figura 9. Descomposición de encoder y decoder en 6 partes

Todos los *encoders* tienen la misma estructura, aunque no comparten pesos. Cada *encoder* está formado por dos capas (Figura 10).

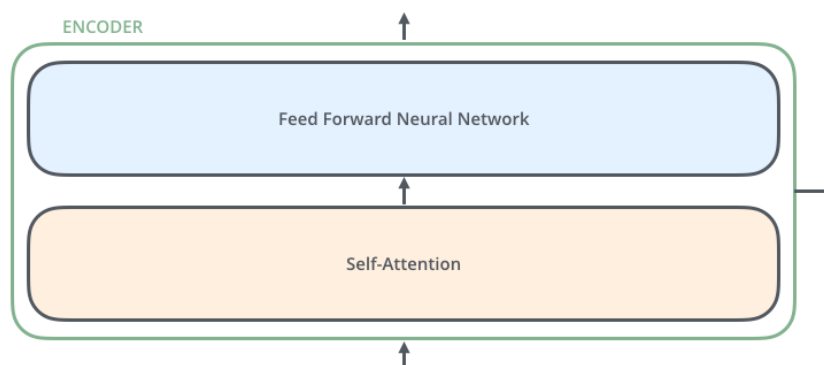


Figura 10. Capas de un encoder

Las entradas a los *encoders* pasan por una capa de auto-atención (*self-attention*), la cual ayuda al *encoder* a mirar hacia otras posiciones de la secuencia de palabras de entrada para entender el contexto de cada palabra, para que el *encoder* pueda codificar mejor las palabras (p.e. para la palabra “eso”, la capa de auto-atención le ayuda a identificar a qué sustantivo se refiere).

Las salidas de la capa de auto-atención son pasadas como entradas a una red neuronal feed-forward (propagación hacia delante).

El *decoder* tiene las mismas dos capas que el *encoder*, pero entre ellas hay una capa de atención (*encoder-decoder attention*) que ayuda al *decoder* a enfocarse en las partes relevantes de la frase de entrada (ver Figura 11).

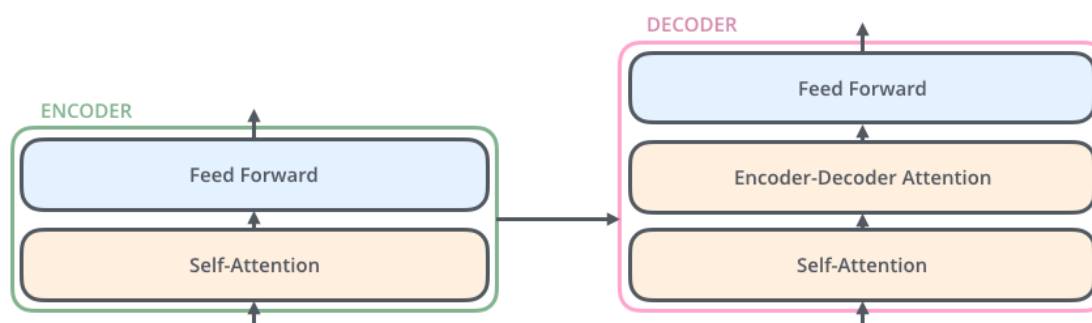


Figura 11. Diferencia entre las capas de un encoder y las de un decoder

El sistema de trabajo con los *transformers* consta de dos fases:

1. **Pre-training:** En la fase de preentrenamiento, el modelo aprende la estructura del lenguaje, así como el significado de las palabras. Este entrenamiento se lleva a cabo de manera no supervisada
2. **Fine-tuning:** Después del preentrenamiento, se le debe añadir algunas capas a la arquitectura, para adaptar el modelo a la tarea concreta que se desee llevar a cabo. Tras esto, se vuelve a entrenar el modelo.

Algunas de las aplicaciones donde los *transformers* han conseguido resultados “estado del arte” son: resumen de textos, generación de texto, clasificación de documentos, sistemas de respuesta a preguntas, identificación de entidades, etc.

El gran crecimiento de estos modelos ha sido posible gracias a los ingentes volúmenes de texto en diversos idiomas que se pueden encontrar en internet, así como el aumento de la capacidad de cómputo de los ordenadores.

2.5. EVALUACIÓN

La evaluación es crucial en la investigación y el desarrollo de sistemas ATS, ya que es necesario establecer un criterio para decidir si un resumen es útil para el contexto o la aplicación para el que es generado. Para ello, los investigadores deben conocer las métricas de evaluación, los enfoques y los datasets existentes, para decidir cuáles se ajustan más a su propósito (Lloret et al., 2018).

En los resúmenes generados por sistemas ATS existe cierta dificultad a la hora de evaluarlos. Esto se debe a que es complicado definir qué hace que un resumen sea bueno o malo. Para evaluar la eficacia de un resumen de manera automática, Lyn y Hovy establecieron dos pautas generales que debe tener un buen resumen (Lin & Hovy, 2003):

- Debe ser más breve que el texto original.
- Debe incluir la información más relevante.

Cuando se tiene un resumen creado por un humano, se puede utilizar este como una referencia para compararlo con el resumen creado por el ordenador. En este caso, se puede utilizar la métrica ROUGE. Para usarla, se debe dividir ambos resúmenes en n-gramas, es decir, cadenas de n elementos. Así, en el caso de los 1-gramas (o unigramas), se compararía las palabras individualmente o, en los 2-gramas (o bigramas) se compararían pares de palabras (en una frase de 4 palabras hay tres bigramas).

En la frase “El gato negro se llama Tobi” se tiene:

Unigramas: ['El', 'gato', 'negro', 'se', 'llama', 'Tobi']

Bigramas: ['El gato', 'gato negro', 'negro se', 'se llama', 'llama Tobi']

Trigramas: ['El gato negro', 'gato negro se', 'negro se llama', 'se llama Tobi']

RECALL

La métrica *recall* cuenta el número de n-gramas coincidentes en el resumen producido y el de referencia, y lo divide entre el número total de n-gramas que hay en el de referencia (Figura 12).

$$recall = \frac{n^{\circ} \text{ de ngramas coincidentes}}{n^{\circ} \text{ de ngramas de la referencia}}$$

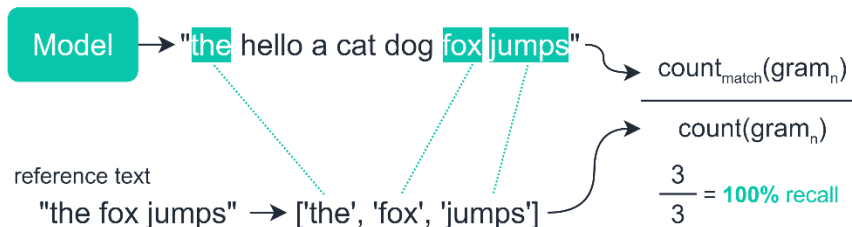


Figura 12. Ejemplo de cálculo del recall (Fuente: Briggs, 2021)

PRECISIÓN

La precisión se calcula de la misma manera, pero en lugar de dividir entre el número de n-gramas del resumen de referencia, se divide entre el número de n-gramas del resumen producido (modelo)(ver Figura 13).

$$precision = \frac{n^{\circ} \text{ de ngramas coincidentes}}{n^{\circ} \text{ de ngramas del modelo}}$$

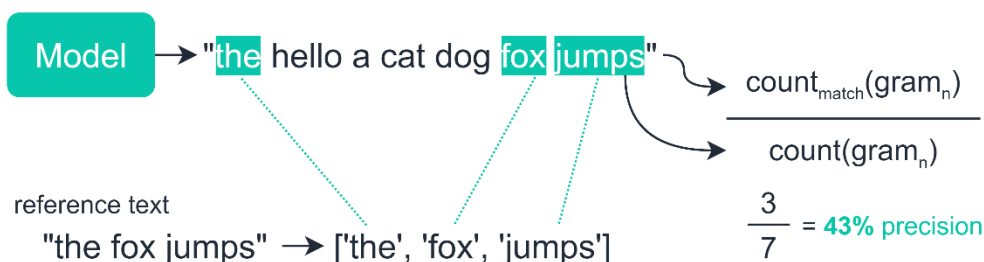


Figura 13. Ejemplo de cálculo de la precisión

F1-SCORE

Para calcular la puntuación F1 se utilizan los resultados de las dos métricas anteriores (Figura 14).

$$F1\ score = 2 * \frac{precision * recall}{precision + recall}$$



$$2 * \frac{0.43 * 1.0}{0.43 + 1.0} = 0.6 \quad \text{60\% f1 score}$$

Figura 14. Ejemplo de cálculo del F1-score

ROUGE

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) consta de un conjunto de métricas y un paquete de software usados para evaluar los resúmenes y las traducciones automáticos en NLP. Estas métricas comparan el resumen o la traducción con un conjunto de resúmenes o traducciones de referencia («ROUGE (Metric)», 2019). Las más comunes son:

- **ROUGE-1:** Calcula el número de unigramas (cada palabra) coincidentes entre el resumen producido y los resúmenes de referencia.
- **ROUGE-2:** Calcula el número de bigramas (cada 2 palabras) coincidentes entre el resumen producido y los de referencia.
- **ROUGE-L:** Calcula la subsecuencia común más larga o LCS (*Longest Common Subsequence*) entre el resumen producido por el modelo y el de referencia. LCS tiene en cuenta la similitud estructural de las frases e identifica los n-gramas coincidentes más largos de manera automática (ver Figura 15)

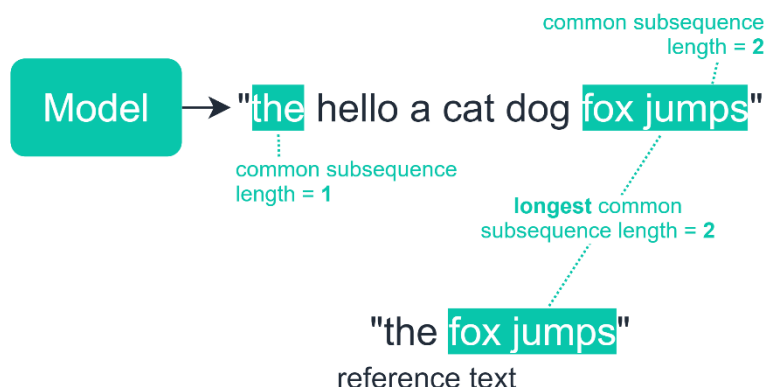


Figura 15. Ejemplo de la obtención de la secuencia común más larga (LCS)

Para aplicar esta métrica, se puede utilizar las métricas anteriores de precisión y recall para utilizar, en lugar del número de n-gramas coincidentes, la longitud de n-grama calculada por LCS.

A pesar de que ROUGE es la métrica de evaluación más extendida para las aplicaciones de resumen y traducción de texto, tiene algunas desventajas. ROUGE es capaz de identificar sinónimos, debido a que hace un análisis sintáctico en lugar de semántico. Por ello, si en el resumen se tiene la misma secuencia pero usando palabras diferentes para expresar lo mismo, se le asignaría una puntuación ROUGE baja.

3. ANTECEDENTES

Los resúmenes han sido una de las aplicaciones más desafiantes del NLP y, por ello, han sido objeto de numerosas investigaciones en este campo. En los siguientes apartados se va a presentar las principales técnicas y modelos utilizados, tanto para el resumen extractivo como el abstractivo. Así, se va a clasificar estos algoritmos en estos dos grupos, ya que necesitan de una diferenciación importante debido a la naturaleza de las técnicas que se han utilizado en la investigación de cada grupo.

3.1. MÉTODOS EXTRACTIVOS

Los métodos de resumen extractivo generan los resúmenes concatenando varias oraciones del texto original. La principal tarea de estos sistemas consiste en determinar qué oraciones son importantes y deben, por ello, ser incluidas en el resumen.

