



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES
INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Desarrollo de un Modelo Conversacional para Asistencia a
Personas de Edad Avanzada con programación RAG sobre
LLMs.

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas
Software

AUTOR/A: Pérez Martín, Ángel

Tutor/a: Monserrat Aranda, Carlos

Cotutor/a: Gil Gómez, José Antonio

CURSO ACADÉMICO: 2023/2024



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Universitat Politècnica
de València

**Departamento de Sistemas Informáticos y
Computación**



Máster Universitario en Ingeniería y Tecnología de Sistemas
Software

Trabajo Fin de Máster

**Desarrollo de un Modelo Conversacional
para Asistencia a Personas de Edad
Avanzada con programación RAG sobre
LLMs.**

Autor: Pérez Martín, Ángel
Tutor: Monserrat Aranda, Carlos
Cotutor: Gil Gómez, José Antonio

Valencia, 07 - 2024

Este Trabajo Fin de Máster se ha depositado en el Departamento de Sistemas Informáticos y Computación de la Universitat Politècnica de València para su defensa.

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas Software

Título: Desarrollo de un Modelo Conversacional para Asistencia a Personas de Edad Avanzada con programación RAG sobre LLMs.

07 - 2024

Autor: Pérez Martín, Ángel

Tutor: Monserrat Aranda, Carlos

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Co-tutor: Gil Gómez, José Antonio

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Agradecimientos

En primer lugar, me gustaría agradecer a Carlos y José Antonio por haberme dado la confianza y la oportunidad de llevar a cabo este proyecto. Asimismo, agradecerles por ayudarme y orientarme en los momentos que han sido oportunos permitiéndome así desarrollar este trabajo con éxito.

Por otro lado, me gustaría agradecer a mis padres y a mi familia, que me han estado apoyado y animado a lo largo de mi vida académica.

Asimismo, agradecer a mis amigos de toda la vida y a compañeros de máster que me han hecho el año más ameno.

Este trabajo representa el cierre de una etapa importante en mi vida y, al mismo tiempo, el inicio de un nuevo capítulo repleto de oportunidades.

Por último, me gustaría expresar mi más sincero agradecimiento a valgrAI - Valencian Graduate School and Research Network of Artificial Intelligence por otorgarme la beca que ha hecho posible la realización de este trabajo y permitirme avanzar en mi carrera académica y profesional.



Resumen

La creciente proporción de personas mayores en la sociedad actual presenta un reto considerable, especialmente en términos de su inclusión tecnológica. Este estudio se centró en mejorar la asistencia a este grupo demográfico mediante el uso de tecnologías avanzadas de Procesamiento del Lenguaje Natural (NLP), implementando la programación Retrieval-Augmented Generation (RAG) en entornos de Large Language Models (LLMs). Se analizaron cuatro LLMs: Mistral7B, PoliGPT, ChatGPT3.5 y ChatGPT4, utilizando herramientas como LangChain para personalizar y optimizar las conversaciones. La metodología incluyó el uso de estos modelos en un entorno controlado, evaluando su efectividad a través de cuestionarios y métricas de similitud. Se realizaron correlaciones entre las evaluaciones automáticas y las valoraciones humanas en diferentes idiomas. Los resultados indicaron que ChatGPT4 y ChatGPT3.5 obtuvieron los mejores resultados en términos de similitud de respuestas y tiempos de respuesta, con ChatGPT4 liderando en ambos aspectos, mientras que Mistral mostró un rendimiento inferior en comparación. Aunque ChatGPT4 fue el más efectivo, su alto costo podría inclinar la preferencia hacia ChatGPT3.5 como una solución más equilibrada. Este estudio subraya la relevancia de integrar tecnologías NLP avanzadas para mejorar la calidad de vida de las personas mayores, proporcionando una base para futuras investigaciones en la interacción asistida por IA para este grupo demográfico.

Abstract

The growing proportion of elderly people today presents a significant challenge, especially in terms of their technological inclusion. This study focused on improving assistance to this demographic group through the use of advanced Natural Language Processing (NLP) technologies, implementing Retrieval-Augmented Generation (RAG) programming in Large Language Models (LLMs) environments. Four LLMs were analyzed: Mistral7B, PoliGPT, ChatGPT3.5, and ChatGPT4, using tools like LangChain to personalize and optimize conversations. The methodology included using these models in a controlled environment, evaluating their effectiveness through questionnaires and similarity metrics. Correlations were made between automatic evaluations and human assessments in different languages. The results indicated that ChatGPT4 and ChatGPT3.5 achieved the best results in terms of response similarity and response times, with ChatGPT4 leading in both aspects, while Mistral showed inferior performance in comparison. Although ChatGPT4 was the most effective, its high cost might favor ChatGPT3.5 as a more balanced solution. This study highlights the relevance of integrating advanced NLP technologies to improve the quality of life of elderly people, providing a foundation for future research in AI-assisted interaction for this demographic group.

Resum

La creixent proporció de persones majors en la societat actual presenta un repte considerable, especialment en termes de la seua inclusió tecnològica. Aquest estudi es va centrar en millorar l'assistència a aquest grup demogràfic mitjançant l'ús de tecnologies avançades de Processament del Llenguatge Natural (NLP), implementant la programació Retrieval-Augmented Generation (RAG) en entorns de Large Language Models (LLMs). Es van analitzar quatre LLMs: Mistral7B, PoliGPT, ChatGPT3.5 i ChatGPT4, utilitzant eines com LangChain per personalitzar i optimitzar les converses. La metodologia va incloure l'ús d'aquests models en un entorn controlat, avaluant la seua efectivitat a través de qüestionaris i mètriques de similitud. Es van realitzar correlacions entre les avaluacions automàtiques i les valoracions humanes en diferents idiomes. Els resultats van indicar que ChatGPT4 i ChatGPT3.5 van obtenir els millors resultats en termes de similitud de respostes i temps de resposta, amb ChatGPT4 liderant en ambdós aspectes, mentre que Mistral va mostrar un rendiment inferior en comparació. Encara que ChatGPT4 va ser el més efectiu, el seu alt cost podria inclinar la preferència cap a ChatGPT3.5 com una solució més equilibrada. Aquest estudi subratlla la rellevància d'integrar tecnologies NLP avançades per millorar la qualitat de vida de les persones majors, proporcionant una base per a futures investigacions en la interacció assistida per IA per a aquest grup demogràfic.

Tabla de contenidos

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura de la Memoria	3
2. Estado del arte	5
2.1. Introducción a la Asistencia a Personas Mayores y Tecnología	5
2.2. Procesamiento del Lenguaje Natural (NLP)	6
2.3. Modelos de Lenguaje Grandes (LLMs)	8
2.4. Retrieval-Augmented Generation (RAG)	9
2.5. Aplicaciones de RAG y LLMs en Diferentes Sectores	12
3. Materiales y métodos	15
3.1. Modelos en local vs en la nube	15
3.2. Modelos utilizados	16
3.3. Desarrollo	17
3.3.1. Problemática de RAG	17
3.3.2. Creación de Documentos	18
3.3.3. Base de Datos y Chunk Splitting	18
3.3.4. Estructura RAG	19
3.3.5. Interfaz Web	24
3.4. Evaluación del Modelo	25
4. Resultados	29
4.1. Evaluación de la Similitud y Correlación	29
4.2. Evaluación del Tiempo de Respuesta	34
4.3. Evaluación de la Naturalidad de las Respuestas	38
5. Discusión	41
5.1. Discusión de la Similitud y Correlación	41
5.2. Evaluación del Tiempo de Respuesta	42
5.3. Evaluación de la Naturalidad de las Respuestas	43
5.4. Conclusiones Discusión	44
6. Conclusiones y trabajos futuros	45
6.1. Relación con las asignaturas	45
6.2. Trabajos futuros	46
Bibliografía	51

Anexo I	53
Anexo II	55
Anexo III	57

Índice de Figuras

2.1. Porcentaje de población mayor de 65.	5
2.2. NLP en un asistente. (Fuente: https://es.shaip.com/blog/what-is-nlp-how-it-works-benefits-challenges-examples/)	7
2.3. Timeline LLMs. (Fuente: Examining User-Friendly and Open-Sourced Large GPT Models: A Survey on Language, Multimodal, and Scientific GPT Models [16])	9
2.4. Estructura de la Generación Aumentada por Recuperación (RAG). (Fuente: https://arxiv.org/pdf/2312.10997)	10
2.5. Arquitecturas RAG. (Fuente: https://arxiv.org/pdf/2312.10997)	11
2.6. Interfaz Web CeliaIA. (Fuente: https://celiatecuida.com/)	13
3.1. Estructura del sistema.	19
3.2. Prompt pregunta o anécdota.	20
3.3. Tipos de memoria.	21
3.4. Prompt reformulación pregunta.	21
3.5. Interfaz web.	24
4.1. Cuestionario similitud.	29
4.2. Distribución datos castellano.	30
4.3. Distribución datos inglés.	30
4.4. Resultados comparativos similitudes.	34
4.5. Resultados tiempos de respuesta.	36
4.6. Resultados naturalidad de los LLMs.	39
4.7. Cuestionario naturalidad.	40

Índice de Tablas

2.1. Comparación entre Fine-Tuning y RAG.	11
3.1. Comparación de modelos en Chatbot Arena ELO Ratings [7].	17
4.1. Comparación de métricas de similitud y correlación en español.	31
4.2. Comparación de métricas de similitud y correlación en inglés.	31
4.3. Similitud media METEOR por idioma y LLM.	32
4.4. Similitud media tokens por idioma y LLM.	32
4.5. Tiempo medio por idioma y LLM.	36
4.6. ANOVA de dos factores sin medidas repetidas.	37
4.7. Resultados de los modelos Bonferri.	38

Capítulo 1

Introducción

1.1. Motivación

La población envejecida representa uno de los desafíos más significativos que la sociedad contemporánea puede enfrentar, no solo desde una perspectiva socioeconómica sino también en el ámbito de la integración tecnológica [23]. A medida que aumenta la proporción de personas mayores, se hace más necesario desarrollar soluciones que mejoren su calidad de vida y promuevan su inclusión en la era digital.

La integración tecnológica es fundamental para garantizar que este grupo demográfico no solo mantenga su independencia y calidad de vida, sino que también se fomente su inclusión social. La tecnología puede ofrecer soluciones innovadoras para enfrentar algunos de sus problemas cotidianos, desde la asistencia médica hasta la comunicación y el entretenimiento. Sin embargo, la brecha digital que existe entre las generaciones más jóvenes y de edad más avanzada sigue siendo un obstáculo significativo. Las personas mayores a menudo se enfrentan a barreras para adoptar nuevas tecnologías, ya sea por falta de conocimientos técnicos, temor a lo desconocido, o simplemente por la ausencia de interfaces amigables y accesibles [11].

Uno de los puntos de mayor motivación es la relevancia que puede aportar este trabajo en el contexto científico actual debido a la intersección de disciplinas tecnológicas y sociales que aborda. Con la población mundial envejeciendo, la necesidad de mantener conversaciones estables y accesibles es necesaria para este grupo social. La integración de tecnologías NLP y RAG en las interacciones diarias ofrece una gran oportunidad para mejorar la calidad de vida de este grupo demográfico. Además, este proyecto contribuye al campo del NLP al explorar y expandir el uso de LLMs en estas áreas.

