COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ESTUDIO Y EVALUACIÓN DE LAS ESTRATEGIAS DE EXTRACCIÓN DE CONOCIMIENTO EN MODELOS LLM PERSONALIZADOS

Autor: Daniel Valverde Gómez

Director: David Contreras Bárcena

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Estudio y evaluación de las estrategias de extracción de conocimiento en modelos LLM personalizados

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2023/24 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Daniel Valverde Gómez Fecha: 14/ 06/ 2024

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: David Contreras Bárcena Fecha: 14/ 06/ 2024

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ESTUDIO Y EVALUACIÓN DE LAS ESTRATEGIAS DE EXTRACCIÓN DE CONOCIMIENTO EN MODELOS LLM PERSONALIZADOS

Autor: Daniel Valverde Gómez

Director: David Contreras Bárcena

Madrid

Agradecimientos

Agradezco principalmente a mi director David Contreras Bárcena por ofrecerme la idea del proyecto y la valiosa orientación y apoyo recibidos durante la realización de este trabajo. Adicionalmente, agradezco a la Universidad Pontificia Comillas – ICAI por facilitarme los recursos hardware con los que se llevó a cabo la prueba de concepto de este proyecto.

Agradezco también a mi familia por su apoyo incondicional y por su comprensión y ánimo durante esta etapa académica. También quiero agradecer a mis compañeros de clase por el intercambio de ideas y compañerismo.

Finalmente, quiero expresar mi gratitud a todas las personas que contribuyeron a este proyecto y cuyo nombre no se menciona aquí, pero su ayuda fue igualmente valiosa.

ESTUDIO Y EVALUACIÓN DE LAS ESTRATEGIAS DE EXTRACCIÓN DE CONOCIMIENTO EN MODELOS LLM PERSONALIZADOS

Autor: Valverde Gómez, Daniel
Director: Contreras Bárcena, David
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Durante los últimos años hemos sido testigos de una revolución en el desarrollo de la Inteligencia Artificial (IA), especialmente en el ámbito de los modelos de procesamiento de lenguaje natural (NLP) popularmente conocidos como *chatbots.* Esta revolución se debe principalmente a la aparición de los modelos grandes de lenguaje o LLM (*Large Language Models* por sus siglas en inglés*)*.

Estos modelos se fundamentan en la arquitectura de los transformadores (*Transformers)* [1] la cual les da la capacidad de "prestar atención" al contexto de la secuencia de entrada y analizar de forma más eficiente grandes cantidades de datos. El propósito de estos modelos reside en la aspiración de dotar a las máquinas con capacidad de comprender nuestro lenguaje y, adicionalmente, de generar textos coherentes en dicho lenguaje de una forma mucho más eficiente.

Mientras se desarrollaba esta arquitectura, surgió una nueva metodología que ha dado paso a muchos de los modelos utilizados en la actualidad: la multimodalidad. En lugar de enfocarse en especializar los modelos en una tarea única, se persigue la creación de modelos con capacidad para realizar múltiples tareas para posteriormente realizar un pequeño reentrenamiento (*fine-tuning*) que los especialice en la tarea que se desea resolver. Esto supuso un punto de inflexión pues los costes de reentrenamiento se vieron considerablemente reducidos y los modelos se volvieron mucho más generales e inteligentes.

Es dentro de este contexto donde se sitúa el presente proyecto. Uno de los mayores problemas a los que se enfrentan las organizaciones o instituciones que desean implementar modelos de IA es el desafío del entrenamiento dado su elevado coste computacional. Esto se debe a que los LLM se componen de miles de millones de parámetros (de ahí su denominación como "grandes modelos de lenguaje"), donde cada uno de los parámetros representa las variables internas del modelo que se ajustan durante el entrenamiento para mejorar su capacidad de entender y generar texto coherente. Una de las alternativas es la de utilizar modelos pre-entrenados ofrecidos por grandes empresas tecnológicas. No obstante, suelen requerir de una licencia para poder ser usados abiertamente, la cual puede acarrear limitaciones en su uso. Es por ello por lo que existe una tendencia creciente al uso de modelos de código abierto (*Open Source),* los cuales han ido ganado terreno debido a sus numerosas ventajas tales como la seguridad de los datos, la gestión autónoma de los modelos o su transparencia.

Este trabajo tiene como objetivo realizar un estudio de las diferentes soluciones *Open Source* disponibles analizando tanto la eficiencia en rendimiento como la viabilidad económica, con el propósito de presentar las herramientas necesarias para adoptar esta tecnología de forma eficaz y eficiente.

Durante el estudio se ha analizado el ecosistema de modelos actuales que implementan tecnologías de código abierto con el fin de obtener una visión global. Entre los más destacados se encontraron:

- Llama: Desarrollado por *Meta.* [16]
- Mixtral: Desarrollado por *Mistral AI.* [84]
- Falcon: Desarrollado por el Instituto de Innovación Tecnológica de los Emiratos Árabes. [11]
- MPT: Desarrollado por *MosaicML* [25]
- Qwen (abreviación de *Tongyi Qianwen* 通义千问) [38]

En un entorno dinámico donde el número de modelos de código abierto está en constante aumento, surge la necesidad de establecer alguna forma de evaluar el rendimiento de los diferentes modelos, para ello se ha recurrido al uso de múltiples métricas para su evaluación, entre ellas, las tablas de evaluación comparativa (*benchmarking).* La realización de un *benchmarking* de un LLM implica el uso de un conjunto de datos cuidadosamente preparado para evaluar el rendimiento del modelo en una tarea específica. Estos conjuntos de datos se diseñan meticulosamente para abarcar una amplia variedad de escenarios y desafíos lingüísticos, permitiendo así una evaluación exhaustiva del rendimiento de un LLM.

	LLaMA 2 70B	**GPT - 3.5**	**Mixtral 8x7B**
MMLU (MCQ in 57 subjects)	69.9%	70.0%	**70.6%**
HellaSwag (10-shot)	87.1%	85.5%	86.7%
ARC Challenge (25-shot)	85.1%	85.2%	**85.8%**
WinoGrande (5-shot)	**83.2%**	81.6%	81.2%
MBPP (pass@1)	49.8%	52.2%	**60.7%**
GSM-8K (5-shot)	53.6%	57.1%	**58.4%**
MT Bench (for Instruct Models)	6.86	**8.32**	8.30

Ilustración 1 – Tabla de benchmarking de los modelos LLama2 (70B), GPT-3.5 y Mixtral 8x7B en algunos de los datsets más conocidos, resaltando los valores más altos. Fuente: [2]

Otro aspecto considerado en el estudio es el de la viabilidad del hardware. Como se ha mencionado anteriormente, estos modelos destacan por su gran tamaño, lo que conlleva

elevados requerimientos computacionales. En respuesta a los desafíos computacionales la comunidad ha desarrollado múltiples herramientas para abordar la ejecución eficiente de los modelos. La técnica más utilizada consiste en cuantizar los parámetros al reentrenarlos para reducir el número de bits con los que se almacenan y procesan los "pesos" del modelo. Esta técnica aumenta la velocidad de inferencia y reduce el coste de GPU. Se podría pensar que esta reducción de precisión tendría un impacto en el rendimiento del modelo, sin embargo, existen técnicas de cuantización que permiten reducir los pesos del modelo hasta 4 bits sin generar ningún tipo de degradación en el rendimiento. Entre las más destacadas se encuentra *QLoRA* [49], que combina LoRA (*Low-Rank adaptation of LLM*) con la cuantización de los parámetros del modelo.

Adicionalmente, se explora una tendencia que ha adquirido considerable relevancia en el último año: los agentes virtuales, basados en generación con recuperación aumentada o RAG (*Retrieval Augmented Generation* por sus siglas en inglés) [5]. Esta arquitectura soluciona algunos de los grandes problemas de los LLM tales como las alucinaciones o la fecha de corte para el entrenamiento. El método se basa en el uso de bases de datos vectoriales, que permite realizar búsquedas basadas en la similitud de los datos. Se añade un paso intermedio cada vez que se realiza una consulta durante el cual se realiza una búsqueda en la base de datos para obtener información relevante adicional y la información obtenida (*retrieval*) se añade al *prompt* inicial (*augmented*) de forma que ahora generará una respuesta generalmente mejor (*generation*).

En la fase conclusiva del proyecto, se llevarán a cabo dos pruebas de concepto (PoC, *Proof of concept* por sus siglas en inglés*)* en las que se realiza una implementación práctica de los conceptos expuestos en el estudio. La primera prueba utiliza el modelo seleccionado (*LLama3-8B-Instruct*) y realiza una cuantización a cuatro (4) bits mediante el método QLoRA, seguida de un reentrenamiento o *fine-tuning* utilizando un conjunto de datos de medicina [46]. En la segunda prueba se hizo uso de una base de datos vectorial y la arquitectura RAG para suministrar varios reportes médicos y que el modelo sea capaz de responder a preguntas relacionadas con el mismo. Se estudiaron tres (3) técnicas (*Naïve*, *ParentDocument*, *MultiQuery*) diferentes de recuperación de documentos para mejorar la calidad de las respuestas del modelo.

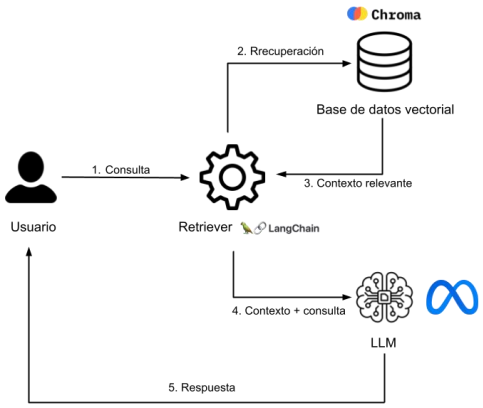

Ilustración 2 - Esquema de los componentes utilizados para la PoC de RAG. Fuente: elaboración propia

STUDY AND EVALUATION OF KNOWLEDGE EXTRACTION STRATEGIES IN CUSTOMIZED LLM MODELS

Author: Valverde Gómez, Daniel.
Supervisor: Contreras Bárcena, David
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

During the last few years, we have witnessed a revolution in the development of Artificial Intelligence (AI), particularly in the field of Natural Language Processing (NLP), commonly known as *ChatBots*. This revolution is primarily attributed to the emergence of Large Language Models (LLMs).

These models are based on the *transformer* architecture [1], which grants them the ability to "pay attention" to the context of the input sequence and analyze large amounts of data mor efficiently. The purpose of these models lies in the aspiration to endow machines with the ability to understand our language and, additionally, to generate coherent texts in that language in a much more efficient manner.

As this architecture was being developed, a new methodology emerged, giving rise to many of the models used today: Multimodality. Instead of focusing on specializing models in a single task, the aim is to create models with the ability to perform multiple tasks, followed by fine-tuning to specialize them in the desired task. This marked a turning point, as the retraining costs were significantly reduced, and the models became much more intelligent.

The current project lies within this context. One of the major challenges faced by organizations or institutions seeking to implement AI models is the training process, given its high computational cost. This is because LLMs consist of billions of parameters (hence their designation as "large language models"), where each parameter represents the internal variables of the model adjusted during the training process to enhance its ability to understand and generate coherent text. One alternative is to use pre-trained models offered by large tech companies. However, they often require a license for open usage, which may come with rate or other limitations. Hence, there is a growing trend towards using open-source models, which have gained ground due to numerous advantages such as data security, autonomous model management and transparency.

This project aims to conduct a study of the various open-source solutions available, analyzing both performance efficiency and economic viability, with the purpose of providing the necessary tools for adopting this technology effectively and efficiently.

Throughout the study, the ecosystem of current models implementing open-source technologies has been analyzed to gain a comprehensive overview. Among the most notable ones, were found:

- Llama: Developed by *Meta.* [16]
- Mistral: Developed by *Mistral AI.* [84]
- Falcon: Developed by the Institute of Technological Innovation of the United Arab Emirates. [11]
- MPT: Developed by *MosaicML* [25]
- Qwen (abbreviation of *Tongyi Qianwen* 通义千问) [38]

In a dynamic environment where the number of open-source models is constantly increasing, there arises the need to establish a way to evaluate the performance of the different models. For this purpose, the use of multiple evaluation metrics has been employed, among them, the use of benchmarking tables. Conducting a benchmarking of an LLM involves the use of a carefully prepared dataset to evaluate the model's performance on a specific task. These datasets are meticulously designed to cover a wide range of scenarios and linguistic challenges, thereby allowing for a comprehensive assessment of an LLM's performance.

	LLaMA 2 70B	**GPT - 3.5**	**Mixtral 8x7B**
MMLU (MCQ in 57 subjects)	69.9%	70.0%	**70.6%**
HellaSwag (10-shot)	87.1%	85.5%	86.7%
ARC Challenge (25-shot)	85.1%	85.2%	**85.8%**
WinoGrande (5-shot)	**83.2%**	81.6%	81.2%
MBPP (pass@1)	49.8%	52.2%	**60.7%**
GSM-8K (5-shot)	53.6%	57.1%	**58.4%**
MT Bench (for Instruct Models)	6.86	**8.32**	8.30

Ilustración 3 – A benchmarking table for the models Llama2 (70B), GPT-3.5, and Mixtral 8x7B on some of the most well-known datasets, highlighting the highest values. Source: [2],

Another aspect considered in this study is the hardware requirements. As mentioned earlier, these models are notable for their large size, which entails high computational needs. In response to these computational challenges, the community has developed multiple tools to address efficient model inference. The most commonly used technique involves quantizing the parameters during fine-tuning to reduce the number of bits used to store and process the model's weights. This technique increases inference speed and reduces GPU costs. One might think that this reduction in precision would impact the model's performance; however, there are quantization techniques that allow reducing the models weights to as low as four (4) bits without causing any performance degradation. Among the most notable is QLoRA [49], which combines LoRA (Low-Rank adaptation of LLM) with parameter quantization.

Finally, the study explores a technique that has gained considerable relevance in the past year: AI agents, based on RAG (Retrieval Augmented Generation). This architecture addresses some major issues of LLMs, such as hallucinations or training cut-off dates. This method is based on vector databases, which allow for searches based on distance similarity. An intermediate step is incorporated each time a query is made by the user, where an additional search is performed in the database for relevant information, and the retrieved information (retrieval) is added to the initial query (augmented). This augmentation typically results in a better-generated response (generation).

In the final part of the project, two Proof of Concept (PoC) test will be conducted, where a practical implementation of the concepts outlined in the study will be performed. The first test utilizes the selected model (*Llama3-8B-Instruct*) and undergoes quantization to 4 bits using the QLoRA method, followed by fine-tuning using a medical dataset [46]. In the second PoC, a vector database and the RAG architecture are used to provide various medical reports, allowing the model to answer related queries. Three (3) different document retrieval techniques (Naïve, ParentDocument, MultiQuery) were studied to improve the quality of the model's response.

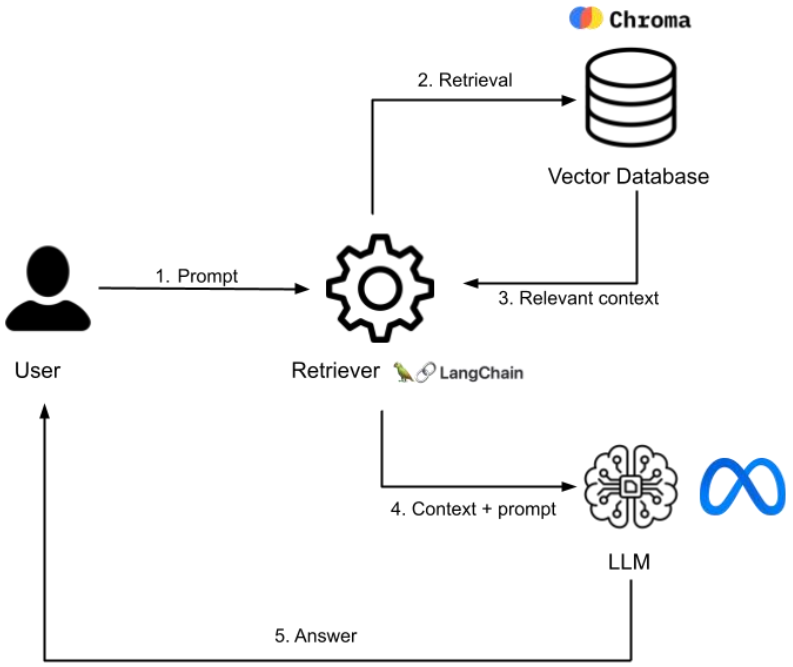

Ilustración 4 – Diagram of the components used for the RAG PoC. Source: Self made

Índice de la memoria

Capítulo 1. Introducción ***5***

Capítulo 2. Descripción de las Tecnologías ***9***
2.1 Los transformer 9
2.2 RAG y Bases de datos vectoriales 10
2.3 MoE (Mixture Of Experts) 13
2.4 Cuantización 16

Capítulo 3. Estado de la Cuestión ***18***
3.1 Modelo Falcon 19
3.2 Modelo Llama 19
3.3 Modelo MPT 21
3.4 Modelo Mistral 21
3.5 Modelo Qwen 23

Capítulo 4. Definición del Trabajo ***24***
4.1 Justificación 24
4.2 Objetivos 24
4.3 Metodología 25

Capítulo 5. Elección del modelo ***27***
5.1 Benchmarks 27
5.2 Chatbot Arena 29
5.3 Características 31
5.4 Especificaciones y requisitos hardware 32

Capítulo 6. Prueba de concepto ***34***
6.1 Fine-Tuning 34
6.1.1 Preparación del dataset *35*
6.1.2 Configuración del modelo *38*
6.1.3 SFT (Supervised Fine-tuning) *39*
6.1.4 Fusión del modelo base con el adaptador entrenado *46*
6.2 RAG 47
6.2.1 Naïve Retriever *50*
6.2.2 Parent Document Retriever *52*

6.2.3 MultiQuery Retriever *54*

Capítulo 7. Análisis de Resultados ***57***

7.1 Resultados del Fine-tuning 57

7.1.1 Comportamiento del modelo *58*

7.2 Resultados de RAG 65

7.2.1 Resultados Naïve Retirver *66*

7.2.2 Resultados Parent Document Retriever *71*

7.2.3 Resultados Multiquery Retriever *75*

Capítulo 8. Conclusiones y Trabajos Futuros ***81***

8.1 Fine-tuning 81

8.2 RAG 81

8.3 Fine-tuning vs RAG 82

8.4 Trabajos Futuros 83

Capítulo 9. Bibliografía ***85***

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS ***92***

ANEXO II: CÓDIGO EMPLEADO ***94***

Índice de figuras

Figura 1 - Diagrama de la arquitectura de los transformers. Fuente: [1] 6
Figura 2- Visualización del funcionamiento del mecanismo de atención para la traducción de una frase del inglés al francés. El color negro representa un peso de atención de 0 y el blanco 1. Fuente: [3] 10
Figura 3 - Flujo conceptual del funcionamiento de RAG con un LLM. Fuente: [6] 12
Figura 4 - Representación del funcionamiento de los embeddings mediante la comparación de varias categorías gramaticales. Fuente: [10] 13
Figura 5 - Esquema del funcionamiento de la arquitectura Mixture of experts (MoE). Fuente: [34] 14
Figura 6 – Proporción de tokens asignados a cada experto durante el entrenamiento de Mixtral 8x7B. Fuente: [31] 15
Figura 7 - Asignación de expertos a cada uno de los tokens en un fragmento de texto. Fuente: [31] 15
Figura 8 – Representación de la distribución de bits para cada una de las diferentes precisiones que presentan los LLM. Fuente: [47] 17
Figura 9 - Rendimiento de los LLM en el benchmark MMLU respecto a su coste por cada millón de tokens. (desde 2022 hasta marzo 2024). Fuente: [4] 18
Figura 10 – Benchmark comparativo de los modelos LLama2 y LLama3 en varias métricas. Fuente: [18] 20
Figura 11 – Benchmark comparativo de los modelos LLama2, Mistral-7B y Mixtral-8x7B en diferentes métricas. Fuente: [2] 22
Figura 12 – Benchmark comparativo de los modelos LLama3-8B-Instruct, Gemma-7b-Instruct y Mixtral-7B-Instruct en diferentes métricas. Fuente: [53] 28
Figura 13 – Disposición del ranking “ChatBot arena” para la fecha del 30 de mayo de 2024. Fuente: [60] 30
Figura 14 – Estimación del número de FLOPs y tokens de entrenamiento necesarios para optimizar el coste de entrenamiento de modelos según su tamaño. Fuente: [61] 32
Figura 15 – Distribución de las muestras del dataset tokenizadas según su frecuencia de aparición. Fuente: elaboración propia 37
Figura 16 – Distribución de las muestras del dataset tokenizadas según su frecuencia de aparición tras aplicar truncamiento (truncation) y relleno (padding) a la longitud de 512 tokens. Fuente: elaboración propia. 38
Figura 17 – Reparametrización de LoRA, durante el entrenamiento solo se actualizan las submatrices A y B. Fuente: [73] 40
Figura 18 – Gráfica comparativa de la métrica train/loss en función de los pasos para diferentes fine-tunings con diferente configuración de hiperparámetros. El entrenamiento de color verde claro (100 step, lr=2e-4) muestra los mejores resultados). Nota: Valores de

learning_rate (tasa de aprendizaje) por encima de $5x10^{-4}$ causaban muchas pérdidas causando que no mejorase el resto de los entrenamientos. Fuente: elaboración propia..... 45
Figura 19 – Distribución del número de chunks en función de su tamaño en tokens, tras la tokenización de los chunks generados sin restricciones de longitud. Es el generador de chunks el que decide los tamaños en base a las características semánticas de estos. Fuente: elaboración propia. 48
Figura 20 – Distribución del número de chunks en función de su tamaño en tokens, tras la tokenización de los chunks generados aplicando una restricción de tamaño máximo de 512 tokens. Fuente: elaboración propia 49
Figura 21 – Esquema del flujo de trabajo de la técnica RAG: naïve retriever. Fuente: elaboración propia 50
Figura 22 – Distribución de los chunks generados (marcados de color verde y amarillo) para un reporte médico aleatorio con un tamaño de 500 tokens y un solape de 50 tokens (marcado en un color gris oscuro). Fuente: [62] 51
Figura 23 – Esquema del flujo de trabajo para la técnica de RAG: Parent Document retriever. Fuente: elaboración propia. 53
Figura 24 – Esquema del flujo de trabajo para la técnica de RAG: Multiquery retriever. Fuente: elaboración propia. 55
Figura 25 - Gráfica comparativa de la métrica train/grad_norm en función de los pasos para diferentes fine-tunings con diferente configuración de hiperparámetros. El entrenamiento de color verde claro (100 step, lr=2e-4) muestra los mejores resultados Fuente: elaboración propia. 57

Capítulo 1. INTRODUCCIÓN

Durante los últimos años ha tenido lugar una revolución en el campo de la Inteligencia Artificial (IA), especialmente en el desarrollo de los modelos de procesamiento de lenguaje natural (NLP por sus siglas en inglés). Esta revolución de la IA se debe principalmente a la aparición de los modelos grandes de lenguaje o LLM (*Large Language Models* por sus siglas en inglés). Modelos de IA capaces de comprender el lenguaje humano y generar textos coherentes en dicho lenguaje con gran precisión.

Las primeras redes neuronales capaces de comprender el lenguaje humano fueron las redes neuronales recurrentes (RNN), las cuales procesaban cada palabra secuencialmente. Esta arquitectura, si bien representó un avance significativo en su momento, presentaba una gran limitación: Al procesar cada palabra por separado enfrentaban problemas para mantener el contexto global de la oración, al aumentar el tamaño de la frase, estas tendían a “olvidar” la información contenida al principio de la oración. Por no mencionar que este tipo de análisis de las oraciones conllevaba altos costes computacionales, lo que limitaba su aplicación a gran escala.

La IA también se enfrentaba a un cuello de botella en el entrenamiento. El método usado era el aprendizaje supervisado, que consiste en entrenar al modelo en un conjunto de datos etiquetados, donde cada entrada está asociada con una etiqueta o respuesta deseada. El modelo se encarga de ajustar sus parámetros para reducir las diferencias entre las predicciones que hace y las respuestas reales con el objetivo de encontrar patrones que le permitan hacer predicciones precisas sobre datos nuevos que no estuviesen incluidos en el *dataset* original. Con esto la red neuronal se especializaba en resolver una única tarea específica para la cual había sido entrenada, lo que limitaba su adaptabilidad a nuevos escenarios. Además, se requería de un trabajo adicional de etiquetación.

Para resolver este problema, surge el aprendizaje auto-supervisado, el cual permie entrenar utilizando datos no etiquetados donde los modelos deben generar sus propias etiquetas a partir de los datos de entrada. Para aprender de estos datos, el modelo puede borrar/ocultar ciertas palabras en la oración para posteriormente tratar de predecir la palabra más adecuada o autocompletar frases. La mayor ventaja de este aprendizaje es su capacidad para aprovechar grandes cantidades de datos no etiquetados como son los que encontramos en

internet. El aprendizaje auto-supervisado ha demostrado ser especialmente efectivo en tareas de procesamiento del lenguaje natural, además, mejora considerablemente el rendimiento del modelo en tareas de aprendizaje supervisado.

Sin embargo, el campo del procesamiento del lenguaje natural experimentó un cambio significativo con la introducción de una nueva arquitectura de redes neuronales revolucionaria: los transformadores o *transformers [1].* A diferencia de las RNN, que procesan palabra por palabra, los *transformers* analizan la frase en paralelo y utilizan el mecanismo de atención, el cual permite a la red "prestar atención" a los datos usados en el entrenamiento para obtener relaciones sintácticas y semánticas más complejas dentro de una oración. Este enfoque permite a los modelos obtener de una manera más efectiva el contexto global.

En este trabajo no se realizará una descripción detallada de la estructura de los *transformers.* No obstante, se considera importante abordar el funcionamiento de la primera etapa de la arquitectura, es decir, el *encoder,* ya que proporciona una comprensión fundamental del funcionamiento de los LLM. Para una descripción detallada del funcionamiento del *encoder,* se remite al Capítulo 2. En la Figura 1 se puede apreciar la arquitectura completa.

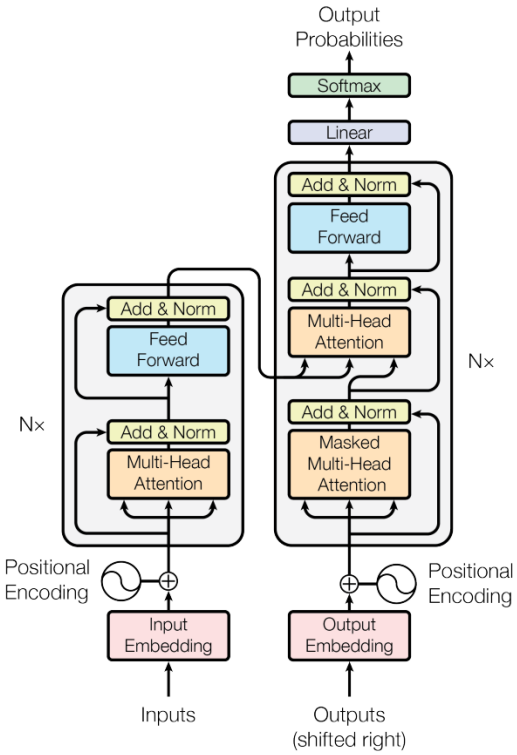

Figura 1 - Diagrama de la arquitectura de los transformers. Fuente: [1]

Los *transformers* presentaban una cualidad que los hacía excepcionalmente potentes: a medida que aumentaba el número de parámetros del modelo, mejoraba tanto su desempeño como su inteligencia. Esta característica fue el motor que impulsó a la creación de los LLM. No obstante, un aumento del tamaño del modelo conlleva un mayor coste computacional y un aumento del tiempo de respuesta ya que se requiere un mayor número de iteraciones a través del *transformer* para ir formando la respuesta. En pocas palabras, los recursos aumentan linealmente con el tamaño del *transformer* por cada token generado. Esto es lo que se conoce como modelos densos, en los que todas las neuronas del modelo trabajan para generar una respuesta. (Densos por la alta cantidad de activaciones que se producen en cada capa).