Durante muchos años, los métodos extractivos han sido el principal enfoque de los investigadores de la comunidad dedicada al resumen de textos. Muchos de los enfoques consistían en una tarea de etiquetado de oraciones, donde cada etiqueta indicaba si la oración debía ser incluida en el resumen o no. Cheng et al. (Cheng & Lapata, 2016) presentaron un *framework* de resumen basado en datos que consistía en redes neuronales. Este *framework* era capaz de resumir documentos individuales y estaba basado en un *encoder* de documentos jerárquico y un extractor basado en la atención.

TEXTRANK

TextRank es un modelo basado en grafos para procesado de texto. Se basa en el concepto de que las palabras que aparecen frecuentemente son importantes. De esta manera, el algoritmo asigna puntuaciones a cada oración e incorpora las oraciones con la puntuación más alta, en el resumen (Mihalcea & Tarau, 2004).

LUHN

El algoritmo de resumen Luhn debe su nombre al conocido Hans Peter Luhn, pionero de los sistemas ATS (Luhn, 1958). Este algoritmo está basado en la técnica TF-IDF (*Term frequency – Inverse document frequency*), frecuencia de término – frecuencia inversa de documento, es decir, la frecuencia de ocurrencia del término en la colección de documentos. Este algoritmo es útil cuando, tanto palabras muy frecuentes (*stopwords*) como poco frecuentes, no son significativas. De esta forma, se dan puntuaciones y las frases más puntuadas se ponen en el resumen.

LEXRANK

Con este algoritmo, cuando una frase es similar a otras, es muy probable de ser considerada importante. El enfoque de LexRank considera que las oraciones que son parecidas deben tener una puntuación más alta. Cuanto más alta es la puntuación, más probabilidades tiene de ser incluida en el resumen (Erkan & Radev, 2004).

LATENT SEMANTIC ANALYSIS (LSA)

LSA es un algoritmo de aprendizaje no supervisado, que puede ser usado para el resumen extractivo de textos (Steinberger & Ježek, 2004). Su funcionamiento se basa en extraer oraciones semánticamente relevantes, aplicando la descomposición en valores singulares (SVD) a la matriz de frecuencia de término en el documento (*term-document frequency*).

KL-SUM

Por último, se va a hablar del algoritmo KL-Sum. Este, selecciona las frases basándose en la divergencia de las palabras del vocabulario del texto de entrada, minimizando este vocabulario (Haghighi & Vanderwende, 2009). Su objetivo es disminuir la divergencia de Kullback-Leibler, la cual cuantifica cuánto difiere una distribución de probabilidad de otra.

3.2. MÉTODOS ABSTRACTIVOS

Los métodos abstractivos son aquellos que, para la creación del resumen, hacen uso de palabras y frases diferentes a las del texto original. Esto requiere de una componente de creatividad y originalidad que no se consigue aplicando las técnicas extractivas.

La última década ha supuesto un punto de inflexión para los métodos abstractivos, debido a la gran cantidad de algoritmos y modelos creados, los cuales han arrojado resultados sorprendentes.

T5

T5 es un modelo de *encoder-decoder*, preentrenado sobre una mezcla de tareas supervisadas y no supervisadas, donde cada tarea se convierte a un formato texto a texto (Raffel et al., 2020).

BERT

El archiconocido *Bidirectional Encoder Representations from Transformers*, o mejor conocido por sus siglas BERT fue publicado en 2018 por Jacob Devlin, Ming-Wei Chang, Kenton Lee y Kristina Toutanova, de Google. Es una representación de lenguaje bidireccional, es decir, tiene en cuenta las palabras a cada lado de los términos (Devlin et al., 2019).

BERT fue preentrenado usando una combinación del modelo de lenguaje de máscara (MLM, *Masked Language Modeling*) y la predicción de la siguiente frase (NSP, *Next Sentence Prediction*), en un corpus formado por la Wikipedia y el Toronto Book Corpus.

Este modelo es eficiente prediciendo *tokens* enmascarados, así como en la comprensión del lenguaje natural (NLU) en general. Sin embargo, no está optimizado para la generación de texto.

Resulta muy relevante la inclusión de este modelo en el estado del arte por su influencia en modelos posteriores.

BART

El modelo BART (Lewis et al., 2019) usa una arquitectura con un *encoder* bidireccional (como BERT) y un *decoder* con flujo de izquierda a derecha (como GPT). La tarea de preentrenamiento implica cambiar aleatoriamente el orden de las oraciones del texto original.

BART es especialmente efectivo en tareas que implican la generación de texto, aunque también en tareas que impliquen la comprensión del texto. Este modelo ha conseguido posicionarse como uno de los modelos que mejores resultados consigue en diálogo abstractivo, respuesta a preguntas y tareas de resumen.

PEGASUS

PEGASUS (Zhang et al., 2020) utiliza la técnica de enmascarar las oraciones relevantes del texto original y entrenar el modelo para que aprenda a generarlas a partir de las demás frases (*gap-sentence generation*). De esta manera, el resumen de salida consistirá en la secuencia de las frases enmascaradas que deberán ser generadas por el algoritmo, por lo que esta salida será similar a la de un resumen extractivo. Aplicando esta técnica, han conseguido resultados “estado del arte” en tareas de resumen abstractivo en 12 datasets de resumen diferentes, usando como referencia de evaluación las puntuaciones ROUGE.

Este algoritmo también ha sido probado en 6 datasets con solo 1000 ejemplos de resumen, en los cuales ha conseguido resultados sorprendentes, demostrando así buen desempeño en datasets escasos.

Finalmente, los resultados de los resúmenes de PEGASUS han sido evaluados por humanos y han demostrado un desempeño igual al de los humanos en múltiples datasets.

LONGFORMER

Los modelos basados en *transformers* son incapaces de procesar largas secuencias de texto debido a la operación de auto-atención (*self-attention*), la cual aumenta cuadráticamente con respecto a la longitud de secuencia. Para hacer frente a esta limitación se creó el Longformer, que cuenta con un mecanismo de atención que aumenta linealmente con respecto a la longitud de secuencia, haciendo posible el procesamiento de documentos con miles de *tokens* de manera sencilla (Beltagy et al., 2020).

Recientemente se introdujo el Longformer-Encoder-Decoder (LED), una variante del Longformer que soporta tareas generativas secuencia-a-secuencia en documentos largos. Su efectividad ha sido demostrada en el dataset de resumen “arXiv”, consiguiendo resultados “estado del arte”.

4. TECNOLOGÍAS PARA EL DESARROLLO

En este apartado se hará una descripción de la tecnología empleada para el desarrollo del sistema de resumen de textos del apartado Experimentación. Esta tecnología está formada por las diversas librerías del lenguaje de programación Python, el cual es el lenguaje de referencia en el ámbito del Machine Learning y la Ciencia de Datos.

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Al ser de código abierto, cuenta con numerosos desarrolladores que están detrás de la creación de las distintas librerías que se han utilizado en este trabajo, las cuales se describirán a continuación.

4.1. LIBRERÍAS PYTHON

NUMPY

NumPy es una librería creada para la computación científica en Python. Con ella se puede crear *arrays* multidimensionales (vectores y matrices) y ofrece una variedad de funciones para hacer operaciones sobre los *arrays*: operaciones matemáticas, lógicas, manipular la forma del array, álgebra lineal básico, operaciones estadísticas, etc. Es muy eficiente y soporta grandes volúmenes de datos.

Por su versatilidad, son muchas librerías las que implementan NumPy internamente, por lo que existe una gran compatibilidad a la hora de usar objetos NumPy con otras librerías.

NLTK

NLTK o *Natural Language Toolkit*, es una plataforma para crear programas con Python para trabajar con el lenguaje humano. Por ello, es fundamental en cualquier aplicación de NLP. Ofrece interfaces para trabajar con una multitud de recursos léxicos como *WordNet*, además de contar con soporte para tareas de procesado de texto, tokenización, *stemming*, lematización, etiquetado, etc.

TENSORFLOW

TensorFlow es una plataforma de código abierto, creada por Google, que permite desarrollar y entrenar modelos de Machine Learning. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que permite a los desarrolladores implementar modelos de Machine Learning.

Debido a la gran cantidad de cálculos que realizan los modelos de Deep Learning durante el entrenamiento, Tensorflow es capaz de ejecutarse sobre GPUs, TPUs, así como de manera distribuida.

KERAS

Keras es una API creada por encima de TensorFlow 2.0, compatible al 100% con la misma, que facilita y agiliza la construcción de modelos de Deep Learning con arquitecturas comunes. Además, ofrece funcionalidades para el entrenamiento y evaluación de dichos modelos.

PYTORCH

PyTorch es un *framework* de código abierto enfocado al Machine Learning que promete acelerar el camino entre la creación de prototipos en la fase de investigación hasta la puesta en producción.

Entre sus principales características están que es un framework “*production ready*” y cuenta con un entrenamiento distribuido, un ecosistema robusto y soporte *cloud*.

TRANSFORMERS

Existe una comunidad enorme de desarrolladores, llamada Hugging Face, que se ha encargado de desarrollar la librería *transformers*. Esta comunidad está formada por una multitud de desarrolladores, que mantienen y aumentan sus funcionalidades, con un modelo open source. Esta librería permite implementar modelos de Machine Learning para crear aplicaciones de resumen, clasificación, etc. Algunos de los modelos son GPT-2, GPT-3, BERT, OpenAI, T5.

La librería *transformers* (antes conocida como *pytorch-transformers* y *pytorch-pretrained-bert*) forma el estado del arte del procesamiento del lenguaje natural. Proporciona arquitecturas de propósito general (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet...) para el NLU (*Natural Language Understanding*) y el NLG (*Natural Language Generation*).

Cuenta con más de 32 modelos preentrenados en más de 100 idiomas, y ofrece una gran interoperabilidad entre TensorFlow 2.0 y PyTorch.

Algunas de sus características es que ofrece un gran rendimiento en tareas de NLU y NLG y ofrece el estado del arte del NLP a todo el mundo, ya que la utilizan investigadores de Deep Learning, practicantes e incluso profesores de inteligencia artificial, Machine Learning y NLP.

Una gran ventaja es que los investigadores pueden compartir los modelos que han entrenado. De esta manera, se ahorran tener que reentrenarlos cada vez.

GENSIM

Gensim es una librería de Python diseñada para aplicar NLP sobre grandes cantidades de datos en formato texto. Esta librería permite crear *word embeddings*, para obtener representaciones de palabras con información sintáctica y semántica.

SUMY

Sumy es una librería de Python que permite crear resúmenes de manera sencilla a partir de páginas HTML o texto plano. Incluye, además, un *framework* para evaluar de manera sencilla estos resúmenes.

Los métodos de resumen que implementa son: Luhn, Edmundson, Latent Semantic Analysis (LSA), LexRank, TextRank, SumBasic, KL-Sum y Reduction.

STREAMLIT

Streamlit es una librería *open-source* de Python que facilita la creación de aplicaciones web para Machine Learning y la ciencia de datos. Con esta herramienta se puede incorporar *widgets* interactivos y personalizar la aplicación de manera sencilla.

BEAUTIFULSOUP

BeautifulSoup constituye una herramienta para extraer datos de archivos HTML y XML. Esta librería de Python navega, busca y modifica la estructura de estos archivos. Es una de las maneras más fáciles de realizar *scraping* de cualquier página web.

4.2. ENTORNO DE TRABAJO

El entorno sobre el que se ha trabajado para la implementación de los diferentes modelos y técnicas de resumen ha sido Jupyter Notebook.

Para desarrollar la aplicación web ha sido necesario usar un IDE de desarrollo. En el caso de este trabajo se ha usado PyCharm.

Finalmente, para desplegar la herramienta desarrollada en este trabajo y ponerla en producción se ha hecho uso de GitHub. Ha sido necesaria la creación de un repositorio para desplegar la página e integrar el proyecto de Streamlit en el mismo.

JUPYTER

El proyecto Jupyter (cuyas letras vienen de la composición de los nombres de algunos de los lenguajes de programación compatibles con el mismo: Julia-Python-R) se creó para poder implementar software *open-source*, estándares abiertos y servicios para la computación interactiva entre múltiples lenguajes de programación. Uno de los módulos de Jupyter es Jupyter Notebook.