Otro de los factores que me han llevado a desarrollar este proyecto han sido los desafíos personales y técnicos que implica el desarrollo de tecnologías avanzadas de NLP y la implementación de RAG utilizando LLMs, tres aspectos totalmente nuevos para mí este año. A lo largo de este proyecto, he enfrentado la dificultad de trabajar con un lenguaje de programación que no había utilizado hasta este año, Python, y adentrarme en el campo de la inteligencia artificial, algo totalmente nuevo para mí. Este proceso ha requerido una dedicación significativa para aprender y dominar nuevas herramientas y técnicas, lo cual ha sido un desafío enriquecedor y motivador. Además, la motivación extra de conseguir que el sistema RAG que pueda proporcionar respuestas de alta calidad y relevancia añade otra capa de complejidad y motivación

al proyecto.

Otra de las motivaciones clave de este trabajo es la evaluación para determinar cuál de los 4 modelos (Mistral [21], PoliGPT (basado en Llama3 [32]), ChatGPT3.5 [33] y ChatGPT4 [14]) a utilizar desempeña un mejor trabajo en la asistencia, esto no solo proporciona un análisis comparativo valioso, sino que también impulsa el desarrollo de soluciones más eficaces y personalizadas. Con lo que se permite identificar las fortalezas y debilidades de cada modelo.

1.2. Objetivos

El objetivo principal de este trabajo es investigar y mejorar la asistencia a las personas mayores mediante el uso avanzado de tecnologías de NLP. Al implementar programación RAG en entornos LLM, se busca facilitar las interacciones, haciéndolas más significativas y contextualizadas para dicho sector demográfico. Esta contextualización se genera a través de unos formularios que rellenan los usuarios donde responden a una serie de preguntas sobre su vida, ya sean experiencias, información personal, anécdotas, etc. Para desarrollar esta contextualización, trabajamos en estrecha colaboración con un grupo de psicólogos. Su objetivo fue asistir en la creación del documento utilizado para extraer información personal de manera efectiva y precisa, ver en el anexo 6.2. Para el desarrollo se utilizan frameworks que facilitan la construcción de la estructura como puede ser LangChain [37], con el que se adaptan y optimizan conversaciones, con el fin de proporcionar una experiencia comunicativa personalizada mediante la aplicación práctica de los modelos.

Otro objetivo clave es evaluar los cuatro modelos seleccionados para determinar cuál de ellos ofrece el mejor desempeño en la asistencia a personas mayores. Esta evaluación comparativa ha permitido identificar las fortalezas y debilidades de cada modelo, proporcionando información valiosa para el desarrollo de soluciones más eficaces y personalizadas.

En base a todo lo mencionado, para este proyecto se han establecido los siguientes objetivos:

- **Desarrollar un sistema de Generación Aumentada por Recuperación (RAG) utilizando los Modelos de Lenguaje Grandes (LLMs):** El primer objetivo es crear un sistema RAG que combine las capacidades de recuperación de información con la generación de texto mediante LLMs. Este sistema permitirá acceder a bases de datos de conocimiento para proporcionar respuestas más precisas y contextualmente relevantes a las necesidades de los usuarios.
- **Evaluar el desempeño de los modelos seleccionados:** Un objetivo fundamental de este trabajo es realizar una evaluación exhaustiva de los modelos seleccionados: Mistral, PoliGPT, ChatGPT3.5 y ChatGPT4. Al llevar a cabo estas pruebas, se podrán comparar las capacidades de cada modelo en la asistencia a personas mayores.
- **Identificar las fortalezas y debilidades de cada modelo:** El análisis detallado ayudará a determinar qué modelos son más adecuados para ciertas tareas o contextos específicos de conversaciones, en tiempo real, con este grupo demográfico.
- **Mejorar la calidad de vida de las personas mayores:** Uno de los objetivos más

importantes de este proyecto es utilizar las tecnologías avanzadas de NLP y RAG para mejorar la calidad de vida de las personas mayores. Esto incluye facilitar la comunicación, ofrecer compañía virtual, entretenimiento y ayuda.

- **Superar los desafíos técnicos y personales asociados con el desarrollo del proyecto:** El desarrollo de este proyecto presenta varios desafíos tanto técnicos como personales. Desde aprender nuevas tecnologías y lenguajes de programación hasta implementar modelos complejos y optimizar su rendimiento.

1.3. Estructura de la Memoria

El presente documento se divide en varios capítulos, los cuales se presentan y desarrollan de la siguiente forma:

- **Introducción:** Se introduce el proyecto, la motivación del trabajo, los objetivos que se pretenden alcanzar y la estructura del documento.
- **Estado del Arte:** Se revisan las herramientas y tecnologías existentes utilizadas en la creación de modelos conversacionales, con un enfoque en la asistencia a personas mayores. Se identifican sus limitaciones y áreas donde podrían mejorarse.
- **Metodología:** Se describen los modelos a utilizar, cómo se han utilizado y cómo se ha desarrollado el sistema RAG paso a paso. Además, se describe la metodología seguida para la evaluación de los modelos conversacionales.
- **Resultados:** Se presentan los resultados obtenidos en las distintas evaluaciones de los modelos. Los resultados se muestran en gráficos y tablas, proporcionando un análisis detallado de cada métrica.
- **Discusión:** Se interpretan y analizan en profundidad los resultados presentados en el capítulo anterior. Se comenta lo que significan estos resultados, su relevancia, sus implicaciones, y cualquier limitación que pueda haber en los experimentos realizados.
- **Conclusiones:** Se resumen las conclusiones del proyecto, evaluando si los objetivos planteados en la introducción han sido alcanzados. Se destacan los modelos que demostraron mejor rendimiento y se sugieren las siguientes líneas de trabajo futuro para continuar mejorando la asistencia a personas mayores mediante el uso de modelos conversacionales avanzados.

Capítulo 2

Estado del arte

2.1. Introducción a la Asistencia a Personas Mayores y Tecnología

El envejecimiento en la población es un problema social que cada vez va más en aumento. En el 2000, el 12.4 % de la población mundial tenía 65 años o más, para el 2030 se espera que la cifra aumente hasta el 19 % y para el 2050 en los países que tengan un mayor desarrollo y una mejor calidad de vida se espera que el porcentaje de la población sea de un 33 % [12], esto es visible en la figura 2.1. Esto hace necesario y urgente empezar a prolongar la vida tecnológica de las personas de edad avanzada permitiendo que sigan manteniendo su nivel de autonomía.

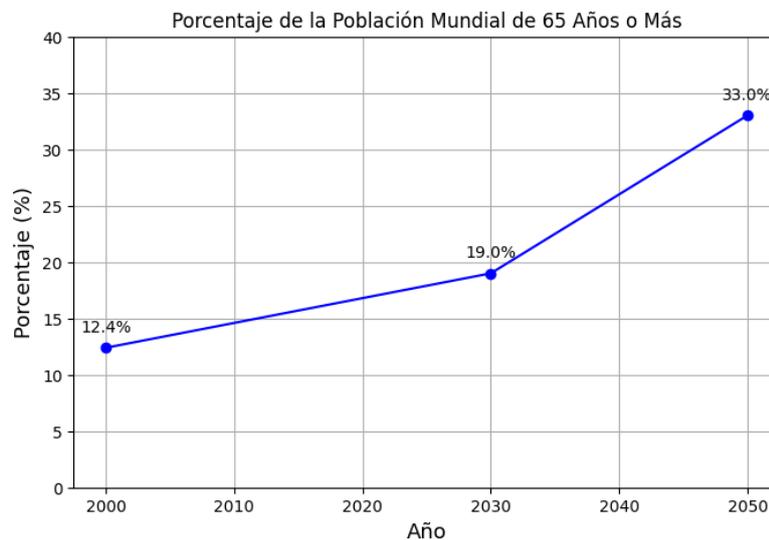


Figura 2.1: Porcentaje de población mayor de 65.

Ese aumento de esperanza de vida ha llevado a que este sector de la población tenga ciertos problemas funcionales y barreras tecnológicas [31], como pueden ser los problemas físicos que van apareciendo con la edad o la resistencia que las personas pueden generar a las nuevas tecnologías o nuevas formas de interacción con el mundo.

Por ello, la inclusión social de este sector de la población en la era digital es un desafío crucial al que hay que enfrentarse. Gracias a la tecnología se puede ofrecer oportunidades únicas de desarrollo de proyectos para mejorar la calidad de vida haciendo su vida en ciertos niveles más independiente y más sociables, puesto que hoy en día hay muchos problemas a los que enfrentarse, como son la soledad, la gestión de la salud, el acceso a los servicios, la interacción con los mismos, etc.

A pesar de tener numerosos beneficios, las personas mayores se enfrentan a una gran cantidad de barreras, como pueden ser la falta de conocimientos técnicos para poder desenvolverse en estos entornos, ya que muchas de estas personas no han tenido la oportunidad de interactuar con la tecnología a lo largo de sus vidas, generando así un rechazo al usuario. Otro de los problemas a destacar sería la falta de concienciación a la hora de realizar el diseño de las interfaces para que sean amigable y accesibles. Hoy en día la mayoría de las interfaces están pensadas para gente que esta habituada a utilizar la tecnología, donde la interacción puede ser muy avanzada, como en el caso de los atajos, lo que hace que sea totalmente opuesto a lo que busca este sector de la población. Dicho sector busca interfaces simples con botones, textos legibles, navegación intuitiva, etc.

Un estudio [39] identificó que además de los factores ya comentados, otro de los grandes inconvenientes que encuentran es el rechazo a la privacidad y la seguridad. El uso de las tecnologías les genera una preocupación por la protección de sus datos, el peligro a que pueda ser un fraude, generándoles muchas inseguridades. Debido a esto, muchas de estas personas no están dispuestas a sacrificar la privacidad por la mejora en la calidad de vida al sentirse vigilados, en cambio, hay otro porcentaje inferior que sí está dispuesto a sacrificar dicha faceta.

Para lograr una buena aceptación, es necesario que los desarrolladores sean capaces de realizar un diseño empático [10] con los usuarios que van a utilizarlos. No hay que tratar solo como se sienten o cuáles son sus emociones al utilizar la aplicación, sino que deben ser capaces de entender qué es lo que quieren, cómo lo quieren y cuáles son sus preferencias. En otras palabras, se trata de ver de la misma forma que lo hacen ellos y plantear una interfaz acorde a las necesidades que requieran. Hay que diseñar y desarrollar tecnologías que sean aceptadas por los usuarios, donde se busque y se promueva la independencia, la reducción del aislamiento de personas mayores, una mejor comunicación social, etc.

2.2. Procesamiento del Lenguaje Natural (NLP)

El procesamiento del lenguaje natural (NLP) se ha convertido en una tecnología de vanguardia que fusiona la inteligencia artificial con la capacidad de los ordenadores que les permite entender, interpretar y responder al lenguaje humano de manera efectiva. NLP se encarga de procesar de forma automática los datos de voz, texto, correos electrónicos, redes sociales, etc. con la intención de analizar, a través del desarrollo de algoritmos y modelos, el sentimiento y la intención del mensaje para responder de una forma natural y precisa. Esta tecnología es fundamental para crear sistemas de interacción humano-computadora (HCI) [22], facilitando la comunicación fluida entre humanos y máquinas.

Uno de los puntos más importantes de la tecnología NLP es que no solo mejora la usabilidad de los sistemas tecnológicos, sino que esto también abre un abanico de

posibilidades y de nuevas oportunidades para proporcionar una mejor accesibilidad e inclusión social a las tecnologías. Esto es algo fundamental si se quiere proporcionar una mejor adopción de la tecnología para las personas de edad avanzada. NLP permite mejorar la accesibilidad eliminando la necesidad de requerir conocimientos sobre el uso de una interfaz web, ya que se podría utilizar la voz para ir avanzando o preguntando qué es lo que debe hacer. Además, NLP permite transcribir voz a texto y viceversa, permite traducción automática y asistencia virtual, lo cual automatiza muchas de las tareas que se llevarían a cabo en un desarrollo normal.