Los elevados costes computacionales de los modelos más potentes llevaron a un cambio en la metodología: el uso de modelos multimodales pre-entrenados. En lugar de entrenar modelos especializados en una única tarea como sumarización de texto o traducción, se persigue la creación de modelos pre-entrenados capaces de realizar múltiples tareas para posteriormente realizar un reentrenamiento (denominado *fine-tuning*) que les de la especialización deseada. Con esta nueva metodología los costes de entrenamiento de modelos se vieron considerablemente reducidos ya que se podía partir de un “modelo base” para comenzar el entrenamiento. Esta es una de las bases de la democratización de la IA.

Con el creciente desarrollo de la IA, muchas organizaciones e instituciones han empezado a mostrar interés por implementar soluciones basadas en esta tecnología. Sin embargo, esta conlleva elevados costes computacionales, por lo que se opta por utilizar modelos pre-entrenados ofrecidos por grandes empresas tecnológicas. No obstante, suelen requerir de una licencia para poder ser usados abiertamente, la cual puede acarrear limitaciones en su uso. Es por ello por lo que existe una tendencia creciente al uso de modelos de código abierto u *Open Source,* los cuales han ido ganado terreno debido a sus numerosas ventajas tales como la gestión autónoma de los modelos, la transparencia o la capacidad de controlar su nivel de seguridad. Este proyecto se centrará en el estudio de dichos modelos *Open Source*.

Otro de los temas más relevantes de la actualidad es la huella de carbono de la IA dado el aumento de su uso en diversas industrias. Esta huella de carbono viene derivada de varias fuentes como es el consumo energético de los modelos, la producción del hardware necesario para la ejecución de los modelos de IA o la refrigeración necesaria de los centros de procesamiento de datos o

datacenters que manejan estos modelos por lo que reducir la huella de carbono es crucial para reducir costes y abordar problemas como el cambio climático. Este proyecto busca fomentar el uso energéticamente eficiente de la IA y proponer una solución para las organizaciones que deseen implementar soluciones basadas en IA. Para ello, se llevará a cabo un estudio detallado de los diferentes modelos disponibles que implementan soluciones *Open Source* donde se abordarán los avances más recientes en el campo, las metodologías utilizadas para el diseño de los modelos y su viabilidad tanto en términos de hardware como económicos. Posteriormente, se realizará una prueba de concepto o PoC (*Proof of Concept* por sus siglas en inglés) con el objetivo de mostrar el procedimiento de ajuste fino o *fine-tuning* y comparar el rendimiento del modelo antes y después de éste. Adicionalmente, se explorará una de las técnicas que está ganando considerable popularidad en la actualidad, conocida como RAG (*Retrieval Augmented Generation* por sus siglas en inglés) para la misma tarea de análisis de reportes médicos.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 LOS TRANSFORMER

Los *transformers* son un tipo de red neuronal, y las redes neuronales normalmente solo reciben números como parámetros de entrada (*inputs)*. Para convertir las palabras a números se utiliza la codificación de palabras o *word embeddings.* Estas son redes neuronales muy sencillas cuya tarea consiste en *tokenizar* las frases de entrada y convertirlas a vectores numéricos denominados *embeddings.* Los *embeddings* proporcionan una elevada eficiencia de almacenamiento y recuperación de grandes cantidades de información ya que permiten representar palabras en un espacio dimensional menor. Además, aumenta la velocidad de entrenamiento de los modelos.

Para mantener el orden de las palabras, los *transformers* utilizan una técnica llamada *positional encoding.* Mediante codificación binaria se genera un vector que identifica la posición de la palabra en la frase y su dimensión. Posteriormente, se representan en funciones sinusoidales de diferente frecuencia dependiendo de la palabra. Debido a que cada *embedding* puede tener una longitud totalmente diferente, cada palabra obtiene una codificación de posición única, lo que permite a la red aprender a asociar diferentes posiciones con diferentes patrones en la representación sinusoidal.

Por último, encontramos el mecanismo de atención (*self-attention),* donde el *transformer* aprende a encontrar relaciones entre las palabras. Este mecanismo se encarga de analizar cómo de similar es cada palabra respecto a las demás palabras de la frase, incluida la palabra que está siendo analizada y asignarle un peso basado en la relevancia para la tarea en cuestión.

En la ilustración de la Figura 2 se muestran los pesos asignados a la traducción de frases del inglés al francés, se puede apreciar como la tendencia de las palabras es 1:1, sin embargo, si se observa en el panel (a) las palabras “European”, “Economic” y “Area” no han sido traducidas respectivamente, sino que el modelo ha tenido que prestar atención a cada palabra en un orden diferente para poder deducir el contexto adecuado. Por otro lado, observando el panel (d), las palabras “the” y “man” corresponden con “l” y “homme”, sin embargo, “the” puede ser traducido al francés como “le”, “la”, “les”, o “l” dependiendo del contexto. Para decidir qué hacer, el modelo atribuye un mayor

peso a la palabra “man”, prestándole una mayor atención para obtener el contexto de la frase y poder traducir correctamente la palabra “the”.

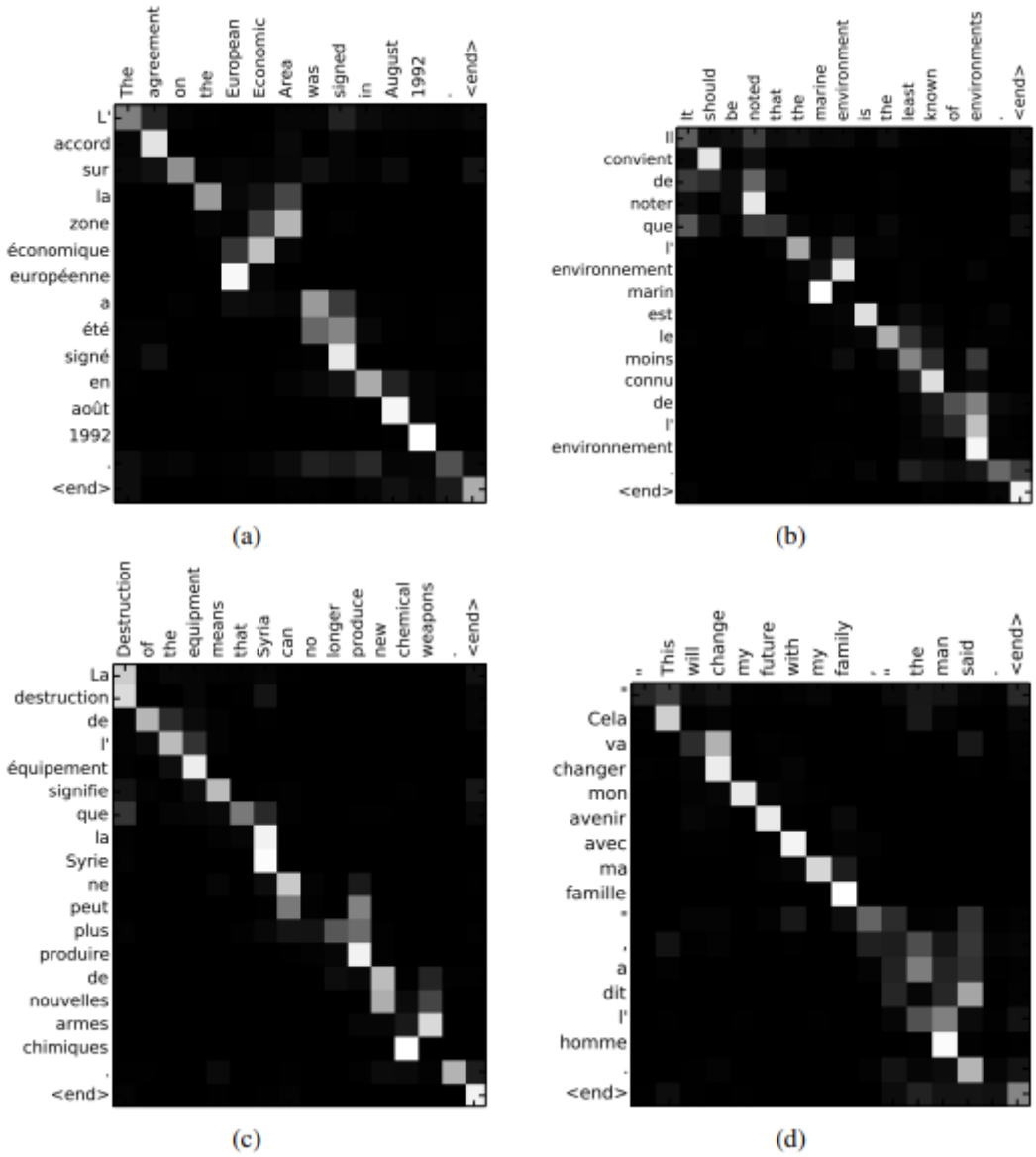

Figura 2- Visualización del funcionamiento del mecanismo de atención para la traducción de una frase del inglés al francés. El color negro representa un peso de atención de 0 y el blanco 1. Fuente: [3]

2.2 RAG Y BASES DE DATOS VECTORIALES

La naturaleza de los LLM introduce imprevisibilidad en las respuestas que genera y a menudo, si no conocen la respuesta correcta, tienden a dar respuestas que

parecen correctas pero no lo son, lo que se conoce como alucinaciones. Además, los datos de entrenamiento son estáticos e introducen una fecha límite en los conocimientos que poseen. Para algunas instituciones interesadas en implementar asistentes virtuales capaces de responder preguntas específicas de los usuarios es necesario realizar un preprocesamiento de los datos de los que dispone el modelo para que se ajuste a la información de la empresa. Sin embargo, realizar ajustes finos (*fine-tuning)* del modelo cada vez que se quiera añadir nueva información puede resultar muy costoso. Además, existe el riesgo de que el modelo comience a desviarse de su rendimiento original tras múltiples ajustes finos.

La técnica RAG (*Retrieval Augmented Generation*) [5] busca resolver estos problemas. Esta técnica consiste en recuperar información de bases de conocimiento externas, como puede ser una base de datos o el propio internet, e introducirla en la entrada (*prompt)* inicial de forma que el modelo adquiera un mayor contexto e intentar que devuelva información más precisa y actualizada. En pocas palabras, se incorpora un paso intermedio cada vez que se realiza un *prompt* en el que se realiza una búsqueda en la base de conocimientos externa y la información obtenida (*retrieval*) se añade al *prompt* inicial para mejorarlo (*augmented*) de forma que ahora generará una respuesta generalmente mejor (*generation*) y con un contexto más amplio. La clave para diseñar un RAG eficiente reside en el primer paso (*retrieval)* y existen diversas técnicas, algunas más avanzadas que otras, para llevar a cabo esta tarea según el tipo de datos con los que se trabaje. En la Figura 3 se muestra el enfoque más básico de todos, conocido como "*naïve retrieval*" denominado así porque al realizar la consulta a la base de conocimiento, se espera de manera ingenua que devuelva los documentos o fragmentos más relevantes. Esta representación permite tener una idea general del funcionamiento de RAG.

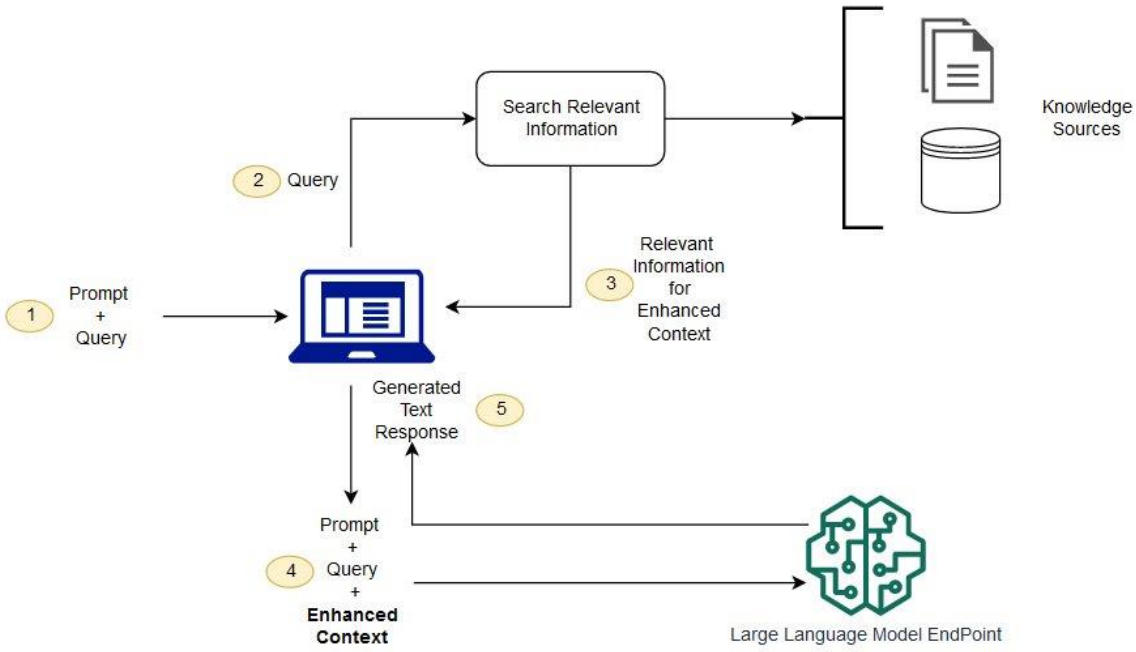

Figura 3 - Flujo conceptual del funcionamiento de RAG con un LLM. Fuente: [6]

Una de las bases de conocimiento más usadas son las **bases de datos vectoriales.** Estas bases de datos se fundamentan en el concepto de inclusiones o *embeddings* que son representaciones matemáticas de los datos en forma de vectores multidimensionales que capturan información semántica y contextual. Las bases de datos vectoriales se encargan de almacenar los *embeddings* en un espacio vectorial.

La mayoría de los modelos de IA generativa utiliza el concepto de los *embeddings* para analizar la información ya que permite representar conceptos abstractos de manera más compacta y facilita su procesamiento. Generalmente se usará un LLM externo para generar los vectores y por la forma en la que se entrenan estos LLM mediante aprendizaje auto-supervisado, palabras similares tendrán vectores similares. Esto se debe a que estas palabras suelen aparecer en contextos similares dentro del texto de entrenamiento, lo que permite al modelo aprender las relaciones semánticas entre palabras, así como la estructura y el contexto del lenguaje natural. Esta cualidad permite almacenar y recuperar grandes cantidades de información de forma altamente eficiente ya que realiza la búsqueda basándose en las distancias entre vectores. En la Figura 4 se puede observar una representación del funcionamiento de los *embeddings* y las relaciones que guardan entre ellos.

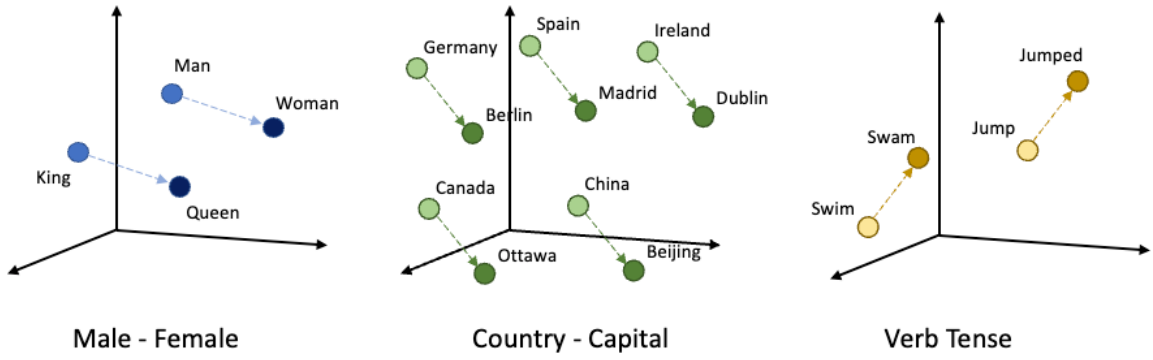

Figura 4 - Representación del funcionamiento de los embeddings mediante la comparación de varias categorías gramaticales. Fuente: [10]

2.3 MoE (Mixture Of Experts)

Como se ha mencionado en la introducción, la arquitectura de los *transformers* presentaba la cualidad de aumentar la inteligencia del modelo linealmente con el número de parámetros, sin embargo, esto acarreaba un aumento de costes de computación. A largo plazo, esta cualidad supondría un problema para la viabilidad y la escalabilidad de la implementación de estos modelos, lo que llevó a la búsqueda de una arquitectura más eficiente. *Mixture-of-Experts* [32] (MoE) presentaba una solución eficiente para aumentar de forma significativa las capacidades de un modelo sin introducir una cantidad proporcional de sobrecarga computacional. La idea principal era dividir las neuronas del modelo en subgrupos, cada uno especializado en una tarea específica, y denominado "experto". Cada "experto" es una red neuronal independiente que durante el entrenamiento se entrena con un subconjunto de datos en el cual alcanzar el rendimiento óptimo para enfocar mejor las capacidades del modelo. La principal ventaja de dividir el modelo en expertos es que, durante la inferencia, solo se activan los expertos necesarios para responder a la pregunta inicial, lo que reduce considerablemente el coste computacional de la ejecución de los modelos y mejora la calidad de las respuestas. Se estima que el modelo GPT-4 [82] se trata de un MoE de grandes dimensiones, de ahí su elevado rendimiento.

En la Figura 5 se puede apreciar un esquema del funcionamiento general de los MoE. La arquitectura se divide en dos partes: los expertos, cada uno de ellos es una red neuronal del tipo *feed-forward* y la red de enrutamiento o *router,* que se encarga de coordinar los distintos expertos y decidir a cuál enviar la entrada. Esto se realiza mediante un sistema de puntuación en el que se elige a los expertos más capaces para la tarea. Esta red de enrutamiento surge como una

extensión de los MoE, conocida como la "*Sparsely-Gated Mixture-of-Experts Layer*" [33], la cual permite seleccionar únicamente unos pocos expertos (normalmente 2) para procesar la entrada, lo que permite preservar la eficiencia computacional.

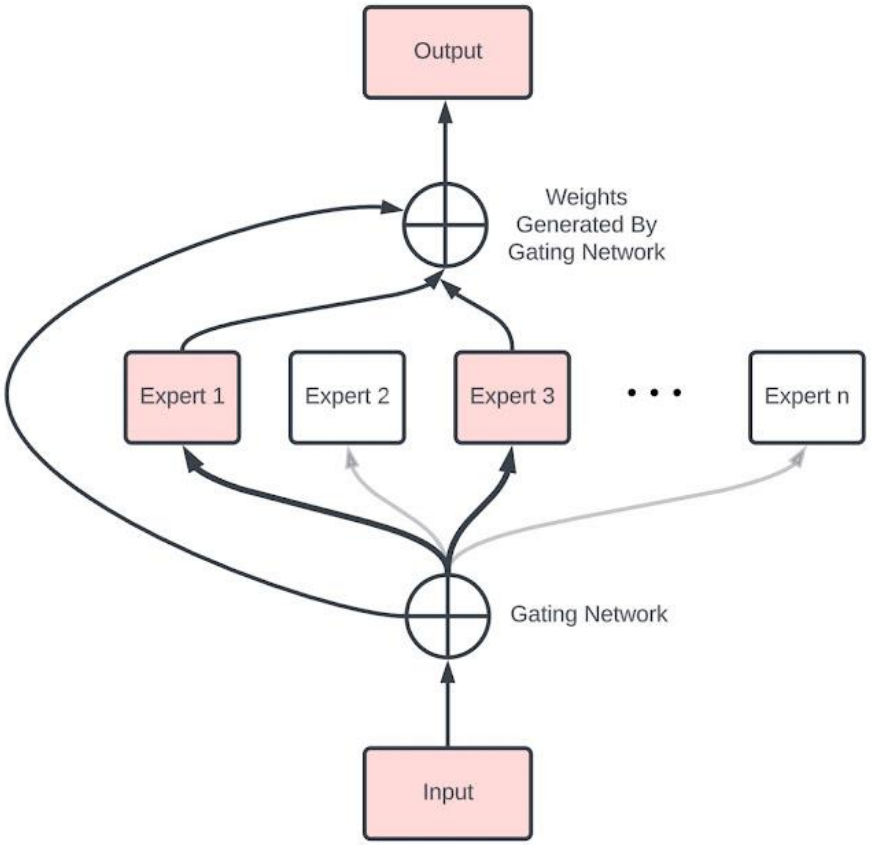

Figura 5 - Esquema del funcionamiento de la arquitectura Mixture of experts (MoE). Fuente: [34]

La red de enrutamiento o *router* se entrena junto a todo el modelo para seleccionar al mejor experto durante el proceso de entrenamiento. Mediante la adición de ruido y la penalización al *router* si no selecciona de forma equitativa a los expertos, se asegura que ningún experto sea sistemáticamente mejor que los otros. Este comportamiento se puede apreciar en la Figura 6, procedente del *paper* del modelo *Mixtral 8x7B*. En dicha figura se puede observar cómo en distintas capas del modelo, los símbolos o *tokens* siempre son asignados a los mismos expertos y en la Figura 7 se observa como en un fragmento de texto, cada experto atiende a una parte diferente, dependiendo de su especialización.

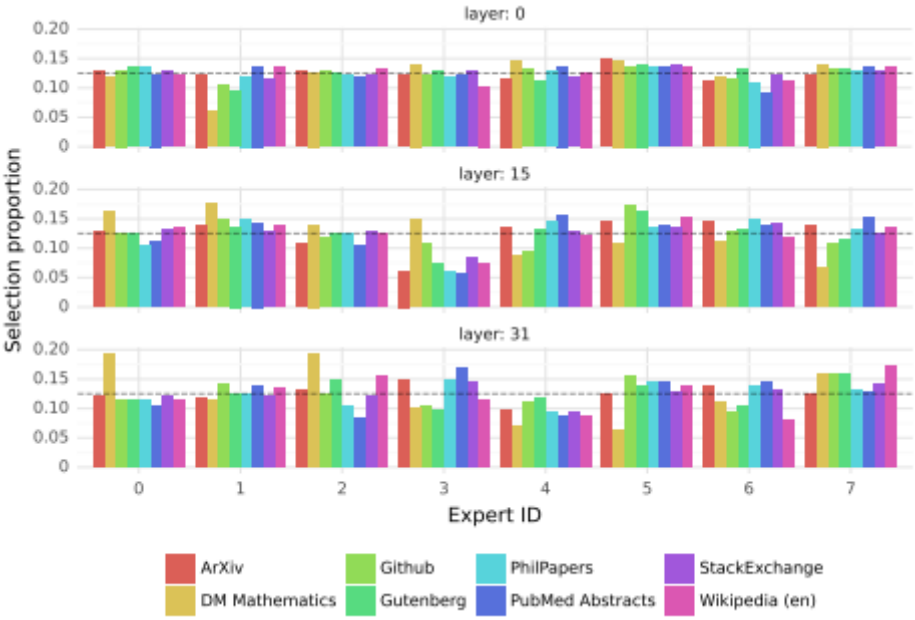

Figura 6 – Proporción de tokens asignados a cada experto durante el entrenamiento de Mixtral 8x7B. Fuente: [31]

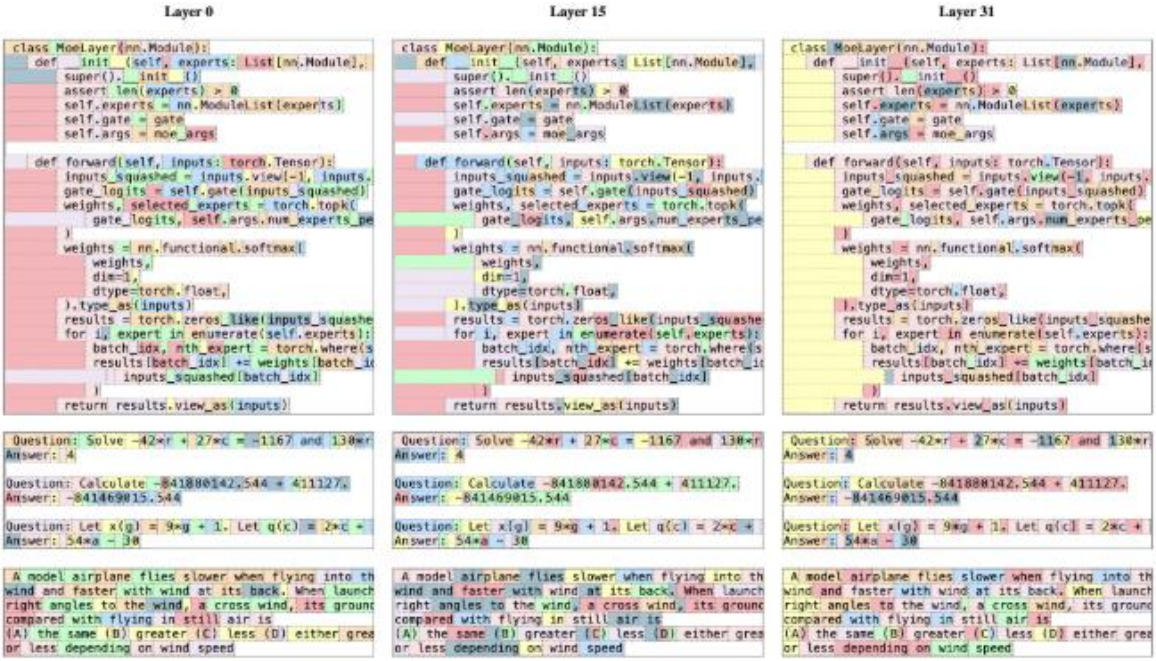

Figura 7 - Asignación de expertos a cada uno de los tokens en un fragmento de texto. Fuente: [31]

2.4 CUANTIZACIÓN

La memoria de acceso aleatorio (RAM) de video (VRAM) de las unidades de procesamiento gráfico (GPU) es un recurso crítico en el entrenamiento y despliegue de LLMs ya que se encarga de almacenar los datos necesarios para los cálculos realizados durante el entrenamiento y la inferencia, incluyendo los pesos del modelo y las activaciones neuronales intermedias. Las GPU están optimizadas para realizar operaciones de punto flotante (comunes en el entrenamiento de redes neuronales) a gran velocidad, aprovechando sus capacidades de cálculo en paralelo. No obstante, debido al elevado tamaño y complejidad de los LLM, se requieren grandes cantidades de VRAM, lo que puede limitar su implementación en dispositivos con recursos limitados. Es posible cargar el modelo en la CPU o incluso cargarlo compartiendo GPU y CPU, sin embargo, la velocidad de inferencia se reduce considerablemente.