Jupyter Notebook es una aplicación web de código abierto que permite al usuario crear y compartir documentos, en directo, que incluyen código, ecuaciones, visualizaciones y texto narrativo. Es la aplicación de referencia para la programación en la ciencia de datos y el Machine Learning.

PYCHARM

PyCharm es uno de los entornos de desarrollo más usados y completos para programar en Python. Forma parte del conjunto de herramientas de la empresa checa JetBrains, antes conocida como IntelliJ. Este IDE proporciona análisis del código, un *debugger* gráfico, testing unitario, integración con sistemas de control de versiones como Git, etc.

GITHUB

GitHub es una plataforma de desarrollo colaborativo, que permite alojar proyectos haciendo uso del sistema de control de versiones Git. Esta plataforma se utiliza principalmente para crear código fuente de programas. GitHub se fundó en los Estados Unidos y fue adquirida en 2018 por Microsoft. GitHub constituye la plataforma de colaboración más importante para proyectos *open-source*.

Capítulo III.

Experimentación

CAPÍTULO III. EXPERIMENTACIÓN

6. ENFOQUES

En este apartado se experimentará con diferentes técnicas y enfoques para hacer un resumen tomando un texto de entrada. Los enfoques utilizados son variados y corresponden con diferentes métodos utilizados a lo largo de los últimos años. Este apartado resulta interesante para ver cómo actúa cada algoritmo a la hora de generar el resumen y, por ello, se analizará el resultado de cada uno de ellos.

5.1. TEXTO

El texto que se utilizará para experimentar con las diferentes técnicas de resumen será la página de Inteligencia Artificial de Wikipedia («Artificial Intelligence», 2021), donde habla de manera muy completa de la inteligencia artificial, abarcando desde la historia hasta las herramientas, aplicaciones, desafíos, regulaciones, filosofía o ética.

Para utilizar el texto, primero es necesario extraerlo de la página deseada, en este caso Wikipedia. Para ello se ha creado una función en Python para hacer *scraping* web de la página deseada utilizando las librerías *urllib.request* para abrir la página deseada y *BeautifulSoup* para extraer el texto de la misma.

A pesar de que algunas técnicas de ATS tienen compatibilidad con el idioma español, la mayoría de ellos han sido creados especialmente para el idioma inglés, por lo que resulta más interesante analizarlo en este idioma, ya que arrojarán los mejores resultados y se podrá obtener su máximo potencial.

PREPROCESAMIENTO

El texto que se desea resumir, antes de pasarlo como entrada al algoritmo de resumen, tiene que pasar por una fase de preprocesamiento. En el NLP, el procesamiento constituye una fase esencial para cualquier aplicación. En esta fase se prepara el texto para una aplicación determinada, de modo que este debe convertirse a un formato analizable por la aplicación.

El preprocesamiento para una aplicación concreta de NLP puede convertirse en una pesadilla para otra tarea, por lo que el preprocesamiento no es transferible de una tarea a otra.

En el caso que ocupa a este trabajo, el preprocesamiento del texto inicial ha consistido en una limpieza del mismo. Se ha borrado, haciendo uso de la librería *re* con su función *re.sub()*, los números o letras entre corchetes que tanto aparecen en los artículos de Wikipedia, o los espacios en blanco redundantes. De esta manera, haciendo uso de las famosas *regex* o expresiones regulares (*regular expressions*, en inglés), se consigue filtrar los caracteres y cadenas no deseados.

En el caso de los algoritmos extractivos, no es necesario preprocesar el texto más allá de una limpieza general, ya que los algoritmos que se utilizarán de las librerías *Sumy* y *Gensim* aceptan como entrada el texto en bruto y deben devolver en el resumen las frases tal como están escritas en el texto original.

Para el caso de los algoritmos de resumen abstractivos que se verán a continuación, al estar todos implementados con la librería *transformers*, son preprocesados de una misma manera. Esta consiste en lo que se denomina “Tokenizer” o tokenizador en español. Los *tokenizers* de *transformers* se pueden crear usando la clase del tokenizador asociada al modelo que se desea usar (p.e. *PegasusTokenizer* para los modelos de “google/pegasus”).

El tokenizador dividirá el texto en palabras (o signos de puntuación), también llamados *tokens*. Tras esto, convertirá estos tokens en números para poder crear un tensor y pasárselo al modelo.

Utilizar el tokenizador asociado a cada modelo es importante para usar el mismo vocabulario que el que se usó para entrenar el modelo.

POSTPROCESAMIENTO

Al igual que con el texto de entrada al algoritmo, el texto de salida también debe pasar por una fase de procesamiento. Esto es importante ya que algunos algoritmos, al generar el resumen, introducen algunos caracteres no deseados. Algunos ejemplos son “<pad>”, “<n>” o “</s>”.

De la misma manera que en la fase de preprocesamiento, se han utilizado técnicas de limpieza haciendo uso de *regex*. El resultado de este filtrado formará el resumen.

5.2. DATASETS

DUC2001

El dataset *Document Understanding Conference* (DUC2001) consiste en un total de 303 documentos que contienen noticias de periódicos, agrupadas en 30 categorías.

CNN / DAILYMMAIL

El *CNN / Dailymail Dataset* es un dataset en inglés que contiene más de 300.000 artículos de noticias, escritos por periodistas del CNN y del Daily Mail. La versión actual soporta tanto resúmenes extractivos como abstractivos, sin embargo, originalmente fue creado para lectura y comprensión de máquina, así como respuesta a preguntas de manera abstractiva.

La media de tokens por artículo se sitúa en 781 y de los resúmenes en 56 tokens.

Este dataset cuenta con tres campos:

- Id: Contiene una cadena de texto con el hash SHA1 de la URL del artículo periodístico.
- Artículo: Contiene una cadena de texto con el cuerpo del artículo.
- Entrada: Contiene una cadena de texto con la entrada o primer párrafo del artículo, que corresponde a un resumen del cuerpo.

5.3. TÉCNICAS EXTRACTIVAS

A continuación, se implementará diferentes técnicas de resumen extractivo, en las que la salida consistirá en la concatenación de las oraciones más relevantes del texto original. No están ordenadas por importancia sino según su posición en el texto.

GENSIM

La librería Gensim, de Python, incorpora un módulo para resumir texto basado en el algoritmo TextRank.

TEXTRANK

Se va a demostrar su funcionamiento mediante un ejemplo. Importando el módulo *summarize* de la librería *gensim.summarization* y pasándole el texto original a este módulo, este devolverá el resumen aplicando la técnica anteriormente descrita.

Ajustando los parámetros de la función *summarize*, se puede cambiar el tamaño deseado del resumen. Así, se puede indicar el ratio (0-1) del resumen con respecto al texto original, o el número de palabras que se desea.

Resumen (200 palabras):

Nota: Las que están marcadas con un asterisco [] son las 3 frases más importantes según el algoritmo.*

*Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations.

*The traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects.

*Many researchers predict that such "narrow AI" work in different individual domains will eventually be incorporated into a machine with artificial general intelligence (AGI), combining most of the narrow skills mentioned in this article and at some point even exceeding human ability in most or all these areas.

Economist Herbert Simon and Allen Newell studied human problem-solving skills and attempted to formalize them, and their work laid the foundations of the field of artificial intelligence, as well as cognitive science, operations research and management science.

Researchers at MIT (such as Marvin Minsky and Seymour Papert) found that solving difficult problems in vision and natural language processing required ad hoc solutions—they argued that no simple and general principle (like logic) would capture all the aspects of intelligent behavior.

Una utilidad muy interesante que ofrece la librería Gensim con su algoritmo TextRank, es la función *keywords()* que, como su nombre indica, devuelve las N palabras o combinaciones de palabras más puntuadas. En el caso del texto de la Wikipedia, estas son las 15 palabras con la puntuación más alta, una vez aplicada la lematización:

1. Humanity
2. Researching
3. Intelligent
4. Machine
5. Includes
6. Learnings
7. Compute
8. Problems
9. Knowledge
10. Approach
11. Artificial
12. Generating
13. Systems
14. Robot
15. Logical

SUMY

Existe una librería de Python llamada Sumy, la cual es realmente útil ya que incorpora múltiples algoritmos para resumir texto. Además, al instanciar las funciones de dichos algoritmos se puede ajustar parámetros como el número de frases que debe contener el resumen. En este caso se ha establecido una longitud de 3 y 5 frases para los algoritmos de Sumy.

Algunos de los algoritmos que se probarán a continuación son los siguientes:

- Luhn
- LexRank
- Latent Semantic Analysis (LSA)
- KL-Sum

LUHN

Se empezará por el algoritmo de resumen Luhn. Este algoritmo es útil cuando, tanto palabras muy frecuentes (*stopwords*) como poco frecuentes, no son significativas. De esta forma, se dan puntuaciones y las frases más puntuadas se ponen en el resumen.

El módulo utilizado deberá ser *LuhnSummarizer* de la librería *sumy.summerizers.luhn*.

Resumen (5 frases):

Nota: Las que están marcadas con un asterisco [] son las 3 frases más importantes según el algoritmo.*

In the twenty-first century, AI techniques have experienced a resurgence following concurrent advances in computer power, large amounts of data, and theoretical understanding; and AI techniques have become an essential part of the technology industry, helping to solve many challenging problems in computer science, software engineering and operations research.

*A simple example of an algorithm is the following (optimal for first player) recipe for play at tic-tac-toe: Many AI algorithms are capable of learning from data; they can enhance themselves by learning new heuristics (strategies, or "rules of thumb", that have worked well in the past), or can themselves write other algorithms.

*Moravec's paradox generalizes that low-level sensorimotor skills that humans take for granted are, counterintuitively, difficult to program into a robot; the paradox is named after

Hans Moravec, who stated in 1988 that "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility".

Many researchers predict that such "narrow AI" work in different individual domains will eventually be incorporated into a machine with artificial general intelligence (AGI), combining most of the narrow skills mentioned in this article and at some point even exceeding human ability in most or all these areas.

*Wendell Wallach introduced the concept of artificial moral agents (AMA) in his book *Moral Machines*. For Wallach, AMAs have become a part of the research landscape of artificial intelligence as guided by its two central questions which he identifies as "Does Humanity Want Computers Making Moral Decisions" and "Can (Ro)bots Really Be Moral".

LEXRANK

El enfoque de LexRank considera que las oraciones que son parecidas deben tener una puntuación más alta. Cuanto más alta es la puntuación, más probabilidades tiene de ser incluida en el resumen.

Para implementar el algoritmo, se debe usar la función *LexRankSummarizer* de la librería *sumy.summarizers.lex_rank*.

Resumen (5 frases):

Nota: Las que están marcadas con un asterisco [] son las 3 frases más importantes según el algoritmo.*

Among the most difficult problems in knowledge representation are: Intelligent agents must be able to set goals and achieve them.

*Many of the problems in this article may also require general intelligence, if machines are to solve the problems as well as people do.

A number of researchers began to look into "sub-symbolic" approaches to specific AI problems.

*Computationalism is the position in the philosophy of mind that the human mind or the human brain (or both) is an information processing system and that thinking is a form of computing.

*If a machine can be created that has intelligence, could it also feel?

LATENT SEMANTIC ANALYSIS (LSA)

LSA es un algoritmo de aprendizaje no supervisado. Para demostrar su funcionamiento, se pasará el texto a la función *LsaSummarizer* de la librería *sumy.summarizers.lsa*.

Resumen (5 frases):

Nota: Las que están marcadas con un asterisco [] son las 3 frases más importantes según el algoritmo.*

Deep Blue's Murray Campbell called AlphaGo's victory "the end of an era... board games are more or less done and it's time to move on."

They can be nuanced, such as "X% of families have geographically separate species with color variants, so there is a Y% chance that undiscovered black swans exist".

*Commonsense knowledge bases (such as Doug Lenat's Cyc) are an example of "scruffy" AI, since they must be built by hand, one complicated concept at a time.