A este respecto, las tecnologías de Procesamiento de Lenguaje Natural (NLP) [2] juegan un papel crucial. Estas tecnologías pueden transformar la manera en que este grupo demográfico interactúa con dispositivos y servicios digitales, permitiendo crear interfaces más intuitivas y accesibles. De hecho, los avances en NLP permiten el desarrollo de asistentes virtuales [30] capaces de comprender, responder a preguntas y comandos de manera natural, lo que puede ser particularmente beneficioso para aquellas personas que tienen dificultades a la hora de manejarse con un estilo de interfaces más tradicionales. Un ejemplo del funcionamiento es en la figura 2.2 donde se puede ver la interacción del usuario con el sistema, cómo va evolucionando la entrada de información y cómo, posteriormente, se devuelve al usuario.

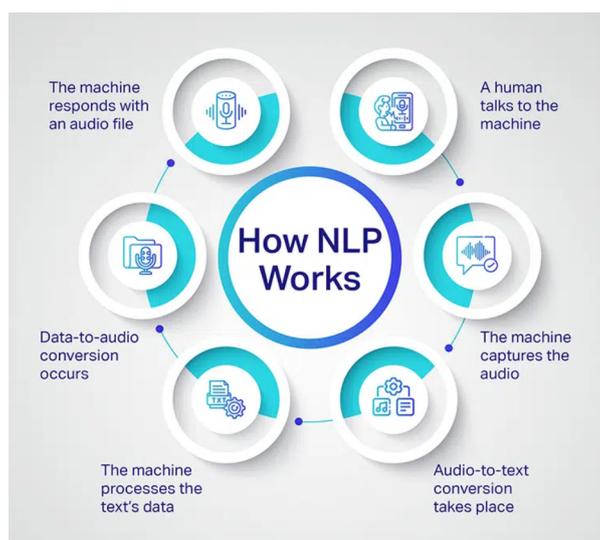


Figura 2.2: NLP en un asistente. (Fuente: <https://es.shaip.com/blog/what-is-nlp-how-it-works-benefits-challenges-examples/>)

Además de los asistentes virtuales que se pueden generar con NLP, también se destacan otras áreas de aplicación, como pueden ser la robótica [18], la automatización del hogar o juegos [4] [35]. En estas, a través de comandos de voz se pueden identificar y realizar operaciones facilitando las tareas. Otro ejemplo de área de aplicación es la recuperación y extracción de información [5] gracias a técnicas de comparación de información y acceso a las mismas.

Sin embargo, esta tecnología tiene un alto margen de mejora, puesto que a pesar de la cantidad de aplicaciones o nuevas oportunidades que se pueden crear, sigue teniendo sus defectos. En el caso del reconocimiento de voz debido a que en el habla humana hay una gran variedad de acentos, distintos volúmenes en el habla y/o la

presencia de ruidos de fondos acaba añadiendo muchas dificultades a las capacidades de interpretación del sistema. Esto se debe abordar para un futuro próximo puesto que la incorrecta escucha o captación de lo que se expresa puede llegar a acarrear problemas graves en el caso de un hogar inteligente o en la interacción con una web.

En conclusión, NLP es una de las soluciones actuales con mejores resultados en el mundo de la interacción humano-computador, donde se continúan realizando investigaciones a diario para ir aumentando la precisión y la capacidad de reconocimiento de los sistemas.

2.3. Modelos de Lenguaje Grandes (LLMs)

Los Modelos de Lenguaje Grandes (LLMs) son una categoría de modelos de inteligencia artificial, considerada IA Generativa [13], que han revolucionado el campo del NLP y no solo este campo sino también el mundo y la metodología de trabajo en general. Estos modelos se entrenan en grandes conjuntos de datos textuales, donde se utilizan técnicas avanzadas de NLP, siendo capaces de generar texto coherente y contextualmente relevante. Los LLMs son unas de las tecnologías que mayor impacto han generado en la sociedad en los últimos años, siendo utilizados por cualquier usuario y mostrando al mundo lo que son capaces de hacer, llevando la inteligencia artificial a todo el mundo.

En la actualidad, es ampliamente aceptado reconocer a ChatGPT como un modelo de lenguaje grande (LLM) debido a su capacidad para generar respuestas coherentes en conversaciones. No obstante, es crucial señalar que ChatGPT no es un LLM en sí mismo. Más bien, es una aplicación web diseñada para facilitar la interacción con un LLM subyacente, como el modelo GPT-3 de OpenAI, que es el responsable de procesar las conversaciones utilizando diversas técnicas.

Los LLMs nacen de una evolución constante de la inteligencia artificial generativa y el procesamiento del lenguaje natural, a partir de la década del 2010 aparecen las redes neuronales recurrentes (RNN) [27], diseñadas para procesar secuencias de datos, como texto o series temporales. Y es que en 2018 OpenAI, lanzó GPT-1, uno de los primeros LLMs cuya arquitectura se basaba en los Transformers, introducida en el artículo "Attention is All You Need" [38], los Transformers son una red neuronal, pero que trabaja de forma paralela, a diferencia de las RNNs que trabajan de forma secuencial. Esta arquitectura desde ese momento se ha ido estableciendo como la dominante para los LLMs, gracias a la captura de dependencias a largo plazo.

La evolución de los LLMs ha sido notable en los últimos años. GPT-3, desarrollado por OpenAI, fue uno de los primeros modelos en demostrar capacidades avanzadas en generación de texto. GPT-4 y otros modelos han continuado esta tendencia, ofreciendo mejoras significativas en precisión y coherencia. En la figura 2.3 se puede apreciar la evolución de los modelos desde el 2018 donde empezaron a aparecer los primeros modelos como Bert y GPT-1, hasta mediados de 2023. Hoy en día sigue aumentando la cifra de LLMs considerablemente y cada vez con una mejor precisión, más parámetros y una mayor eficiencia.

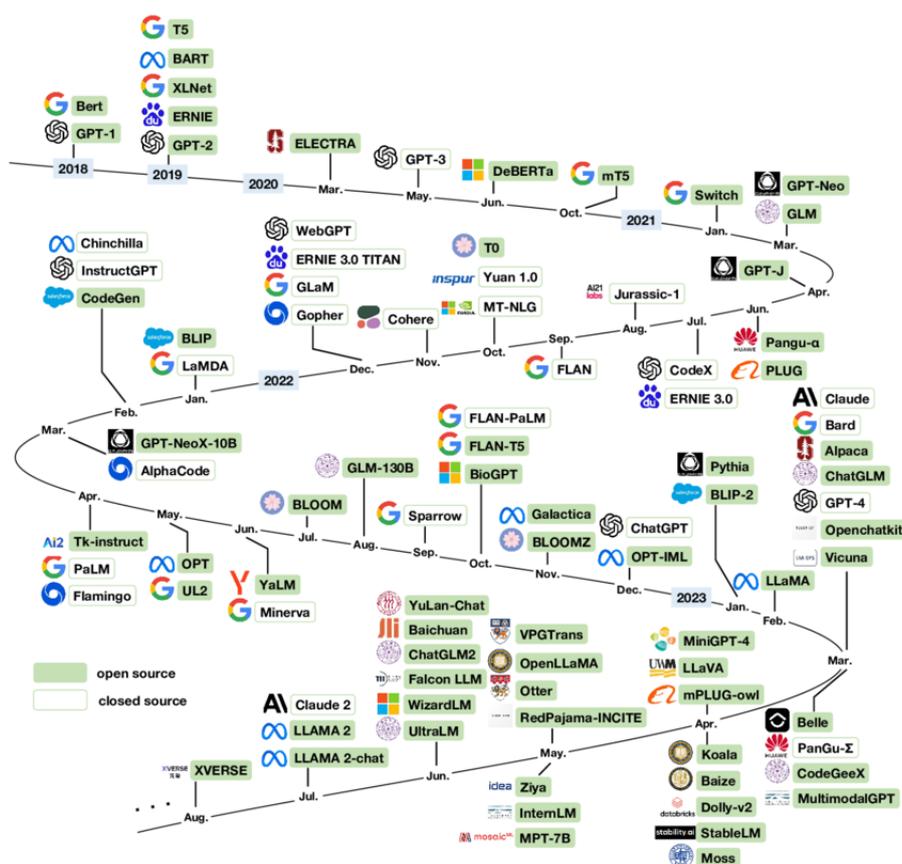


Figura 2.3: Timeline LLMs. (Fuente: Examining User-Friendly and Open-Sourced Large GPT Models: A Survey on Language, Multimodal, and Scientific GPT Models [16])

En contraposición, uno de los problemas más latentes de los LLMs son los sesgos que puedan existir en los datos de entrenamiento, perpetuando actitudes o juicios. Otro de los problemas es la sostenibilidad e impacto ambiental que generan el uso de los LLMs, ya que para entrenar y operar con dichos modelos se tiene un impacto muy alto en el medio ambiente, lo que plantea buscar una nueva forma de uso a largo plazo, ya que los modelos requieren de una cantidad de recursos y potencia computacional muy elevada [34].

2.4. Retrieval-Augmented Generation (RAG)

Además de la tecnología NLP, en 2020 se publicó un artículo titulado "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks"[25] por Facebook AI, hoy en día conocido como Meta, en el que se trataba la programación con Retrieval-Augmented Generation (RAG) [17]. Dicha programación combina la recuperación de información con la generación de texto, permitiendo a los Large Language Models (LLMs) [8] acceder a bases de datos de conocimiento para mejorar la calidad y precisión de sus respuestas. Esta tecnología está aumentando su uso exponencialmente desde la fecha de publicación del artículo.

La Generación Aumentada por Recuperación (RAG) es una técnica emergente en el campo del Procesamiento de Lenguaje Natural (NLP) que combina la capacidad de

recuperación de información con la generación de texto. Esta integración tiene el potencial de mejorar significativamente la precisión, relevancia y contextualización de las respuestas generadas por los modelos de lenguaje.

RAG se podría resumir en sus dos componentes o ideas principales, la recuperación de la información y la generación de texto. Respecto a la recuperación de la información, esta parte se encarga de buscar y recuperar la información más relevante en una base de datos o árbol de conocimiento de documentos, buscando siempre las coincidencias de las palabras clave y similitud semántica entre lo que se pide y la información a devolver. Respecto a la generación de texto, este se apoya en los modelos de lenguaje grandes con los que busca generar respuestas coherentes y contextualizadas con la información extraída.

En la figura 2.4 se puede apreciar la estructura del sistema RAG, donde el usuario realiza una query y el retrieval se encarga de encontrar los documentos que tengan una mayor similitud con dicha query. Una vez se ha hecho ese proceso, se pasa la información al LLM con un prompt y este se encarga de generar una respuesta.