Para abordar este desafío, se recurre a la cuantización de los pesos del modelo, lo que permite optimizar el uso de VRAM al coste de reducir la precisión del modelo. La cuantización consiste en utilizar un menor número de bits para representar y procesar los parámetros del modelo. Esto permite mejorar la velocidad de inferencia ya que requiere menos tiempo para realizar los cálculos. Sin embargo, la principal motivación de la cuantización es la reducción de los costes energéticos asociados a los LLM, por lo que se da gran importancia al estudio de las técnicas de cuantización.

La precisión en la que haya sido entrenado el modelo tendrá un impacto en su rendimiento, las representaciones de punto flotante más comunes en los modelos base son:

- FP32: Utiliza 32 bits para representar un parámetro y aunque proporciona un alto grado de precisión, la desventaja es su alto consumo computacional y huella de carbono.
- FP16: Utiliza 16 bits para representar un parámetro, esta precisión acelera los cálculos, pero puede tener un impacto en los números muy pequeños o grandes.
- BF16: Utiliza 16 bits también, pero la distribución de bits es diferente comparada con FP16. Este formato permite mantener la precisión para valores pequeños sin afectar al rendimiento y está especialmente diseñado para tareas de aprendizaje profundo o *deep learning.*

En la Figura 8 se encuentra representada la distribución de bits para cada una de las representaciones tratadas anteriormente:

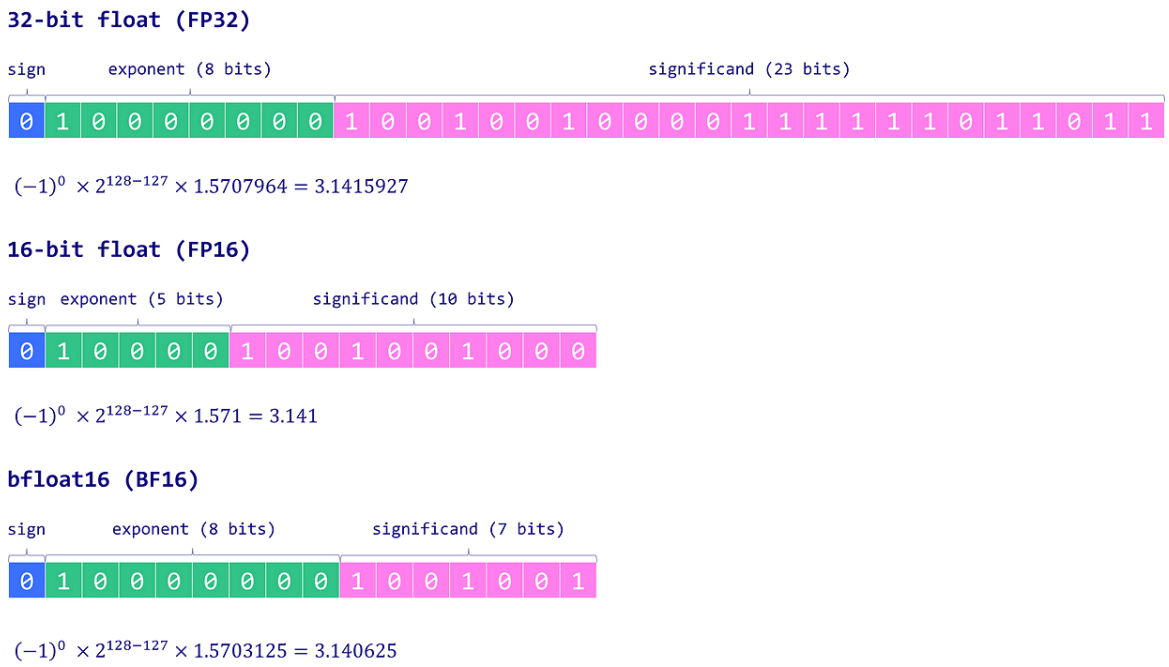

Figura 8 – Representación de la distribución de bits para cada una de las diferentes precisiones que presentan los LLM. Fuente: [47]

A la hora de cuantizar un modelo preentrenado, se suelen llevar a cabo dos técnicas diferentes:

- Post-Training Quantization (PTQ) o cuantización a *posteriori*: Esta técnica se aplica después de que el modelo haya sido entrenado, lo que la hace más sencilla y rápida ya que no se requiere de un entrenamiento adicional. Sin embargo, puede resultar en una pérdida de precisión significativa, especialmente si el modelo original es sensible a los cambios de precisión.
- Quantization-Aware Training (QAT) o entrenamiento dependiente de la cuantización: Esta cuantización se lleva a cabo durante el entrenamiento del modelo, de manera que se simula la cuantización en cada paso para aumentar su robustez ante la reducción de precisión. Esto permite al modelo adaptarse mejor a futuras cuantizaciones, pero implica un mayor tiempo de entrenamiento y uso de recursos computacionales.

La técnica más común es PTQ ya que se han desarrollado métodos de cuantización altamente eficientes que permiten reducir la precisión a ocho, cuatro e incluso hasta 2 bits sin perder apenas rendimiento. (Uno de los últimos avances en la cuantización ha conseguido cuantizar a 1.58 bits sin pérdidas drásticas de rendimiento [51]).

Capítulo 3. ESTADO DE LA CUESTIÓN

La Figura 9 permite obtener una visión global de las tendencias de los diferentes LLMs que han ido surgiendo desde el año 2022 hasta la actualidad (marzo 2024). En esta gráfica se compara el rendimiento del modelo frente al coste por cada millón de tokens de su API. Para evaluar el rendimiento del modelo se ha hecho uso de la comparativa *Massive Multitask Language Understanding* (MMLU) [7], una forma de evaluación de los modelos muy usada actualmente que permite evaluar la inteligencia del modelo en diferentes ámbitos académicos.

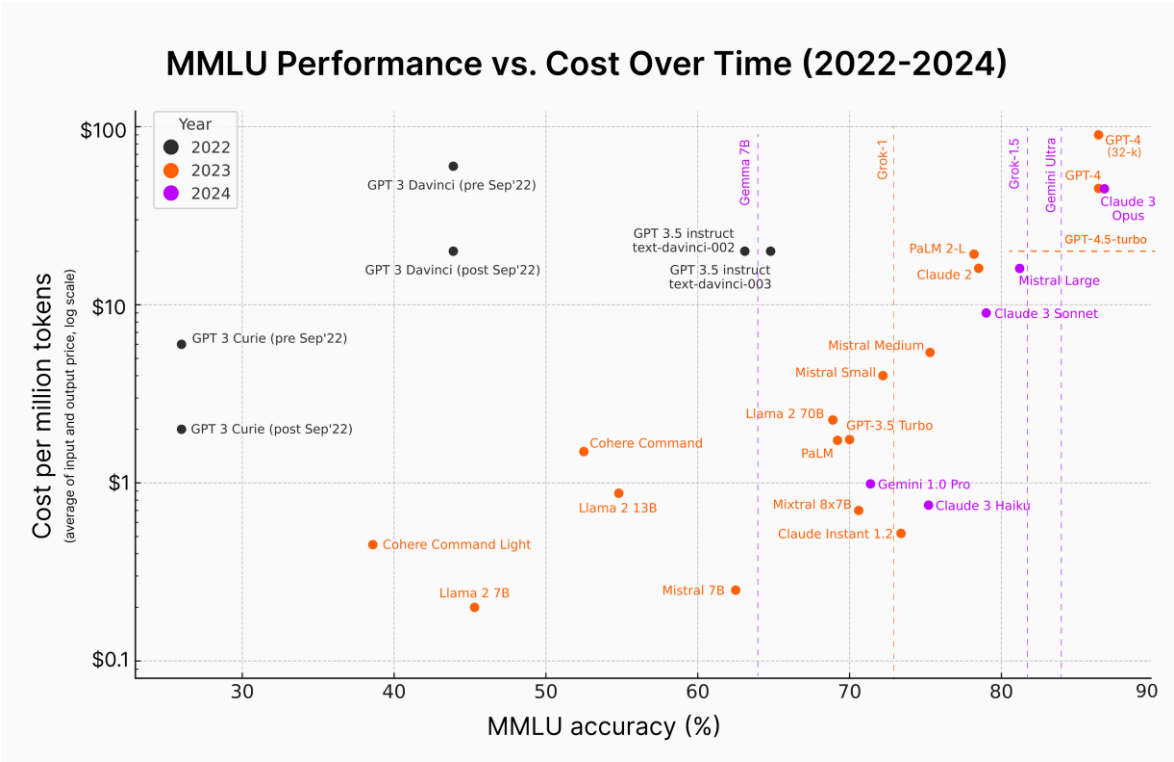

Figura 9 - Rendimiento de los LLM en el benchmark MMLU respecto a su coste por cada millón de tokens. (desde 2022 hasta marzo 2024). Fuente: [4]

Si se analiza la gráfica se puede observar una tendencia a aumentar el rendimiento de los modelos. Inicialmente los modelos eran costosos y propietarios (GPT 3), y rápidamente fueron superados por modelos de código abierto (Llama 2 7-13-70B, Mistral 7-8x7B) mucho más inteligentes y a un coste mucho menor. En la actualidad (2024) se observa el mismo comportamiento, los modelos más inteligentes (GPT 4, Claude) vuelven a tener un coste elevado, además, el estudio realizado revela otro aspecto de las tendencias de los

modelos: tanto los modelos comerciales como los de código abierto están convergiendo en un punto. En base a esta gráfica, cabría predecir que los modelos de código abierto alcanzarán o superarán a los modelos propietarios actuales. Avanzando tan solo un mes más, hasta mayo, se confirma la validez de las predicciones. Modelos como LLama3 [9] y Yi-1.5 [8] se acercan mucho al rendimiento de los modelos comerciales más potentes del mes anterior.

En esta sección del trabajo se realizará una breve descripción sobre los modelos de código abierto disponibles actualmente con el objetivo de obtener una visión global del ecosistema actual de los diferentes modelos *Open Source.*

3.1 MODELO FALCON

La familia de modelos *Falcon* [11], desarrollada por el Instituto de Tecnología de los Emiratos Árabes Unidos, se presenta en dos versiones diferentes. La más antigua, *Falcon* incluye cuatro modelos base con tamaños de 1.3B, 7.5B, 40B y 180B parámetros (B = miles de millones, *billions* en inglés) y la más reciente, *Falcon 2*, incluye un modelo de 11B. Los modelos de 40B e inferiores fueron entrenados en 1.5T (T (*trillion*) = billones) tokens y el modelo de 180B fue entrenado en 3.5T tokens de los cuales más del 80% pertenecen al conjunto de datos *RefinedWeb* [12], basado en *CommonCrawl* [13]. Ha sido entrenado en varios lenguajes (inglés, alemán, español y francés) y posee capacidades limitadas en italiano, portugués, polaco, rumano, checo y sueco. Una cualidad interesante de estos modelos es que utilizan atención multi-búsqueda o *multiquery attention* [14] que permite compartir pares de *embeddings* clave-valor en las diferentes capas de atención, lo que mejora la escalabilidad de la inferencia.

Todos los modelos excepto *Falcon* 180B están sujetos a la licencia *Apache 2.0* [28] la cual permite utilizar los modelos con amplia libertad. El modelo 180B puede usarse libremente para investigación o uso personal, pero requiere de un acuerdo comercial cuando se superan los beneficios totales de 1M$ al año y no permite utilizar el modelo como un servicio en la nube [15].

3.2 MODELO LLAMA

Los modelos *Llama,* desarrollados por la empresa *Meta,* se presentan en tres versiones diferentes: *Llama* [16]*, LLama2* [17] *y LLama3* [18], cada una más potente que la anterior. La versión más antigua tiene una licencia muy restrictiva y no se considera de código abierto. La versión *Llama 2* posee tres modelos base

de tamaños 7B, 13B y 70B entrenados en 2T tokens y con una ventana de contexto de 4096 tokens. La versión más reciente, *LLama3*, se presenta con dos modelos base de 8B y 70B (aunque ha anunciado que habrá una versión de 400B) de parámetros, con una ventana de contexto de 8.000 tokens. Fueron entrenados en un *dataset* de 15T tokens de datos dentro de los cuales se encuentran 10 millones de ejemplos anotados por humanos. Adicionalmente, un 5% del *dataset* consiste en información en 30 lenguajes diferentes.

Todos los modelos presentan versiones *fine-tuned* (preajustadas) para tareas de chat, código o seguimiento de instrucciones y han sido entrenados utilizando la técnica de atención a preguntas agrupadas o *Grouped Query Attention* [19], una versión mejorada de *multiquery attention* de los modelos *Falcon* que reduce el coste del mecanismo de atención y permite una ventana de contexto mayor. Adicionalmente, los modelos *Llama3-Instruct* utilizan aprendizaje reforzado con retroalimentación humana *Reinforcement Learning With Human Feedback* (RLHF) [44], una técnica que permite a los modelos aprender y ajustarse a partir de comentarios humanos obtenidos de las conversaciones con el modelo. Es decir, el modelo ajusta sus respuestas para alinearse mejor con lo que los humanos consideran correcto o útil.

En la Figura 10 se puede observar una comparación de las versiones *Llama2 y Llama3* en varios *benchmarks* [20]. Esta tabla muestra la gran diferencia entre ambas versiones.

Category	Benchmark	Llama 3 8B	Llama2 7B	Llama2 13B	Llama 3 70B	Llama2 70B
General	MMLU (5-shot)	66.6	45.7	53.8	79.5	69.7
	AGIEval English (3-5 shot)	45.9	28.8	38.7	63.0	54.8
	CommonSenseQA (7-shot)	72.6	57.6	67.6	83.8	78.7
	Winogrande (5-shot)	76.1	73.3	75.4	83.1	81.8
	BIG-Bench Hard (3-shot, CoT)	61.1	38.1	47.0	81.3	65.7
	ARC-Challenge (25-shot)	78.6	53.7	67.6	93.0	85.3
Knowledge reasoning	TriviaQA-Wiki (5-shot)	78.5	72.1	79.6	89.7	87.5
Reading comprehension	SQuAD (1-shot)	76.4	72.2	72.1	85.6	82.6
	QuAC (1-shot, F1)	44.4	39.6	44.9	51.1	49.4
	BoolQ (0-shot)	75.7	65.5	66.9	79.0	73.1
	DROP (3-shot, F1)	58.4	37.9	49.8	79.7	70.2

Figura 10 – Benchmark comparativo de los modelos LLama2 y LLama3 en varias métricas. Fuente: [18]

Los modelos *Llama* han sido entrenados con información pública disponible en internet y anotaciones humanas, lo que los hace especialmente útiles para la creación de asistentes virtuales. Además, estos modelos incorporan diversas medidas de seguridad para evitar respuestas inapropiadas y ponen un gran énfasis en la capacidad del modelo para rechazar solicitudes malintencionadas. Existe un modelo adicional denominado *LLama Guard 2* [21] que ha sido ajustado usando *Llama3* y que está enfocado en detectar respuestas que puedan ser consideradas poco seguras.

Respecto a la licencia [22] [23], ésta permite su redistribución y uso comercial siempre y cuando no se superen los 700 millones de usuarios activos mensuales. Además, si se utiliza el modelo *LLama3* para proyectos derivados se deberá atribuir incluyendo “Llama 3” al inicio del nombre y “Built with Meta Llama 3” en la documentación. Tampoco puede usarse para generar datos de entrenamiento para ajustar o mejorar otros LLM.

3.3 MODELO MPT

Los modelos MPT desarrollados por *MosaicML* están constituidos por un modelo base de 7B parámetros ajustado para seguimiento de instrucciones, conversación (*chat*) y lectura y creación de historias (este último con una ventana de contexto de 65.000 tokens) [25]. Fueron entrenados en 1T de tokens de datos y utilizan *FlashAttention* [26] *y FasterTransformer* [27] para optimizar el entrenamiento y la inferencia.

De los modelos disponibles, el modelo base y el modelo *StoryWriter* poseen la licencia Apache 2.0 [28], las versiones *Chat* e *Instruct* están sujetos a la licencia *CC-By-SA-3.0 y CC-By-NC-SA-4.0* [29] de las cuales, la última no permite su uso comercial.

3.4 MODELO MISTRAL

Los modelos *Mistral* de código abierto de la empresa *Mistral AI* [84] se presentan con dos arquitecturas diferentes: el modelo base *Mistral 7B* [30] basado en la arquitectura *Transformers y* los modelos *Mixtral 8x7B y 8x22B* [31], que utilizan la arquitectura MoE (*Mixture of Experts,* más información en el Capítulo 2.3). El modelo *Mistral 7B* posee una ventana de contexto de 32.000 tokens y utiliza las técnicas de *Grouped Query Attention* (GQA) [19] para mejorar la velocidad de inferencia y atención de ventana deslizante Sliding Window Attention (SWA) [35] para manejar secuencias largas de forma más eficiente. La técnica SWA permite

reducir el número de cálculos requeridos para calcular los pesos asignados en el mecanismo de atención ya que divide el texto en "ventanas" y calcula los pesos de las palabras dentro de la "ventana". Además, se superponen estas ventanas para que el modelo incluya el contexto de las ventanas adyacentes. Esto habilita a las capas superiores el acceso a información más distante en el pasado.

El modelo *Mixtral 8x7B* consiste en 8 expertos donde cada experto es un modelo *Mistral 7B* (de ahí su nombre). El tamaño total es de 45B parámetros, sin embargo, durante la inferencia solo utiliza 12.9B, lo que significa que posee la misma potencia que un modelo de 45B al coste de un modelo de 12.9B. Posee una ventana de contexto de 32.000 tokens y es capaz de generar textos en inglés, francés, italiano, alemán y español de forma fluida. El modelo *Mixtral 8x22B* es similar al modelo *Mixtral 8x7B*, pero utiliza expertos de 22B y cuenta con una ventana de contexto de 64.000 tokens. De los 141B de parámetros activos, utiliza solo 39B en la inferencia. El mayor problema de los MoE es que se requiere que todos los expertos estén cargados para poder realizar la inferencia, lo que supone un mayor coste de VRAM, aunque existen técnicas para des-cargar los expertos que no estén siendo usados [36]. Es importante considerar este último aspecto para decidir qué tipo de modelo utilizar. Los MoE son recomendables en escenarios en los que se trabaja con varias máquinas o se posee una elevada capacidad computacional, en casos de poca VRAM se recomienda utilizar modelos densos. Hasta la salida de *Llama3*, los modelos *Mixtral* se encontraban entre los de mejor rendimiento. En la Figura 11 se puede observar un *benchmark* entre la familia de modelos *Llama2 y Mistral,* demostrando el potencial de los modelos de la familia *Mistral* y de los MoE.

Model	Active Params	MMLU	HellaS	WinoG	PIQA	Arc-e	Arc-c	NQ	TriQA	HumanE	MBPP	Math	GSM8K
LLaMA 2 7B	7B	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	17.5%	56.6%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	13B	55.6%	80.7%	72.9%	80.8%	75.2%	48.8%	16.7%	64.0%	18.9%	35.4%	6.0%	34.3%
LLaMA 1 33B	33B	56.8%	83.7%	76.2%	82.2%	79.6%	54.4%	24.1%	68.5%	25.0%	40.9%	8.4%	44.1%
LLaMA 2 70B	70B	69.9%	**85.4%**	**80.4%**	82.6%	79.9%	56.5%	25.4%	**73.0%**	29.3%	49.8%	13.8%	69.6%
Mistral 7B	7B	62.5%	81.0%	74.2%	82.2%	80.5%	54.9%	23.2%	62.5%	26.2%	50.2%	12.7%	50.0%
Mixtral 8x7B	12B	**70.6%**	84.4%	77.2%	**83.6%**	**83.1%**	**59.7%**	**30.6%**	71.5%	**40.2%**	**60.7%**	**28.4%**	**74.4%**

Figura 11 – Benchmark comparativo de los modelos LLama2, Mistral-7B y Mixtral-8x7B en diferentes métricas. Fuente: [2]

Todos los modelos de código abierto de la familia *Mistral* están sujetos a la licencia Apache 2.0 [28] y poseen versiones ajustadas para seguimiento de instrucciones [37]. Destacan en la generación de código, resumen de textos, clasificación y completado de texto, además de su facilidad para el *fine-tuning.*

3.5 MODELO QWEN

Los modelos *Qwen* entrenados por la organización *Alibaba Cloud* se presentan en dos versiones, *Qwen* [38] *y Qwen 1.5* [39]. La versión *Qwen* cuenta con cuatro modelos de 1.8B, 7B, 14B y 72B parámetros preentrenados en 3T tokens de datos multilinguísticos en varios dominios, priorizando el inglés y el chino. Todos los modelos presentan una ventana de contexto de 32.000 tokens, excepto el modelo de 14B cuya ventana es de 8.000 tokens. Adicionalmente, ofrecen versiones ajustadas para *chat* y cuantizadas.

La versión *Qwen 1.5* es una mejora de su versión anterior que añade tres modelos más de 0.8B, 4B y 110B y un modelo MoE denominado *Qwen1.5-MoE-A2.7B.* Este último consiste en expertos de 1.8B obteniendo 14.3B parámetros totales y activando 2.7B en la inferencia, lo que le permite alcanzar el rendimiento de *Qwen1.5-7B* utilizando solo un cuarto de los recursos. Los modelos de esta versión poseen una ventana de contexto de 32.000 tokens y son capaces de comprender imágenes y audio, interpretar roles, actuar como agentes y RAG. En el entrenamiento se utilizaron las técnicas RoPE [40], y SwiGLU [41].

Los modelos *Qwen-72B, 14B y 7B* están sujetos a la licencia *Tongyi Qianwen LICENSE AGREEMENT* [42], la cual permite su comercialización siempre y cuando no se superen los cien millones de usuarios mensuales. El modelo *Qwen-1.8B* usa la licencia *Tongyi Qianwen RESEARCH LICENSE AGREEMENT* [43], la cual no permite su comercialización. En la versión 1.5 todos los modelos por debajo de 7B parámetros solo pueden ser usados para investigación o uso personal, el resto están sujetos a la licencia *Tongyi Qianwen LICENSE AGREEMENT* [42].

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

En el panorama actual de la IA, los LLM han demostrado ser herramientas transformadoras en el procesamiento del lenguaje natural. A pesar de su potencial, los modelos más avanzados han sido desarrollados por grandes corporaciones y están disponibles solo a través de servicios de pago o con restricciones de uso significativas. Es por esto que se prefiere el uso de modelos de código abierto ya que ofrecen el mejor balance entre coste y rendimiento. Además, promueven la transparencia, permitiendo comprender el funcionamiento interno del modelo y la naturaleza de los datos utilizados para su entrenamiento. Esto último resulta vital para evitar alucinaciones, información errónea o respuestas sesgadas en la salida del modelo.

El objetivo es proporcionar información práctica y estratégica que pueda ser utilizada por empresas, investigadores y desarrolladores para optimizar la selección y despliegue de modelos de IA en una amplia gama de aplicaciones y contextos. Esto representa una oportunidad para aprovechar el crecimiento explosivo en el campo de la IA y el procesamiento del lenguaje natural ya que conocer los últimos avances permite a las organizaciones desarrollar productos innovadores y competitivos para liderar el mercado.

Una de las últimas tendencias en el ámbito de los LLM es el desarrollo de agentes virtuales o *AI Agents* [81], por lo que este proyecto centrará gran parte del trabajo en estudiar este concepto y mostrar algunas de las técnicas utilizadas actualmente en su funcionamiento, como RAG. Se busca proporcionar una comprensión profunda y práctica del funcionamiento de esta técnica, de manera que pueda servir como base para el diseño de agentes virtuales o, en su defecto, para mejorar el rendimiento de los modelos de IA que las empresas ya tienen implementados.

4.2 OBJETIVOS

El proyecto tiene el objetivo de identificar el modelo de código abierto con el mejor equilibro entre coste y rendimiento para posteriormente utilizar el lenguaje Python y las librerías *Transformers* [69] y *LangChain* [70] para alcanzar los siguientes objetivos:

- Llevar a cabo un *fine-tuning* del modelo seleccionado en un conjunto de datos de preguntas y respuestas de medicina [46] para modificar el comportamiento del modelo.
- Desarrollar un sistema RAG sobre una serie de reportes médicos y estudiar la capacidad del modelo para mantener el contexto intra-documento e inter-documento.
- Estudiar el rendimiento del modelo en varias técnicas de recuperación avanzadas.
- Comparar las técnicas RAG y *fine-tuning.*
- Servir como herramienta para la toma de decisiones a la hora de implementar soluciones basadas en IA.

4.3 METODOLOGÍA

Debido al constante desarrollo de la tecnología de los LLM, resulta complicado establecer una planificación para el estudio de los modelos ya que constantemente aparecen nuevos modelos mejores que los anteriores, así como versiones mejoradas de los actuales. Por ello, se optó por seguir una metodología iterativa o ágil, en la cual se adaptan los recursos conforme se desarrollan nuevos modelos.

En la primera fase se llevó a cabo una revisión bibliográfica y un análisis de los modelos existentes para desarrollar un entendimiento de las diferentes soluciones disponibles y seleccionar aquel modelo que presente las mejores especificaciones. Además, se llevó a cabo un seguimiento continuo para poder registrar las últimas actualizaciones en la información, y las tendencias más recientes.

El modelo seleccionado fue aquel que, a la fecha de redacción de este proyecto, presentó las mejores especificaciones y el óptimo equilibrio entre coste y rendimiento. Para argumentar la elección, se examinaron los resultados del modelo en diversos *benchmarks,* su valoración por parte de la comunidad y sus características y especificaciones.

Con el modelo seleccionado se estudiaron las librerías, técnicas de cuantización y ajuste de hiperparámetros más adecuadas para llevar a cabo una prueba de concepto en la que se entrenó al modelo para que respondiera a las consultas del usuario con la personalidad de un doctor. Además, se evalúan los resultados del *fine-tuning,* tales como la precisión, la pérdida o error de entrenamiento y el comportamiento del modelo resultante.

Posteriormente se han estudiado las diferentes técnicas de recuperación disponibles para seleccionar aquellas más adecuadas para la interacción con reportes médicos. Las técnicas de recuperación seleccionadas fueron *Naïve, ParentDocument y Mulitquery*, y se especializaron en la recuperación de documentos con el objetivo de mejorar las respuestas del modelo. Las métricas utilizadas para evaluar los modelos se centraron en varios aspectos, como la precisión de las respuestas, la calidad, la relevancia o la utilidad de las respuestas. Además, se realizó un análisis comparativo entre las tres técnicas para determinar aquellas más adecuadas para diferentes casos de uso.

Finalmente, se ha llevado a cabo una comparativa detallada entre las técnicas de RAG y *fine-tuning* para mostrar las distintas ventajas y desventajas de ambas técnicas con el objetivo final de proporcionar una guía informada para la implementación de modelos de IA en aplicaciones específicas. Esta comparación abarca aspectos como el rendimiento, la eficiencia computacional, la facilidad de implementación, y la capacidad de manejar datos actualizados y específicos, ofreciendo así una visión integral para la toma de decisiones en proyectos futuros.

Capítulo 5. ELECCIÓN DEL MODELO

Esta sección del proyecto tiene como objetivo argumentar la elección del modelo seleccionado *LLama3,* más concretamente ***LLama3-8B-Instruct***, el cual se usará en la posterior prueba de concepto. La elección de este modelo se basa en sus destacadas capacidades en la comprensión y generación de lenguaje natural. A continuación, se detallan los métodos y criterios utilizados para llegar a esta decisión.