*Humans, who are limited by slow biological evolution, couldn't compete and would be superseded. In his book *Superintelligence*, philosopher Nick Bostrom provides an argument that artificial intelligence will pose a threat to humankind.

*He argues that sufficiently intelligent AI, if it chooses actions based on achieving some goal, will exhibit convergent behavior such as acquiring resources or protecting itself from

KL-SUM

Por último, la librería Sumy incluye un método para aplicar el algoritmo KL-Sum.

Su funcionamiento sobre el texto original se puede ver a continuación, donde se ha hecho uso de la función *KLSummarizer* de la librería *sumy.summarizers.kl*.

Resumen (5 frases):

Nota: Las que están marcadas con un asterisco [] son las 3 frases más importantes según el algoritmo.*

*This marked the completion of a significant milestone in the development of Artificial Intelligence as Go is a relatively complex game, more so than Chess.

Unsupervised learning is the ability to find patterns in a stream of input, without requiring a human to label the inputs first.

*Many of the problems in this article may also require general intelligence, if machines are to solve the problems as well as people do.

This question is closely related to the philosophical problem as to the nature of human consciousness, generally referred to as the hard problem of consciousness.

*For the danger of uncontrolled advanced AI to be realized, the hypothetical AI would have to overpower or out-think all of humanity, which a minority of experts argue is a possibility far enough in the future to not be worth researching.

EVALUACIÓN

A simple vista, se puede observar que cada algoritmo pondera las frases de manera diferente. Al tratarse de un texto largo que se compone de 10776 *tokens*, reducirlo a tan solo 5 frases supone comprimir mucho la información original. Además, al tratarse de un texto que contiene información muy diversa dentro de un mismo tema, los resúmenes reflejan este hecho, mostrando una secuencia de frases que no siguen un mismo hilo.

En la siguiente tabla (Tabla 1) se muestra los resultados de las 3 métricas principales de ROUGE.

Tabla comparativa – DUC2001 dataset

	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	40.42	15.40	24.62
Luhn	42.07	16.81	25.82
LexRank	42.29	15.80	25.29
LSA	35.85	11.99	21.10
KL-Sum	35.85	11.70	21.58

Tabla 1. Puntuaciones ROUGE de algunos algoritmos extractivos sobre el dataset DUC2001 (Fuente: Victor et al., 2019)

5.4. TÉCNICAS ABSTRACTIVAS

Como ya se ha visto anteriormente, las técnicas abstractivas aplican formas de resumir más parecidas a cómo lo hacen los humanos, utilizando palabras y frases que no figuran en el texto original. Sin embargo, las técnicas actuales, a pesar de haber conseguido resultados sorprendentes, siguen sin tener errores de coherencia y cohesión.

En este apartado, se va a implementar algunas de las técnicas que representan el estado del arte. Seguidamente, se analizará los resultados de estos algoritmos.

¿Cómo se puede implementar de manera sencilla un resumen abstractivo?

TRANSFORMERS

La pregunta anterior tiene una fácil respuesta: *transformers*. Como ya se ha comentado anteriormente, esta librería permite implementar modelos de Machine Learning para crear aplicaciones de resumen, clasificación, etc. Algunos de los modelos son GPT-2, GPT-3, BERT, BART, Longformer, OpenAI, T5, PEGASUS, etc.

En los siguientes apartados se mostrarán los resúmenes que se pueden conseguir utilizando algunos de los mejores modelos preentrenados de *transformers*.

Se ha establecido un margen de longitud de entre 150 y 300 *tokens*.

T5

T5 funciona muy bien en una amplia variedad de tareas y su aplicación pasa por añadir al inicio de la entrada de texto la tarea que se desea llevar a cabo. En este caso, para la tarea de resumen, hay que añadir al inicio de la secuencia de texto original “summarize: ”. De esta forma, el modelo T5 devuelve el resumen de dicho texto.

Como ya se ha comentado en el apartado Estado del Arte, T5 es un *encoder-decoder*, lo que significa que primero codifica el texto de entrada y luego lo decodifica. Para ello, el texto de entrada se debe convertir a una secuencia de *ids* o *input-ids*. Esto se consigue mediante la función *encode()*, y posterior *decode()* para convertir los *ids* a texto.

Para la implementación, se ha elegido los modelos preentrenados “t5-small”, “t5-base” y “t5-large”, ordenados por tamaño de memoria de los modelos. mediante la función *from_pretrained()*. Previamente hay que importar las librerías *T5ForConditionalGeneration* para el modelo y *T5Tokenizer* para el tokenizador.

Se puede observar grandes similitudes entre las palabras y frases elegidas, pero con ligeros cambios. Es posible incluso apreciar que el más grande de los tres modelos, “t5-large” genera en el resumen información más amplia y variada.

Sin embargo, también se puede observar que al final de los textos introducen algunas combinaciones de letras y caracteres extraños.

Resumen (“t5-small”):

'Strong' AI is labelled as artificial general intelligence (AGI) while attempts to emulate 'natural' intelligence have been called artificial biological intelligence (ABI) the term "artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving" modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games ...'art' AI 'a. ... 'i.e. 'n' a . a human intelligence .

Resumen (“t5-base”):

artificial intelligence (AI) is intelligence demonstrated by machines, unlike natural intelligence . leading AI textbooks define the field as the study of "intelligent agents" the term "artificial intelligence" is often used to describe machines that mimic "cognitive" functions . some people consider AI to be a danger to humanity if it progresses unabated, others believe it will create a risk of mass unemployment . the field was founded on the assumption that human intelligence can be so precisely described that a machine can

Resumen (“t5-large”):

artificial intelligence (AI) is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals . leading textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of achieving its goals . modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems . after alphago defeated a professional go player in 2015, artificial intelligence once again attracted widespread global attention - ed a . - n aa ena a- a. na n- en aen en-ena

BART

La librería Transformers también ofrece la posibilidad de implementar resúmenes con modelos BART. Para ello, hay que importar los módulos *BartForConditionalGeneration* y *BartTokenizer*. Mediante la función *from_pretrained()*, al igual que con el modelo T5, se importa el modelo preentrenado. En este caso se ha utilizado el modelo “facebook/bart-large-cnn” de Facebook y uno de los modelos más descargados actualmente, “sshleifer/distilbart-cnn-12-6”, los cuales han sido ajustados (*fine-tuned*) al dataset CNN / Dailymail.

Al igual que con T5, primero se hace codifica (*encode*) el texto de entrada a una secuencia de *ids*, el modelo genera el resumen a partir de los *ids* y, finalmente, se decodifica para obtener este resumen en forma de texto.

Resumen:

Nota: Para esta entrada de texto, el resumen generado es idéntico para ambos modelos.

Artificial intelligence (AI) is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals. 'Strong' AI is usually labelled as artificial general intelligence (AGI) while attempts to emulate 'natural' intelligence have been called artificial biological intelligence (ABI) Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations. The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence.

LONGFORMER

El modelo *Longformer2Roberta* es un modelo que ha sido ajustado para ser usado en tareas de resumen. Soporta un máximo de 4096 *tokens*. Este modelo consiste en una arquitectura Longformer2RoBERTa, la cual utiliza el Longformer como *encoder* y RoBERTa como *decoder*.

Para cargar el modelo *Longformer2Roberta* se ha importado la librería *LongformerTokenizer* y *EncoderDecoderModel*. Como modelo, se ha utilizado para la implementación el modelo preentrenado “patrickvonplaten/longformer2roberta-cnn_dailymail-fp16” mediante la función *from_pretrained()*. Este ha sido ajustado (*fine-tuned*) con el dataset CNN / Dailymail, práctica común en tareas de resumen de textos. Como tokenizador, se ha usado “allenai/longformer-base-4096”. El resultado es el siguiente:

Resumen:

Artificial intelligence is intelligence demonstrated by machines that mimic "cognitive" functions . The term "artificial intelligence" is often used to describe machines that imitate "cognition" The field is being divided into sub-fields that often fail to communicate with each other . Some people consider AI to be a danger to humanity if it progresses unabated . Many people consider it a danger for humanity if its progress unabated unabated, such as previous technological revolutions . AI techniques have been developed in the past, but many are still considered to be incomplete . and are still needed to solve problems . In the The Systems of AI is among the field's long-term goals . ABI field

PEGASUS

Para implementar un resumen con PEGASUS, se ha recurrido a varios modelos como “google/pegasus-xsum”, “google/pegasus-cnn_dailymail”, “google/pegasus-large” o “google/bigbird-pegasus-large-arxiv”. Estos modelos han sido implementados por Google, empresa que creó el algoritmo Pegasus. Soportan como entrada 512, 1024, 1024 y 4096 *tokens* respectivamente.

Para los resúmenes se ha establecido una longitud mínima de 150 *tokens* y una longitud máxima de 300 *tokens*.

Resumen (“pegasus-xsum”):

Artificial intelligence is the study of "intelligent agents": machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving". Colloquially, the term "artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving".

Resumen (“pegasus-cnn_dailymail”)

Artificial intelligence (AI) is intelligence demonstrated by machines, unlike the natural intelligence displayed by humans and animals . 'Strong' AI is usually labelled as artificial general intelligence (AGI). Attempts to emulate 'natural' intelligence have been called artificial biological intelligence (ABI)

Resumen (“pegasus-large”)

\Strong\ AI is usually labelled as artificial general intelligence (AGI) while attempts to emulate \natural\ intelligence have been called artificial biological intelligence (ABI). Colloquially, the term "artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving". Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations. Artificial intelligence was founded as an academic discipline in 1955, and in the years since has experienced several waves of

Este último modelo ha replicado las frases del texto original. Esto se debe a que el proceso de pre-entrenamiento de Pegasus es muy similar al proceso de resumir, ya que en el pre-entrenamiento se enmascaran las frases importantes del documento de entrada para luego generarlas en el proceso de resumen, como secuencia de frases importantes, de manera similar al resumen extractivo.

Resumen (“bigbird-pegasus-large-arxiv”)

this paper presents a brief history of the field of artificial intelligence (), which was founded as an academic discipline in the 1950s . since then , has been divided into subfields that often fail to communicate with each other . <n> this paper presents a brief history of the field of artificial intelligence (), which was founded as an academic discipline in the 1950s . since then , has been divided into subfields that often fail to communicate with each other . <n> this paper presents a brief history of the field of artificial intelligence (), which was founded as an academic discipline in the 1950s . since then , has been divided into subfields that often fail to communicate with each other . <n> this paper presents a brief history of the field of artificial intelligence (), which was founded as an academic discipline in the 1950s . since then , has been divided into subfields that often fail to communicate with each other . <n> this paper presents a brief history of the field of artificial intelligence (), which was founded as an academic discipline in the 1950s . since then , has been divided into subfields that often fail to communicate with each other .

EVALUACIÓN

Algunos de los resúmenes generados por las técnicas abstractivas han arrojado unos resultados sorprendentes. Sin embargo, estos se han conseguido utilizando como texto de entrada el primer párrafo o introducción de la página de la Wikipedia, en lugar de toda la página, el cual está compuesto por 681 *tokens*. Incluso para algún modelo, como “pegasus-xsum”, se ha tenido que reducir aún más la entrada.

Esta elección se debe a la limitación que sufren los modelos utilizados, que soportan entradas de un número de *tokens* limitado, ya sea de 512, 1024, 2048 o 4096. Este problema se desarrollará en el siguiente punto.

Como orientación se pueden observar los resultados de las 3 métricas principales de ROUGE en los modelos anteriores sobre el dataset “CNN/Dailymail” (Tabla 12).

Tabla comparativa – CNN / Dailymail dataset

	ROUGE-1	ROUGE-2	ROUGE-L
T5	43.52	21.55	40.69
BART	44.16	21.28	40.90
PEGASUS	44.17	21.47	41.11

Tabla 2. Puntuaciones ROUGE de algunos algoritmos abstractivos sobre el dataset CNN / Dailymail (Fuente: <https://paperswithcode.com>).