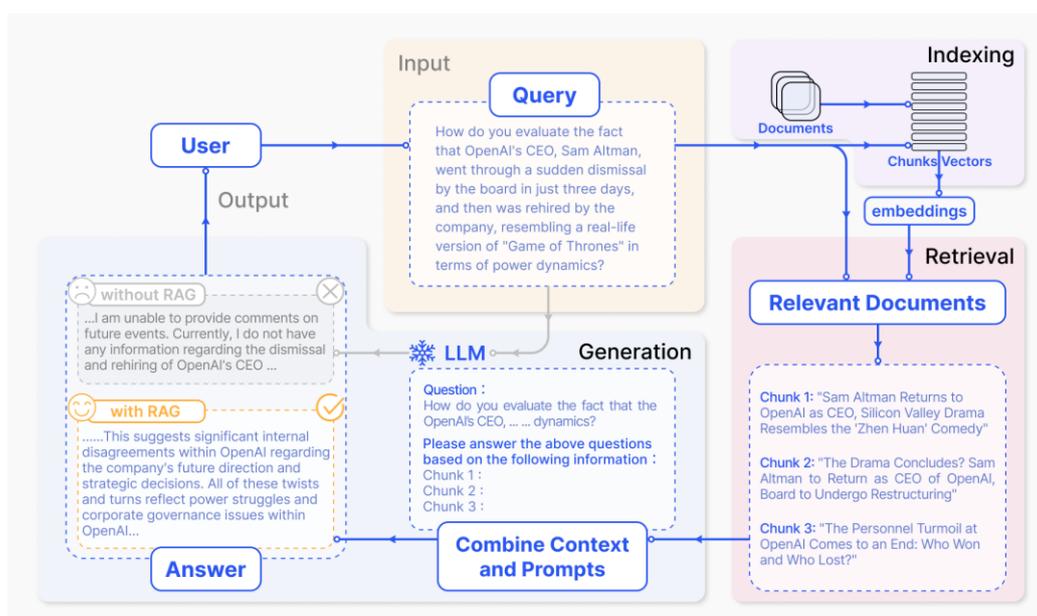


Figura 2.4: Estructura de la Generación Aumentada por Recuperación (RAG). (Fuente: <https://arxiv.org/pdf/2312.10997>)

La estructura comentada en el párrafo anterior se puede incorporar a varios tipos de arquitecturas RAG, donde en la figura 2.5 se pueden apreciar 3 tipos diferentes: naive RAG, advanced RAG y modular RAG. De entre estas, para este proyecto se va a buscar aplicar la arquitectura avanzada. Esta elección está basada en que, a pesar de ser más compleja que la arquitectura naive, permite tener una mejor precisión en la recuperación y realizar una mayor optimización de los chunks y de los fragmentos relevantes antes de enviarlos al LLM. La arquitectura modular de RAG tiene un muy elevado coste tanto computacional como de desarrollo, así como de conocimientos técnicos para poder implantarla.

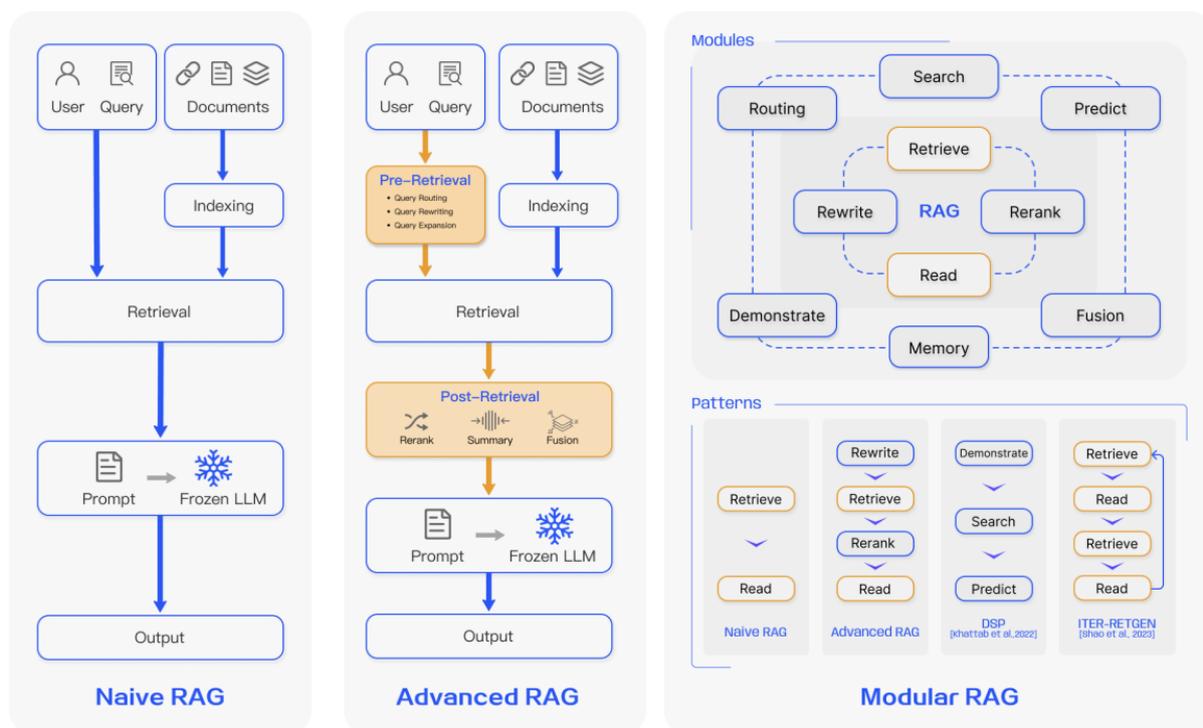


Figura 2.5: Arquitecturas RAG. (Fuente: <https://arxiv.org/pdf/2312.10997>)

Para llevar a cabo este proyecto se valoró la posibilidad de aplicar un enfoque basado en Fine-tuning [20]. Sin embargo, se descartó dado el enfoque dinámico del contexto a utilizar y el coste computacional que implica el Fine-tuning. En este sentido, el Fine-tuning busca ajustar un modelo para ser capaz de responder mejor a tareas específicas utilizando datos de entrenamiento adicionales (haciendo que aprenda más el modelo) más relacionados con el contexto. El problema de esta aproximación es que para que el modelo aprenda este nuevo contexto se necesitan muchos ejemplos y, además, resulta bastante poco flexible a los cambios del mismo. Dado que lo que se quiere generar es un modelo conversacional adaptado a cliente, el Fine-tuning resulta poco adecuado en este contexto. En la tabla 2.1 donde se puede ver una comparativa entre ambos enfoques.

Tabla 2.1: Comparación entre Fine-Tuning y RAG.

Aspecto	Fine-Tuning	RAG
Proceso	Ajuste de un modelo preentrenado con datos específicos.	Combina recuperación de información y generación de texto.
Ventajas	Adaptación precisa, buen rendimiento en tareas específicas.	Maneja preguntas abiertas, acceso a gran corpus de datos.
Desventajas	Requiere datos etiquetados, riesgo de sobreajuste.	Complejidad en implementación, infraestructura de recuperación necesaria.
Aplicaciones	Clasificación de texto, análisis de sentimientos, generación en dominios específicos.	Sistemas de preguntas y respuestas, chatbots avanzados.

RAG tiene la capacidad de ser incorporado en muchos sectores de nuestra sociedad, como la asistencia personalizada, investigación, atención al cliente, cuidados de la salud, etc. Sin embargo, a pesar de tener muchas posibilidades de adopción, hay que tener mucho cuidado con su desarrollo y con la actualización de las bases de datos de donde se extrae el contexto. De este modo, para proporcionar las mejores respuestas, se necesita ser capaces de escalar el sistema eficientemente sin que se vea comprometido al aumentar el volumen de datos y la complejidad a la hora de incorporar distintos tipos de datos(texto, imágenes, audio, etc.).

2.5. Aplicaciones de RAG y LLMs en Diferentes Sectores

Respecto a la aplicación de la tecnología RAG, la mayoría de los sistemas basados en el modelo de Generación Aumentada por Recuperación (RAG) se centran más en otro tipo de aplicaciones diferentes a la planeada en este proyecto, como puede ser las aplicaciones empresariales [6] donde se busca mejorar la eficiencia o el acceso a informes para trabajadores de recursos humanos [1]. Estas aplicaciones empresariales han demostrado ser extremadamente beneficiosas para automatizar tareas repetitivas y proporcionar acceso rápido a información crítica. En este sentido, la programación RAG puede ayudar a filtrar grandes volúmenes de solicitudes de empleo, encontrar información relevante sobre empleados, o generar informes detallados de desempeño de manera eficiente.

También se han encontrado usos en otros sectores, como entornos médicos para detectar enfermedades o en campos de energía y sostenibilidad [15]. Respecto al ámbito médico, RAG puede ser utilizado para revisar y analizar grandes cantidades de datos de pacientes, ayudando a los médicos a diagnosticar enfermedades [28] con mayor precisión y rapidez. En cuanto al campo de la energía, RAG puede ayudar a optimizar la producción y el consumo de energía mediante el análisis de datos en tiempo real.

No obstante, las aplicaciones mencionadas anteriormente no abordan directamente las necesidades interpersonales o contextuales específicas del sector de la sociedad que se cubre en este proyecto. Estas necesidades pueden requerir interacciones más intuitivas. La personalización y contextualización de las respuestas son cruciales para que las personas mayores se sientan comprendidas y apoyadas. Las aplicaciones empresariales y técnicas a menudo priorizan la eficiencia y funcionalidad por encima de la empatía y la relación personal, lo que las hace menos adecuadas para este propósito.

Aunque no existen proyectos que utilicen sistemas RAG, hay iniciativas basadas en la tecnología NLP que buscan mejorar la inclusión de las personas mayores en la sociedad. Por ejemplo, el proyecto DIAL, conocido como "Tecnologías asistidas de voz multiplataforma para la detección y abordaje de soledad no deseada en personas mayores", impulsado por la Universidad de Valencia, donde se desarrolla un dispositivo de asistencia por voz. Este dispositivo ayuda a combatir la soledad no deseada en personas mayores, aumentando su autonomía y conexión social.

Otro proyecto interesante que utiliza LLM y NLP es Celia IA, desarrollado en atlantTtic en la Universidad de Vigo. Sin emplear un sistema RAG, este proyecto aprovecha la inteligencia artificial generativa y la integración de LLMs, junto con biometría de voz y NLP. Gracias a estas tecnologías, han creado un asistente virtual que puede mantener

conversaciones sobre temas de salud, noticias, familia, curiosidades, entre otros. Esto permite que el sistema se adapte a las preferencias del usuario de manera continua. La figura 2.6 muestra cómo es la interfaz del proyecto.

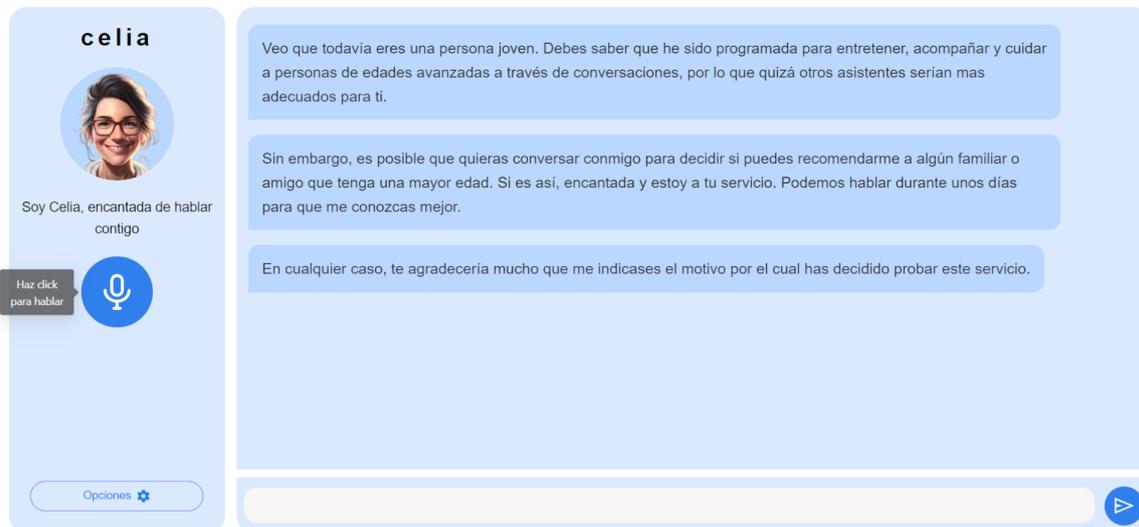


Figura 2.6: Interfaz Web CeliaIA. (Fuente: <https://celiatecuida.com/>)

Capítulo 3

Materiales y métodos

3.1. Modelos en local vs en la nube

Antes de iniciar a comentar cómo se ha llevado a cabo el desarrollo del proyecto, es importante destacar las diferencias que existen a la hora de implementar un LLM ya sea a nivel local o accediendo a la nube a través de llamadas a su API. Cada uno de estos tipos de implementación de modelos tiene sus ventajas y sus desafíos a la hora de implementarlos.