5.1 BENCHMARKS

En el ecosistema actual de los modelos LLM *OpenSource* se encuentra una gran variedad de modelos disponibles que ofrecen un punto de partida para desarrollar soluciones basadas en IA. Sin embargo, existe la necesidad de determinar el rendimiento de los diferentes modelos para decidir si uno es mejor que otro. Aunque se podría someter al modelo a algunas preguntas y evaluar personalmente si sus respuestas son satisfactorias, esta aproximación puede resultar subjetiva. Por lo tanto, resulta crucial recurrir a evaluaciones objetivas para obtener una medida imparcial del rendimiento del modelo y así fundamentar su calidad de manera más fiable. Para ello se hace uso de las tablas de *benchmarking,* que permiten evaluar el conocimiento general de los modelos, su capacidad de razonamiento, matemáticas, programación, etc.

Para llevar a cabo un *benchmark* del modelo existen dos enfoques diferentes:

- *Few-shot prompting:* En este enfoque el modelo recibe ejemplos de preguntas junto con sus soluciones dentro del mismo prompt. Ejemplo: "prompt: *Pregunta: ¿Cuánto es 10+12? Respuesta: 22, Pregunta: ¿Cuánto es 2+17? Respuesta: 19*, *Pregunta: ¿Cuánto es 29+3? Respuesta: ¿?*"

- *Zero-shot prompting:* En este caso el modelo recibe únicamente la pregunta, sin proporcionar ejemplos ni soluciones adicionales. Ejemplo: "prompt: *Pregunta: ¿Cuánto es 29+3? Respuesta: ¿?*"

Ambos enfoques tienen sus ventajas y limitaciones, y la elección del enfoque dependerá de los objetivos específicos y del contexto del entrenamiento del modelo.

A continuación se describen los principales *benchmarks* utilizados en la evaluación de *Llama3-8B-Instruct,* indicados en la tabla de *benchmarking* de la Figura 12.

	Meta Llama 3 8B	Gemma 7B - It Measured	Mistral 7B Instruct Measured
MMLU 5-shot	68.4	53.3	58.4
GPQA 0-shot	34.2	21.4	26.3
HumanEval 0-shot	62.2	30.5	36.6
GSM-8K 8-shot, CoT	79.6	30.6	39.9
MATH 4-shot, CoT	30.0	12.2	11.0

Figura 12 – Benchmark comparativo de los modelos LLama3-8B-Instruct, Gemma-7b-Instruct y Mixtral-7B-Instruct en diferentes métricas. Fuente: [53]

MMLU (Massive Multitask Language Understanding): El *benchmark* MMLU [7] consiste en una prueba o examen que abarca 57 materias diferentes, incluyendo matemáticas, STEM, humanidades, o ciencias sociales, entre otras. La dificultad varía desde un nivel de primaria hasta un nivel profesional avanzado, evaluando tanto el conocimiento del mundo como la capacidad de resolución de problemas. *LLama3* utiliza *MMLU 5-shot,* lo que signifca que se le presentan al modelo cinco problemas y después se le pide resolver uno adicional. Se evalúa el porcentaje de respuestas correctas. Este *benchmark* es usado como referencia por muchos modelos para evaluar su rendimiento general.

GPQA (Graduate-Level Google-Proof Q&A): Este *benchmark* [54] se compone de 448 preguntas de opción múltiple escritas por expertos en biología, física y química. Estas preguntas están diseñadas para ser extremadamente complejas,

de forma que no puedan ser resueltas aun teniendo acceso a internet. De hecho, modelos más avanzados como GPT-4 solo han conseguido alcanzar un 39% en esta prueba. El hecho de que el modelo *Llama3* haya sido capaz de alcanzar un resultado de 34.2% demuestra una notable capacidad para manejar tareas complejas. No obstante, este *benchmark* también demuestra el margen de mejora que presentan los modelos de IA.

HumanEval: A pesar de su nombre, este *benchmark* [55] está diseñado para evaluar las habiliades de programación de un modelo. Según los datos de evaluación de *Llama3* [20], el modelo ha sido evaluado en función de la métrica *pass@k* [55]. Esta métrica indica la probabilidad de que al menos una de las respuestas generadas por el modelo entre las primeras k sea correcta. Es decir, el modelo tiene k oportunidades para generar una respuesta, *pass@k* medirá el porcentaje de veces que al menos una de ésas k oportunidades es correcta. En el caso de *Llama3*, se utiliza *pass@1* = 62.2, lo que indica que hay un 62.2% de probabiliad de que la primera respuesta generada por el modelo sea correcta.

GSM-8k (Grade School Math 8k): Este *benchmark* [56] se compone de un *dataset* de 8.500 problemas de matemáticas de educación primaria. Estas preguntas evaluan la capacidad del modelo para resolver problemas matemáticos que requieren varios pasos para ser resueltos. *Llama3* utiliza la métrica *maj@1* mostrada en el *paper*, en la que se evalúa la probabilidad dé que el modelo de una respuesta que se encuentra en las respuestas correctas más frecuentes. En la página 35 de la publicación original se encuentran algunos ejemplos *few-shot.*

Math: Se compone de un conjunto de datos o *datatset* [57] de 12.500 preguntas de matemáticas de nivel competición. Cada problema tiene una solución paso a paso que puede ser usada para enseñar a los modelos a generar resoluciones y explicaciones de respuestas. *Llama3* utiliza la misma métrica que en GSM-8k, *maj@1.*

5.2 CHATBOT ARENA

Es importante considerar que cada organización o grupo de investigación tiene la autonomía para determinar cuáles *benchmarks* proporcionan los resultados más óptimos para su modelo y utilizarlos para resaltar sus capacidades y rendimiento. Aunque existen *benchmarks* comunmente utilizados, uno de los mas importantes es la evaluación humana, donde se analiza la calidad del modelo mediante una valoración directa de expertos o usuarios finales. Esta evaluación cualitativa es fundamental, ya que proporciona una perspectiva

integral sobre la efectividad y aplicabilidad del modelo en situaciones del mundo real, más allá de los parámetros cuantitativos obtenidos del *benchmark.* Existe una plataforma muy conocida denominada *Chatbot Arena* [58] [59] donde se evalúa a los modelos basandose en preferencias humanas. Los modelos mejor valorados se muestran ordenados por puntuación (*ranking*), y a menudo aquellos que ocupan las primeras posiciones suelen ser los que tienen un mayor rendimiento. En la Figura 13 se muestra la clasificación del ranking a finales de Mayo 2024, donde se puede observar al modelo *Llama3* como el modelo de código abierto mejor valorado.

Rank* (UB)	Model	Arena Elo	95% CI	Votes	Organization	License	Knowledge Cutoff
1	GPT-4o-2024-05-13	1287	+4/-4	26899	OpenAI	Proprietary	2023/10
2	Gemini-1.5-Pro-API-0514	1268	+5/-4	20181	Google	Proprietary	2023/11
2	Gemini-Advanced-0514	1267	+4/-4	22132	Google	Proprietary	Online
4	Gemini-1.5-Pro-API-0409-Preview	1258	+3/-3	55731	Google	Proprietary	2023/11
4	GPT-4-Turbo-2024-04-09	1256	+3/-3	58147	OpenAI	Proprietary	2023/12
5	GPT-4-1106-preview	1252	+2/-3	78286	OpenAI	Proprietary	2023/4
6	Claude 3 Opus	1248	+3/-3	118351	Anthropic	Proprietary	2023/8
6	GPT-4-0125-preview	1246	+3/-2	71547	OpenAI	Proprietary	2023/12
9	Yi-Large-preview	1239	+3/-4	30787	01 AI	Proprietary	Unknown
9	Gemini-1.5-Flash-API-0514	1232	+4/-6	18317	Google	Proprietary	2023/11
11	Bard (Gemini Pro)	1208	+7/-5	11853	Google	Proprietary	Online
11	Llama-3-70b-Instruct	1208	+3/-2	118874	Meta	Llama 3 Community	2023/12
12	Claude 3 Sonnet	1201	+3/-2	92706	Anthropic	Proprietary	2023/8

Figura 13 – Disposición del ranking "ChatBot arena" para la fecha del 30 de mayo de 2024. Fuente: [60]

Al analizar la clasificación, se puede observar que el modelo *Llama3* lidera como el mejor modelo de código abierto. Se observa que la versión *8B-Instruct* se posiciona por encima de modelos con un tamaño cinco veces mayor, como el modelo *Mistral 8x7B.* Además, éste supera a modelos comerciales como *GPT3.5-Turbo* o *Gemini Pro.* Por otro lado, el modelo de 70B parámetros supera a la primera versión de *GPT4* y a *Claude 3 Sonnet,* y destaca por haber sido desarrollado a un costo significativamente menor.

Adicionalmente existe otra clasificación [75] donde se evaluan modelos que han sido ajustados por la comunidad. Este ranking evalua todos los modelos en las mismas seis *benchmarks* diferentes, lo que permite estandarizar la evaluación de los modelos. Además, ofrece una retrospectiva acerca del rendimiento de los diferentes modelos ajustados y qué modelos base proporcionan mejores resultados.

5.3 CARACTERÍSTICAS

La versión *Llama3* utiliza la estructura estándar de los *transformers,* al igual que las versiones anteriores de *Llama.* Sin embargo, esta versión utiliza un *tokenizador* con un tamaño de vocabulario de 128.000 *tokens*, lo que permite reconocer un mayor número de palabras únicas o *tokens.* Esto incluye palabas raras, refranes, frases y otras estructuras lingüísticas. Al tener un vocabulario tan amplio, el número de tokens necesarios para analizar una frase se reduce considerablemente, ya que un *token* puede comprender varias palabras en una misma frase. Esta característica mejora significativamente el rendimiento del modelo. Adicionalmente, posee múltiples medidas de seguridad para evitar respuestas con riesgos en el ámbito CBRNE (*Chemical, Biological, Radiological, Nuclear and high yield Eplosives)* y en ciberseguridad. Además, incluye instrucciones para rechazar *prompts* aparentemente benignos que puedan ser malinterpretados o utilizados de manera inapropiada.

Sin modificar la arquitectura clásica de los modelos *Llama, LLama3* ha obtenido excelentes resultados en los *benchmark* evaluados. El secreto detrás del elevado rendimiento del modelo *Llama3* radica en su entrenamiento, más concretamente en la calidad y en la cantidad de los datos utilizados. Al aumentar la cantidad de datos de programación, aumenta en gran medida la capacidad de razonamiento del modelo, como se demuestra en un estudio [63] reciente sobre el impacto de los datos de programación en la capacidad de razonamiento de los LLM.

Sin embargo, la principal razón del elevado rendimiento reside en la cantidad de datos utilizados para su entrenamiento. En un estudio [61] realizado por la compañía *Google DeepMind* se investigó la relación entre el número de parámetros de un modelo y la cantidad de datos con los que debe ser entrenado para alcanzar la rentabilidad óptima. En este estudio se entrenaron modelos en un rango de 70B a16B parámetros en 5B a 500B tokens de datos para encontrar este óptimo y probarlo en un modelo conocido como *Chinchilla.* En la tabla de la Figura 14 se muestran los resultados.

Parameters	FLOPs	FLOPs (in *Gopher* unit)	Tokens
400 Million	1.92e+19	1/29,968	8.0 Billion
1 Billion	1.21e+20	1/4,761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

Figura 14 – Estimación del número de FLOPs y tokens de entrenamiento necesarios para optimizar el coste de entrenamiento de modelos según su tamaño. Fuente: [61]

En esta tabla se muestra la cantidad óptima de *tokens* con los que entrenar un modelo para alcanzar el equilibrio entre rendimiento y coste. Utilizar una cantidad mayor de *tokens* a la indicada haría que el entrenamiento no fuese rentable, y se debería considerar el uso de un modelo con más parámetros. El modelo *Llama3* fue entrenado con 15T *tokens* de datos, superando por mucho el óptimo recomendado. Esto demuestra que la empresa *Meta* gastó una suma considerable de dinero en el entrenamiento de los modelos *LLama3* para obtener modelos con un elevado rendimiento y un bajo coste de inferencia. A largo plazo, resulta una inversión ya que la comunidad podrá mejorar el modelo debido a su asequibilidad. Esta nueva metodología puede suponer un nuevo punto de inflexión en cuanto al diseño de los LLM.

5.4 ESPECIFICACIONES Y REQUISITOS HARDWARE

Para determinar el coste computacional de un modelo es necesario conocer el peso de sus parámetros. Para ello es necesario conocer la precisión a la que fue entrenado y el número de parámetros que posee (Más información en el Capítulo 2.4). Por ejemplo, un modelo de 7B parámetros entrenado con una precisión FP32 (4 bytes) requerirá de

$$7 * 10^9\ parámetros * 4\ bytes = 28\ GB\ VRAM$$

En el caso del modelo *LLama3*, los tensores del modelo se encuentran en una precisión BF16, donde cada parámetro BF16 pesa 2 bytes o 16 bits. Por lo tanto,

el modelo *LLama3-8B* requerirá aproximadamente 16 GB de memoria VRAM para cargarlo completamente.

$$8 * 10^9\ parámetros * 2\ bytes = 16\ GB$$

Es recomendable llevar a cabo un análisis detallado de la capacidad computacional disponible al escoger el modelo a utilizar, ya que esto tendrá un impacto significativo en la velocidad del modelo.

Para este proyecto se dispone de una tarjeta gráfica de 64GB RAM, I7 3ª Gen RTX 4070 TI con 12 GB VRAM por lo que será necesario cuantizar el modelo seleccionado (*Llama3-8B-Instruct)* para poder cargarlo completamente en la memoria VRAM (GPU).

Capítulo 6. PRUEBA DE CONCEPTO

En esta sección, se presenta una prueba de concepto destinada a explorar la eficacia y la aplicabilidad de dos enfoques clave al trabajar con LLMs: el ajuste fino o *fine-tuning* y la técnica RAG. La prueba de concepto se centra en evaluar cómo estas técnicas pueden mejorar el rendimiento de un LLM, en este caso el modelo seleccionado anteriormente *LLama3-8B-Instruct*, en tareas específicas.

6.1 FINE-TUNING

La técnica del *fine-tuning* permite adaptar un modelo preentrenado a una tarea específica mediante el reentrenamiento de una pequeña parte de sus parámetros en un *dataset* más específico. Este método resulta ideal para mejorar la capacidad del modelo en tareas específicas sin tener que reentrenar un modelo desde cero. Además, permite mejorar el rendimiento del modelo en términos de precisión, conocimiento e incluso especificaciones, como el aumento de ventana de contexto o la censura de ciertas respuestas del modelo. Sin embargo, supone un coste técnico y computacional mayor, ya que se requiere de ajustar múltiples hiperparámetros, como la tasa de aprendizaje, el tamaño de lote (*batch-size*) y el número de épocas. Esto puede ser un desafío y requerir de un conocimiento experto para obtener los mejores resultados dado que no existen métodos estandarizados. Además, múltiples ajustes finos pueden causar que el modelo derive de su rendimiento original.

Para esta prueba de concepto, se utilizó un *dataset* [46] de aproximadamente 100.000 conversaciones reales entre pacientes y doctores. El objetivo es realizar un ajuste fino al modelo en dicho *dataset* para ajustar su comportamiento y adecuar sus respuestas de modo que se asemejen a las de un doctor. Con esto se busca demostrar el potencial de esta técnica para ajustar el comportamiento de los modelos al uso deseado. Se utilizó el servicio *HuggingFace* [65], un servicio de versionado de modelos similar a *GitHub*, para descargar los pesos del modelo y el *dataset*. Para monitorizar el entrenamiento del modelo y visualizar los resultados se hizo uso de la plataforma *Weights & Biases* [64]. Esta plataforma ofrece herramientas para visualizar las métricas en tiempo real, lo que permitió identificar cualquier problema y ajustar los hiperparámetros de manera eficiente. Todas las funciones y la lógica para llevar a cabo el *fine-tuning* se basó en la librería *Transformers* [69], ofrecida por *HuggingFace.*

Para este proyecto se realizó un PTQ (Capítulo 2.4) haciendo uso del método de cuantización *QLoRA* [49]. Este método reduce el rango de las matrices de autoatención utilizando precisiones de 8 y 4 bits, lo que ayudará a reducir la complejidad computacional y el consumo de memoria al limitar el número de parámetros entrenables. Además, también incorpora regularización y *fine-tuning* para mitigar la pérdida de rendimiento asociada a la cuantización, garantizando que el modelo mantenga un alto nivel de rendimiento.

Una pregunta importante antes de realizar el *fine-tuning* es qué versión del modelo seleccionado utilizar. La mayoría de los modelos ofrecen una versión base del modelo y una versión ajustada para seguimiento de instrucciones, denominada *Instruct.* El modelo base es más adecuado en situaciones donde se requiere una adaptación considerable del comportamiento del modelo para tareas muy específicas. En cambio, el modelo *Instruct* es preferible cuando la tarea principal implica seguir instrucciones humanas, proporcionando un rendimiento inicial superior en este tipo de tareas. El *dataset* que se utilizará para esta prueba de concepto se compone de preguntas médicas específicas y respuestas detalladas. Por lo tanto, la versión más adecuada del modelo es la versión ***Instruct,*** ya que está afinada para seguir instrucciones y proporcionar respuestas útiles, lo que hará que el *fine-tuning* sea más eficiente.

A continuación, se detallan los pasos a seguir para llevar a cabo el *fine-tuning.*

6.1.1 PREPARACIÓN DEL DATASET

Los modelos *instruct,* al haber sido entrenados en un formato diferente, suelen seguir un formato especial para la conversación entre el usuario y el modelo. Es importante seguir este formato para obtener el mejor rendimiento y aprovechar todas las características de los modelos *instruct.* En el caso de *LLama3-Instruct*, el formato es el siguiente:

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

{{ system_prompt }}<|eot_id|><|start_header_id|>user<|end_header_id|>

{{ user_msg_1 }}<|eot_id|><|start_header_id|>assistant<|end_header_id|>

{{ model_answer_1 }}<|eot_id|>
```

Debido a que se utilizará un *dataset* de preguntas y respuestas, es necesario preparar el *dataset* para que se ajuste a dicha plantilla. En algunos modelos, la plantilla viene incluida en la configuración del *tokenizador* [67] y se puede implementar como una función que *tokenize* el *dataset* con el formato adecuado. En este caso, el *dataset* seleccionado [46] ya se encuentra en el formato

adecuado, por lo que no es necesario llevar a cabo un preprocesamiento del *dataset.* A continuación, se muestra un ejemplo de una de las consultas contenidas en el *dataset.*

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|> You are a helpful
assistant helping individuals with their medical queries

<|eot_id|><|start_header_id|>user<|end_header_id|> I am 57 yrs old and have
been taking nebicard sm daily to control my BP. My cholesterol level is 233
and Triglylcerides is 211. I am not taking any other medicines. Kindly let me
know if the triglycerides levels go up because of the BP medicine and if so
what other medicine can I take as a substitute?

<|eot_id|><|start_header_id|>assistant<|end_header_id|> Hi & Welcome. Nebicard
SM contains a beta blocker (Nebivolol) & a calcium channel blocker Amlodipine
commonly prescribed to treat high BP. Increased cholesterol & triglycerides
appear to have been caused by Nebivolol. Beta blockers like Nebivolol can
affect your cholesterol & triglyceride levels causing a slight increase in
their levels. These changes are often temporary. You shouldn't abruptly stop
taking a beta blocker because doing so could increase your risk of a heart
attack or other heart problems. Consult your treating doctor for an alternate,
but a safer medication like Losartan to lower your high BP. Hope this will
help. <|eot_id|>
```

Se observan tres roles diferentes, el '*system'* que corresponde con el *system prompt* o la instrucción principal del modelo, el '*user'* que corresponde con la consulta del usuario y por último el '*assistant'* que será la respuesta del modelo.

Un parámetro de los modelos cuyo ajuste es muy importante es la longitud máxima de secuencia, este parámetro representa el número máximo de *tokens* que el modelo puede procesar en una entrada e influye directamente en la cantidad de memoria necesaria para procesar una entrada. Para manejar secuencias de diferentes longitudes, es común usar *padding (*relleno) para las secuencias más cortas y *truncation* (truncamiento) para las más largas, con el objetivo de que todas las secuencias tengan la misma longitud. La forma óptima de determinar esta longitud es hacerlo en base a la distribución del *datatset.* Para ello, se *tokenizará* primero el *dataset* sin añadir relleno ni truncamiento, lo que permitirá analizar la distribución de las longitudes de las secuencias y establecer una longitud adecuada.

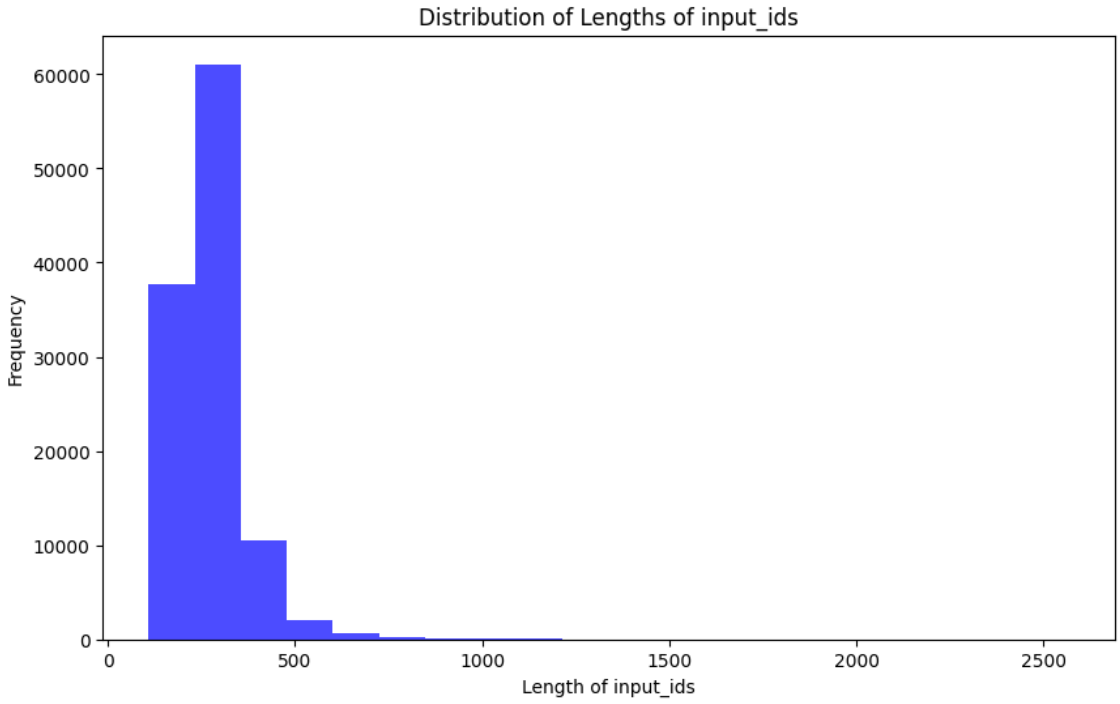

Figura 15 – Distribución de las muestras del dataset tokenizadas según su frecuencia de aparición. Fuente: elaboración propia

Como se puede observar en la Figura 15, la mayoría de las muestras poseen una longitud inferior a 500, por lo que se utilizará un valor de longitud máxima de secuencia de **512.** Tras volver a *tokenizar* el *datatset* aplicando el truncamiento y el relleno, el resultado es el mostrado en la Figura 16.

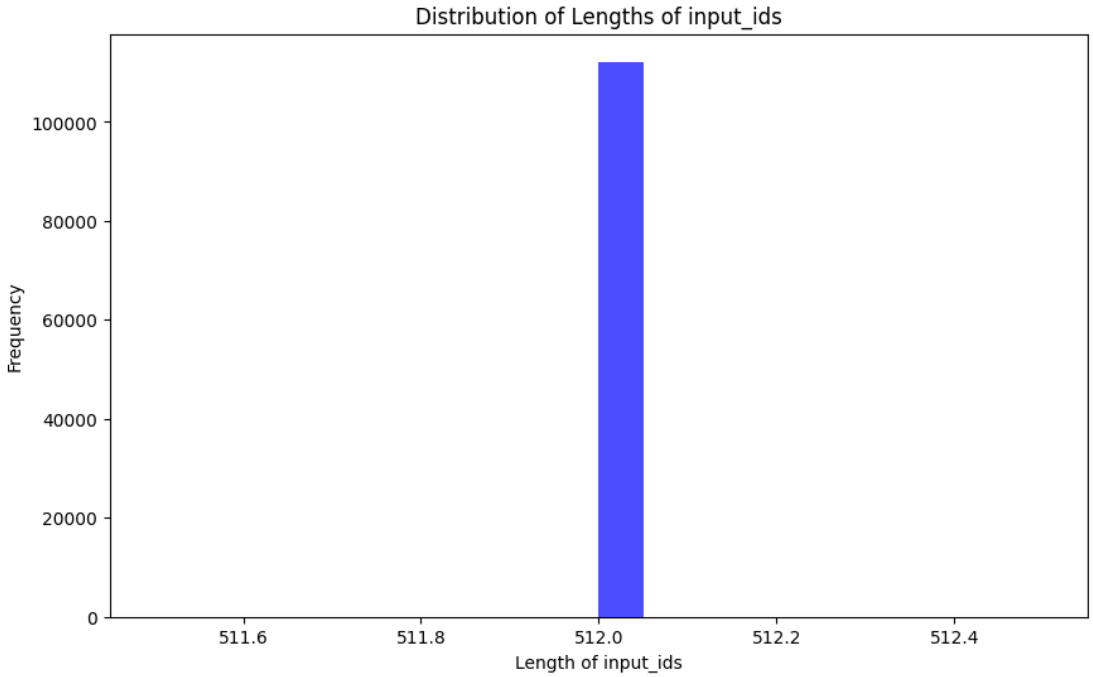

Figura 16 – Distribución de las muestras del dataset tokenizadas según su frecuencia de aparición tras aplicar truncamiento (truncation) y relleno (padding) a la longitud de 512 tokens. Fuente: elaboración propia.

Con esta nueva distribución de las muestras, el modelo podrá procesar lotes de secuencias de diferentes longitudes sin problemas.

6.1.2 CONFIGURACIÓN DEL MODELO

Como se ha mencionado en el capítulo 5.4, es necesario realizar una cuantización del modelo para poder llevar a cabo el *fine-tuning* con el equipo disponible. En concreto, se realizará una cuantización a cuatro bits, ya que el método QLoRA [49] reduce de forma altamente eficiente el uso de memoria durante el *fine-tuning,* permitiendo cuantizar modelos a 4 bits sin apenas pérdidas de rendimiento. Al cuantizar a 4 bits, el requisito de VRAM se reduce a:

$$8 * 10^9\ parámetros * 0.5\ bytes = 4\ GB$$

Este método emplea un tipo de datos para almacenamiento (normalmente *4-bit NormalFloat)* y otro tipo para la computación (16-bit *BrainFloat),* lo que permite una mayor velocidad de entrenamiento. Para llevar a cabo la cuantización se utilizó la librería *BitsAndBytes* [68] con la siguiente configuración:

```
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
```

```
    bnb_4bit_compute_dtype=torch.bfloat16,
    bnb_4bit_use_double_quant=True
)
```

El formato “nf4” implica que los valores de los bits son representados en una distribución normal y distribuidos de manera que puedan ser representados con alta precisión incluso en un formato de baja precisión como es cuatro (4) bits. Durante los cálculos, los pesos cuantizados en “nf4” se descuantizan en un formato de mayor precisión, en este caso 16-bit *bfloat.* Adicionalmente se realiza una segunda cuantización tras la primera para ahorrar 0.4 bits por parámetro.