Nota 1: Los resultados de Longformer han sido calculados sobre el dataset arXiv, por lo que no se han incluido en esta tabla.

Nota 2: Los resultados de PEGASUS son para su modelo “pegasus-large”.

5.5. LIMITACIONES

En los puntos anteriores se ha podido comprobar el buen funcionamiento de algunas técnicas abstractivas consistentes en *transformers*. Sin embargo, estos modelos pre-entrenados sufren de una limitación y es que soportan una entrada de texto de una longitud limitada. Algunos modelos como “pegasus-cnn_dailymail” o “pegasus-large” soportan hasta 1024 *tokens*. Otros, como BART o T5, únicamente permiten un máximo de 512 *tokens*, por lo que son inservibles a la hora de resumir textos de un tamaño mayor.

5.6. SOLUCIÓN

Como el objetivo de este trabajo es obtener un algoritmo multifuncional, es decir, que se pueda aplicar a cualquier tipo y tamaño de texto, se propone una solución que sea capaz de soportar tamaños grandes. Esta solución consiste en la creación de un algoritmo que combine las técnicas extractivas y abstractivas, de forma que la técnica extractiva se encargue de reducir el texto de entrada al tamaño soportado por el modelo, y este produzca el resumen abtractivo a partir de las frases más importantes extraídas previamente. Esta solución permite aprovechar el gran potencial de los *transformers*, sea cual sea el tamaño del texto de entrada.

De esta manera, el algoritmo elegido para el resumen extractivo ha sido TextRank, de la librería Gensim. Esta elección se debe a que permite establecer el número de palabras que se desea para el resumen extractivo, mientras que los 4 algoritmos vistos anteriormente, de la librería Sumy, permiten establecer el número de frases. Como se quiere establecer el número de *tokens* que tome como entrada el algoritmo abtractivo, por la limitación de este último, el algoritmo TextRank es ideal para este cometido, además de que sus resultados son bastante buenos.

Para el segundo paso, se usará un algoritmo abtractivo, a elección del usuario. Como ya se ha comentado anteriormente, una vez que se haya reducido el texto original a un tamaño soportado por el modelo de *transformers* elegido, este generará el resumen abtractivo final.

A continuación, se mostrarán algunos ejemplos para poder observar el desempeño del algoritmo con los diferentes modelos. Esta vez, en lugar de utilizar el primer párrafo o introducción de la página de Wikipedia de Inteligencia Artificial, se utilizará todos los párrafos de la página web al completo.

Se ha establecido un tamaño de resumen de entre 150 y 300 *tokens*.

Resumen BART (“facebook/bart-large-cnn”)

The term "artificial intelligence" is often used to refer to the ability of computers to mimic human thinking and behavior. In the past, the term was used to describe systems that were able to solve complex problems. In recent years, it has been used to mean systems that are able to understand and act on complex information in a way that is consistent with human thinking. The term is also used to talk about systems that can be taught to think in a specific way, such as a computer's ability to understand a complex set of words. The word "artificially intelligent" means that a system can learn and act in a certain way to solve a complex problem. It can also mean that it can be programmed to do certain things in a particular way, for example, to learn a new language.

Resumen DistilBART (“sshleifer/distilbart-cnn-12-6”)

The term "artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind . Traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects . Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations . Research in this area includes machine ethics, artificial moral agents, friendly AI and discussion towards building a human rights framework is also in talks . In his book Moral Machines For Wallach,

Resumen T5 (“t5-small”)

"artificial intelligence" is commonly used to describe machines that mimic "cognitive" functions that humans associate with the human mind . modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems . traditional problems (or goals) of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects . by 2020, Natural Language Processing systems such as the enormous GPT-3 were matching human performance on pre-existing benchmarks ...»»» ... "artificial intelligence" is the . human intelligence .com

Resumen T5 (“t5-base”)

"artificial intelligence" is often used to describe machines that mimic "cognitive" functions . many researchers predict that such "narrow AI" work in different individual domains will eventually be incorporated into a machine with artificial general intelligence (AGI) by 2020, natural language processing systems such as the enormous GPT-3 were matching human performance on pre-existing benchmarks . researchers at MIT argued that no simple and general principle (like logic) would capture all of the aspects of intelligent behavior, resulting in "n- - " n . . "

Resumen T5 (“t5-large”)

"artificial intelligence" is often used to describe machines that mimic "cognitive" functions . AI techniques have become an essential part of the technology industry . many researchers predict that such "narrow AI" work in different domains will eventually be incorporated into a machine with artificial general intelligence (agi) this would combine most of the narrow skills mentioned in this article and at some point even exceed human ability . a new generation of machines will be able to mimic human behavior, but they won't be perfect a .- a- na aa n aen a. are a "a ena -a

Artificial intelligence is used to study human behavior and take action . The field of AI has experienced a resurgence following the success of expert systems . In the early 1980s, AI research was revived by the commercial success of experts . Many of the problems in AI are related to artificial intelligence, says John Sutter . Sutter: AI research is a way to help humans and machines . and that's a great idea . and a great start . and it's time to move on from the current state of AI . He says the research goal of AI is to create technology that allows computers to function in an intelligent AAASAs are complex, but they are not immediately recognized . A few ideas are being considered .

[illegible]

Resumen PEGASUS (“google/pegasus-cnn_dailymail”)

"artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind . Traditional AI problems include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects . In the twenty-first century, AI techniques have experienced a resurgence following advances in computer power, large amounts of data, and theoretical understanding . High-profile examples of AI include autonomous vehicles (such as drones and self-driving cars), medical diagnosis, creating art (such as poetry), proving mathematical theorems, playing games (such as Chess or Go), search engines (such as Google Search), online assistants (such as Siri), image recognition in photographs, spam filtering, predicting flight delays, prediction of judicial decisions, targeting online advertisements, and energy storage .

Resumen PEGASUS (“pegasus-large”)

Modern machine capabilities generally classified as AI include successfully understanding human speech, competing at the highest level in strategic game systems (such as chess and Go), and also imperfect-information games like poker, self-driving cars, intelligent routing in content delivery networks, and military simulations. In the twenty-first century, AI techniques have experienced a resurgence following concurrent advances in computer power, large amounts of data, and theoretical understanding; and AI techniques have become an essential part of the technology industry, helping to solve many challenging problems in computer science, software engineering and operations research. By 2020, Natural Language Processing systems such as the enormous GPT-3 (then by far the largest artificial neural network) were matching human performance on pre-existing benchmarks, albeit without the system attaining commonsense understanding of the contents of the benchmarks.

this paper presents a philosophical analysis of the field of machine learning . <n> [section]
[section] [section] [section] [section] [section] [section] [section] [section] [section
] [section] [section] [section] [section] [section] [section] [section] [section] [section]
section] [section] [section] [section] [section] [section] [section] [section] [section
] [section] [section] [section] [section] [section] [section] [section] [section]

Se puede observar que este último modelo, aunque promete funcionar con entradas de texto grandes de hasta 4096 *tokens*, en ciertos casos sufre de graves problemas y no consigue generar un resumen coherente.

RESULTADOS Y DISCUSIÓN

Llegado este punto, ya se ha experimentado con una multitud de algoritmos de resumen, tanto extractivo como abstractivo. En este proceso se ha comprobado el funcionamiento de técnicas que ponderan frases del texto original y sacan como resumen las más puntuadas, así como técnicas SOTA como lo son los *transformers* que mediante diferentes modelos pre-entrenados y ajustados para la tarea de resumen obtienen, mediante su arquitectura *encoder-decoder*, unos resultados realmente sorprendentes.

Tras generar resúmenes con diferentes modelos de *transformers*, se puede decir que todos los que se han utilizado en este trabajo generan resúmenes bastante coherentes y cohesivos, e incluyen información variada pero muy relevante del texto. En algunos modelos, el resumen acaba sin finalizar la última frase. Otros, como T5, introducen al final del resumen caracteres raros, los cuales hay que filtrar ya que el modelo por sí mismo no lo hace.

Se puede observar que los modelos de Pegasus, en general, contienen información muy amplia y relevante del texto, permitiendo obtener una idea general del texto de un simple vistazo.

Los tres modelos de T5 han generado resúmenes diferentes a pesar de consistir en el mismo modelo. Sin embargo, al haber sido preentrenados y contar con unos pesos inicializados de diferente manera, devuelven resultados un tanto distintos. Bien es cierto que los resúmenes contienen información muy relevante del texto original.

El *transformer* BART es un modelo muy consistente y ello se ve reflejado en sus resultados. El Longformer es, en cambio, un modelo muy reciente, pero genera resúmenes que constan de oraciones bastante relevantes.

Como el objetivo final de este trabajo es obtener un algoritmo funcional para una amplia variedad de textos de diferentes tamaños, se ha desarrollado una aplicación que es capaz de cumplir este propósito.

6. APLICACIÓN WEB

Para dar una salida a este trabajo y que el código no quede olvidado entre las carpetas del ordenador, se ha desarrollado una aplicación web. Esta aplicación ha sido desarrollada mediante la librería Streamlit, de Python, la cual ha resultado útil para desarrollar la interfaz y poner la página en producción a través de un servidor que la propia plataforma ofrece.

La aplicación y el código desarrollado están disponibles para cualquier usuario a través del enlace que se ha dejado en el Anexo de este trabajo.

El aspecto de esta interfaz de la app se puede ver en la Figura 16.