En primer lugar, los modelos locales ofrecen varias ventajas significativas, en el caso de este proyecto se utiliza el modelo local Mistral 7B. La implementación local otorga a los usuarios un control total sobre su hardware, sus datos y los propios LLM. Además, los LLM locales pueden reducir el tiempo de respuesta del modelo al no tener que hacer consultas al exterior. Esto es especialmente útil para aplicaciones que se ejecuten en tiempo real y requieran respuestas rápidas. Otra ventaja crucial es la mejora en la privacidad y la seguridad de los datos, ya que la información confidencial permanece bajo el control del usuario y en el dispositivo donde se ejecuta, sin ser compartida con un proveedor externo.

Sin embargo, ejecutar LLM localmente también presenta ciertos desafíos, como puede ser la configuración inicial del servidor local donde se ejecuta el modelo, esto puede ser muy costoso y requiere una inversión inicial significativa en hardware y software. Además, tener un LLM local puede incluir el ajuste y la capacitación personalizada de los modelos, lo cual requiere de conocimientos técnicos avanzados y una experiencia previa para ser capaz de dominarlos.

Por otro lado, los modelos basados en la nube, como PoliGPT, ChatGPT3.5 y ChatGPT4, accedidos a través de APIs. Proporcionadas por la Universidad Politécnica de Valencia y OpenAI, ofrecen una serie de ventajas respecto a los modelos locales. Las plataformas en la nube se destacan por proporcionar recursos escalables, además pueden manejar sin esfuerzo las demandas de hardware y disponen un gran almacenamiento de datos, permitiendo así ajustar los recursos según sea necesario. Asimismo, los servicios en la nube simplifican el proceso de desarrollo con un simple acceso a la API, agilizando así la creación y la implementación de los modelos.

No obstante, los LLM basados en la nube también tienen sus inconvenientes. Los usuarios pueden encontrarse con un control limitado sobre la infraestructura y los

detalles de implementación cuando dependen de este tipo de servicios, es decir, no se puede realizar una personalización tan alta en el modelo. Puesto que el modelo no es manejado por el usuario, sino que eso es problema del proveedor. Asimismo, el uso de modelos en la nube implica tener un menor nivel de seguridad, puesto que se está compartiendo datos con el servicio.

3.2. Modelos utilizados

Para el desarrollo del proyecto, se han utilizado cuatro LLMs: Mistral7B, PoliGPT, ChatGPT3.5 y ChatGPT4.

Mistral 7B es un modelo de lenguaje con 7 mil millones de parámetros diseñado para ofrecer un buen rendimiento y eficacia. Mistral 7B supera a otros modelos más grandes como pueden ser Llama2 13B o Llama 1 34B. A pesar de ser un modelo con un menor número de parámetros, Mistral está asociado con herramientas como Hugging Face para facilitar la implementación y tiene licencia de Apache 2.0. Para este proyecto, se ha utilizado el modelo desde la plataforma Google Colab para poder ejecutarlo con garantías, debido a la falta de un hardware y unas especificaciones como es la GPU que permite ejecutar el modelo. El hardware utilizado para ejecutar el modelo en este entorno virtual está formado por el tipo de GPU NVIDIA Tesla T4, con una arquitectura de tipo Turing, una memoria GPU de 16 GB GDDR6 y un total de 40 núcleos, esta GPU es conocida por su capacidad de procesamiento eficiente y su adecuada memoria para la ejecución de modelos de lenguaje de gran tamaño.

Llama-3 es un modelo de lenguaje con 70 mil millones de parámetros, este es el modelo que está dentro de PoliGPT, modelo al que se accede a través de una API que ofrece la Universidad Politécnica de Valencia. Dicho modelo ofrece características superiores a la media y una muy alta eficiencia. Llama-3-70b-Instruct teóricamente supera a otros modelos de menor tamaño como son Mistral 7B y GPT-3.5. Este modelo pertenece a Meta y tiene la licencia de Llama 3 Community.

GPT-3.5 es un modelo de lenguaje que destaca por su equilibrio entre rendimiento y eficiencia. Este modelo supera a otros modelos más pequeños como Mistral 7B, aunque queda por detrás de modelos más grandes como Llama-3-70b-Instruct. La integración de GPT-3.5-Turbo-0125 con su API proporcionada por OpenAI facilita su implementación sin la necesidad de mantenimiento de hardware o software local como ya se ha comentado anteriormente y al igual que pasa con PoliGPT y GPT4, dicha API es de pago.

GPT4 es un modelo de lenguaje de última generación con un número de parámetros significativamente mayor, diseñado para proporcionar el mejor rendimiento y precisión del mundo de los modelos.

Como se puede ver en la tabla 3.1 ahí aparece el rating que tiene cada modelo en la plataforma de Chatbot Arena, donde GPT4 está considerado el mejor de los 4, seguido de Llama3, GPT3.5 y Mistral en último lugar. Posteriormente con los resultados se comprobará si los resultados de los modelos concuerdan con el ofrecido por dicha plataforma.

Tabla 3.1: Comparación de modelos en Chatbot Arena ELO Ratings [7].

Model	Chatbot Arena ELO Rating
GPT-4o-2024-05-13	1287
Llama-3-70b-Instruct	1207
GPT-3.5-Turbo-0125	1104
Mistral-7B-Instruct-v0.2	1072

3.3. Desarrollo

A continuación, se comentará las fases del proyecto y que es lo que se ha ido realizando durante su desarrollo, desde la implementación de cada una de las partes del sistema RAG, junto con sus desafíos, hasta el despliegue final en una web interactiva para probar el sistema junto con los modelos.

3.3.1. Problemática de RAG

Los sistemas RAG tienen un potencial muy alto y cada vez irá aumentando su uso en más ámbitos de nuestra sociedad, sin embargo, tienen un problema muy importante al usar LLMs y son las alucinaciones. Las alucinaciones se podrían describir como las respuestas que genera el modelo que no se basan en la información recuperada por el contexto o información irreal, ya sea a través de datos inexactos o inventados y que por lo tanto afectan al correcto funcionamiento del sistema.

Las alucinaciones son muy importantes y deben ser tratadas puesto que pueden provocar la falta de confianza del usuario, haciendo que empiece a dudar de la efectividad al ver los fallos, afectando a la adopción del proyecto y de su utilidad. Asimismo, puede afectar a que los usuarios le pregunten algo y les dé una respuesta que no debía dar, pudiendo llegar a causar serios problemas en el usuario si se trata algún tema delicado.

Sin embargo, para evitar esta problemática hay que tener en cuenta los siguientes puntos clave, con el objetivo de reducir las alucinaciones.

- **Prompts de calidad:** Prompts claros y específicos para que el modelo pueda proporcionar las respuestas precisas.
- **Calidad de los datos:** Utilizar datos de entrada de una alta calidad que estén libres de errores y sean datos consistentes.
- **Chunking de datos:** Establecer un buen estilo de fragmentación entre los disponibles para el procesamiento por el modelo a utilizar, evitando siempre la pérdida de información por una mala fragmentación.
- **Recuperación de la información:** Tener un buen ranker que permita recuperar la información de mayor calidad.
- **Transformación de consultas:** Implementar la reformulación de consultas para mejorar la precisión cuando se pregunte sobre algo y se haya comentado anteriormente y poder referenciar el tema, evitando que acceda a un contexto erróneo, proporcionando una mayor precisión.
- **Memoria conversacional:** Incorporar memoria al sistema y manejarla de forma

que esté disponible para la reformulación de consultas y se mantenga por debajo de los límites de los tokens permitidos.

Todos estos aspectos se irán aplicando a lo largo del desarrollo del proyecto para buscar la mejor precisión del sistema RAG y evitar las alucinaciones.

3.3.2. Creación de Documentos

Para poder tener un contexto al que acceder es necesario generar una plantilla de documentos, ya que como información a almacenar en la base de datos del sistema RAG se necesita la creación de un pequeño cuestionario a las personas que vayan a utilizar el proyecto. En este cuestionario, las preguntas formuladas para las personas mayores que se utilizarán en el sistema RAG han sido cuidadosamente diseñadas por una psicóloga, la cual se enfocó en crear un conjunto de preguntas para formar una conversación completa e interesante, adaptada a las capacidades y necesidades de los participantes. Todo esto con el objetivo de guardar los aspectos más relevantes de sus vidas y poder almacenar algunas de sus experiencias. Estas preguntas abarcan una amplia variedad de temas, donde se puede encontrar información básica, antecedentes familiares, intereses y pasatiempos, etc. Este documento puede verse en el anexo 6.2.

3.3.3. Base de Datos y Chunk Splitting

Antes de empezar a comentar la estructura del sistema, es importante comentar el tipo de base de datos que se va a utilizar en el sistema RAG y como se van a almacenar los documentos para su posterior acceso.

En primer lugar, respecto a la base de datos existían dos posibilidades, una base de datos vectorial o un grafo de conocimiento, ambas opciones son las más utilizadas dentro del mundo RAG, sin embargo, cada una de ellas tiene sus ventajas o inconvenientes dependiendo del planteamiento que se haga. Para este proyecto proporcionaba una mejor forma de trabajo las bases de datos vectoriales puesto que proporcionan mejores resultados cuando se manejan datos no estructurados como lo es el texto plano. Además, ofrecen una alta escalabilidad y una recuperación rápida, lo cual es un punto clave para aplicaciones que requieran responder a consultas en tiempo real. En contraposición, los árboles de conocimiento proporcionan peores resultados en estos aspectos. Otra de las características que decantaron el uso de este tipo de base de datos es que además de ser muy eficiente con texto plano, también rinde con otro tipo de datos, como tablas, imágenes, etc.

Una vez escogida la base de datos, se pasa al almacenamiento de los documentos. En este trabajo, se ha implementado la segmentación de documentos utilizando el método `RecursiveCharacterTextSplitter` con unos parámetros de `chunksize` de 500 y `chunkoverlap` de 200. Este método lo proporciona la biblioteca de `LangChain`, que permite tener en cuenta la estructura del documento al ir dividiendo el documento por párrafos y en el caso de que sea necesario por frases. De este modo se consigue que cada chunk (segmento) mantenga un contexto coherente. Configurar el método de división de los documentos con un `chunksize` de 500 y un `chunkoverlap` de 200 equilibra la necesidad de contextos suficientemente largos para que el modelo sea capaz de procesar la información y sepa que es lo que se está tratando. Además, el solapamiento de 200 caracteres garantiza que los puntos clave o ideas importantes que podrían aparecer al final de un párrafo no se pierdan en fragmentación y aparezcan

en el siguiente fragmento. La fragmentación de los documentos es una de las partes más importantes de preparación a la estructura RAG, permitiendo así que cuando se acceda la información, se obtenga información de calidad. Fragmentar el texto mientras se mantiene la semántica y el contexto de cada parte mejora positivamente los resultados en la fase de recuperación, aumentando así la precisión y relevancia de los datos recuperados.