6.1.3 SFT (SUPERVISED FINE-TUNING)

Para este proyecto se realizará un *fine-tuning* supervisado, es decir, se utilizarán datos etiquetados que han sido comprobados previamente. Para ello se hará uso de la subclase “SFTTrainer” de la clase “Trainer” de la librería “trl” [66], ofrecida por HuggingFace. Esta clase será la encargada de actualizar los parámetros, preparar el *dataset*, añadir la configuración de LoRA [73] y preparar el modelo y su tokenizador para el entrenamiento entre otras cosas. La configuración utilizada es la siguiente:

```
trainer = SFTTrainer(

    model = base_model,

    train_dataset = tokenized_train_dataset,

    eval_dataset = tokenized_eval_dataset,

    peft_config = peft_config,

    dataset_text_field = "text" # Fiel name in training dataset containing
data,

    max_seq_length = 512,

    tokenizer = tokenizer,

    args = training_arguments,

    packing = False,

)
```

Esta configuración permite inicializar la clase “SFTTrainer” con todos los componentes necesarios para llevar a cabo el *fine-tuning* de un modelo preentrenado (*LLama3-8B-Instruct).* El *dataset* seleccionado anteriormente será dividido en un 80% para entrenamiento y un 20% para validación. Se utiliza la longitud máxima de secuencia seleccionada en el apartado 6.1.1 y se establece

el parámetro “packing” a *False*. Este parámetro se encarga de empaquetar las secuencias del *dataset* a la longitud máxima debido a que este paso ya ha sido realizado en la preparación del *dataset*, no es necesario que la clase “SFTTrainer” realice el trabajo.

6.1.3.1 Configuración de LoRA

LoRA pertenece a la familia de los métodos PEFT [72] (*Parameter Efficient Fine-Tuning).* Estos métodos resultan cruciales para poder llevar a cabo el *fine-tuning* cuando no se dispone de una elevada capacidad computacional. Mientras que el entrenamiento completo de un LLM entrena todos los parámetros del modelo, PEFT consiste en entrenar únicamente un subconjunto de parámetros (normalmente entre el 15-20% de los pesos originales) elegidos estratégicamente y “congelar” el resto, lo que ahorra una gran cantidad de memoria en los cálculos de gradientes y costes de almacenamiento en VRAM. Las técnicas de PEFT más avanzadas son capaces de obtener el mismo rendimiento que un *fine-tuning* completo.

Una de las técnicas PEFT más utilizadas es LoRA (Low-Rank Adaptation) [73] que consiste en descomponer las transformaciones lineales del modelo en otras de menor rango. Los pesos del modelo se almacenan en matrices con un determinado rango, LoRA divide cada una de estas matrices en productos de matrices más pequeñas de menor rango. Durante el entrenamiento, en lugar de ajustar todos los elementos de la matriz original, se ajustan solo los de las matrices pequeñas. Esto se basa en la idea de que muchos parámetros en los LLM tienen un impacto menor y, por lo tanto, no necesitan ser ajustados con precisión. Una vez se entrenan estas matrices pequeñas, se vuelven a multiplicar y se añaden al modelo original, lo que garantiza que la arquitectura del modelo original no se vea afectada. En la Figura 17 se ilustra este procedimiento.

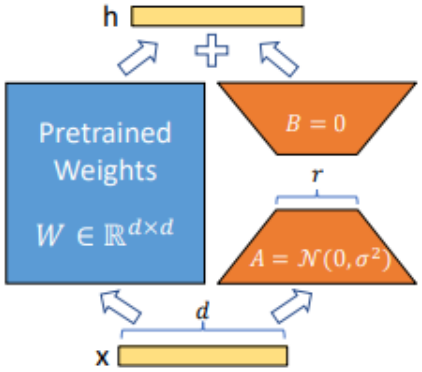

Figura 17 – Reparametrización de LoRA, durante el entrenamiento solo se actualizan las submatrices A y B. Fuente: [73]

La configuración de LoRA utilizada es la siguiente:

```
peft_config = LoraConfig(
    lora_alpha = 64,
    lora_dropout = 0.05,
    r = 32,
    target_modules=[
        "q_proj",
        "k_proj",
        "v_proj",
        "o_proj",
        "gate_proj",
        "up_proj",
        "down_proj",
        "lm_head",
    ],
    bias="none" # Determines how bias terms are handled in LoRA,
    task_type="CAUSAL_LM",
)
```

Los parámetros clave son el rango “r” y alfa “lora_alpha” que determinarán el tamaño de las matrices y el comportamiento del *fine-tuning.*

- El parámetro “r” controla el rango de las submatrices generadas y, por ende, determina el número de parámetros que serán entrenados. Un rango más alto permite capturar relaciones más complejas en los pesos originales, mientras que un valor más bajo puede limitar la habilidad del modelo para adaptarse.
- El parámetro “lora_alpha” es un factor de escalado para los pesos de las matrices, se utiliza para ajustar la contribución de las matrices de bajo rango.

Se seleccionaron los valores *r = 32* y *lora_alpha = 64* para dar más énfasis a los nuevos datos, al mismo tiempo que se reduce la complejidad computacional. En el caso de *LLama3,* las matrices tienen un tamaño de 4096x4096 [74], los que significa que tienen aproximadamente 16 millones de parámetros cada una.

Aplicando LoRA con un valor de *r=32*, se dividen en dos matrices de 4096x32 y 32x4096, en total:

$$4096 * 32 + 32 * 4096 = 262.144\ parámetros$$

Esto representa aproximadamente un 1.56% del tamaño original de la matriz W.

$$\frac{262.144}{16.777.216} \approx 1.56\%$$

Otro parámetro importante es los módulos diana (*target_modules),* el cual representa los módulos del modelo que serán entrenados. Éstos dependen de cada modelo y se pueden ver en la configuración interna del modelo. En el caso de *LLama3* los módulos son los siguientes:

```
LlamaForCausalLM(

  (model): LlamaModel(

    (embed_tokens): Embedding(128256, 4096)

    (layers): ModuleList(

      (0-31): 32 x LlamaDecoderLayer(

        (self_attn): LlamaSdpaAttention(

          (q_proj): Linear4bit(in_features=4096, out_features=4096,
bias=False)

          (k_proj): Linear4bit(in_features=4096, out_features=1024,
bias=False)

          (v_proj): Linear4bit(in_features=4096, out_features=1024,
bias=False)

          (o_proj): Linear4bit(in_features=4096, out_features=4096,
bias=False)

          (rotary_emb): LlamaRotaryEmbedding()

        )

        (mlp): LlamaMLP(

          (gate_proj): Linear4bit(in_features=4096, out_features=14336,
bias=False)

          (up_proj): Linear4bit(in_features=4096, out_features=14336,
bias=False)

          (down_proj): Linear4bit(in_features=14336, out_features=4096,
bias=False)
```

```
          (act_fn): SiLU()
        )
        (input_layernorm): LlamaRMSNorm()
        (post_attention_layernorm): LlamaRMSNorm()
      )
    )
    (norm): LlamaRMSNorm()
  )
  (lm_head): Linear(in_features=4096, out_features=128256, bias=False)
)
```

6.1.3.2 Configuración del entrenamiento

La parte más importante del *fine-tuning* consiste en ajustar los distintos hiperparámetros. La clase *"TrainingArguments"* [71] de la librería *Transformers* [69] permite configurar los distintos hiperparámetros del proceso de *fine-tuning.* La configuración de esta clase depende en gran medida del modelo y el *dataset* que se utilicen y si requiere de ajustes específicos para garantizar un rendimiento óptimo. La configuración utilizada para esta prueba de concepto es la siguiente:

```
training_arguments = TrainingArguments(
    output_dir = output_dir,
    num_train_epochs = 1,
    per_device_train_batch_size = 1,
    gradient_accumulation_steps = 1,
    optim = "paged_adamw_32bit" # Default optimizer,
    save_steps = 0 # Save a checkpoint every x steps,
    logging_steps = 1 # Log every step,
    learning_rate = 2e-4,
    weight_decay = 0.001 # To prevent overfitting,
    fp16 = False,
    bf16 = True,
    max_grad_norm = 0.3 # Prevents excessively large updates,
    max_steps = 100,
```

```
    group_by_length = True # Groups training examples by length during
batching to improve efficiency,

    lr_scheduler_type = ”constant”,

    report_to = "wandb"

)
```

La configuración de estos parámetros depende de múltiples factores, tales como el modelo base, el *dataset* o el objetivo del *fine-tuning.* En esta prueba de concepto se busca demostrar el efecto del *fine-tuning* en el comportamiento del modelo por lo que los parámetros se ajustaron en base a este objetivo.

El número de épocas (*num_train_epochs)* determina cuántas veces se pasa el *dataset* completo por el modelo. Debido a que éste ya posee conocimientos médicos avanzados, no se consideró necesario realizar una iteración completa por el *dataset.* En su lugar, se estableció un número de pasos o *steps (max_steps),* que prevalece sobre el número de épocas. Cada paso corresponde a una iteración a través de las diferentes capas del modelo, tanto en la fase de propagación hacia adelante como en la de retropropagación. Los pasos se agrupan en lotes, por lo que el tamaño de lote (*per_device_train_batch_size*) y la acumulación de gradiente (*gradient_acumulation_steps*) tienen una influencia considerable en el uso de memoria y la velocidad de entrenamiento. Ambos parámetros deberán ajustarse según el *hardware* disponible.

La metodología de trabajo consiste en establecer un número elevado de pasos al inicio y guardar un punto de control (*checkpoint*) a intervalos regulares para evitar la pérdida del progreso. Se monitoriza el rendimiento del modelo en cada paso y se analiza el punto en el que comienza a degradarse. Utilizando los puntos de control, se reentrena el modelo a partir de éstos hasta alcanzar el número óptimo de pasos o el rendimiento deseado.

Una de las métricas usadas durante el *fine-tuning* es la métrica “*train/loss”* para evaluar el rendimiento del modelo en el *dataset* de entrenamiento. Esta métrica, conocida como “pérdida de entrenamiento” o “error de entrenamiento”, indica el valor de la función de pérdida (*loss function)* en cada uno de los pasos del entrenamiento. La función de pérdida cuantifica la diferencia entre las predicciones del modelo y los valores reales esperados. Un valor más bajo de esta métrica sugiere que el modelo está mejorando en su capacidad para hacer predicciones precisas sobre los datos de entrenamiento. En la Figura 18 se muestran los resultados para esta métrica de varios *fine-tunings* con diferentes pasos y tasas de aprendizaje.

Figura 18 – Gráfica comparativa de la métrica train/loss en función de los pasos para diferentes fine-tunings con diferente configuración de hiperparámetros. El entrenamiento de color verde claro (100 step, lr=2e-4) muestra los mejores resultados). Nota: Valores de learning_rate (tasa de aprendizaje) por encima de $5x10^{-4}$ causaban muchas pérdidas causando que no mejorase el resto de los entrenamientos. Fuente: elaboración propia.

Se observa como la pérdida de entrenamiento desciende gradualmente y se estabiliza después de los primeros pasos de entrenamiento, lo que indica que el modelo está mejorando rápidamente sus predicciones y aprendiendo de los datos de entrenamiento y sugiere que no es necesario entrenar al modelo con más de 50-60 pasos.

Los parámetros fp16 y bf16 permiten realizar un entrenamiento de precisión mixta con el objetivo de optimizar la eficiencia computacional mediante el uso de formatos numéricos de menor precisión para ciertas variables. Generalmente, se utiliza fp32 (*fp16=False, bf16=False).* Sin embargo, las GPUs más modernas, como las A100, permiten y se benefician de este tipo de entrenamiento. Se puede comprobar la compatibilidad mediante la siguiente función:

```
torch.cuda.is_bf16_supported()
```

La función del optimizador consiste en actualizar los pesos del modelo durante el entrenamiento, diferentes optimizadores y configuraciones pueden afectar a la convergencia y el rendimiento final del modelo. En este caso se utilizó *"AdamW_32bit"*, una variante eficiente en memoria que suele venir por defecto y

que proporciona una precisión de 32 bits. Existen variantes que cuantizan hasta 8 bits para obtener una mayor velocidad de entrenamiento.

La tasa de aprendizaje del optimizador está controlada por la tasa de aprendizaje del modelo (*learning_rate),* esta depende de la complejidad del modelo (modelos con más capas pueden requerir tasas de aprendizaje más bajas), el *dataset* o el tamaño de lote. Ajustar este parámetro es a menudo una tarea iterativa que requiere de experimentación y múltiples *fine-tunings.* Para ajustar este parámetro se estudió la estabilidad del modelo en los resultados del *fine-tuning* (Figura 18). Adicionalmente, se pueden establecer parámetros de calentamiento (*warmup_ratio)* o decaimiento (*decay*) para ajustar la estabilidad del entrenamiento según se necesite. Si se observa la Figura 18, se puede observar una convergencia en las tasas de aprendizaje, siendo la más adecuada $2x10^{-4}$.

El tipo de programador de la tasa de aprendizaje (*lr_scheduler_type*) se encarga de controlar cómo varía la tasa de aprendizaje durante el entrenamiento, establecerlo en "*constant"* significa que la tasa de aprendizaje permanece fija a lo largo del proceso de entrenamiento (tras los primeros pasos del *warmup_ratio).* Se eligió este parámetro por razones de simplicidad y estabilidad. Otras opciones son "*linear", "cosine", "plynomial" o "step"* que reducen la tasa de aprendizaje según diferentes funciones matemáticas.

6.1.4 FUSIÓN DEL MODELO BASE CON EL ADAPTADOR ENTRENADO

Durante el *fine-tuning*, solo se entrenan los parámetros adicionales (denominados adaptadores) generados por el método PEFT seleccionado (en este caso *LoRA* [73]) y no los parámetros completos del modelo base para reducir significativamente la cantidad de recursos computacionales necesarios. Al finalizar el *fine-tuning* es necesario fusionar estos adaptadores con el modelo base para incorporar los ajustes específicos aprendidos durante el entrenamiento y obtener el modelo final. En el caso de *LoRA* los pesos finales *W'* se calculan combinando las matrices de baja dimensión A y B con los pesos del modelo base W (Ver Figura 17):

$$W' = W + A * B$$

Otra aplicación de la fusión de modelos también es la de crear LLMs multimodales, combinando la inteligencia de todos los modelos fusionados sin aumentar el tamaño total de estos. En el siguiente trabajo [77] se propone la idea de combinar varios modelos mediante las medias de sus pesos.

6.2 RAG

La técnica RAG permite combinar la generación de texto de los LLM con la recuperación de información de una base de datos. En lugar de depender del conocimiento del modelo, esta técnica busca mejorar la generación de respuestas aumentando el contexto del modelo añadiendo más información a la consulta inicial (Información más detallada sobre RAG en el Capítulo 2.2).

Para llevar a cabo esta prueba de concepto se ha utilizado un *dataset* compuesto de múltiples reportes médicos pertenecientes a distintos departamentos en el idioma inglés. Los distintos reportes están en formato *.xml* y poseen una serie de campos comunes que identifican la información relevante del paciente y el documento. Esta información será trasladada a los metadatos de los documentos de forma que quede solo el cuerpo del documento. Los reportes poseen información censurada para cumplir con las normas de privacidad y protección de datos, por lo que se generaron varios reportes sintéticos mediante IA para facilitar la obtención de resultados.

Para generar los *embeddings* se ha utilizado el modelo de código abierto ***bge-small-en-v1.5*** [51], capaz de entender múltiples idiomas. Los *embeddings* generados se indexaron utilizando la base de datos vectoriales ***ChromaDB*** [50], una base de datos sencilla y eficiente, enfocada en IA. Toda la lógica RAG será gestionada mediante la librería *LangChain* [70], una herramienta de código abierto que permite "encadenar" LLMs para crear aplicaciones que ejecuten una serie de funciones secuenciales.

Aunque lo ideal sería introducir el *dataset* completo al LLM para que tenga todo el contexto, muchos LLM tienen una ventana de contexto limitada. Además, muchos LLM experimentan el problema conocido como pérdida intermedia, "*lost in the middle*", donde el modelo presta más atención a los datos al principio y al final del documento y tiende a olvidar la información intermedia. Una prueba común para medir el rendimiento del modelo en este ámbito se conoce como "aguja en un pajar", *Needle in a HayStack.* Para hacer frente a este problema, se dividen los documentos en fragmentos o *chunks,* que serán introducidos en la base de datos. Cuando el usuario realice una pregunta esta se introducirá en el mismo espacio vectorial de la base de datos, permitiendo recuperar los *chunks* más relevantes.

Cuando se trabaja con RAG, un aspecto crítico es el tamaño de los *chunks*, ya que influye significativamente en el rendimiento del sistema. Si los *chunks* son muy pequeños, es posible que no contengan contexto suficiente para ser útiles.

Por otro lado, si son demasiado grandes, podrían abarcar temas diversos, lo que puede dificultar la recuperación de información relevante y precisa. Además, aumenta el coste computacional al tener que procesar más texto.

Este parámetro está condicionado por múltiples factores, incluyendo el modelo utilizado para generar los *embeddings,* la ventana de contexto y el tipo de documentos empleados. Para determinar el tamaño de *chunk* más apropiado, se puede evaluar el rendimiento mediante métricas como el tiempo de respuesta promedio, la fidelidad promedio o la relevancia de las respuestas. Sin embargo, la mejor estrategia es a menudo la experimentación empírica.

Uno de los parámetros cruciales para determinar el tamaño de los *chunks* es la longitud máxima de secuencia (*max_seq_length*) del modelo utilizado para generar los *embeddings.* Este parámetro indica el número máximo de *tokens* que el modelo puede procesar, lo que significa que el tamaño de los *chunks* no podrá exceder este valor. En la Figura 19 se muestra el resultado de *tokenizar* sin establecer una limitación en el tamaño de los *chunks,* se observa que algunos *chunks* alcanzan una longitud de hasta 2000 *tokens,* y además se obtiene un aviso indicando que se producirán errores de indexación.

```
Token indices sequence length is longer than the specified maximum sequence length for this model (1776 > 512). Running this sequence through the model will
result in indexing errors
```

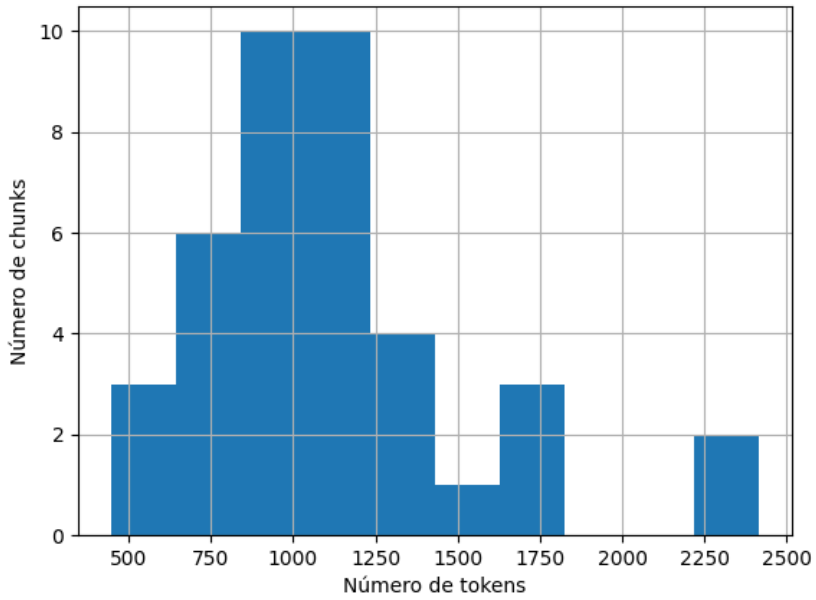

Figura 19 – Distribución del número de chunks en función de su tamaño en tokens, tras la tokenización de los chunks generados sin restricciones de

longitud. Es el generador de chunks el que decide los tamaños en base a las características semánticas de estos. Fuente: elaboración propia.

Es importante distinguir entre *tokens* y caracteres, ya que la longitud de los *chunks* dependerá de cuál de estas unidades se utilice. El modelo utilizando para realizar los *embeddings* posee una longitud máxima de secuencia de **512** *tokens*, por lo que los *chunks* deberán tener un tamaño inferior a este valor.

Adicionalmente, se define un solapamiento entre *chunks,* con el objetivo de dotar al modelo de contexto acerca de la relación entre *chunks* consecutivos, de forma que, si una idea abarca varios *chunks*, esta información no se pierda. No obstante, el solapamiento puede introducir redundancia en los datos con información repetida, por lo que su valor deberá ser significativamente inferior al del tamaño de los *chunks.*

Para mayor comodidad, se definirán los *chunks* en base al número de *tokens* en lugar de caracteres. Además, se utilizará una división recursiva de texto (*Recursive Text Splitter* [83]*)* para tratar de mantener los párrafos lo más juntos posible. Este tipo de división tiene en cuenta la estructura del documento, utilizando separadores como el retorno de carro, los espacios o los saltos de párrafo. En la Figura 20 se muestra la nueva distribución de *chunks* al separar los documentos en fragmentos de 500 tokens con un solpamiento de 50 tokens.

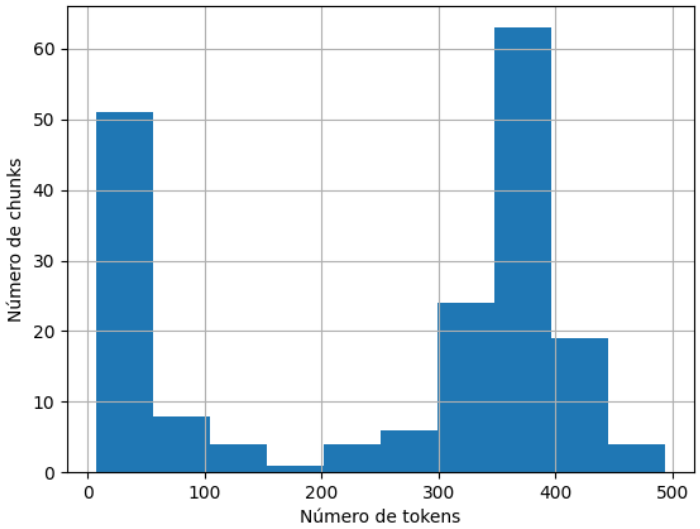

Figura 20 – Distribución del número de chunks en función de su tamaño en tokens, tras la tokenización de los chunks generados aplicando una restricción de tamaño máximo de 512 tokens. Fuente: elaboración propia

Para esta prueba de concepto se estudiaron tres técnicas de RAG diferentes con el objetivo de llevar a cabo una comparativa utilizando métricas como la precisión en la recuperación y la coherencia en la generación de respuestas. Estas técnicas están enfocadas en la recuperación de documentos y se exploran diferentes métodos gestionados por un *retriever*, que es la clase encargada de la lógica de recuperación.

6.2.1 NAÏVE RETRIEVER

El método *naïve* o ingenuo es el más simple de todos. Su nombre proviene de la expectativa "ingenua" de que, al realizar una búsqueda en la base de datos, se devuelvan los fragmentos (*chunks)* más relevantes para la pregunta introducida. Este método es el más básico de RAG, consiste en convertir el *prompt* del usuario a un *embedding* utilizando el mismo modelo empleado para crear la base de datos y posteriormente, recuperar los documentos más relevantes basados en la consulta realizada por el usuario. En la Figura 21 se puede observar un esquema del flujo de trabajo de esta técnica.

Figura 21 – Esquema del flujo de trabajo de la técnica RAG: naïve retriever. Fuente: elaboración propia

Debido a la estructura interna de los documentos con los que se trabaja, se establecerá un tamaño de *chunks* de **500** *tokens* con un solape de una décima parte, es decir, de aproximadamente **50** *tokens.* Con esta configuración, se dividen los documentos en dos *chunks*, añadiendo un pequeño solape para que el modelo sea capaz de relacionar los *chunks* entre sí. En la Figura 22 se puede observar la distribución de *chunks* para uno de los documentos utilizados.

```
**INSTITUTION
Consultation
Name: John Doe
Acct #: 123456
MRN: 654321
Admitted:
Date: May 20 2024
Dict: Dr. Emily Smith
Attend: Dr. Robert Johnson
REASON FOR CONSULTATION: Severe chest pain.
We are asked to evaluate this pleasant 75-year-old man with a history of
diabetes and hypertension. The patient was recently admitted to City Hospital
for chest pain. The patient was noted at that time to have an elevated troponin level.
As part of the patient's assessment, he underwent a coronary angiography which revealed
significant stenosis for which he was treated with stenting. Initially, he did well.
However, he subsequently developed recurrent chest pain and was evaluated by Dr. Sarah Brown
and sent to the emergency department for further evaluation and treatment. Mr. Doe
reports that he has had no significant fevers or chills. He does have a mild cough which is
non-productive. He has a remote smoking history, having quit more than 30 years ago. The patient
admits to a diet high in fatty foods. He reports no significant changes in bowel or bladder function.
He denies any skin lesions, rashes, changes in vision or hearing. He denies any constitutional symptoms
other than chest pain.
REVIEW OF SYSTEMS: Otherwise negative.
PAST MEDICAL HISTORY: His past medical history is notable for type 2 diabetes mellitus,
hypertension, coronary artery disease status post myocardial infarction, and hyperlipidemia.
MEDICATIONS: His outpatient medications include
1. Metformin.
2. Lisinopril.
3. Atorvastatin.
4. Aspirin.
5. Clopidogrel.
6. Metoprolol.
SOCIAL HISTORY: The patient is retired, lives with his wife, has two daughters. No alcohol,
remote tobacco use as mentioned.
FAMILY HISTORY: Positive for coronary artery disease and diabetes.
REVIEW OF SYSTEMS: As outlined.
PHYSICAL EXAMINATION: A pleasant elderly Caucasian male, in no acute distress.
Vital Signs: Pulse 74, blood pressure 150/80, respiratory rate 16, saturation 96% on room air.
Oropharynx is clear without exudate or erythema. No jugular venous distention. Chest reveals
clear breath sounds bilaterally. Cardiac Exam: Regular rhythm, no murmurs, rubs, or gallops.
Abdomen is soft, non-tender, no hepatosplenomegaly. Lower Extremities: No edema, no clubbing or cyanosis.
Neurologically: Alert and oriented, no focal deficits.
Chest x-ray: No acute infiltrates, no pleural effusions. Additional laboratories notable for
sodium of 138, potassium 4.0, serum CO2 of 25, BUN and creatinine 18 and 1.1 respectively.
White count 6000, hemoglobin 13.5, platelets 250. Troponin on admission was 0.2, repeat was less than 0.01.
IMPRESSION: A 75-year-old male admitted with recurrent chest pain. The patient has
a history of coronary artery disease and recent stenting. Chest pain likely secondary to
coronary artery spasm. Will continue current medical management and monitor.
No need for further invasive procedures at this time. Follow-up with cardiology recommended.
Thank you for allowing us to participate in this patient's care.
_____
Dr. Emily Smith, MD
HS Job # 987654 / 321654 / 123456
SHY # 654321
D: May 20 2024 14:25
T: May 20 2024 16:10
```

Figura 22 – Distribución de los chunks generados (marcados de color verde y amarillo) para un reporte médico aleatorio con un tamaño de 500 tokens y un solape de 50 tokens (marcado en un color gris oscuro). Fuente: [62]

El método *naïve* permite establecer métricas para la recuperación de *chunks* tales como el algoritmo utilizado para calcular las distancias entre vectores o el número de *chunks* que se recuperarán de la base de datos. Para la recuperación de documentos se empleará el *reranking,* un proceso adicional que reordena los fragmentos recuperados priorizando aquellos de mayor relevancia. Se seleccionarán los tres documentos más relevantes, con el objetivo de minimizar el error de recuperación y aumentar la precisión entre documentos.