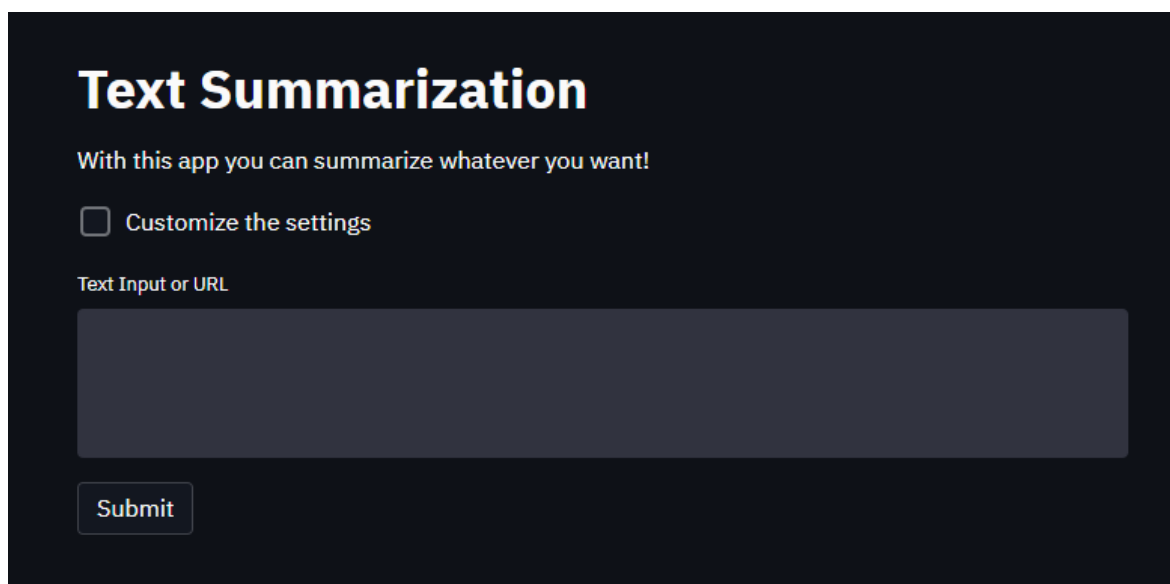
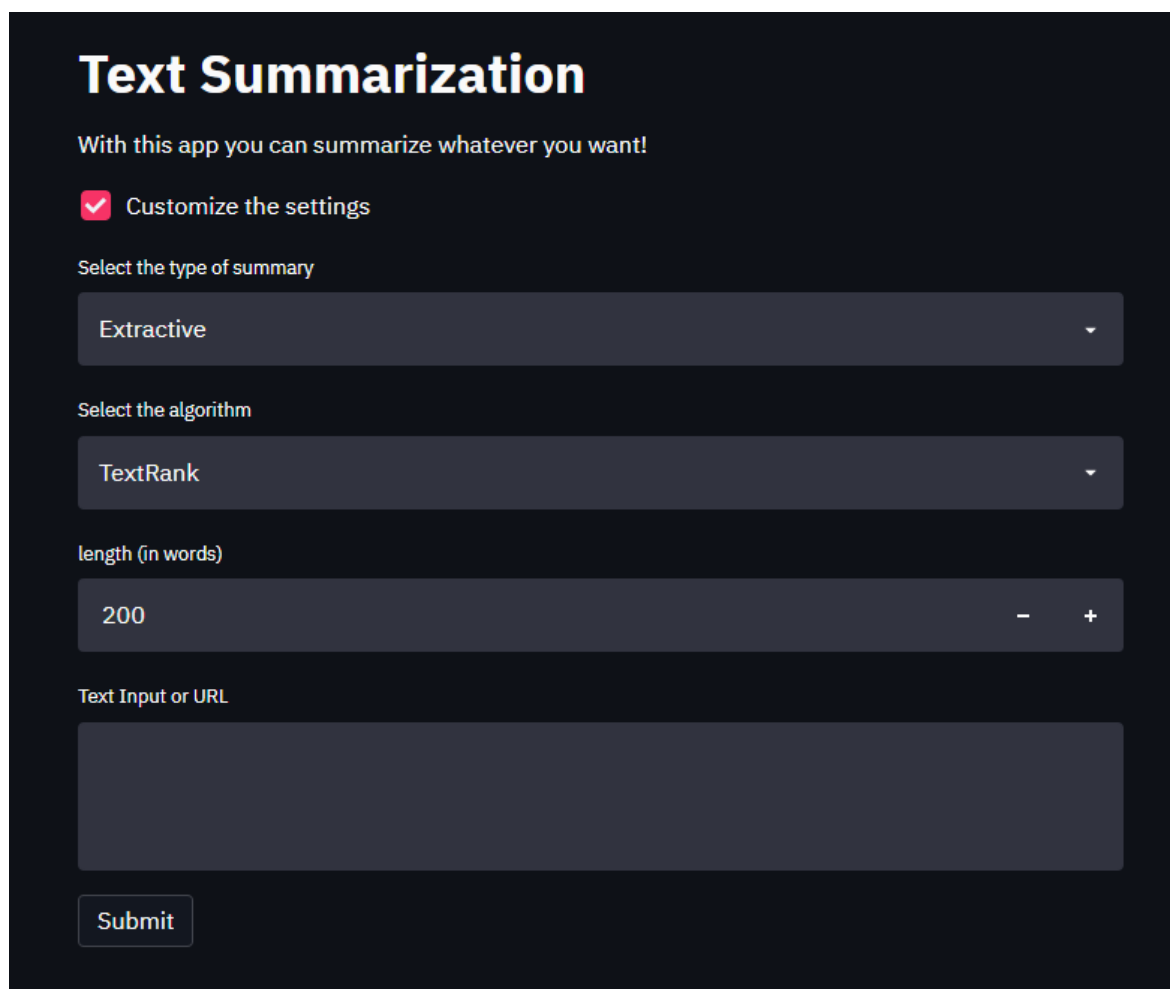


Figura 16. Interfaz de la app

Se puede observar que la interfaz consiste en un espacio donde el usuario puede poner un texto plano o un enlace a una página de internet. Si la entrada del usuario es un texto, este pasa por un proceso de preprocesado donde se limpia el texto y, a continuación, se pasa este texto al algoritmo de resumen. Una vez que el algoritmo ha generado el resumen, este es pasado por un proceso de postprocesado en el cual vuelve a pasar por un proceso de limpieza ya que algunos algoritmos generan caracteres que no se desea mostrar al usuario.

Por defecto, el algoritmo encargado de realizar el resumen es TextRank de la librería Gensim, el cual generará un resumen extractivo. Sin embargo, es posible seleccionar el algoritmo deseado para la tarea de resumir. Para ello se debe hacer click en la casilla “Customize the settings”. Una vez seleccionada (ver Figura 17) aparecen varias opciones para customizar el algoritmo de resumen.



Text Summarization

With this app you can summarize whatever you want!

☒ Customize the settings

Select the type of summary

Extractive

Select the algorithm

TextRank

length (in words)

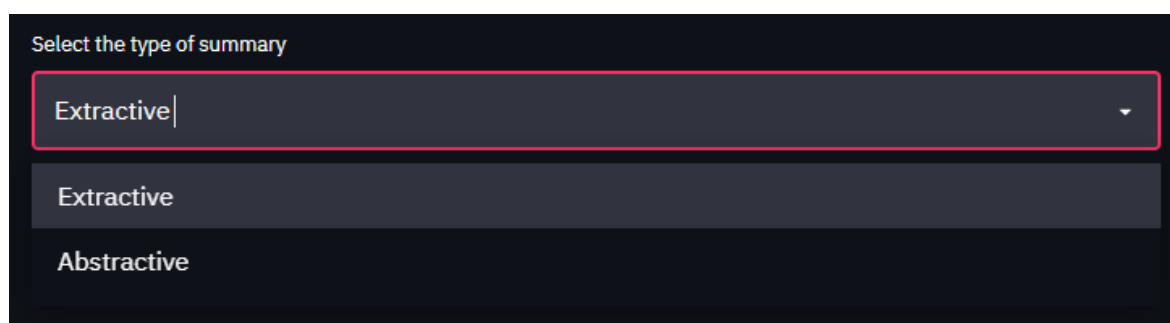
200

Text Input or URL

Submit

Figura 17. Opciones de customización

En la opción “Select the type of summary” se puede seleccionar si se desea utilizar un algoritmo extractivo o abstractivo (Figura 18).



Select the type of summary

Extractive

Extractive

Abstractive

Figura 18. Menú desplegable Extractivo/Abstractivo

En caso de estar seleccionada la opción “Extractive”, la siguiente casilla le permite al usuario elegir el algoritmo extractivo de entre una lista de algoritmos utilizada en este trabajo (Figura 19).

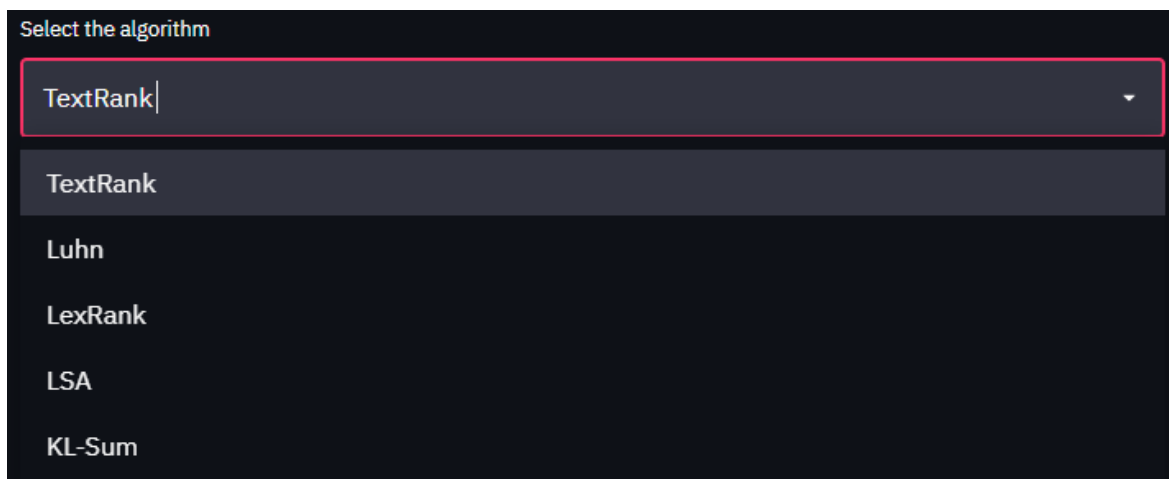
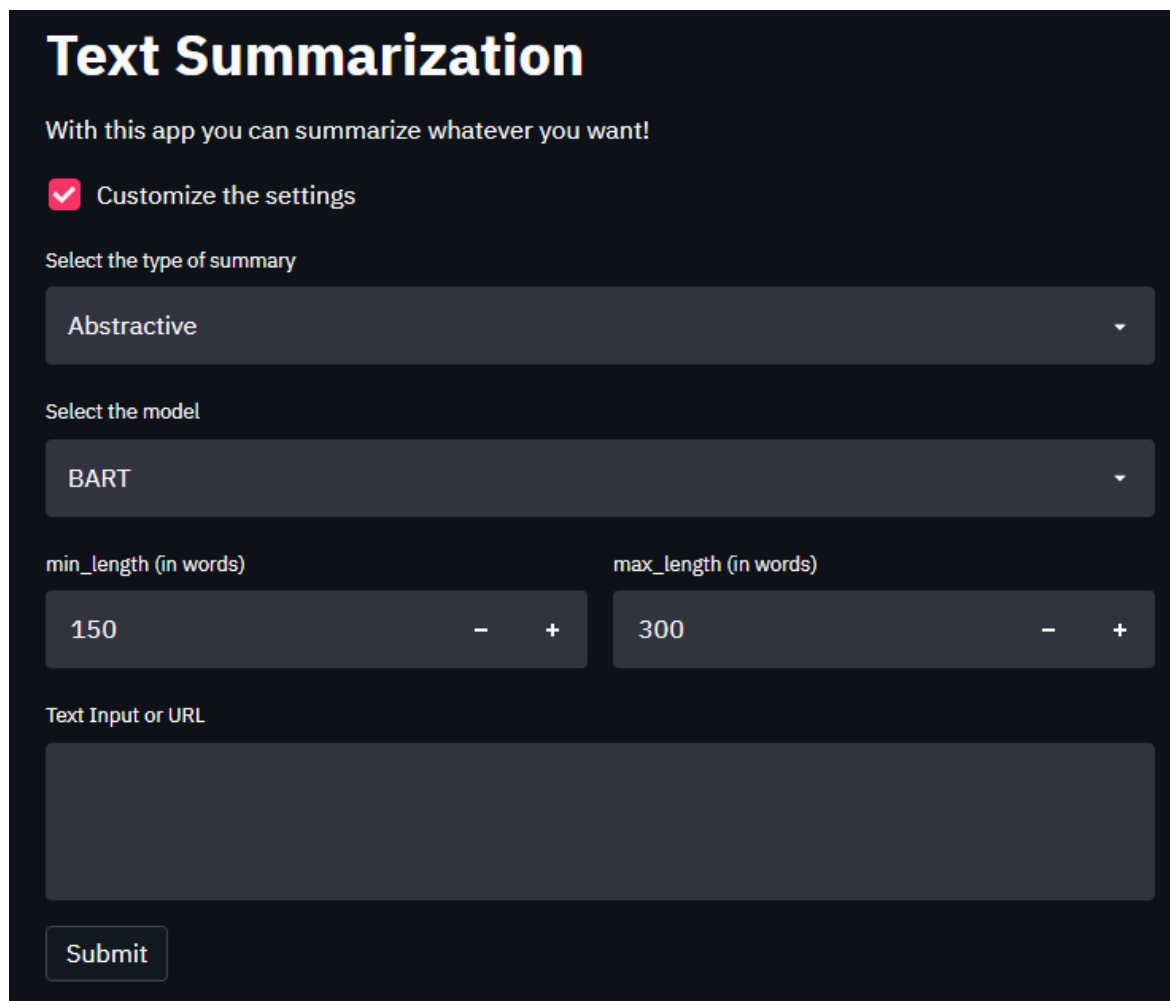


Figura 19. Menú desplegable de algoritmos extractivos

Mientras que el algoritmo TextRank permite elegir el número de palabras del resumen, los otros cuatro algoritmos, de la librería Sumy, permiten elegir el número de frases. Esto se debe a los diferentes parámetros que ofrecen las librerías Gensim y Sumy.