3.3.4. Estructura RAG

La figura 3.1 muestra la integración del sistema de Generación Aumentada por Recuperación (RAG) en los Modelos de Lenguaje Grande (LLMs), enfatizando tanto los componentes de la interfaz del chatbot como su arquitectura fundamental. Este sistema está optimizado para chatbots que deben navegar entre preguntas y respuestas complejas, confiando exclusivamente en datos no estructurados. Los diferentes elementos de la aproximación basada en la programación RAG son:

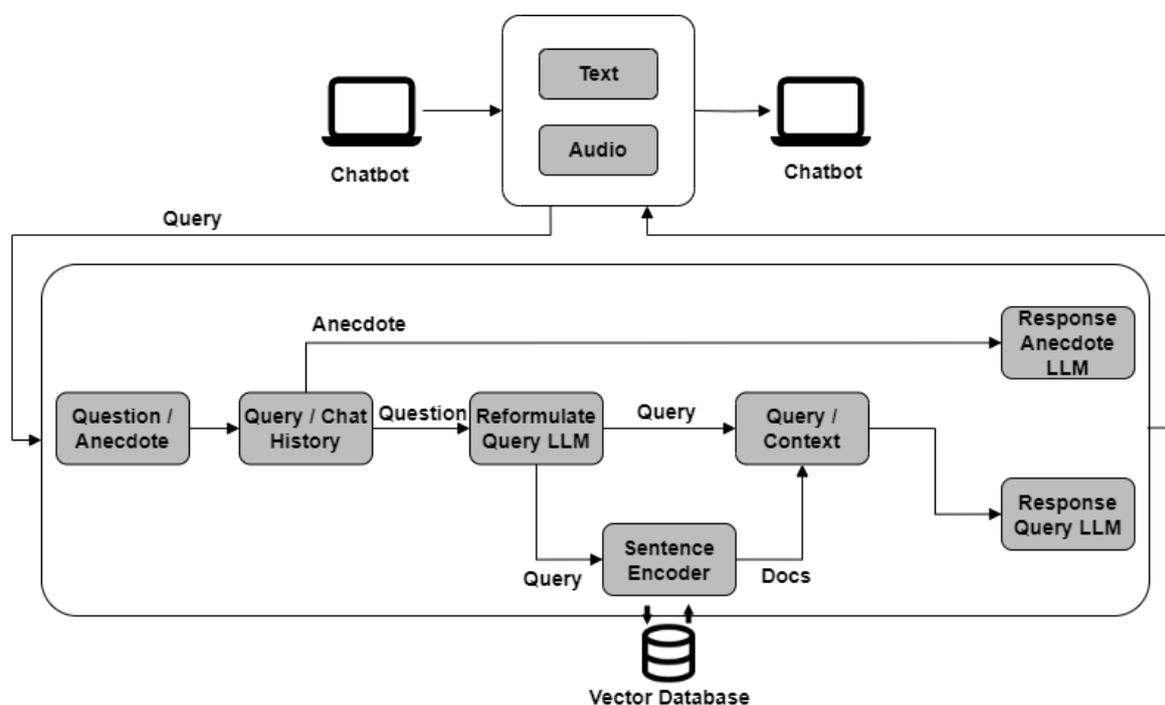


Figura 3.1: Estructura del sistema.

- **Entrada del usuario:** El proceso de interacción se inicia cuando un usuario se relaciona con el chatbot, proporcionando una entrada en forma de texto o audio a través de la interfaz web, esto se comentará más adelante en la sección de la interfaz web. El soporte del sistema para múltiples canales de entrada garantiza una experiencia de usuario más inclusiva y versátil en diferentes contextos.
- **Procesamiento de entrada:** Independientemente del formato de entrada, el sistema estandariza toda la información recibida a texto. Se utilizan tecnologías avanzadas de reconocimiento de voz para las entradas de audio para garantizar que el texto resultante refleje con precisión el mensaje original hablado, esto también se comentará en la sección de interfaz web.

• **Consulta inicial:** El texto procesado se utiliza para formular una consulta inicial. Esta puede ser una pregunta directa, una curiosidad o una anécdota compartida por el usuario. El sistema está adaptado para manejar ambos tipos de entradas, ajustando sus respuestas a la naturaleza contextual de la consulta, es decir, el sistema a través de una consulta al modelo detectará el tipo de query y dependiendo de eso realizará una llamada al modelo con un prompt distinto para proporcionar una respuesta más precisa. En el caso de que sea anécdota pasará directamente a realizar la consulta sin reformular la query como se puede ver en la figura 3.1, sin embargo, si es el caso de una pregunta pasaría a reformular la pregunta. En la figura 3.2 se puede apreciar el prompt para diferenciar entre anécdota y pregunta.

```
messages = [
  {'role': 'system', 'content':
    f'Determina si la siguiente oración es una pregunta sobre la vida de una persona o una anécdota.'
    f'Permítanme compartir un par de ejemplos que serán importantes.'
    f'Oración: ¿Cómo conocí a mi mejor amigo?'
    f'Respuesta: pregunta'
    f'Oración: ¿Puedes recordarme cuál es mi lugar de nacimiento?'
    f'Respuesta: pregunta'
    f'Oración: Una vez me encontré con mi mejor amigo en un parque y pasamos todo el día jugando.'
    f'Respuesta: anécdota'
    f'Frase: Cuando era pequeño, una tarde conocí a Pelé, el mejor jugador de la historia de Brasil.'
    f'Respuesta: anécdota'
    f'Ahora, con esos ejemplos, aquí está la oración de entrada real.'
    f'Oración: {user_input}'
    f'Simplemente responde la palabra anécdota o pregunta'
  }
]
```

Figura 3.2: Prompt pregunta o anécdota.

No obstante, se intentó llevar a cabo la implementación de agentes de LangChain para gestionar de manera más eficiente las interacciones y ser capaces de diferenciar entre los distintos tipos de interacción y mejorar la precisión en la reformulación de consultas. Sin embargo, estos agentes presentaron un nivel de complejidad más elevado de lo esperado y debido a restricciones de recursos y temporales se descartaron. De esta forma, se focalizaron los esfuerzos en la implementación de soluciones menos complejas (aunque menos escalables) y más adaptadas a las necesidades específicas del proyecto.

• **Análisis y reformulación de la consulta:** En los casos en que el usuario presenta una pregunta, el LLM identifica los elementos clave mencionados a lo largo de la conversación y reformula la pregunta por si hace referencia a algo tratado anteriormente para mejorar la precisión cuando se haga la recuperación. Este paso es crucial para asegurar consultas bien definidas y dirigidas al sistema de recuperación. Sin embargo, hay que tener mucho cuidado con esta práctica puesto que los modelos tienen un límite de tokens de entrada a la hora de leer. Por lo tanto, es importante vigilar que nunca se sobrepase dicho límite y se permita el correcto funcionamiento del sistema. Por ello, se estudió cual era la mejor forma de mantener un buen uso de la memoria como se puede ver en la figura 3.3, donde se destacan ConversationSummaryMemory. Esta se basa en ir almacenando la conversación en forma de resumen para no sobrecargar la memoria. También se usó el ConversationBufferWindowMemory, el cual funciona como una ventana temporal que retiene información de las últimas k interacciones que se le indiquen como parámetro y que, en el caso de la figura 3.3, se establece en 12. Ambos tipos de memoria conversacional tienen unos resultados parecidos, los

cuales permiten tener una buena cantidad de información en la memoria y no superar los límites de tokens, siendo las dos opciones factibles para su implementación[36].

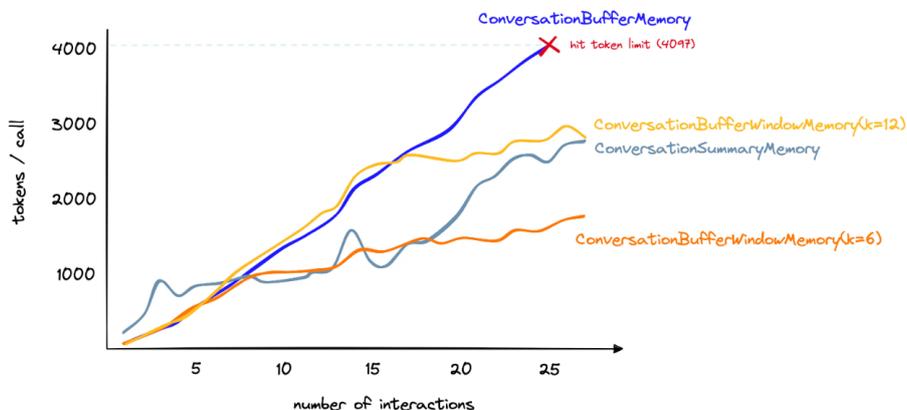


Figura 3.3: Tipos de memoria.

En la figura 3.4 se puede observar el prompt para poder reformular la consulta en el caso que sea necesario.

```
messages = [
  {'role': 'system', 'content':
    f'Dado un historial de chat y la conversación con el usuario '
    f'que podría hacer referencia al contexto en el historial de chat, formule una pregunta independiente '
    f'que se puede entender sin el historial de chat. NO respondas la pregunta, '
    f'simplemente reformúlelo si es necesario y, en caso contrario, devuélvalo tal como está. '
    f'Utilice siempre la PRIMERA PERSONA cuando reescriba la pregunta porque debe ser el ser humano.'

    f'Permítanme compartir un par de ejemplos que serán importantes.'

    f'Si no ve ningún historial de chat, DEBE devolver la "Pregunta" tal como está:'

    f'Historial de chat:'
    f'Pregunta: ¿Cuál es mi nombre?'
    f'Pregunta independiente:'
    f'¿Cuál es mi nombre?'

    f'Si esta es la segunda pregunta en adelante, deberías reformular correctamente la pregunta de esta manera:'

    f'Historial de chat:'
    f'Humano: ¿Cuál es mi nombre?'
    f'AI:'
    f'Tu nombre es Miguel López'

    f'Pregunta: ¿Cuántos años tengo?'
    f'Pregunta independiente:'
    f'¿Cuántos años tengo, si soy Miguel López?'

    f'Ahora, con esos ejemplos, aquí está el historial de chat real y la pregunta de entrada.'
    f'Chat: {memory}'
    f'Devuelve SOLO la PREGUNTA, nada más'},
  {
    'role': 'user', # Este es el prompt, la pregunta o instrucción del usuario
    'content': user_input
  }
]
```

Figura 3.4: Prompt reformulación pregunta.

Un prompt en un LLM es una instrucción inicial que se le da al modelo para guiar su respuesta. Es esencial puesto que determina cómo el modelo interpretará y responderá a la pregunta, es decir, en el caso de la figura 3.4 se puede observar el prompt, que en este caso es un prompt de ejemplo para que el modelo sea capaz de comprender que es lo que debe hacer con el input del usuario.