Cuando se trabaja con RAG, es común crear plantillas de *prompt (prompt template),* estas plantillas son estructuras predefinidas que se utilizan para estructurar las consultas o preguntas que se le hacen al modelo. Esta plantilla podría interpretarse como el *system prompt*, es decir, una instrucción o conjunto de instrucciones dadas al modelo antes de que comience a interactuar con las consultas del usuario. En el contexto de RAG, sirven para mejorar la calidad y la precisión del *prompt,* así como ayudar al modelo a proporcionar una respuesta más precisa. En este caso, es necesario que la plantilla tenga el formato de *Llama3-Instruct.* Para este método se utilizó la siguiente plantilla:

```
prompt_template = """

    <|start_header_id|>system<|end_header_id|> Answer as an expert doctor.
Analyze the context provided below to answer the query.

    If you don't know the answer, just say that you don't know, don't try to
make up an answer.

    Context:

    {context}

    <|eot_id|>

    <|start_header_id|>user<|end_header_id|> This is the Query:
{question}<|eot_id|>"""
```

Al comienzo se instruye al modelo para que actúe como un “doctor experto” y se le pide que utilice la información proporcionada en la sección “context” para responder a la consulta realizada por el usuario. En la sección “context” será donde se introducirán los tres *chunks* más relevantes recuperados de la base de datos. Además, se indica al modelo que no invente las respuestas, lo que reduce la probabilidad de alucinaciones.

6.2.2 PARENT DOCUMENT RETRIEVER

Esta técnica consiste en primero dividir los documentos en *chunks* de tamaño grande (*parent document)* y a su vez dividir estos últimos en *chunks* más

pequeños (*child documents)* que serán usados para realizar la búsqueda. A la hora de devolver los resultados, se realizará la búsqueda en base a los *chunks* pequeños y se devolverán los *chunks* grandes a los que pertenecen. Para este caso de uso, al estar utilizando reportes médicos de pequeña longitud (una página), no es necesario realizar la primera división (*parent chunks).* Esto es debido a que podemos considerar los documentos originales como un *chunk* en sí mismo. En este caso, los *chunks* más pequeños se guardarán en la base de datos y los documentos en memoria.

Con esta técnica, se obtiene el balance perfecto entre precisión y contexto, ya que se realiza una búsqueda precisa de *chunks* y se proporciona un contexto completo al devolver el documento íntegro. No obstante, esta técnica resulta útil únicamente si los documentos no son muy grandes, ya que pueden superar la ventana de contexto del modelo o aportar información irrelevante. En la Figura 23 se muestra el flujo de trabajo.

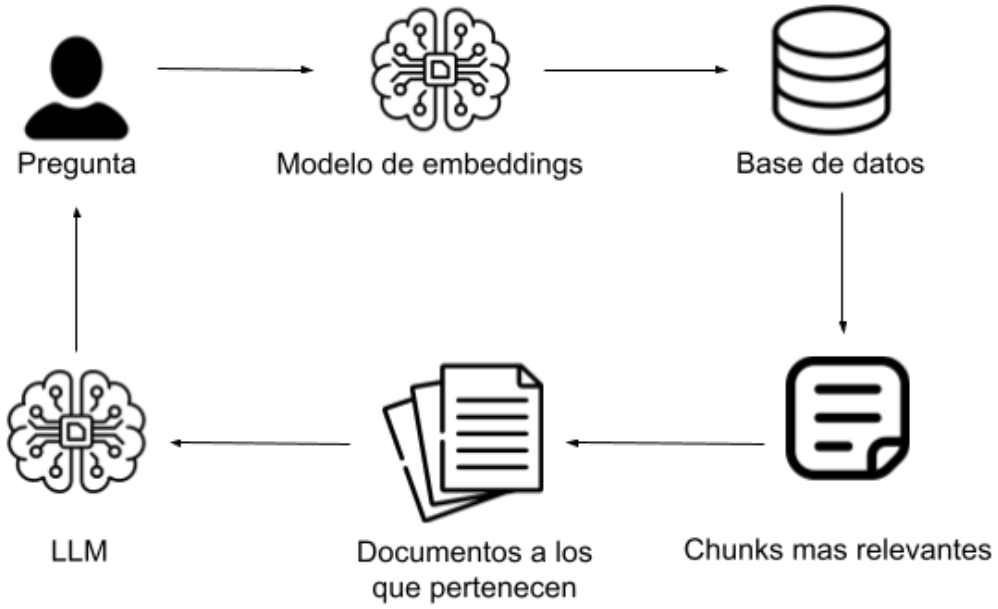

Figura 23 – Esquema del flujo de trabajo para la técnica de RAG: Parent Document retriever. Fuente: elaboración propia.

En este método se prioriza la precisión en la recuperación de los *chunks,* dado que el contexto se obtiene al recuperar el documento completo, haciendo innecesario que cada fragmento contenga toda la información relevante para la consulta. Por esta razón, se ha decidido utilizar un tamaño de *chunk* de **400** tokens, lo que permite aumentar la precisión sin comprometer el rendimiento que se logra con *chunks* de mayor tamaño. Además, se eliminó completamente el

solapamiento entre *chunks.* Tras varias pruebas se observó que el solapamiento causaba errores en la recuperación de los documentos finales, probablemente debido a la redundancia y la confusión introducidas por la superposición de contenido, al tomar partes de *chunks* pertenecientes a otros documentos adyacentes.

Se hará uso del *reranking*, pero eligiendo únicamente los dos documentos más relevantes, es decir, aquellos que obtengan la mayor puntuación ya que añadir múltiples documentos a la entrada o *input* del modelo puede causar confusión o pérdida de contexto. La plantilla del *prompt* también se modificará ligeramente para adaptarla al tipo de información que se espera recuperar de la base de datos y añadir el formato de *Llama3-Instruct*.

```
TEMPLATE = """

    <|start_header_id|>system<|end_header_id|>

    You are an expert doctor. Use the information in the medical reports
provided below to answer the query. Be specific and detailed.

    If the information is not in the report, just say that you don't know.
Don't try to make up an answer.

    Medical report:

    {context}

    <|eot_id|>

    <|start_header_id|>user<|end_header_id|> Query: {question}<|eot_id|>"""
```

En este caso, se indica al modelo que recibirá un reporte médico y que deberá responder a la consulta en base a este. Además, debido a que recibirá un mayor contexto, se le pidió que fura especifico y detallado en la respuesta. Por último, se le añadió una instrucción para reducir la posibilidad de alucinaciones.

6.2.3 MULTIQUERY RETRIEVER

En ocasiones, las distintas técnicas de RAG pueden producir resultados erróneos debido al cálculo de similitudes ya sea porque los *embeddings* no capturan correctamente la semántica de las palabras o debido a cambios sutiles en la redacción de la consulta. Esto se puede solucionar realizando un trabajo de ingeniería de *prompt* con la plantilla, pero puede acabar siendo tedioso. Esta técnica busca automatizar este proceso mediante la generación de múltiples consultas diferentes pero relacionadas. En general suelen ser variaciones de la consulta original que exploran diferentes perspectivas de la información buscada, proporcionando una visión más completa y diversa. Por cada variación

de la consulta original se recuperan una serie de *chunks* para posteriormente obtener la unión entre todos ellos y obtener un set de *chunks* relevantes. En la Figura 24 se muestra el flujo de trabajo para esta técnica.

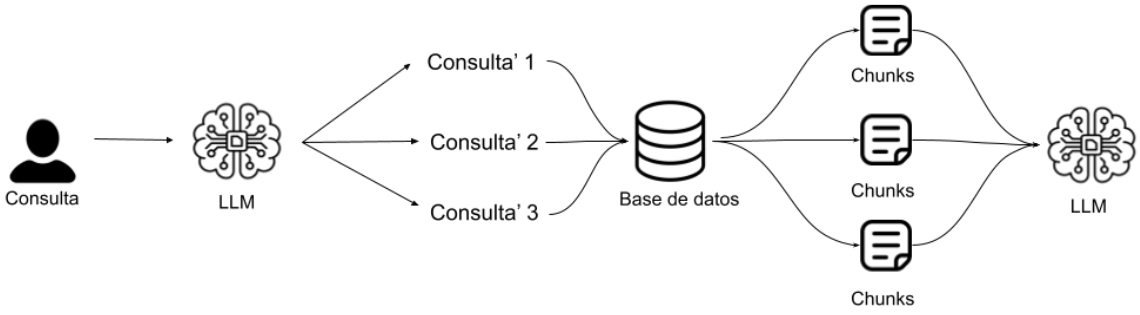

Figura 24 – Esquema del flujo de trabajo para la técnica de RAG: Multiquery retriever. Fuente: elaboración propia.

Esta técnica genera una gran cantidad de *chunks* por lo que es crucial garantizar la precisión de estos para reducir al mínimo la información irrelevante. Para lograrlo, se estableció un tamaño de *chunks* de **500** tokens con un solape de **100** *tokens*. Este método es similar a la técnica *Naïve*, en tanto depende de la información en los *chunks* para dar la respuesta adecuada. Para evitar pérdidas de contexto o confusión con *chunks* similares es importante definir una longitud de *chunk* que abarque toda la información que pueda ser relevante y un solapaiento que permita mantener el contexto entre *chunks* adyacentes.

Este método posee una plantilla de *prompt* por defecto para indicar al modelo las instrucciones para generar las consultas adicionales, la cual se aplica al invocar la función correspondiente. No obstante, debido a que se utiliza el modelo *LLama3-Instruct*, es necesario generar una plantilla con el formato adecuado para obtener el mejor rendimiento posible. La plantilla utilizada para esta técnica de RAG es la siguiente:

```
template = """

    <|start_header_id|>system<|end_header_id|>

    You are an AI language model assistant. Your task is to generate three (3)
different versions of the given user question to retrieve relevant documents
from a vector database. By generating multiple perspectives on the user
question, your goal is to help the user overcome some of the limitations of
the distance-based similarity search.

    Provide these alternative questions separated by newlines.

    <|eot_id|>
```

```
    <|start_header_id|>user<|end_header_id|> Original question:
{question}<|eot_id|>"""
```

Inicialmente se le indica al modelo su rol y posteriormente la instrucción para generar tres (3) variaciones de la pregunta original con el objetivo de recuperar documentos de una base de datos vectorial. Es necesario pedir al modelo que dé las respuestas en una nueva línea cada una ya que la librería *LangChain* [70] requiere de este formato para funcionar correctamente.

Capítulo 7. ANÁLISIS DE RESULTADOS

7.1 RESULTADOS DEL FINE-TUNING

En esta sección del proyecto se presentan y analizan los resultados obtenidos del *fine-tuning* del modelo *Llama3-8B-Instruct* en un *dataset* [46] especializado en preguntas y respuestas entre pacientes y doctores. El objetivo principal fue ajustar el modelo para que se comporte como un médico al interactuar con diferentes *prompts* relacionados con el ámbito de la medicina, sin necesariamente incrementar su conocimiento en el área.

La gráfica de la Figura 25 muestra la métrica "*train/grad_norm"* [79] o "norma del gradiente de entrenamiento" que indica la magnitud de los gradientes calculados en cada iteración del entrenamiento. Esta métrica es crucial para detectar problemas como la explosión o desaparición de gradientes [80] (al realizar el cálculo del gradiente, puede darse el caso de que la norma incremente exponencialmente causando que el algoritmo del descenso del gradiente sea imposible de ejecutar). Valores bajos indican que el modelo está aprendiendo de manera estable y efectiva, evitando inestabilidades que podrían degradar el rendimiento.

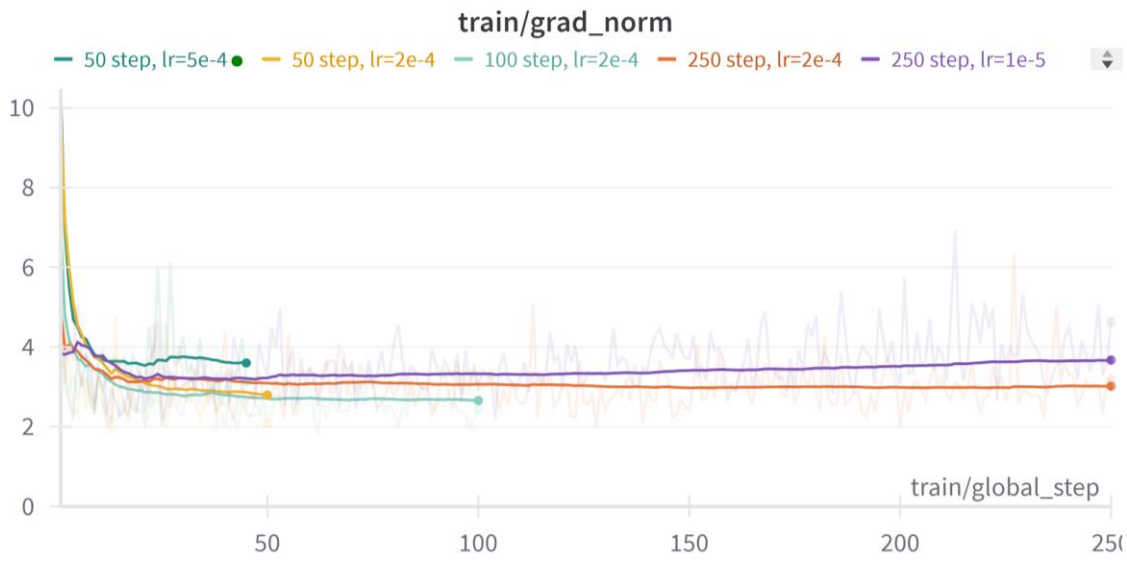

Figura 25 - Gráfica comparativa de la métrica train/grad_norm en función de los pasos para diferentes fine-tunings con diferente configuración de

hiperparámetros. El entrenamiento de color verde claro (100 step, lr=2e-4) muestra los mejores resultados Fuente: elaboración propia.

En la gráfica se observa como los diferentes modelos ajustados presentan diversos resultados, aquel con el menor valor será el que se utilice para la prueba de concepto.

El tiempo de entrenamiento depende de numerosos factores, entre los cuales se incluyen la tasa de aprendizaje (*learning_rate*) y el tamaño del *dataset.* En las distintas pruebas llevadas a cabo, se observaron variaciones significativas en la duración del proceso. La prueba identificada con el color morado fue la más prolongada, con una duración de 4h y 42 minutos, mientras que la prueba marcada con el color verde fue la más breve, completándose en tal solo 12 minutos. Ajustar correctamente los hiperparámetros es crucial para lograr un equilibrio óptimo entro el rendimiento del modelo y el coste computacional. Una configuración adecuada puede reducir significativamente el tiempo de entrenamiento sin comprometer la calidad del modelo, lo que es esencial para una implementación efectiva y eficiente.

7.1.1 COMPORTAMIENTO DEL MODELO

En esta sección se procederá a analizar el efecto del *fine-tuning* en el comportamiento del modelo, comparando las respuestas obtenidas antes y después de este proceso. Primero se probarán *prompts* directamente extraídos del *dataset* para evaluar si el modelo ha memorizado los datos o sigue utilizando su conocimiento previo. Posteriormente, se utilizarán *prompts* más sencillos con información no presente en el *dataset*, y se variará el *system prompt* para observar cualquier cambio en el comportamiento del modelo.

7.1.1.1 Comportamiento antes del fine-tuning (modelo base)

Nota: A no ser que se indique lo contrario, el *system prompt* utilizado es el siguiente*: You are a helpful assistant helping individuals with their medical queries.*

Prompt: *I have been having a lot of catching, pain, and discomfort under my right rib. If I twist to either side especially my right it feels like my rib catches on something and at times, I have to stop try to catch my breath and wait for it to subside. There are times if I am laughing too hard that it will do the same thing but normally its more so if I have twisted or moved a certain way.*

Respuesta:

```
I'm so sorry to hear that you're experiencing discomfort and catching
sensations under your right rib. It's understandable that you're concerned
about this, and I'm here to help you explore possible causes and potential
solutions.

Based on your description, it's possible that you might be experiencing some
kind of ribcage or diaphragmatic issue. Here are a few potential causes that
come to mind:

1. Ribcage inflammation or irritation: The ribcage can become inflamed or
irritated due to various reasons such as overuse, poor posture, or underlying
conditions like fibromyalgia or costochondritis. This can cause pain and
discomfort, especially when you twist or move in a certain way.

2. Diaphragmatic irritation or dysfunction: The diaphragm is a muscle that
separates the chest cavity from the abdominal cavity. If it becomes irritated
or dysfunctional, it can cause discomfort, pain, and difficulty breathing,
especially when you twist or move. This might be related to underlying
conditions like asthma, chronic cough, or respiratory issues.

3. Costochondral syndrome: This is a condition where the cartilage that
connects the ribs to the breastbone becomes inflamed or irritated, causing
pain and discomfort under the ribcage.

4. Hernia or other internal issues: Although less likely, it's possible that
you might have an internal issue like a hernia or a diaphragmatic hernia,
which could cause discomfort and catching sensations under your ribcage.

To better understand what might be going on, I have a few questions for you:

* Have you experienced any recent trauma or injury to your ribcage or
diaphragm?

* Do you have any underlying medical conditions, such as asthma, chronic
cough, or respiratory issues?

* Have you noticed any other symptoms, such as difficulty breathing, shortness
of breath, or chest tightness?

* Have you experienced any recent changes in your diet or exercise routine
that might be contributing to your symptoms?

It might be helpful to consult with a healthcare professional, such as a
primary care physician or a specialist like a pulmonologist or a physical
therapist, to get a more detailed evaluation and diagnosis. They can help you
identify the underlying cause of your symptoms and develop a personalized
treatment plan to address your discomfort and catching sensations.

In the meantime, you might want to try some gentle exercises to help loosen up
your ribcage and diaphragm. You can try some deep breathing exercises, ribcage
stretches, and gentle twists to help alleviate your symptoms.
```

Prompt: *Hi, I've just had an embryo replacement 5 days ago. Since then, I have had pain down my right leg near my groin, a kind of cold tinkly discomfort. Today I started getting reasonable stomach pains as well Is this suggestive of anything?*

does it mean the embryo has it taken? I have my blood test to confirm pregnancy on day ten to come with thanks.

Respuesta:

```
Hi there! I'm happy to help you with your concerns.

Firstly, congratulations on taking the first step towards building your
family!

Regarding your symptoms, it's not uncommon to experience some discomfort or
pain after an embryo replacement procedure. However, the symptoms you're
describing are quite specific, and I'd like to address each one separately:

1. Pain down your right leg near your groin: This could be related to the
embryo transfer procedure itself. The transfer process can cause some
discomfort, especially if the embryo is placed in the correct location. It's
also possible that you might be experiencing some minor bleeding or spotting,
which can cause cramping or discomfort in the lower abdomen or groin area.
However, it's essential to note that this symptom is not typically a sign of
embryo failure.

2. Cold, tinkly discomfort: This is a bit more unusual, but it's possible that
it's related to the embryo implantation process. Implantation can cause some
discomfort, which can manifest as a cold or tingling sensation. It's also
possible that this is related to minor bleeding or spotting, which can cause
cramping or discomfort.

Regarding your stomach pains, it's possible that they're related to the embryo
implantation process as well. Implantation can cause some cramping, bloating,
or discomfort in the abdomen, which can be similar to menstrual cramps.

It's essential to note that these symptoms are not necessarily a sign of
embryo failure. Embryos can implant and develop successfully even if you're
experiencing some discomfort.

However, it's always better to be safe than sorry, and it's good that you have
a blood test scheduled for day 10 to confirm pregnancy. This test will provide
a more accurate indication of whether the embryo has implanted and is growing.

In the meantime, it's essential to monitor your symptoms and report any
changes or concerns to your healthcare provider. They may recommend some pain
management or provide further guidance on managing any discomfort.

Remember, every woman's experience is unique, and it's essential to prioritize
your health and well-being during this time. If you have any further questions
or concerns, feel free to ask!
```

Prompt: *What is the underlying physiological mechanism of pulmonary hypertension?*

Respuesta:

```
Pulmonary hypertension (PH) is a complex condition characterized by increased
blood pressure in the pulmonary arteries, which can lead to right ventricular
failure and ultimately, heart failure. The underlying physiological mechanism
of PH is multifactorial and involves various pathways. Here's a breakdown of
the key mechanisms:

1. **Vascular remodeling**: In PH, the pulmonary arteries undergo significant
remodeling, including thickening of the arterial wall, medial hypertrophy, and
intimal fibrosis. This leads to increased resistance to blood flow and
increased pressure in the pulmonary arteries.

2. **Oxidative stress and inflammation**: PH is associated with increased
levels of reactive oxygen species (ROS) and inflammatory mediators, such as
cytokines and chemokines. These factors contribute to the development of
vascular remodeling and increased pulmonary artery pressure.

3. **Endothelial dysfunction**: The pulmonary endothelium plays a crucial role
in regulating blood flow and pressure. In PH, endothelial dysfunction leads to
decreased production of nitric oxide (NO), a potent vasodilator, and increased
production of vasoconstrictors, such as endothelin-1.

4. **Vasopressor activation**: The sympathetic nervous system and the renin-
angiotensin-aldosterone system (RAAS) are activated in PH, leading to
increased production of vasopressors, such as norepinephrine and angiotensin
II. These hormones contribute to increased pulmonary artery pressure and
vasoconstriction.

5. **Hypoxia**: In PH, chronic hypoxia (low oxygen levels) in the lungs leads
to increased production of erythropoietin, which stimulates red blood cell
production. This, in turn, increases blood viscosity and reduces blood flow to
the lungs, further exacerbating PH.

6. **Right ventricular dysfunction**: As PH progresses, the right ventricle
(RV) becomes hypertrophied and fails, leading to decreased cardiac output and
increased mortality.

7. **Microvascular disease**: In PH, the pulmonary microvasculature is
affected, leading to increased permeability, inflammation, and fibrosis. This
contributes to the development of PH and worsens its progression.

These mechanisms are interconnected and can interact with each other to
exacerbate PH. Understanding the underlying physiological mechanisms is
crucial for the development of effective treatments and therapeutic strategies
for this complex condition.
```

7.1.1.2 Análisis de las respuestas del modelo antes del fine-tuning

Las respuestas del modelo antes del *fine-tuning*, es decir, el modelo base, son notablemente más elaboradas y organizadas, reflejando la naturaleza propia de los modelos *instruct.* Estas respuestas muestran un mayor nivel de empatía, lo que permite conectar mejor con el usuario y generar una experiencia más personalizada y humana. Además, exploran diversas posibilidades y ofrecen soluciones más generales.

7.1.1.3 Comportamiento tras el fine-tuning

Prompt: *I have been having a lot of catching, pain, and discomfort under my right rib. If I twist to either side especially my right it feels like my rib catches on something and at times, I have to stop try to catch my breath and wait for it to subside. There are times if I am laughing too hard that it will do the same thing but normally its more so if I have twisted or moved a certain way.*

Respuesta del modelo:

```
Hi. Thanks for your query. I have gone through your question. It seems that
you are suffering from musculoskeletal problem. You should consult a doctor
and get done X-ray chest, lateral view and CT thorax. Based on the results of
these investigations, you may need to undergo a surgical procedure. Hope I
have answered your query. If you have doubt or have additional questions, I
will be happy to answer. Please feel free to contact me. Wish you good health.
Take care.   Regards.   Chat Doctor.
```

Respuesta del *dataset:*

```
Hi thanks for asking question. Here you are complaining pain in particular
position esp. While turning to a side. So strong possibility is about moderate
degree muscular strain. It might have occurred by heavyweight lift or during
some activities. Simple analgesic taken. Take rest. Sleep in supine position.
Second here Costco Chat Doctor. Ribs are tender to touch.x-ray also useful. If
cough, cold, sore throat present then respiratory infections also has to be
ruled out. Treat it symptomatically. If still seems serious then x-ray done
for chest. CBC will also be done. If you have yellow sclera, right abdomen
pain, anorexia then do your serum liver enzyme study for detecting liver
pathology. Avoid stress as it can aggravate pain. I hope you will understand
my concern
```

Prompt: *Hi, I've just had an embryo replacement 5 days ago. Since then, I have had pain down my right leg near my groin, a kind of cold tinkly discomfort. Today I started getting reasonable stomach pains as well Is this suggestive of anything? does it mean the embryo has it taken? I have my blood test to confirm pregnancy on day ten to come with thanks.*

Respuesta del modelo:

```
Hello, Thanks for using Chat Doctor. I have gone through your query, and I can
understand your concern. There is no relation between your symptoms and embryo
replacement. You may be having a pelvic inflammatory disease, and that may be
causing the symptoms. I suggest you take a course of antibiotics, such as
amoxicillin and clavulanate potassium, and metronidazole, for 7 days. Take a
painkiller, such as ibuprofen, and a muscle relaxant, such as diclofenac
sodium, as needed. Hope this helps.   Let me know if I can assist you further.
Take care.   Wish you a very good health.   Regards,   Chat Doctor.
```

Respuesta del *dataset:*

```
Hello and welcome to Chat Doctor The tingling sensation in the groin and
stomach pain does not indicate success or failure of embryo replacement. These
two symptoms have some other pathology. Stomach pain can occur due to various
causes- ingestion, muscular pain, spasm of stomach muscles. Infact it is due
to the organ which lies at the point of pain. Abdomen is a big area and many
organs are present in the abdominal cavity like liver, pancreas, spleen,
intestine, etc. You need to visit the GP to know the exact cause. You may need
an ultrasound abdomen. Regarding the tingling sensation in the groin, it could
be due to nerve compression while performing the embryo replacement. If it is
not relieved in few days please consult your GP for complete examination.
Thanks and take care Chat Doctor.
```

Prompt: *What is the underlying physiological mechanism of pulmonary hypertension?*

Respuesta:

```
Pulmonary hypertension (PH) is a condition characterized by high blood
pressure within the pulmonary arteries. The underlying physiological mechanism
of PH involves a complex interplay between various factors. The primary
factors involved are:

1. Inflammation: Inflammation is the first step in the development of PH. This
inflammation can be caused by various factors such as smoking, exposure to
pollutants, or infections. The inflammation triggers the release of various
chemical mediators that contribute to the development of PH.

2. Hypoxia: Hypoxia, or low blood oxygen levels, is another primary factor
involved in the development of PH. When blood oxygen levels are low, the body
responds by increasing blood pressure to maintain blood flow to the lungs.
This increased blood pressure in the lungs causes the blood vessels to
constrict and eventually leads to the development of PH.

3. Vasoconstriction: Vasoconstriction is another factor involved in the
development of PH. When the blood vessels in the lungs constrict, blood
pressure increases, and this increase in blood pressure contributes to the
development of PH.

4. Right ventricular hypertrophy: As blood pressure increases, the heart must
work harder to pump blood to the lungs. The right ventricle, which is
responsible for pumping blood to the lungs, becomes hypertrophied (thickened)
in response to the increased workload. This hypertrophy contributes to the
development of PH.

5. Increased pulmonary vascular resistance: Increased pulmonary vascular
resistance is another factor involved in the development of PH. This increased
resistance is caused by the constriction of blood vessels in the lungs and
contributes to the development of PH.

6. Endothelial dysfunction: Endothelial dysfunction is the loss of function of
the endothelial cells that line the blood vessels. This dysfunction
contributes to the development of PH by causing blood vessels to constrict and
increasing blood pressure.
```

```
7. Increased thrombotic activity: Increased thrombotic activity, or the
formation of blood clots, is another factor involved in the development of PH.
Blood clots can cause blood vessels to constrict and increase blood pressure,
leading to the development of PH.

In summary, the underlying physiological mechanism of PH involves a complex
interplay between various factors, including inflammation, hypoxia,
vasoconstriction, right ventricular hypertrophy, increased pulmonary vascular
resistance, endothelial dysfunction, and increased thrombotic activity.
Understanding these factors is important for the development of effective
treatments for PH.   I hope this helps.   Take care.   Chat Doctor.
```

Prompt: *Is it safe to travel pregnant?*

System prompt: *Answer truthfully*

Respuesta:

```
Hi, Thanks for choosing Chat Doctor. It is safe to travel pregnant, but it is
always advisable to avoid traveling during the first trimester. If you are
already pregnant, then you should avoid traveling to areas with high altitude,
high temperature, and high humidity. These areas can cause dehydration, which
may lead to pregnancy complications. If you have to travel, then you should
take necessary precautions such as taking multivitamin, wearing comfortable
clothes, and taking regular breaks. Hope this helps. Take care.   Chat Doctor.
```

Prompt: *How is a quadratic equation solved?*

System prompt: *Answer truthfully*

Respuesta:

```
Hello there! I can help you with that. Quadratic equations can be solved by
using the quadratic formula. This formula is given as follows:

x = (-b ± √(b2 - 4ac)) / 2a

Here, a is the coefficient of x2, b is the coefficient of x, and c is the
constant term. This formula is used to find the roots of the quadratic
equation. If you have any doubt, feel free to ask.   Thanks for using Chat
Doctor. Wish you a very good health. Take Care.   Chat Doctor.
```

7.1.1.4 Análisis de las respuestas del modelo tras el fine-tuning

Se observa una clara reducción en la elaboración de las respuestas, los cual está directamente relacionado con la naturaleza del *dataset* utilizado. Las respuestas del modelo ajustado reflejan la brevedad y directividad presentes en los datos de entrenamiento. Además, se puede observar como el modelo implementa los saludos y despedidas comunes en el *dataset*, lo que sugiere que el modelo base ha aprendido correctamente de los datos del *dataset* y ha modificado sus respuestas para asemejarse a ellos. Se observa que este comportamiento no

depende del *system prompt* o del tipo de consulta realizada por el usuario, sino que se ha vuelto un conocimiento interno del modelo.