Seleccionando como tipo de resumen el abstractivo se queda la interfaz de la siguiente manera (ver Figura 20):



The screenshot shows a web application titled "Text Summarization". It has a dark theme. The main heading is "Text Summarization" in a large, bold, white font. Below it, a subtitle says "With this app you can summarize whatever you want!". There is a checked checkbox labeled "Customize the settings". Below this, there are two sections: "Select the type of summary" with a dropdown menu showing "Abstractive", and "Select the model" with a dropdown menu showing "BART". Below these are two input fields for "min_length (in words)" and "max_length (in words)". The first field has a value of 150 and the second has a value of 300. Both fields have minus and plus buttons for adjustment. Below these fields is a large text input area labeled "Text Input or URL". At the bottom left is a "Submit" button.

Figura 20. Opciones de customización con el tipo de resumen abstractivo

Se puede observar que en este caso las opciones para elegir el tamaño del resumen consisten en elegir el número de palabras mínimo y máximo. Es necesario establecer un rango en los *transformers* ya que, al ser un resumen abstractivo y, por ello, generar frases nuevas, el número de palabras final será variable dependiendo del tamaño que se desee.

En la Figura 21 se puede observar el desplegable que se abre a la hora de elegir el modelo con el que se desee realizar el resumen.

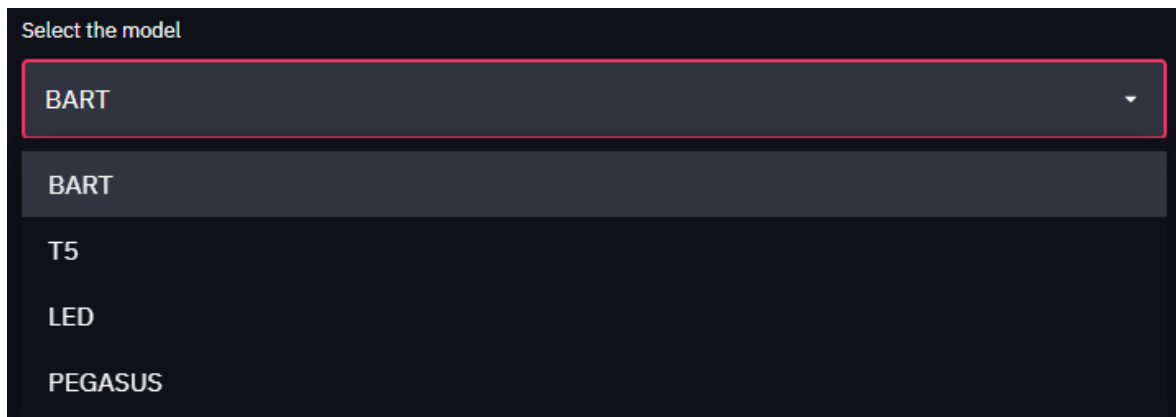


Figura 21. Menú desplegable de algoritmos abstractivos

Para algunos algoritmos como BART, T5 o Pegasus, se abre un menú desplegable para que el usuario pueda elegir qué modelo preentrenado desea usar para obtener el resumen (Figura 22, Figura 23 y Figura 24).

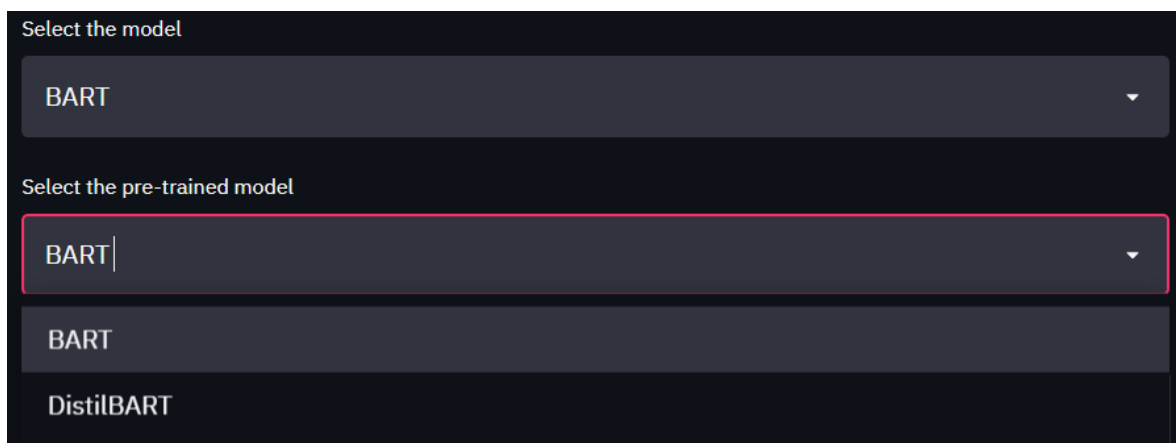


Figura 22. Menú desplegable modelos de BART preentrenados

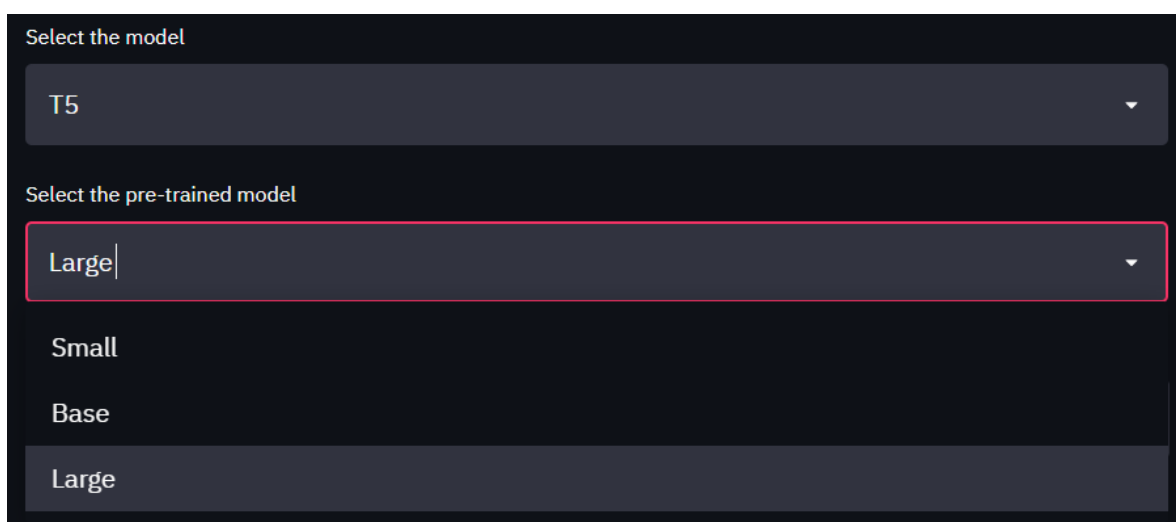
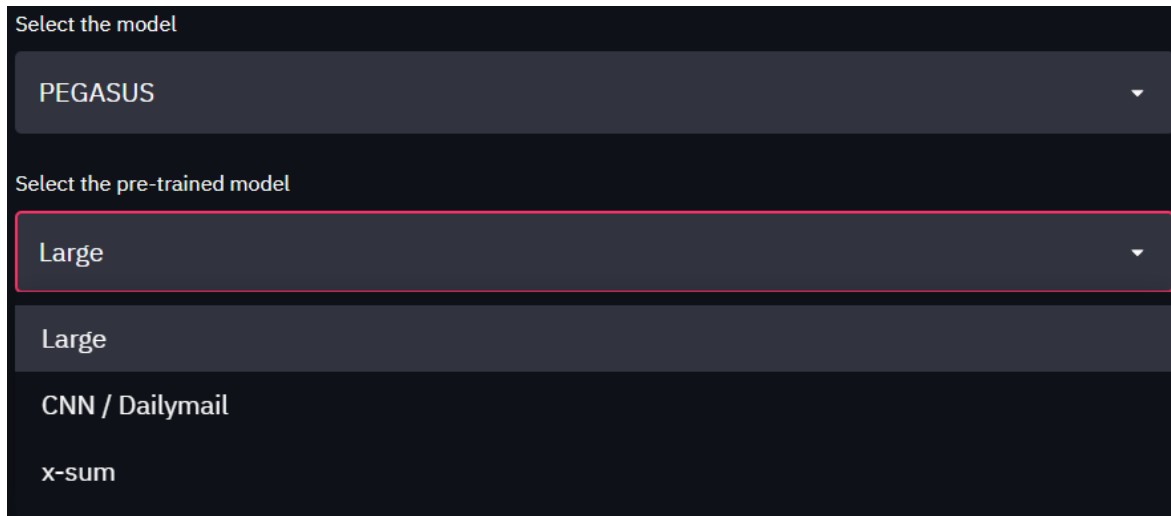


Figura 23. Menú desplegable modelos de T5 preentrenados



The image shows a dark-themed web interface for selecting models. It contains two dropdown menus. The first menu, titled "Select the model", has "PEGASUS" selected. The second menu, titled "Select the pre-trained model", has "Large" selected and is highlighted with a red border. Below this menu, the text "Large", "CNN / Dailymail", and "x-sum" are visible, likely representing other options in the dropdown.

Figura 24. Menú desplegable modelos de Pegasus preentrenados

Una vez que se tenga seleccionadas las opciones de personalización al gusto y haber puesto el enlace o texto a resumir hay que hacer click en “Submit”. El resultado final con el resumen generado se puede ver en la Figura 25.

Text Summarization

With this app you can summarize whatever you want!

☒ Customize the settings

Select the type of summary

Abstractive

Select the model

PEGASUS

Select the pre-trained model

CNN / Dailymail

min_length (in words) 80 - + max_length (in words) 150 - +

Text Input or URL

https://en.wikipedia.org/wiki/Artificial_intelligence

Submit

Your text is being summarized...

Summary

"artificial intelligence" is often used to describe machines that mimic "cognitive" functions that humans associate with the human mind . Traditional AI problems include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects . In the twenty-first century, AI techniques have experienced a resurgence following advances in computer power, large amounts of data, and theoretical understanding . Research in this area includes machine ethics, artificial moral agents, friendly AI and discussion towards building a human rights framework is also in talks .

Generated in 1 min, 39 s

Figura 25. Resultado de generar un resumen con la app

Tras haber realizado diferentes pruebas de esta aplicación, se puede decir que cumple con su propósito inicial. La aplicación permite realizar resúmenes extractivos y abstractivos, de cualquier tamaño, utilizando diversas técnicas y modelos. Permite, además, customizar diversos parámetros para obtener un resumen a la medida del usuario.

Si bien es cierto que la herramienta es capaz de extraer el texto de una gran cantidad de páginas, en otras no consigue extraer el contenido de las mismas. Esto se debe a que las páginas están escritas en código HTML y su estructura varía de una página a otra, por lo que establecer reglas para extraer información de ciertos apartados del código es una tarea complicada.

CONCLUSIONES FINALES

En este trabajo se ha realizado un recorrido histórico por las principales técnicas y métodos utilizados para la tarea de resumen automático de textos o ATS, para los que se ha empleado desde técnicas estadísticas como TF-IDF hasta redes neuronales, modelos de atención y *transformers*.

Además de lo anterior, se ha realizado una experimentación utilizando diferentes modelos que componen el estado del arte. Se ha mostrado cómo implementar estas técnicas de manera sencilla, utilizando diferentes librerías de Python como Sumy, Gensim o Transformers, además de otras como TensorFlow o PyTorch, necesarias para el funcionamiento de los *transformers*.

Finalmente, se ha desarrollado una herramienta que permite al usuario obtener un resumen del texto de cualquier página de internet en inglés o de cualquier texto plano. Para poder resumir todas estas páginas, se ha propuesto una solución que consiste en la combinación de técnicas extractivas y abstractivas, y que mediante esta combinación es capaz de resumir de manera abstractiva textos muy largos, lo cual no es posible de conseguir actualmente directamente con un *transformer*. Esto se consigue ya que previamente se reduce el texto con la técnica extractiva para proporcionarle un texto con un número de *tokens* acorde a la entrada que el modelo abstractivo es capaz de soportar.