Por lo que el prompt es uno de los elementos más importantes en el buen desarrollo y funcionamiento del LLM, el prompt puede suponer que el LLM sea capaz de entender de la forma en la que debe actuar, proporcionando una respuesta concisa y precisa o en caso contrario devolver un resultado que no sea el esperado. En consecuencia, se investigó la mejor forma de hacer un buen prompt sobre todo para el modelo Mistral 7B que era en el que más visible era este problema. Pasando de no responder a lo que se pedía, a establecer un buen prompt a partir de prompt engineering [19] y que fuese capaz de responder de manera correcta. Respecto al resto de modelos una vez se estableció el prompt para Mistral, el resto requerían de alguna pequeña modificación, pero la estructura del prompt era la misma.

- **Codificación de la oración:** Después de que la consulta ha sido reformulada, se somete a una codificación pasando a ser una representación vectorial, transformando el texto en un vector semántico. Esta representación vectorial facilita la comparación con entradas en una base de datos vectorial.

- **Búsqueda en la base de datos vectorial:** Utilizando la representación vectorial, generada de la query, el sistema ejecuta una búsqueda dentro de la base de datos vectorial para recuperar fragmentos de documentos que exhiban la mayor similitud con la consulta del usuario. Para ello se utiliza el retriever proporcionado por la base de datos, en este caso FAISS de Meta, donde se almacenan los documentos, el retriever transforma esta consulta en una representación vectorial utilizando un modelo de embeddings (sentence-transformers/paraphrase-multilingual-mpnet-base-v2) [24], diseñado para poder detectar similitudes semánticas para más de 50 idiomas. Este modelo utiliza la técnica de similitud por coseno, en la que se escogen las que tengan un ángulo más próximo entre ellos.

La similitud por coseno entre dos vectores A y B se define como:

$$\text{sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \quad (3.1)$$

La similitud por coseno varía entre 0 y 1, donde:

- 1 indica que los vectores apuntan en la misma dirección.
- 0 indica que los vectores son ortogonales (no tienen similitud).

- **Generación de respuesta:** Basado en la información recuperada, el sistema emplea el modelo del que disponga el sistema para generar una respuesta adaptada a la query del usuario. El proceso de generación de respuestas varía dependiendo del tipo de consulta (ya sea una anécdota o una pregunta), utilizando indicaciones e información apropiadas para ofrecer la respuesta más relevante y precisa según el contexto proporcionado. En el caso de una anécdota el sistema no accederá al contexto puesto que no se requiere y simplemente se encargará de responder de una forma correcta y coherente a lo que se está contando, en cambio, para una pregunta sí

que será necesario el acceso a la base de datos vectorial para la extracción de los documentos, con el objetivo de tener el contexto necesario para responder.

Una vez que se devuelve la respuesta al usuario, el sistema procede a guardar tanto la respuesta como la pregunta en la memoria para mantener el historial y la conversación en la base de datos. En esta parte se observó que cuándo se realiza un almacenamiento en la base de datos de una conversación, existía la posibilidad de almacenarla entera o detectar si existían documentos previos los cuales ya tenían información similar. Por lo que sería interesante no almacenarlos y, de ese modo, optimizar la necesidad de recursos. Además de esto, otro de los problemas consistía en saber cuándo era buen momento para almacenar la información en la base de datos para no molestar al usuario durante las interacciones. El proceso de almacenamiento de los datos en la base de datos es un proceso que, si tiene una conversación extremadamente larga, puede ralentizar el funcionamiento del chatbot. Por ello, a partir de la experimentación y el análisis, se determinó cuáles eran los mejores momentos para almacenar una conversación. De esta forma, se concluyó que lo mejor era en cada respuesta del LLM se almacenase la interacción y así reducir el tiempo de carga en la base de datos y no ralentizar en exceso el tiempo de interacción. Asimismo, así se evita que cualquier usuario cierre sin querer el chatbot y se pierda toda la conversación con la consiguiente pérdida de información.

- **Almacenamiento conversación en base de datos:** Para almacenar la conversación primero se genera un documento concatenando las entradas del usuario y las respuestas generadas durante la interacción. Posteriormente, se utiliza el modelo de embeddings de la base de datos vectorial para calcular la similitud entre el nuevo documento y los documentos existentes en la base de datos. Si encuentra que el nuevo documento es superior al umbral de 0.75, que delimita la similitud con los textos ya almacenados. En ese caso, indica que ya hay algo similar a la información actual y, por lo tanto, no se almacenaría. En el caso de que sea menor se añade a la base de datos para mantener un registro actualizado de las interacciones. Finalmente, se guarda la base de datos actualizada localmente para futuras consultas.

Este método no solo permite almacenar las conversaciones para análisis posterior, sino que también optimiza el almacenamiento al evitar duplicados o contenido demasiado similar, asegurando que solo se guarden interacciones únicas o significativamente distintas en la base de datos.

En cualquier caso, tanto las anécdotas como cualquier otro tipo de interacción se utilizarán para retroalimentar la base de datos del sistema. De esta manera, con el tiempo, será posible generar interacciones más personalizadas adaptadas al contexto de cada usuario.

- **Salida al usuario:** Finalmente, la respuesta generada se convierte nuevamente en formatos de texto y audio y se presenta al usuario, completando el ciclo de interacción.

En cuanto a la implementación de los modelos tanto a nivel local en el anexo 6.2, como en la nube en el anexo 6.2, se pueden ver las diferencias en la complejidad a la hora de acceder por ejemplo al LLM en local donde había que realizar una amplia configuración, añadiendo pipelines y posteriormente incorporar dicho pipeline a una cadena de LLM. No obstante, al LLM que se accede en la nube, pasando la apikey y la URL y pasándole un mensaje donde se adjunten la query y el prompt ya nos devuelve la respuesta.

3.3.5. Interfaz Web

Para el desarrollo de la interfaz web para los usuarios se utilizó la biblioteca Streamlit, esta biblioteca destaca por su facilidad de uso y su simplicidad a la hora de programar, ya que se puede programar una interfaz sin la necesidad de conocer tecnologías adicionales más complejas. Además, permite personalizar y adaptar la interfaz a las necesidades del desarrollador, tanto a nivel visual como a nivel de organización de los componentes.

Gracias a todo esto se consigue desarrollar una interfaz sencilla de utilizar e intuitiva para las personas de edad avanzada donde, además de ser sencilla visualmente, se les permite poder interactuar con esta a través del audio. De esta manera, las personas puedan conversar con el modelo clicando en el botón del micrófono en la figura 3.5 que muestra la interfaz. Además de permitir hablar por audio también se permite escuchar la respuesta del modelo (como se puede ver en la parte inferior del envío del mensaje). De esta forma, se facilita y naturaliza la tarea de interacción a aquellas personas que no tengan ya la capacidad tanto de escribir como de leer.

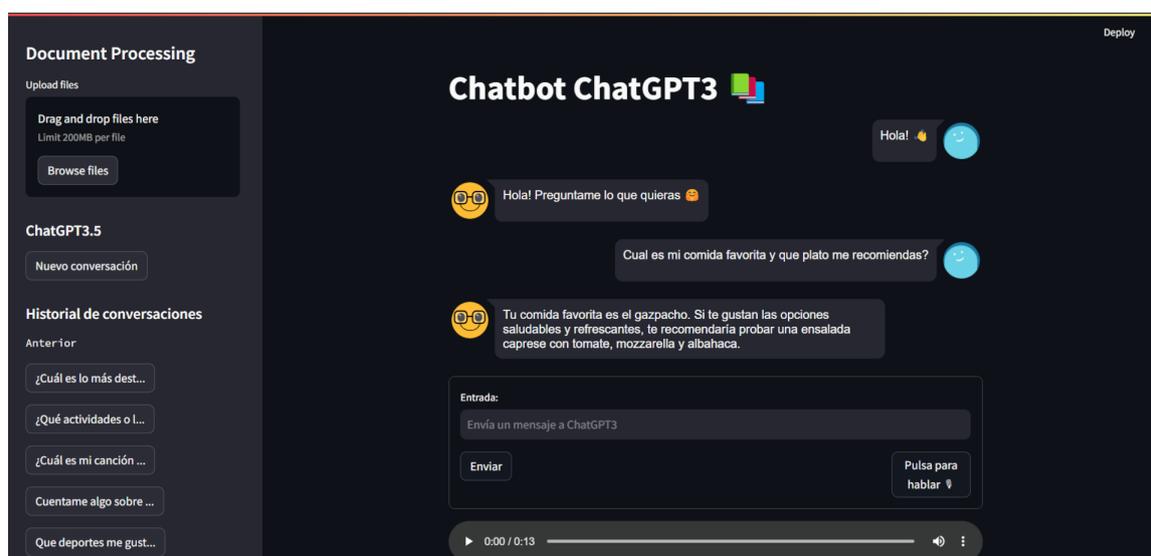


Figura 3.5: Interfaz web.

Respecto a la implementación del sistema de audio se probaron dos opciones, el paquete gTTS que permite convertir el texto a audio, con la ayuda de speech recognition para captar el audio y el servicio en la nube de Amazon Polly al que se accede a través de una API. Ambas opciones proporcionaban una salida de audio aceptable, sin embargo, con el paquete de gTTS la salida que proporcionaba de audio era mucho más comprensible. El servicio de Amazon Polly, a pesar de ser un servicio de una gran empresa, ofrecía resultados peores, tanto a nivel de comprensibilidad como en la fluidez con la que habla el bot que, en ciertos momentos, se hacía difícil de entender. Además, este último tiene una implementación mucho más compleja que la otra opción en la que, con un par de líneas de código ya estaba preparado para su uso. Destacar que ambas opciones eran de ámbito gratuito y ninguna de las dos es capaz de mostrar una entonación o sentimientos mientras está leyendo el mensaje, opciones que sí que ofrecen algunas alternativas de pago.

Además, la interfaz implementada incluye un historial de conversaciones que organiza automáticamente las interacciones por días: Hoy, Ayer, Hace 7 días y Anterior. Este historial permite a los usuarios acceder fácilmente a conversaciones pasadas, facilitando el seguimiento de los temas discutidos y proporcionando una referencia útil para futuras interacciones.

3.4. Evaluación del Modelo

Para evaluar la efectividad y aplicabilidad de los modelos de lenguaje grande (LLMs) con programación RAG, se utilizaron varios modelos para determinar la similitud entre las respuestas esperadas y las generadas. Estas métricas incluyen el modelo BERT, el modelo basado en tokens, el modelo de similitud coseno, el modelo Sentence-BERT, un evaluador Lang-Smith y las métricas más comunes para este tipo de evaluaciones, Rouge, Bleu y Meteor. Además de estas métricas, también se utilizará un cuestionario para que un grupo de personas responda con un valor indicando el grado de similitud de las respuestas. Todas las métricas serán utilizadas para los idiomas castellano e inglés, identificando así en que idioma el sistema RAG proporciona mejores resultados.

A continuación, se mostrará con más al detalle cada uno de los modelos que se utilizaron en la evaluación. Todas las métricas utilizadas proporcionan un valor más próximo a 1 cuanto más cercano sea el resultado y 0 cuanto más lejano.

El modelo de **similitud coseno** mide la similitud entre la respuesta esperada (v_e) y la respuesta generada (v_g), vectorizando los textos con TfidfVectorizer y calculando el coseno del ángulo entre los vectores resultantes.

La fórmula de la similitud coseno se expresa como:

$$\text{Similitud Coseno}(v_e, v_g) = \frac{v_e \cdot v_g}{\|v_e\| \|v_g\|} \quad (3.2)$$

El modelo **BERT** utiliza embeddings generados por BERT para representar los textos y posteriormente calcular la similitud de coseno entre estos embeddings.