Los cambios en el tono y la formalidad de las respuestas son evidentes: las respuestas del modelo ajustado se vuelven más formales y distantes en comparación con las del modelo base, que eran más empáticas y detalladas. Es interesante destacar que el modelo conserva su conocimiento básico y no ha memorizado las respuestas del *dataset.* Aunque las respuestas son más directas, el modelo todavía hace uso de sus conocimientos preexistentes para responder de manera coherente.

7.2 RESULTADOS DE RAG

Para esta sección del análisis de resultados se analizarán los *prompts* obtenidos mediante la aplicación de las tres técnicas de RAG estudiadas anteriormente. Resulta relativamente difícil evaluar el rendimiento de los sistemas RAG, especialmente si se utilizan varios documentos para realizar las consultas. Debido a la falta de métricas para medir estos sistemas, se suele medir mediante evaluación humana en base a las respuestas del modelo. En este caso, se evaluarán las respuestas generadas por el modelo en términos de precisión, calidad, utilidad y relevancia [81], destacando los hallazgos más significativos y realizando un análisis crítico de los mismos.

Las siguientes consultas fueron utilizadas para evaluar las respuestas generadas por cada técnica:

1. "What *medications were administrated to patients with chronic obstructive pulmonary disease?"*
2. "*What treatment or management plan is proposed for a patient's closed head injury and multiple contusions?"*
3. "*What is the physical examination of Mark Johnson?*"
4. "*The patient has burning in the tongue, what may be the cause?"*
5. "*What are the recommended medical interventions for a patient experiencing severe lower abdominal pain?"*

Estas preguntas se formularon con el objetivo de evaluar diferentes aspectos de las respuestas en base a los documentos recuperados por cada sistema RAG. A continuación, se detallan estos aspectos:

<u>Consulta 1</u>: Se evalúa la precisión y relevancia en la recuperación de información sobre tratamientos específicos para una enfermedad crónica. Ésta aparece en varios reportes médicos como síntomas o historial del paciente, por lo que el

modelo deberá ser capaz de identificar aquellos medicamentos relacionados con la enfermedad y que estén siendo usados por pacientes que la padecen.

Consulta 2: Dado que las lesiones en la cabeza y contusiones múltiples son comunes en informes médicos, se evalúa la capacidad del modelo de, basándose en la diagnosis y el tratamiento de un paciente con estas características, establecer un plan para tratar a pacientes similares.

Consulta 3: Esta consulta se centra en la precisión en la recuperación de datos personales y específicos del paciente. Acceder a la medicación actual de un paciente puede resultar de gran ayuda para facilitar la diagnosis y llevar un seguimiento adecuado.

Consulta 4: Con esta pregunta se evalúa la capacidad del modelo para responder a preguntas muy generales, de modo que pueda servir para realizar un diagnóstico inicial cuando se carece de suficiente especificación.

Consulta 5: El dolor abdominal puede tener muchas causas y suele ser una queja común en medicina. Además, puede resultar potencialmente grave. Con esta consulta se busca que el modelo ofrezca un tratamiento general para las diferentes causas del dolor abdominal que pueden observarse en los reportes.

7.2.1 RESULTADOS NAÏVE RETIRVER

Consulta: *What medications were administrated to patients with chronic obstructive pulmonary disease?"*

Respuesta generada:

```
Based on the provided documents, I found the following medications
administered to patients with Chronic Obstructive Pulmonary Disease (COPD):

From Document 1:

* Os-Cal with vitamin D (for osteoporosis, which may be related to COPD)

From Document 2:

* None explicitly mentioned, but the patient has a history of COPD.

From Document 3:

* None explicitly mentioned, but the patient has a history of interstitial
lung disease, which can be related to COPD.

Please note that these medications might not be directly related to COPD
treatment, but rather are part of the overall medication regimen for other
conditions. Additionally, it's essential to consult the full medical records
```

```
and consult with healthcare professionals for accurate information about
patient treatment plans.
```

- Análisis: El sistema RAG ha identificado correctamente los fragmentos (*chunks)* en los que se menciona la enfermedad (Nota: el modelo se refiere a los *chunks* cuando habla de documentos). Sin embargo, debido a la falta de contexto, probablemente causada porque el *chunk* no abarcaba toda la información relevante, la respuesta carece de contexto suficiente para ser completamente precisa. A pesar de esto, la respuesta es cuidadosa al aclarar la relación indirecta de los medicamentos con la enfermedad y mencionar la importancia de consultar los registros completos.

Consulta: *What treatment or management plan is proposed for a patient's closed head injury and multiple contusions?*

Respuesta generada:

```
Based on the provided medical report, the patient with a closed head injury
and multiple contusions is being managed by the Trauma Service.

The report states that the patient's injury was evaluated and treated in the
Emergency Department under the supervision of the attending physician. The
patient's occipital scalp laceration was repaired with staples.

The patient will be admitted to the Trauma Service for further evaluation,
management, and observation. The report does not specify a specific treatment
plan for the patient's closed head injury and multiple contusions, but it
implies that the Trauma Service will provide ongoing care and monitoring to
manage the patient's condition.

In general, the treatment plan for a patient with a closed head injury and
multiple contusions may include:

* Close monitoring of the patient's neurological status, including regular
checks of their Glasgow Coma Scale (GCS) score

* Management of any associated symptoms such as headache, nausea, or vomiting

* Administration of medications as needed to control symptoms and prevent
complications

* Imaging studies such as computed tomography (CT) scans or magnetic resonance
imaging (MRI) scans to assess the extent of the injury and monitor for any
changes over time

* Rehabilitation therapy to help the patient recover from their injuries and
regain functional abilities

* Follow-up appointments with the Trauma Service and other specialists as
needed to ensure the patient's continued recovery and well-being.
```

```
It is important to note that every patient's situation is unique, and the
specific treatment plan will depend on the individual patient's needs and
circumstances.
```

- Análisis: El sistema RAG ha identificado correctamente los *chunks* más relevantes y ofrece un contexto adecuado para el modelo. La calidad de la respuesta es alta, ya que proporciona una descripción completa y bien estructurada del manejo del paciente. Además, incluye detalles que reflejan el proceso de atención adecuado para este tipo de lesiones, con un plan detallado sobre las acciones llevadas a cabo con el paciente. Esta información resulta de utilidad para realizar un tratamiento eficiente del paciente.

Consulta: *What is the physical examination of Mark Johnson?*

Respuesta generada:

```
Based on the provided documents, here is the physical examination of Mark
Johnson:

"PHYSICAL EXAMINATION

VITAL SIGNS: Pulse 84, respirations 14, blood pressure 120/88.

HEENT exam was normocephalic and atraumatic. There was no significant scalp
tenderness. His oropharynx is clear and moist. Sclerae and conjunctivae are
clear. Tympanic membranes are clear. Mastoid area was nontender without
erythema. External auditory canal appeared clear. There was no significant
focal neck tenderness or limitation of neck movement.

He did have certain discomfort when moving his shoulder and certain motion
such as anterior flexion.

In testing muscle function and sensation in the distal right upper extremity
there was no focal weakness. He had no respiratory distress. Motor and sensory
function in the lower extremities was normal and gait was normal."
```

- Análisis: La respuesta proporciona un resumen del examen físico de un paciente, detallada y completa. Sin embargo, la información es incorrecta, el sistema RAG recuperó los *chunks* relacionados con un examen físico, pero ese examen no se corresponde al del paciente requerido. El problema es similar al de la primera consulta, el *chunk* en el que se menciona al paciente no abarca todo el contexto perteneciente a este. La respuesta correcta es la siguiente:

```
PHYSICAL EXAMINATION

VITALS: Blood pressure is 130/80, temperature 37.5, pulse 85, respiratory rate
18. HEENT: Normocephalic/atraumatic, no jaundice. ABDOMEN: Distended, tender
in the lower quadrants, positive rebound tenderness. Bowel sounds present but
```

```
hypoactive. EXTREMITIES: No edema. NEUROLOGIC: Alert and oriented, no focal
deficits.
```

Consulta: *The patient has burning in the tongue, what may be the cause?*

Respuesta generada:

```
Based on the provided medical report, it seems that the patient has reported
burning in the tongue. According to the report, the patient was diagnosed with
an allergic reaction to sulfa, and the doctor mentioned that the tongue
burning may be due to sulfa decreasing the B vitamin levels.

It's possible that the burning sensation in the tongue is related to the
allergic reaction caused by the sulfa medication. Sulfa medications can cause
various adverse effects, including oral manifestations such as glossitis
(inflammation of the tongue) or stomatitis (mouth sores). The burning
sensation in the tongue might be a symptom of these conditions.

However, without further information or a detailed examination, it's difficult
to pinpoint the exact cause of the burning sensation in the tongue. It's
always best to consult a healthcare professional for a proper diagnosis and
treatment plan.
```

- Análisis: En este caso, se recuperaron los *chunks* relacionados con un ardor en la lengua, y lo relaciona con una alergia a la medicación específica (sulfa). Debido a que la palabra *tongue* aparece en varios párrafos seguidos, se ha podido obtener una relación entre los diferentes *chunks* para ofrecer un contexto más completo. Además, se incluyen posibles efectos adversos y recomendaciones. La respuesta se mantiene enfocada en el tema y proporciona información útil para obtener una posible causa de los síntomas.

Consulta: *What are the recommended medical interventions for a patient experiencing severe lower abdominal pain?*

Respuesta generada:

```
Based on the provided documents, here are the recommended medical
interventions for a patient experiencing severe lower abdominal pain:



**Patient 1 (Laura Gutierrez)**

* Continue IV antibiotics (implied)

* Surgical consultation for possible drainage (recommended)

* Maintain on a clear liquid diet and advance as tolerated (recommended)
```

```
**Patient 2 (Consultation Report)**

* Continue IV antibiotics (recommended)

* Surgical consultation for possible drainage (recommended)

* Maintain on a clear liquid diet and advance as tolerated (recommended)

* Monitor blood glucose levels closely (recommended)

* Adjust insulin as needed (recommended)

* Continue current antihypertensive regimen (recommended)

* Monitor hemoglobin levels (recommended)

* Continue iron supplements (recommended)

In general, the recommended medical interventions for a patient experiencing
severe lower abdominal pain may include:

1. Antibiotics to treat potential infections

2. Pain management using medications such as ibuprofen or opioids

3. Fluid replacement therapy to prevent dehydration

4. Rest and relaxation to reduce discomfort

5. Dietary modifications, such as a clear liquid diet, to manage symptoms

6. Surgical intervention, if necessary, to drain abscesses or repair damaged
tissues

7. Monitoring of vital signs, laboratory tests, and imaging studies to assess
the severity of the condition and guide treatment decisions.

Please note that these recommendations are based on the limited information
provided and should be considered as a general guideline only. A thorough
evaluation and diagnosis by a qualified healthcare professional are essential
to determine the best course of treatment for each individual patient.
```

- Análisis: Se recuperan los *chunks* de dos pacientes en los que se menciona el dolor abdominal y sus intervenciones médicas. El modelo utiliza la información recibida para proporcionar directrices generales que puedan ser de utilidad para el tratamiento de estos síntomas. Además, ofrece una gama completa de intervenciones médicas, lo que resulta de mucha utilidad.

7.2.1.1 Conclusiones

La técnica *naïve* proporciona respuestas basadas en la información directamente extraída de los tres *chunks* obtenidos de la base de datos, por lo que su conocimiento estará limitado al contexto proporcionado por la recuperación. El

principal problema observado es que existe una limitación en el contexto que puede abarcar la recuperación, dado que los *chunks* tienen un tamaño limitado. Esta técnica funciona mejor con preguntas más generales y menos precisas, ya que los documentos presentan un elevado grado de similitud y es altamente probable que, si la información no se encuentra contenida en un solo *chunk*, se devuelva un *chunk* con relevancia parcial. Como es el caso de la consulta 3, donde se encuentra al paciente, pero no toda su información y devuelve un *chunk* en el que se menciona un examen físico sin relación directa con la pregunta. En cambio, en el caso de la consulta 4, al estar la información repartida por el mismo documento, se pudieron extraer los *chunks* necesarios para mantener el contexto global del documento.

7.2.2 RESULTADOS PARENT DOCUMENT RETRIEVER

Consulta: *What medications were administrated to patients with chronic obstructive pulmonary disease?*

Respuesta generada:

```
Based on the medical report, the patient with chronic obstructive pulmonary
disease (COPD) was administered the following medications:

* Albuterol inhaler as needed

* Prednisone 10 mg daily

Note that these medications are likely being used to manage the patient's COPD
symptoms and prevent exacerbations.
```

- Análisis: En este caso, la técnica *ParentDocument* recuperó un solo documento con información relevante a un paciente con COPD, del que se extrajo la información relevante sobre su mediación. Se da la respuesta más concisa con la información disponible. Además, el modelo ha sido capaz de diferenciar entre las medicaciones para tratar el COPD y las que está recibiendo el paciente, aunque sería mejor si se añadiesen a la respuesta para ofrecer más información. El contenido de los medicamentos del documento original es el siguiente:

```
CURRENT MEDICATIONS

1. Albuterol inhaler as needed.

2. Lisinopril 20 mg daily.

3. Atorvastatin 40 mg daily.

4. Metformin 500 mg twice daily.
```

```
5. Prednisone 10 mg daily.

6. Doxycycline 100 mg IV q.12 h.
```

Consulta: *What treatment or management plan is proposed for a patient's closed head injury and multiple contusions?*

Respuesta generada:

```
Based on the medical report, the patient with a closed head injury and
multiple contusions is being admitted to the Trauma Service for further
evaluation, management, and observation.

The proposed treatment or management plan is not explicitly stated in the
report. However, it can be inferred that the patient will likely receive:

1. Close monitoring for potential complications such as increased intracranial
pressure, seizures, or neurological deficits.

2. Pain management for the multiple contusions, possibly with medication or
other interventions.

3. Imaging studies, such as computed tomography (CT) scans, to assess the
extent of the injuries and monitor for any changes.

4. Potential surgical intervention for the 2 cm occipital scalp laceration,
which was repaired in the Emergency Department.

5. Admission to the intensive care unit (ICU) or a monitored bed for close
observation and management.

It is also important to note that the patient's admission diagnosis includes a
possible compression fracture at T8, which may require additional imaging
studies and management.

Please consult with a qualified healthcare professional for a more
comprehensive treatment plan and management strategy for this patient.
```

- Análisis: La información recuperada es precisa y completa. A pesar de no haber un plan de tratamiento indicado explícitamente, el modelo es capaz de responder de forma precisa con las intervenciones que recibió el paciente. Aunque falte generalización, resulta de utilidad para realizar un tratamiento inicial.

Consulta: *What is the physical examination of Mark Johnson?*

Respuesta generada:

```
Based on the medical report, the physical examination of Mark Johnson is:

"VITALS: Blood pressure is 130/80, temperature 37.5, pulse 85, respiratory
rate 18.
```

```
HEENT: Normocephalic/atraumatic, no jaundice.

ABDOMEN: Distended, tender in the lower quadrants, positive rebound
tenderness. Bowel sounds present but hypoactive.

EXTREMITIES: No edema.

NEUROLOGIC: Alert and oriented, no focal deficits."
```

- Análisis: La información es correcta y precisa, al haber recuperado el reporte completo del paciente, el modelo tiene acceso a toda la información relevante y puede responder a preguntas con un elevado grado de precisión.

Consulta: *The patient has burning in the tongue, what may be the cause?*

Respuesta generada:

```
Based on the medical report, the patient has been taking Bactrim DS
(sulfamethoxazole-trimethoprim) for 3 weeks, and has developed a burning
sensation in the tongue. This symptom is likely related to the medication.

The report mentions that the patient's tongue is "almost geographic in
nature", which suggests that the tongue has become inflamed and irritated.
This is a common side effect of sulfonamide antibiotics like Bactrim DS, which
can cause a condition known as sulfonamide-induced glossitis (inflammation of
the tongue).

The burning sensation in the tongue is likely caused by the sulfonamide
component of Bactrim DS, which can irritate the mucous membranes of the mouth
and tongue. This reaction is usually reversible and resolves after
discontinuation of the medication or switching to an alternative antibiotic.

It's worth noting that the attending physician suggested a multivitamin to
help alleviate the tongue burning, but it's unclear whether this was
specifically recommended to counteract the effects of Bactrim DS.
```

- Análisis: El *retriever* recupera correctamente el documento relevante para la consulta e identifica la causa probable del ardor en la lengua. A diferencia de la técnica *naïve*, al poseer el documento completo, esta técnica es capaz de ofrecer una visión más específica sobre la relación entre el medicamento y los síntomas del paciente.

Consulta: *What are the recommended medical interventions for a patient experiencing severe lower abdominal pain?*

Respuesta generada:

```
Based on the medical reports provided, here are some recommended medical
interventions for a patient experiencing severe lower abdominal pain:
```

```
1. **Acute Diverticulitis**: For patients with acute diverticulitis, the
consulting physician recommends:

       * Continuing IV antibiotics

       * Surgical consultation for possible drainage

       * Maintaining a clear liquid diet and advancing as tolerated

2. **Diabetes Mellitus**: For patients with diabetes mellitus, the consulting
physician recommends:

       * Monitoring blood glucose levels closely

       * Adjusting insulin as needed

3. **Hypertension**: For patients with hypertension, the consulting physician
recommends:

       * Continuing the current antihypertensive regimen

4. **Anemia**: For patients with anemia, the consulting physician recommends:

       * Monitoring hemoglobin levels

       * Continuing iron supplements

Additionally, based on the patient's chief complaint of severe abdominal pain,
the attending physician may recommend:

1. **IV Fluids**: Administering intravenous fluids to help manage dehydration
and electrolyte imbalances.

2. **Pain Management**: Providing pain relief medication, such as ibuprofen or
opioids, as needed.

3. **Antibiotics**: Prescribing antibiotics if there is suspicion of infection
or abscess formation.

4. **Imaging Studies**: Ordering imaging studies, such as computed tomography
(CT) scans or magnetic resonance imaging (MRI), to rule out complications like
perforation, abscess, or obstruction.

5. **Surgical Consultation**: Consulting with a surgeon if there is concern
about surgical intervention, such as appendectomy or colonic resection.

Please note that these recommendations are based on the specific circumstances
of the patient and may vary depending on individual factors and clinical
judgment. It is essential to consult with a qualified healthcare professional
for personalized advice.
```

- Análisis: En este caso se recuperaron documentos de pacientes que acusaban dolor abdominal, el modelo interpretó los síntomas del paciente y su historial como posibles causas del dolor abdominal. Adicionalmente, se lista una serie de recomendaciones para tratar el dolor. La información

de la respuesta del modelo posee una buena estructuración, haciendo que sea sencillo entender el procedimiento para cada enfermedad.

7.2.2.1 Conclusiones

Este método resulta útil cuando se busca realizar consultas que requieren de un contexto completo o para atender consultas más precisas en las que el modelo posee toda la información relevante. Ofrece mucha más precisión y contexto, pero puede recuperar documentos no relevantes y sufrir problemas de contexto irrelevante. La mayor limitación de este método aparece cuando la información se encuentra distribuida en muchos documentos.

7.2.3 RESULTADOS MULTIQUERY RETRIEVER

Consulta: *What medications were administrated to patients with chronic obstructive pulmonary disease?*

Consultas adicionales generadas:

What medications have been prescribed for individuals diagnosed with chronic obstructive pulmonary disease (COPD)?

Which pharmaceutical treatments have been used to manage symptoms in patients with chronic obstructive pulmonary disease?

Can you provide information on the medication regimens employed in clinical trials or real-world settings for patients with chronic obstructive pulmonary disease (COPD)?

Respuesta generada:

```
Based on the provided documents, I can analyze the medication information.

From the documents, I found the following medications administered to patients
with chronic obstructive pulmonary disease (COPD):

1. Albuterol inhaler (as needed)

2. Lisinopril (20 mg daily)

3. Atorvastatin (40 mg daily)

4. Metformin (500 mg twice daily)

5. Prednisone (10 mg daily)

These medications are mentioned in the discharge diagnoses and treatment plans
for patients with COPD. Specifically:
```

```
* Document 1: "Albuterol inhaler as needed" is mentioned in the treatment plan
for a patient with severe community-acquired pneumonia and COPD exacerbation.

* Document 2: Lisinopril, atorvastatin, metformin, and prednisone are listed
under "Current Medications" for a patient with COPD exacerbation.

Please note that these medications may not be exhaustive, as there might be
additional documents or information not included in the provided dataset.
```

- Análisis: Los *chunks* recuperados son relevantes para la consulta, se identifica correctamente un paciente que posee la enfermedad y el modelo es capaz de identificar los medicamentos e incluso dividirlos entre los que sirven para tratar COPD y los que han sido recetados para el paciente. No obstante, se puede observar que los medicamentos recuperados no están completos, probablemente debido a que el *chunk* corta justo en el último de ellos (El modelo muestra una comprensión de las limitaciones de los datos proporcionados).

Consulta: *What treatment or management plan is proposed for a patient's closed head injury and multiple contusions?*

Consultas adicionales generadas:

What are the recommended medical interventions for a patient with a closed head injury and multiple contusions?

Can you provide information on the standard treatment protocols for patients who have suffered a closed head injury and multiple contusions?

What is the typical course of care for a patient diagnosed with a closed head injury and multiple contusions, including any necessary rehabilitation or follow-up care?

Respuesta generada:

```
Based on the provided reports, it appears that the patient was treated for a
closed head injury and multiple contusions following a motor vehicle accident.

The treatment plan proposed for the patient includes:

* Admission to the Trauma Service for further evaluation, management, and
observation

* Repair of the 2-cm occipital scalp laceration in the Emergency Department
under the supervision of the attending physician

* Further evaluation and management of the possible compression fracture at T8
```

```
* Follow-up with the orthopedic surgeons and the chronic brain clinic for
further management of the chronic issue

* Prescription of Percocet for acute pain relief, with instructions to follow
up with the orthopedic surgeon and the chronic pain clinic

It is important to note that the specific treatment plan may vary depending on
the individual circumstances of each patient, and this information should not
be considered as medical advice.
```

- Análisis: Al igual que en los métodos anteriores, la información recuperada es correcta y está bien estructurada.

Consulta: *What is the physical examination of Mark Johnson?*

Consultas adicionales generadas:

What are the key findings in Mark Johnson's physical examination report?

What can be learned about Mark Johnson's health status based on his physical examination results?

Can you provide information on the common physical examination procedures used for patients like Mark Johnson?

Respuesta generada:

```
The document is:

`Document(page_content='[Report de-identified (Safe-harbor compliant) by De-ID
v.6.22.07.0]', metadata={'admit_diagnosis': '782.1', 'chief_complaint':
'ALLERGIC REACTION', 'deid': 'v.6.22.06.0', 'discharge_diagnosis':
'693.0,E931.0,709.8,781.1,V45.89,681.00,041.10,311,', 'downlaod_time':
'','source':'report25.xml','start_index': 4168,'subtype': '', 'type': 'ER',
'year': '2007'})`

In this document, under the "PHYSICAL EXAMINATION" section, it states:

* General: Appears well, visibly weak.

* Vital Signs: Vital signs are per the nursing notes.

* HEENT: Normocephalic atraumatic. Pupils are equal and reactive. Extraocular
muscles are intact.

* Neck: Neck is supple and nontender. No JVD.

* Lungs: Decreased breath sounds bilaterally.

* Heart: A 3/6 systolic murmur is noted.

* Abdomen: Abdomen soft. Moderate epigastric tenderness. No guarding, no
rebound, or mass.
```

```
* Extremities: Extremities atraumatic. No pedal edema.

* Skin: Skin is warm and dry. Fair turgor.

* Back Examination: Severe kyphoscoliosis is noted.

Please note that there may be additional documents containing information
about Mark Johnson's physical examination, but based on the provided data,
this is the only relevant document I could find.
```

- Análisis: A pesar de que ofrece una información mucho más extensa y desarrollada que el método *naïve*, es errónea. La causa es la misma que en el método *naïve*, los *chunks* no abarcan el contexto suficiente para poder relacionar dos piezas de información muy distantes en el mismo documento.

Consulta: *The patient has burning in the tongue, what may be the cause?*

Consultas adicionales generadas:

What medical conditions or diseases are commonly associated with a sensation of burning in the tongue?

Can you provide information on potential causes of a burning sensation in the mouth, specifically related to the tongue?

What are some possible explanations for a patient experiencing a burning feeling in their tongue, and how can it be diagnosed and treated?

Respuesta generada:

```
Based on the provided reports, there are several possible causes of burning
sensation in the tongue mentioned:

1. **Allergic reaction**: Patient reported a rosy pink macular papular rash,
which is generalized and somewhat pruritic, suggesting an allergic reaction.
The medication Bactrim DS 4 times a day might be the culprit.

2. **Sulfa-induced reaction**: Another patient's report mentions taking sulfa
medication, which can cause a burning sensation in the tongue due to decreased
B vitamin levels.

3. **Medication side effect**: Paxil, Seroquel, and Bactrim DS medications
were mentioned in another patient's report. These medications can have various
side effects, including oral manifestations like burning sensations.

To provide a more specific answer, it would be helpful to know the exact
medication(s) taken by the patient and their duration of use. Additionally, a
thorough medical history, physical examination, and laboratory tests would be
necessary to determine the underlying cause of the burning sensation in the
tongue.
```

- Análisis: Esta respuesta proporciona una comparación más detallada al analizar posibles causas de la sensación de ardor en la lengua, alcanzando un grado de precisión mayor que la técnica *Parent Document* ya que considera información de otros documentos. Este enfoque se centra en identificar las posibles razones detrás de los síntomas del paciente. Se destaca la importancia de considerar los diferentes medicamentos que ha tomado el paciente, un enfoque que incide en la comprensión del estado del paciente y la necesidad de atención médica individualizada.