Todavía queda un largo trayecto hasta que los resúmenes abstractivos sean comparables a los que es capaz de realizar un humano. Sin embargo, los resultados son muy prometedores.

7. TRABAJO FUTURO

Este trabajo da lugar a una multitud de ampliaciones. Entre ellas está mejorar la aplicación web, haciéndola más robusta frente a diversos formatos de entrada. Así, una mejora interesante sería incorporar un botón para que el usuario pueda subir un archivo de texto y que la herramienta genere el resumen de este documento.

Otra aplicación potencial para este trabajo es crear una extensión de un navegador de internet (Chrome, Firefox, etc.) que, utilizando una de las técnicas de resumen extractivo utilizadas, obtenga las frases más relevantes de una página web y las marque de un color. De esta manera el usuario podrá obtener de un vistazo la información más importante de la página, artículo, noticia, etc.

Por supuesto, ya se habla de nuevos modelos de Transformers que prometen mejorar los resultados de los actuales y soportar entradas más grandes. Un ejemplo de ello es el modelo Reformer que, una vez que esté disponibles y haya sido ajustado para su uso en el pipeline de resumen de textos, podrá ser usado de la misma manera que los otros modelos. Este y otros avances que vayan surgiendo en el ámbito del resumen de textos se incluirán en la herramienta, de modo que esta se vaya adaptando a las mejoras de los algoritmos de resumen automático de textos.

Otra aplicación futura podría consistir en tomar como entrada no solo texto escrito, sino también audio. Esto se conseguiría con un paso previo al proceso empleado en este trabajo, en el cual se tendría que realizar una conversión previa de audio a texto, mediante librerías como *wav2vec*.

Finalmente, también se tomará en consideración los futuros desarrollos que se vayan haciendo para mejorar el resumen automático de textos en el idioma español, lo cual resultaría de un gran interés.

BIBLIOGRAFÍA

Alammar, J. (2018). *The Illustrated Transformer*. <http://jalammar.github.io/illustrated-transformer/>

Artificial intelligence. (2021). En *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Artificial_intelligence&oldid=1025930579

Barzilay, R., & McKeown, K. R. (2005). Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3), 297-328. <https://doi.org/10.1162/089120105774321091>

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. *arXiv:2004.05150 [cs]*. <http://arxiv.org/abs/2004.05150>

Bing, L., Li, P., Liao, Y., Lam, W., Guo, W., & Passonneau, R. J. (2015). Abstractive Multi-Document Summarization via Phrase Selection and Merging. *arXiv:1506.01597 [cs]*. <http://arxiv.org/abs/1506.01597>

Briggs, J. (2021, marzo 4). *The Ultimate Performance Metric in NLP*. Medium. <https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460>

Cheng, J., & Lapata, M. (2016). Neural Summarization by Extracting Sentences and Words. *arXiv:1603.07252 [cs]*. <http://arxiv.org/abs/1603.07252>

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*. <http://arxiv.org/abs/1810.04805>

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based Lexical Centrality as Saliency in Text Summarization. *Journal of Artificial Intelligence Research*, 22, 457-479. <https://doi.org/10.1613/jair.1523>

Haghighi, A., & Vanderwende, L. (2009). Exploring Content Models for Multi-Document Summarization. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 362-370. <https://www.aclweb.org/anthology/N09-1041>

Hahn, U., & Mani, I. (2000). The challenges of automatic summarization. *Computer*, 33(11), 29-36. <https://doi.org/10.1109/2.881692>

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv:1910.13461 [cs, stat]*. <http://arxiv.org/abs/1910.13461>

- Lin, C.-Y., & Hovy, E. (2003). Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, 150-157. <https://www.aclweb.org/anthology/N03-1020>
- Lloret, E., Plaza, L., & Aker, A. (2018). The challenging task of summary evaluation: An overview. *Language Resources and Evaluation*, 52(1), 101-148. <https://doi.org/10.1007/s10579-017-9399-2>
- Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2), 159-165. <https://doi.org/10.1147/rd.22.0159>
- Maybury, M. T. (1995). Generating summaries from event data. *Information Processing & Management*, 31(5), 735-751. [https://doi.org/10.1016/0306-4573\(95\)00025-C](https://doi.org/10.1016/0306-4573(95)00025-C)
- Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing Order into Text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404-411. <https://www.aclweb.org/anthology/W04-3252>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. 9.
- Moratanh, N., & Chitrakala, S. (2017). A survey on extractive text summarization. *2017 international conference on computer, communication and signal processing (ICCCSP)*, 1-6.
- Nazari, N., & Mahdavi, M. A. (2019). A survey on Automatic Text Summarization. *Journal of AI and Data Mining*, 7(1), 121-135. <https://doi.org/10.22044/jadm.2018.6139.1726>
- Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the Special Issue on Summarization. *Computational Linguistics*, 28(4), 399-408. <https://doi.org/10.1162/089120102762671927>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv:1910.10683 [cs, stat]*. <http://arxiv.org/abs/1910.10683>
- Red neuronal artificial. (2021). En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Red_neuronal_artificial&oldid=133525230
- ROUGE (metric). (2019). En *Wikipedia*. [https://en.wikipedia.org/w/index.php?title=ROUGE_\(metric\)&oldid=913825951](https://en.wikipedia.org/w/index.php?title=ROUGE_(metric)&oldid=913825951)
- Steinberger, J., & Ježek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. *In Proc. ISIM '04*, 93-100.

Torres, J. (2019). Deep Learning, Introducción práctica con Keras (SEGUNDA PARTE). *Jordi TORRES.AI*. <https://torres.ai/deep-learning-inteligencia-artificial-keras-2a-parte/>

Vaca, A. (2020, mayo 6). Transformers en Procesamiento del Lenguaje Natural—IIC. *Instituto de Ingeniería del Conocimiento*. <https://www.iic.uam.es/innovacion/transformers-en-procesamiento-del-lenguaje-natural/>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [cs]*. <http://arxiv.org/abs/1706.03762>

Victor, D. M., Eduardo, F. F., Biswas, R., Alegre, E., & Fernández-Robles, L. (2019). Application of Extractive Text Summarization Algorithms to Speech-to-Text Media. En H. Pérez García, L. Sánchez González, M. Castejón Limas, H. Quintián Pardo, & E. Corchado Rodríguez (Eds.), *Hybrid Artificial Intelligent Systems* (Vol. 11734, pp. 540-550). Springer International Publishing. https://doi.org/10.1007/978-3-030-29859-3_46

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv:1912.08777 [cs]*. <http://arxiv.org/abs/1912.08777>

ANEXO

Enlace al proyecto en GitHub:

- <https://github.com/YanisNC/Text-Summarization-NLP>

Para ejecutar la aplicación en local, cualquier usuario puede clonarse el repositorio y ejecutar la aplicación con el siguiente comando:

```
streamlit run streamlit_app.py
```

Enlace a la página web de la aplicación:

- <https://share.streamlit.io/yanisnc/text-summarization-nlp/main>

A continuación, se mostrarán más resultados de los resúmenes generados con los diferentes algoritmos abstractivos, reduciendo previamente el texto mediante el algoritmo extractivo TextRank.

En este caso, mientras que anteriormente se ha mostrado el funcionamiento del algoritmo con una página de Wikipedia, esta vez con el fin de mostrar los resultados con otro tipo de página, se resumirá un artículo periodístico del periódico inglés *The Sun*. Así, la página de la cual se va a resumir su contenido será:

- <https://www.thesun.co.uk/news/royal/15332868/prince-charles-grandson-archie-never-prince-fury-harry-meghan/>

El título de la noticia es:

- *ROYAL RUMBLE Prince Charles to 'ensure grandson Archie, 2, will never be a Prince sparking fury with Prince Harry and Meghan Markle'*

y la entrada es:

- *PRINCE Charles is planning to ensure that his grandson Archie will never be given the title of Prince - as part of his plans for a slimmed-down monarchy, reports claim.*

Resumen BART (“facebook/bart-large-cnn”)

The Prince of Wales has allegedly made clear that Prince Harry and Meghan Markle’s two-year-old son will not be at the forefront of the Royal Family when he becomes King. The existing rules for Royal titles were established in Letters Patent dated November 20, 1917, which allowed the title of Prince and Princess to be given to specific relatives. Prince Harry is also thought to have demanded to hand pick at least one journalist to work alongside the British press pack of Royal reporters at the unveiling of the statue dedicated to Princess Diana next month. He is believed to have asked for a journalist to join him at the ceremony. The royal baby is expected to be born in the spring of next year and is due to be the first Royal baby to be named after a monarch.

Resumen DistilBART (“sshleifer/distilbart-cnn-12-6”)

Prince of Wales is planning to ensure that his grandson Archie will never be given the title of Prince . Charles has allegedly made clear that Prince Harry and Meghan Markle’s two-year-old son will not be at the forefront of the Royal Family when he becomes King . The Sussexes are thought to have found out about the plans before sitting down with Oprah Winfrey for their bombshell interview . Charles will likely stay in Scotland when his son jets in from the US at the end of the month . It comes after warring brothers William and Harry have called a truce for the unveiling of a statue to mum Diana . The pair are due at Kensington Palace in London on July 1 — their first meeting since Philip's funeral in April.

Resumen T5 (“t5-small”)

PRINCE Charles is planning to ensure that his grandson will never be given the title of Prince . he is also thought to have demanded to hand pick at least one journalist to work alongside the press pack of royal reporters at the unveiling of the statue dedicated to princess Diana next month . prince of Wales has allegedly made clear that prince Harry and his son will not be at the forefront of the royal family when he becomes King . the existing rules for royal titles were established in Letters patent dated November 20, 1917 .»»». priprince Harry's son 'will never be a Prince ' 'the prince of england's a journalist to be

Resumen T5 (“t5-base”)

news corp is a network of leading companies in the worlds of diversified media, news, education, and information services . reports claim the prince of wales has allegedly made clear that his grandson Archie will not be at the forefront of the royal family when he becomes King . existing rules for royal titles were established in Letters patent dated November 20, 1917, which allowed the title of Prince and Princess to be given to specific relatives . 'when you're the grandchild of the monarch, automatically Archie and .. [[[[.»nl-[-

Resumen T5 (“t5-large”)

prince of wales wants to ensure that his grandson will never be given the title of prince . a source said: "harry and Meghan were told archie would never be a prince, even when Charles became King." existing rules for Royal titles were established in Letters Patent dated November 20, 1917, which allowed the titles of Prince and Princess to be given to specific relatives . prince Harry is also thought to have demanded to hand pick at least one journalist to work alongside the British press pack of Royal reporters at the unveiling of a .- . n a . n .. n. .- .enaen

Resumen Longformer (“patrickvonplaten/longformer2roberta-cnn_dailymail-fp16”)

Prince Charles allegedly made clear that his grandson will never be Prince . Prince Harry and Meghan Markle's son will not be at the forefront of the Royal Family when he becomes King . Sources claim that the Prince and Me will not have the title of Prince or Princess . The existing rules for Royal titles were established in 1917 . Charles is also thought to have demanded to hand pick at least one journalist to work alongside the British press . He also demanded to be handpicked at the unveiling of Diana's statue . Royal reporters will be given a chance to pick a journalist to be the next royal royal .

' But it's not their right to take . ‘Prince or Princess’ .

Resumen PEGASUS (“pegasus-large”)

During one part of the interview, Meghan told Oprah: “They were saying they didn't want him to be a Prince or a Princess.” “You know, the other piece of that convention is, there's a convention – I forget if it was George V or George VI convention – that when you're the grandchild of the monarch, so when Harry's dad becomes King, automatically Archie and our next baby would become Prince or Princess, or whatever they were going to be... But also it's not their right to take it away.” Harry is also thought to have demanded to hand pick at least one journalist to work alongside the British press pack of Royal reporters at the unveiling of the statue dedicated to Princess Diana next month.