Consulta: *What are the recommended medical interventions for a patient experiencing severe lower abdominal pain?*

Consultas adicionales generadas:

What are the most effective medical treatments for patients presenting with acute lower abdominal pain?

Can you provide information on the best practices for managing severe lower abdominal pain in adults?

What are the evidence-based guidelines for diagnosing and treating severe lower abdominal pain in patients, considering factors such as age, sex, and comorbidities?

Respuesta generada:

```
Based on the provided documents, it appears that the patient is experiencing
severe lower abdominal pain due to diverticulitis with abscess formation. The
recommended medical interventions for this patient would likely include:

1. **Antibiotic therapy**: Continue IV antibiotics to treat the underlying
infection.

2. **Surgical consultation**: Consultation with a surgeon for possible
drainage of the abscess.

3. **Clear liquid diet**: Maintain the patient on a clear liquid diet and
advance as tolerated.

4. **Blood sugar monitoring**: Monitor blood glucose levels closely and adjust
insulin as needed.

Please note that these recommendations are based on the provided documents and
may require adjustments according to the specific needs of the patient.
```

- Análisis: En este caso la respuesta está orientada a tratar el dolor abdominal causado por diverticulitis con formación de absceso. Si bien

esta orientación se ajusta a un caso específico, podría considerarse limitada en su utilidad para el manejo general del dolor abdominal. No obstante, ofrece un enfoque detallado para abordar esta condición médica específica.

7.2.3.1 Conclusiones

Este método aumenta el contexto de los modelos al agregar un mayor número de *chunks* al input del LLM. Sin embargo, aumenta la probabilidad de obtener *chunks* irrelevantes. Además, permite capturar múltiples aspectos relevantes de una pregunta, lo que proporciona un contexto más amplio y completo para la recuperación de *chunks.* Esto mejora la precisión y la relevancia de las respuestas obtenidas. Por otro lado, posee una limitación de contexto en el tamaño de los *chunks* y puede acarrear costes computacionales adicionales, así como un impacto en la velocidad de inferencia. Este método resulta útil cuando se trabaja con bases de datos más extensas y con información más variada y no tan similar. Esto ayuda a reducir la probabilidad de confusión de *chunks* similares con información diferente (como se ha observado en la consulta 3).

Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS

A lo largo de este trabajo, se ha llevado a cabo un análisis extenso de dos enfoques prominentes en el campo del procesamiento del lenguaje natural (NLP): la técnica del *fine-tuning* y *Retrieval Augmented Generation* (RAG). El objetivo principal fue estudiar ambas técnicas para facilitar la toma de decisiones a la hora de implementar soluciones basadas en IA.

8.1 FINE-TUNING

La técnica del *fine-tuning* permite que un modelo preentrenado reciba un entrenamiento adicional para aprender algo nuevo o modificar su comportamiento. Es análogo a enviar al modelo a la escuela de nuevo. Esta técnica es particularmente útil para enseñar al modelo a comprender una tarea específica con elevada precisión. Tras el *fine-tuning,* el modelo será capaz de dar respuestas más precisas y alineadas con los objetivos deseados.

No obstante, el *fine-tuning* requiere de una considerable capacidad computacional y tiempo, lo que puede resultar costoso. Además, se necesita contar con conocimientos avanzados de *machine learning* y arquitectura de modelos de lenguaje para realizar un *fine-tuning* eficaz y eficiente. Otro aspecto es que el *dataset* utilizado debe estar correctamente estructurado, y a largo plazo, realizar un número excesivo de *fine-tunings* puede afectar negativamente al rendimiento del modelo.

La prueba de concepto realizada ha demostrado la capacidad de esta técnica para modificar los parámetros internos del modelo y personalizarlo para un caso de uso específico. Además, se ha proporcionado una visión general del proceso de *fine-tuning*, facilitando así la compleja configuración que este método requiere y ofreciendo un punto de partida para su implementación.

8.2 RAG

La técnica RAG combina la inteligencia por defecto del modelo con una base de conocimiento, que a menudo suele ser una base de datos vectorial. El coste computacional y la velocidad de inferencia dependen significativamente de la base de datos utilizada y la cantidad de datos disponibles, por lo que resulta de vital importancia optimizar la recuperación de información de la base de datos

para alcanzar un alto grado de eficacia. Este método resulta especialmente útil en situaciones donde se necesita información actualizada o adicional que el modelo no haya aprendido previamente. RAG ofrece un equilibrio óptimo entre personalización y recursos necesarios, mejorando las respuestas del modelo al proporcionarle más contexto.

Un aspecto importante a considerar es que la calidad de las respuestas dependerá en gran medida de la relevancia y la precisión de la información recuperada, por lo que elegir adecuadamente el método de recuperación según las características del sistema es crucial. Aunque existen múltiples técnicas de recuperación, en este proyecto se estudiaron algunas de las más comunes o adecuadas para el caso de uso.

Dentro de las técnicas de RAG estudiadas, cada una destaca en diferentes aspectos. Si se busca una recuperación más general donde la información se encuentra distribuida por toda la base de datos, las técnicas *naïve* y *multiquery* son las más adecuadas. Por otro lado, si se requiere de una alta precisión en las respuestas y dotar al modelo del mayor contexto posible, es preferible utilizar la técnica *parent document*, ya que, al devolver el documento completo, el contexto se vuelve preciso y detallado.

Finalmente, de las pruebas seleccionas, aquella que proporcionó el mejor rendimiento fue ***ParentDocument*** debido a las características específicas de los documentos con los que se ha trabajado. Esta técnica resulta ser la más adecuada dado que la información se encuentra distribuida en documentos donde toda la información que contienen es relevante. Además, poseen tamaño relativamente pequeño, lo que permite introducir varios de ellos en el contexto del modelo sin riesgo a exceder la ventana de contexto disponible. Al ser documentos con una estructura muy similar, prevalece la precisión en las respuestas y este método ofrece una ventaja significativa al proporcionar el contexto completo.

8.3 FINE-TUNING VS RAG

En conclusión, la selección entre *fine-tuning* o RAG dependerá de los requisitos específicos del proyecto, los recursos disponibles y los resultados deseados. Cada método tiene sus propias fortalezas y limitaciones. No obstante, el método RAG ha demostrado ser aquel que ofrece el mayor equilibrio entre rendimiento y complejidad. Como se ha visto en la prueba de concepto, el desarrollo de RAG está orientado a asistir a los médicos en lugar de reemplazarlos, destacando que la IA debe ser vista como una herramienta complementaria que facilita y mejora

el trabajo de los profesionales. Aunque existe la preocupación de que la IA pueda sustituir empleos, su verdadero potencial radica en potenciar las capacidades humanas y optimizar los procesos, asegurando que aquellos que integren la IA en su práctica pueden ofrecer un mejor servicio y tomar decisiones más adecuadas y precisas.

8.4 TRABAJOS FUTUROS

En el ámbito de la IA y los LLM, existen múltiples direcciones prometedoras para continuar con la investigación y el desarrollo. A continuación, se presentan algunas áreas clave que se podrían explorar para expandir el alcance de este proyecto:

1. Asistentes y aplicaciones IA con memoria de la conversación: Uno de los desafíos actuales en el desarrollo de IA es la capacidad de mantener una conversación coherente y contextual a lo largo de múltiples interacciones con los usuarios. Se podría investigar como integrar mecanismos de memoria a largo plazo en los LLMs para mejorar su capacidad de recordar y utilizar información de conversaciones pasadas.

2. Exploración de más métodos de recuperación RAG: En este trabajo se han explorado solo algunos de los métodos de recuperación que mejor se adaptaban a los datos disponibles (reportes médicos). Sería interesante explorar más técnicas de recuperación para estudiar su rendimiento e incluso adaptar los datos a éstas (como es el caso de *SelfQuery Retriever,* que realiza las búsquedas en base a los metadatos de los documentos)

3. RAG con grandes bases de datos: En este proyecto se ha trabajado con un conjunto de reportes de apenas 40 documentos. Sería interesante investigar cómo afecta a la velocidad de búsqueda e inferencia el uso de un conjunto de datos mucho más extenso y variado ya que en aplicaciones reales la cantidad de información suele ser mucho más extensa.

4. Aumento de la ventana de contexto: El aumento de la ventana de contexto en los LLMs es otra área prometedora de investigación. El modelo utilizado (*LLama3)* cuenta con una ventana de contexto de 8192 *tokens*, lo que limita la capacidad del modelo para manejar contextos extensos. Ampliar esta ventana permitiría a los modelos procesar y entender mejor textos largos y complejos, mejorando así su desempeño con algunas técnicas de RAG que generen gran cantidad de contexto (como puede ser

ParentDocument o Multiquery). Sería interesante estudiar técnicas actualizadas de *fine-tuning* que aumenten la ventana de contexto considerablemente.

5. Censura y moderación del modelo: Un área de interés para organizaciones e instituciones que deseen implementar modelos de IA en atención al cliente es la capacidad de censurar y moderar el contenido generado por los modelos. A medida que los LLM se utilizan en aplicaciones más diversas, es crucial asegurarse de que no generen contenido inapropiado, sesgado o dañino. Una investigación futura podría centrarse en estudiar técnicas avanzadas de censura que vayan más allá de filtros básicos, implementando algoritmos complejos que detecten información dañina en tiempo real. Además, podría ser interesante explorar cómo estos sistemas pueden ser transparentes y ajustables para diferentes contextos y audiencias, garantizando así un equilibrio entre la moderación efectiva y la libertad de expresión.

Capítulo 9. BIBLIOGRAFÍA

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. & Polosukhin, I. (2017, 12 junio). *Attention is All You Need*. https://arxiv.org/abs/1706.03762

[2] Mistral AI team. (2023, 11 de diciembre). Mixtral of experts (A high quality Sparse Mixture-of-Experts). https://mistral.ai/news/mixtral-of-experts/

[3] Bahdanau, D., Cho, K., Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. https://arxiv.org/abs/1409.0473

[4] Nguyen, K. (2024, 29 de marzo). *The cost of AI reasoning over time.* https://semaphore.substack.com/p/the-cost-of-reasoning-in-raw-intelligence

[5] Lewis, P., Perez E. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* https://arxiv.org/pdf/2005.11401

[6] *¿Qué es RAG?: explicación de la generación aumentada por recuperación: AWS*. (s. f.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/retrieval-augmented-generation/

[7] Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., & Steinhardt, J. (2020, 7 septiembre). *Measuring massive multitask language understanding*. arXiv.org. https://arxiv.org/abs/2009.03300

[8] Ai, 0., Young, A., Chen, B., Li, C., Huang, C., Zhang, G., Zhang, G., Li, H., Zhu, J., Chen, J., Chang, J., Yu, K., Liu, P., Liu, Q., Yue, S., Yang, S., Yang, S., Yu, T., Xie, W., . . . Dai, Z. (2024, 7 marzo). Yi: Open Foundation Models by 01.AI. arXiv.org. https://arxiv.org/abs/2403.04652

[9] Meta-Llama. (s. f.). llama3/MODEL_CARD.md at main · meta-llama/llama3. GitHub. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[10] Thcookieh. (2024, 4 marzo). Embeddings + Knowledge Graphs: El futuro de la IA Generativa. Medium. https://medium.com/@thcookieh/embeddings-knowledge-graphs-el-futuro-de-la-ia-generativa-c7795554f1d8

[11] Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, É., Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., & Penedo, G. (2023, 28 noviembre). The Falcon Series of Open Language Models. arXiv.org. https://arxiv.org/abs/2311.16867

[12] Penedo, G., Malartic, Q., Hesslow, D., Cojocaru, R., Cappelli, A., Alobeidli, H., Pannier, B., Almazrouei, E., & Launay, J. (2023, 1 junio).

The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. arXiv.org. https://arxiv.org/abs/2306.01116

[13] Common Crawl - Open Repository of Web Crawl Data. (s. f.). https://commoncrawl.org/

[14] Shazeer, N. (2019, 6 noviembre). Fast Transformer Decoding: One Write-Head is All You Need. arXiv.org. https://arxiv.org/abs/1911.02150

[15] LICENSE.txt · tiiuae/falcon-180b-license at main. (s. f.). https://huggingface.co/spaces/tiiuae/falcon-180b-license/blob/main/LICENSE.txt

[16] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023, 27 febrero). LLaMA: Open and Efficient Foundation Language Models. arXiv.org. https://arxiv.org/abs/2302.13971

[17] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., . . . Scialom, T. (2023, 18 julio). Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv.org. https://arxiv.org/abs/2307.09288

[18] Meta-Llama. (s. f.-b). llama3/MODEL_CARD.md at main · meta-llama/llama3. GitHub. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[19] Ainslie, J., Lee-Thorp, J., Michiel, D. J., Zemlyanskiy, Y., Lebrón, F., & Sanghai, S. (2023, 22 mayo). GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. arXiv.org. https://arxiv.org/abs/2305.13245

[20] Meta-Llama. (s. f.-a). llama3/eval_details.md at main · meta-llama/llama3. GitHub. https://github.com/meta-llama/llama3/blob/main/eval_details.md

[21] Meta-Llama. (s. f.-d). PurpleLlama/Llama-Guard2/MODEL_CARD.md at main · meta-llama/PurpleLlama. GitHub. https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard2/MODEL_CARD.md

[22] META LLAMA 3 COMMUNITY LICENSE AGREEMENT. (2024, 18 abril). Meta AI. https://ai.meta.com/llama/license/

[23] META LLAMA 2 LICENSE AGREEMENT. (2023, 18 julio). Meta AI. https://ai.meta.com/llama/license/

[24] Hull, C. (2023, 5 octubre). Is Llama 2 open source? No – and perhaps we need a new definition of open… OpenSource Connections.

https://opensourceconnections.com/blog/2023/07/19/is-llama-2-open-source-no-and-perhaps-we-need-a-new-definition-of-open/

[25] mosaicml/mpt-7b-storywriter · Hugging Face. (2001, 18 enero). https://huggingface.co/mosaicml/mpt-7b-storywriter

[26] Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022, 27 mayo). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. arXiv.org. https://arxiv.org/abs/2205.14135

[27] Nvidia. (s. f.). GitHub - NVIDIA/FasterTransformer: Transformer related optimization, including BERT, GPT. GitHub. https://github.com/NVIDIA/FasterTransformer

[28] Apache License, Version 2.0. (s. f.). https://www.apache.org/licenses/LICENSE-2.0

[29] CC BY-NC-SA 4.0 Deed | Attribution-NonCommercial-ShareAlike 4.0 International | Creative Commons. (s. f.). https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en

[30] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., De las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W. E. (2023, 10 octubre). Mistral 7B. arXiv.org. https://arxiv.org/abs/2310.06825

[31] Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., De las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M., Stock, P., Subramanian, S., Yang, S., . . . Sayed, W. E. (2024, 8 enero). Mixtral of Experts. arXiv.org. https://arxiv.org/abs/2401.04088

[32] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E. (1991). Adaptive Mixtures of Local Experts. https://www.cs.toronto.edu/~hinton/absps/jjnh91.pdf

[33] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017, 23 enero). Outrageously large neural networks: the Sparsely-Gated Mixture-of-Experts layer. arXiv.org. https://arxiv.org/abs/1701.06538

[34] Mixture of Experts: How an Ensemble of AI Models Act as One | Deepgram. (s. f.). Deepgram. https://deepgram.com/learn/mixture-of-experts-ml-model-guide

[35] Papers with Code - Sliding Window Attention Explained. (s. f.). https://paperswithcode.com/method/sliding-window-attention

[36] Dvmazur. (s. f.). GitHub - dvmazur/mixtral-offloading: Run Mixtral-8x7B models in Colab or consumer desktops. GitHub. https://github.com/dvmazur/mixtral-offloading?tab=readme-ov-file

[37] mistralai (Mistral AI_). (s. f.). https://huggingface.co/mistralai

[38] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., . . . Zhu, T. (2023, 28 septiembre). Qwen Technical Report. arXiv.org. https://arxiv.org/abs/2309.16609

[39] Qwen. (s. f.). https://qwen.readthedocs.io/en/latest/

[40] Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., & Liu, Y. (2021, 20 abril). RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv.org. https://arxiv.org/abs/2104.09864

[41] Papers with Code - SwiGLU Explained. (s. f.). https://paperswithcode.com/method/swiglu

[42] QwenLM. (s. f.). Qwen/Tongyi Qianwen LICENSE AGREEMENT at main · QwenLM/Qwen. GitHub. https://github.com/QwenLM/Qwen/blob/main/Tongyi%20Qianwen%20LICENSE%20AGREEMENT

[43] QwenLM. (s. f.-b). Qwen/Tongyi Qianwen RESEARCH LICENSE AGREEMENT at main · QwenLM/Qwen. GitHub. https://github.com/QwenLM/Qwen/blob/main/Tongyi%20Qianwen%20RESEARCH%20LICENSE%20AGREEMENT

[44] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022, 4 marzo). Training language models to follow instructions with human feedback. arXiv.org. https://arxiv.org/abs/2203.02155

[45] medalpaca/medical_meadow_wikidoc · Datasets at Hugging Face. (s. f.). https://huggingface.co/datasets/medalpaca/medical_meadow_wikidoc

[46] harshith99/HealthCareMagic-100k-llama3 · Datasets at Hugging Face. (s. f.). https://huggingface.co/datasets/harshith99/HealthCareMagic-100k-llama3

[47] Labonne, M. (2023, 9 julio). Introduction to Weight Quantization | Towards Data Science. Medium. https://towardsdatascience.com/introduction-to-weight-quantization-2494701b9c0c

[48] Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2022, 31 octubre). GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv.org. https://arxiv.org/abs/2210.17323

[49] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023, 23 mayo). QLoRA: Efficient Finetuning of Quantized LLMs. arXiv.org. https://arxiv.org/abs/2305.14314

[50] the AI-native open-source embedding database. (s. f.). https://www.trychroma.com/

[51] BAAI/bge-small-en-v1.5 · Hugging Face. (s. f.). https://huggingface.co/BAAI/bge-small-en-v1.5

[52] Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., Dong, L., Wang, R., Xue, J., & Wei, F. (2024, 27 febrero). The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits. arXiv.org. https://arxiv.org/abs/2402.17764

[53] Introducing Meta Llama 3: The most capable openly available LLM to date. (2024, 18 abril). Meta AI. https://ai.meta.com/blog/meta-llama-3/

[54] Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., & Bowman, S. R. (2023, 20 noviembre). GPQA: a Graduate-Level Google-Proof Q&A benchmark. arXiv.org. https://arxiv.org/abs/2311.12022

[55] Chen, M., Tworek, J., Jun, H., Yuan, Q., De Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., . . . Zaremba, W. (2021, 7 julio). Evaluating Large Language Models Trained on Code. arXiv.org. https://arxiv.org/abs/2107.03374

[56] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021, 27 octubre). Training Verifiers to Solve Math Word Problems. arXiv.org. https://arxiv.org/abs/2110.14168

[57] Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., & Misra, V. (2022, 29 junio). Solving Quantitative Reasoning Problems with Language Models. arXiv.org. https://arxiv.org/abs/2206.14858

[58] Chatbot Arena: Benchmarking LLMs in the Wild with Elo Ratings | LMSYS Org. (2023, 3 mayo). https://lmsys.org/blog/2023-05-03-arena/

[59] Chiang, W., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., & Stoica, I. (2024, 7 marzo). Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. arXiv.org. https://arxiv.org/abs/2403.04132

[60] LMSYS Chatbot Arena Leaderboard. (2024, mayo). https://chat.lmsys.org/?leaderboard

[61] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., De las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Van Den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., . . . Sifre, L. (2022, 29 marzo). Training Compute-Optimal large language models. arXiv.org. https://arxiv.org/abs/2203.15556

[62] Chunk Visualizer - a Hugging Face Space by m-ric. (s. f.). https://huggingface.co/spaces/m-ric/chunk_visualizer

[63] Zhang, X., Chen, Z. Z., Ye, X., Yang, X., Chen, L., Wang, W. Y., & Petzold, L. R. (2024, 30 mayo). Unveiling the Impact of Coding Data Instruction Fine-Tuning on Large Language Models Reasoning. arXiv.org. https://arxiv.org/abs/2405.20535v1

[64] Weights & Biases: The AI developer platform. (2024, 28 mayo). Optimize LLMOps and Prompt Engineering with Weights & Biases. Weights & Biases. https://wandb.ai/site/solutions/llm-fine-tuning

[65] Hugging Face – The AI community building the future. (s. f.). https://huggingface.co/

[66] Huggingface. (s. f.). GitHub - huggingface/trl: Train transformer language models with reinforcement learning. GitHub. https://github.com/huggingface/trl

[67] Meta-Llama-3-8B-Instruct / tokenizer_config.json. (s. f.). Meta-Llama-3-8B-Instruct. https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct/blob/e1945c40cd546c78e41f1151f4db032b271faeaa/tokenizer_config.json#L2053

[68] TimDettmers. (s. f.). GitHub - TimDettmers/bitsandbytes: Accessible large language models via k-bit quantization for PyTorch. GitHub. https://github.com/TimDettmers/bitsandbytes

[69] Huggingface. (s. f.-a). GitHub - huggingface/transformers: 🤗 Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX. GitHub. https://github.com/huggingface/transformers

[70] Langchain-Ai. (s. f.-b). GitHub - langchain-ai/langchain: 🦜🔗 Build context-aware reasoning applications. GitHub. https://github.com/langchain-ai/langchain

[71] Trainer — transformers 3.0.2 documentation. (s. f.). https://huggingface.co/transformers/v3.0.2/main_classes/trainer.html#trainingarguments

[72] Huggingface. (s. f.-a). GitHub - huggingface/peft: 🤗 PEFT: State-of-the-art Parameter-Efficient Fine-Tuning. GitHub. https://github.com/huggingface/peft

[73] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021, 17 junio). LoRA: Low-Rank Adaptation of Large Language Models. arXiv.org. https://arxiv.org/abs/2106.09685

[74] Dey, S., PhD. (2024b, mayo 3). Deep Dive into LlaMA 3 by Hand ✍️ - Towards Data Science. Medium. https://towardsdatascience.com/deep-dive-into-llama-3-by-hand-%EF%B8%8F-6c6b23dc92b2

[75] Open LLM Leaderboard - a Hugging Face Space by open-llm-leaderboard. (s. f.). https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

[76] Shangar, G. (2024, 13 febrero). LLaMA 2: A Detailed Guide to Fine-Tuning the Large Language Model. Medium. https://medium.com/@gobishangar11/llama-2-a-detailed-guide-to-fine-tuning-the-large-language-model-8968f77bcd15

[77] Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., & Schmidt, L. (2022, 10 marzo). Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. arXiv.org. https://arxiv.org/abs/2203.05482

[78] Weights & biases. (s. f.). W&B. https://wandb.ai/mrfat/llama3-8b-instruct-medic-v0.1?nw=nwusermrfat

[79] Chen, Z., Badrinarayanan, V., Lee, C., & Rabinovich, A. (2017, 7 noviembre). GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. arXiv.org. https://arxiv.org/abs/1711.02257

[80] Bajaj, A. (2023, 29 agosto). Understanding Gradient Clipping (and How It Can Fix Exploding Gradients Problem). neptune.ai. https://neptune.ai/blog/understanding-gradient-clipping-and-how-it-can-fix-exploding-gradients-problem

[81] LangChain. (2023, 21 diciembre). LangChain State of AI 2023. LangChain Blog. https://blog.langchain.dev/langchain-state-of-ai-2023/

[82] OpenAI GPT-4. (s. f.). https://openai.com/index/gpt-4/

[83] Recursively split by character | 🦜🔗 LangChain. (s. f.). https://python.langchain.com/v0.1/docs/modules/data_connection/document_transformers/recursive_text_splitter/

[84] Ai, M. (s. f.). Technology. Mistral AI | Frontier AI In Your Hands. https://mistral.ai/technology/#models

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Este proyecto se puede emplear para promover la innovación y mejora de la infraestructura tecnológica (ODS 9) así como abogar por la producción y el consumo responsable (ODS 12). Adicionalmente, este proyecto se basa en las alianzas y la colaboración con la comunidad de código abierto para su desarrollo (ODS 17).

ODS 9 – Industria, innovación e infraestructura:

Este proyecto, al centrarse en modelos de lenguaje avanzados y utilizar soluciones de código abierto, contribuye a la promoción de la innovación tecnológica en el campo de la IA. Los LLM son una tecnología de última generación en el campo del procesamiento de lenguaje natural y es por eso por lo que este estudio puede aportar un avance significativo en dicho campo, alineándose con este ODS.

Así mismo, la infraestructura tecnológica es un componente clave en el desarrollo de modelos LLM, este proyecto busca optimizar y desarrollar la infraestructura de estos modelos de manera que permita un mejor rendimiento y a la vez impulse su eficiencia y capacidad, reduciendo el costo computacional y la infraestructura requerida para poder hacer uso de estos modelos.

Adicionalmente, esta investigación contribuye a modernizar y reconvertir la infraestructura de las organizaciones para que sean sostenibles y eficientes, haciendo un uso más eficaz de los recursos y promoviendo la adopción de herramientas avanzadas que mejoren su capacidad tecnológica sin requerir una inversión de capital excesivamente elevada. Asimismo, se hace hincapié en la importancia de optimizar el uso de los recursos existentes, lo cual implica una mayor eficiencia en la asignación de capital, energía y otros recursos valiosos.

El ODS 9 también hace referencia a la creación de empleo y el aumento de capacidad productiva, uno de los objetivos de este proyecto consiste en mejorar la capacidad de las empresas o instituciones para desarrollar soluciones basadas en LLM, lo que contribuiría en gran medida a mejorar su productividad.

ODS 12 - Producción y consumo responsable:

Los modelos de lenguaje que se estudian en este proyecto y las soluciones de código abierto que se ofrecen, sirven para optimizar la producción y el consumo

de recursos tecnológicos en las organizaciones, así como contribuir a la reducción del desperdicio y al uso más eficiente de recursos computacionales. Este TFG aboga por las prácticas responsables en el uso de tecnología, promoviendo la sostenibilidad en el campo de la IA y el procesamiento de lenguaje natural. Adicionalmente, este proyecto permitirá que las personas de todo el mundo tengan la información y los conocimientos pertinentes para el desarrollo sostenible de estas nuevas tecnologías.

ODS 17 - Alianzas para lograr objetivos:

Mucha de la información de este proyecto se ha obtenido en colaboración con comunidades de código abierto y los conocimientos obtenidos de este trabajo serán compartidos con la comunidad académica y la sociedad en general, promoviendo así un enfoque de acceso abierto a los resultados de la investigación. Este proyecto aumentará el intercambio de conocimientos y supondrá un apoyo a la creación de capacidad en materia de ciencia, tecnología e innovación para los países menos adelantados, así como un aumento de la utilización de tecnologías de la información.

ANEXO II: CÓDIGO EMPLEADO

El código empleado para las diferentes pruebas de concepto se encuentra en los siguientes repositorios de GitHub: https://github.com/MrFat1/RAG-TFG y https://github.com/MrFat1/Fine-tunning-TFG