La fórmula de la similitud coseno entre los embeddings generados por BERT se expresa exactamente igual que la fórmula 3.2.

La métrica basada en **tokens** evalúa la similitud contando los tokens comunes entre las respuestas esperadas (T_e) y generadas (T_g). Haciendo la intersección entre ambos y dividiendo entre los tokens de la respuesta esperada. Este método es simple pero efectivo para medir la coincidencia directa de palabras entre dos textos.

La fórmula de la similitud basada en tokens se expresa como:

$$\text{Similitud de Tokens}(T_e, T_g) = \frac{|T_e \cap T_g|}{|T_e|} \quad (3.3)$$

El modelo **Sentence-BERT** mejora la precisión al utilizar embeddings de Sentence-BERT, diseñados específicamente para capturar similitudes semánticas en frases y oraciones. Este modelo también utiliza la similitud coseno para comparar los embeddings de las respuestas esperadas (v_e) y generadas (v_g). Para la evaluación

multilingüaje, se utiliza el modelo calificado ya comentado en la sección de búsqueda en base de datos vectorial de la estructura RAG.

La fórmula de la similitud coseno entre los embeddings generados por Sentence-BERT se expresa exactamente igual que la fórmula 3.2.

El evaluador **Lang-Smith** utiliza un (LLM) para calificar la exactitud de las respuestas generadas en comparación con las respuestas esperadas, basado en un conjunto estructurado de prompts y evaluaciones. Donde con esos datos es capaz de dar un valor 1 en el caso de que crea que coinciden o 0 en el caso de que no, es decir, esta forma de evaluación no tiene valores intermedios entre el 0 y el 1.

La fórmula para calcular la puntuación de exactitud utilizando el evaluador Lang-Smith se puede describir como:

$$\text{Similitud} = \text{LLM}(\text{Pregunta}, \text{Respuesta Correcta}, \text{Respuesta Generada}) \quad (3.4)$$

ROUGE mide la superposición de n-gramas, palabras y subsecuencias entre el texto generado y el de referencia (respuesta generada y esperada respectivamente), centrándose en el recall (recuperación). Es útil para evaluar la cobertura de información relevante en el texto generado [26].

La fórmula de ROUGE-N aplicada individualmente a cada pregunta (Q_i) y respuesta (A_i) se expresa como:

$$\text{ROUGE-N}(Q_i, A_i) = \frac{\sum_{\text{gram}_n \in Q_i} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{\text{gram}_n \in Q_i} \text{Count}(\text{gram}_n)} \quad (3.5)$$

Donde:

- $\text{Count}_{\text{match}}(\text{gram}_n)$ es el número de n-gramas en la referencia, respuesta esperada, que coinciden con la respuesta generada.
- $\text{Count}(\text{gram}_n)$ es el número total de n-gramas en la referencia.

BLEU calcula la precisión de n-gramas en la respuesta generada en comparación con la esperada, utilizando una penalización por longitud para evitar generar respuestas demasiado cortas. BLEU es comúnmente utilizado en la evaluación de traducción automática [29].

La fórmula de BLEU se expresa como:

$$\text{BLEU} = BP \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right) \quad (3.6)$$

Donde:

- BP es el factor de penalización por longitud.
- w_n es el peso asignado a los n-gramas de longitud n .
- p_n es la precisión de los n-gramas.

METEOR considera la alineación de n-gramas entre la respuesta generada y la esperada, y combina precisión, recall y una penalización por la fragmentación. METEOR también tiene en cuenta la similitud semántica y las variaciones morfológicas [3].

La fórmula de METEOR se expresa como:

$$\text{METEOR} = F_{mean} \cdot (1 - \text{Penalty}) \quad (3.7)$$

Donde:

- F_{mean} es la media armónica de la precisión y el recall.
- Penalty es una penalización por fragmentación.

Respecto al cuestionario de evaluación de similitud se utilizó la plataforma de Google Forms. El cuestionario está formado por cuarenta y cuatro preguntas, la mitad en un idioma y la mitad en otro, para poder medir la similitud entre las respuestas esperadas y las respuestas generadas para cada idioma, donde se evalúa del 1 (Ninguna similitud) al 5 (Total similitud).

Además, se intentó utilizar RAGAS de LangChain para la evaluación, sin embargo, debido al tiempo totalmente excesivo que requería para poder proporcionar los resultados se descartó. Ya que el tiempo de respuesta que proporcionaba no era viable para los objetivos del proyecto y poder obtener resultados en un tiempo razonable. Por lo tanto, se optó por utilizar el resto de métricas y herramientas de evaluación que se han comentado anteriormente.

Una vez comentadas los modelos, volvemos con la evaluación donde se escogerá uno de los LLMs seleccionados, en este caso PoliGPT, para ser evaluado analizando la correlación entre los resultados obtenidos por el cuestionario basado en una evaluación humana y los valores obtenidos de todas los modelos con dicho LLM. Permitiendo así que el análisis identifique el modelo más adecuado para evaluar los LLMs. Este proceso se basa en la hipótesis de que la evaluación humana tiene una mayor validez que el resto a la hora de valorar las respuestas, por lo que se buscaba qué métrica correlaciona mejor con las evaluaciones humanas. Una vez se haya hecho eso, se utilizará la métrica con una mejor correlación para evaluar cada uno de los LLMs y determinar cuál tiene el mayor porcentaje de similitud.

Además de ese análisis también se realizará un estudio sobre el tiempo de respuesta de cada LLM, obteniendo así el tiempo medio de cada modelo y ver la evolución con el aumento de la memoria en el LLM, analizando si esto afecta mucho al tiempo de respuesta o no. Donde se hará un análisis significativo de los tiempos para confirmar las hipótesis.

Finalmente, se realizará otro cuestionario, donde se buscará determinar que LLM ofrece una mayor naturalidad en sus respuestas, evaluando específicamente la fluidez, coherencia y comprensibilidad del texto generado.

En base a los resultados y análisis realizados, se determinó cuál de los LLMs analizados era el más eficiente y efectivo en el contexto del problema abordado en el presente proyecto.

Capítulo 4

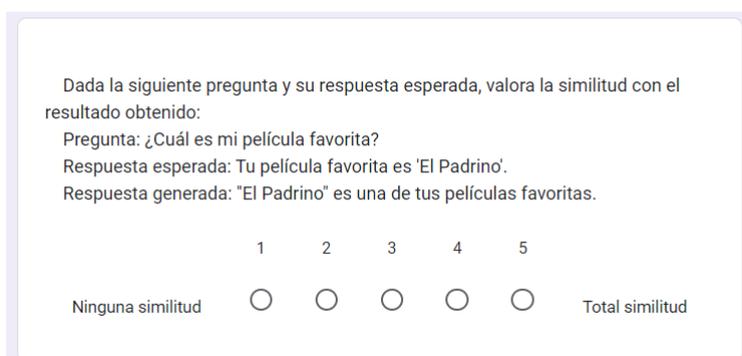
Resultados

En esta sección se presentan los hallazgos obtenidos a partir de los experimentos desarrollados durante la evaluación de los LLMs. Estos resultados incluyen la evaluación de similitud y correlación de los modelos en dos idiomas, español e inglés, así como un análisis detallado de la velocidad de respuesta y la naturalidad de las respuestas generadas por cada modelo. El objetivo principal de estos experimentos es determinar que modelo es mejor en cada uno de estos ámbitos y encontrar cual es el que mejores resultados proporciona en global.

4.1. Evaluación de la Similitud y Correlación

Con esta evaluación lo que se quería era ver que modelo tenía una mejor valoración según la métrica que sea más similar a la evaluación manual y que proporcione una mejor media para los LLMs. Pudiendo así extrapolarla a todos los modelos para poder obtener una valoración similar pero de forma automática y poder determinar el mejor LLM.

En primer lugar, se comentarán los resultados del cuestionario que mide la similitud de las respuestas generadas respecto a las esperadas a partir de la evaluación humana para el modelo PoliGPT. Este cuestionario consistía en cuarenta y cuatro preguntas, veintidós para cada idioma, donde la similitud media (para valores de similitud escalados entre 0 y 1) de cada idioma es de un 0.75 para el castellano y un 0.69 para el inglés. En la figura 4.1 se puede apreciar un ejemplo del cuestionario.



Dada la siguiente pregunta y su respuesta esperada, valora la similitud con el resultado obtenido:

Pregunta: ¿Cuál es mi película favorita?

Respuesta esperada: Tu película favorita es 'El Padrino'.

Respuesta generada: "El Padrino" es una de tus películas favoritas.

1 2 3 4 5

Ninguna similitud Total similitud

Figura 4.1: Cuestionario similitud.

Una vez extraídos los resultados del cuestionario y tener la similitud que se le da a cada respuesta, se realizó la correlación con los resultados de cada una de las métricas comentadas en el apartado de evaluación del modelo.

Para poder realizar una correcta correlación, se realizó un pequeño estudio sobre la muestra de los datos. De este modo, se analizó si estos seguían una distribución normal o si existe alguna relación lineal, para permitir escoger la mejor correlación posible. Las figuras 4.2 y 4.3 muestran los datos resultantes del análisis del cuestionario en castellano y en inglés respectivamente. En estos gráficos se puede ver como en el gráfico de dispersión de ambos idiomas se muestran los puntos azules que indican el valor de la similitud, la línea roja (regresión lineal) y las líneas verdes (regresión polinomial). Para ambos gráficos de dispersión, tanto para el inglés como el castellano, se observa como las valoraciones siguen una tendencia alineada, pero con ciertas variaciones, lo que nos hace indicar que la mejor opción es Spearman.

Por otro lado, se encuentran los histogramas, donde a simple vista se puede ver como no siguen una distribución normal perfecta, ya que hay ciertas variaciones, sobre todo para los valores entre 0 y 0.2, lo que hace que se respalde la idea de aplicar la correlación de Spearman en lugar de la típica correlación de Pearson (para la que es preferible disponer de una distribución normal en los datos).

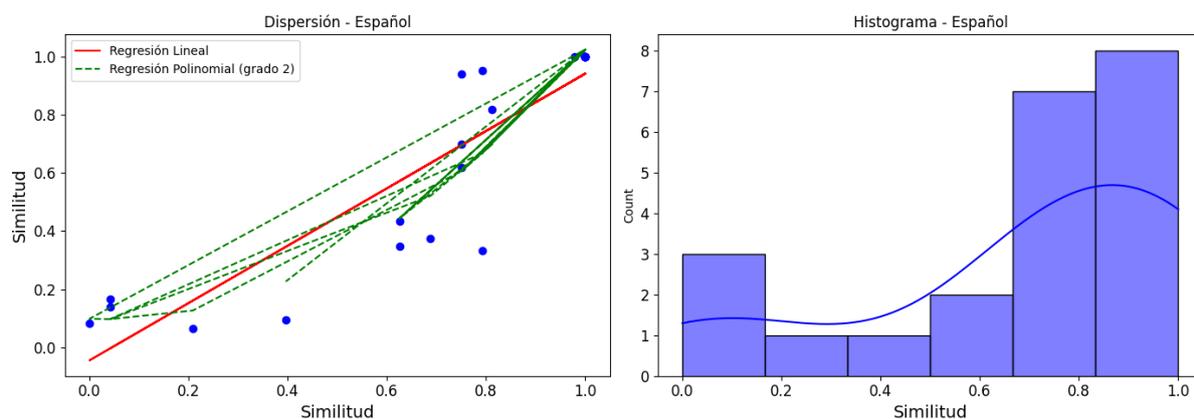


Figura 4.2: Distribución datos castellano.

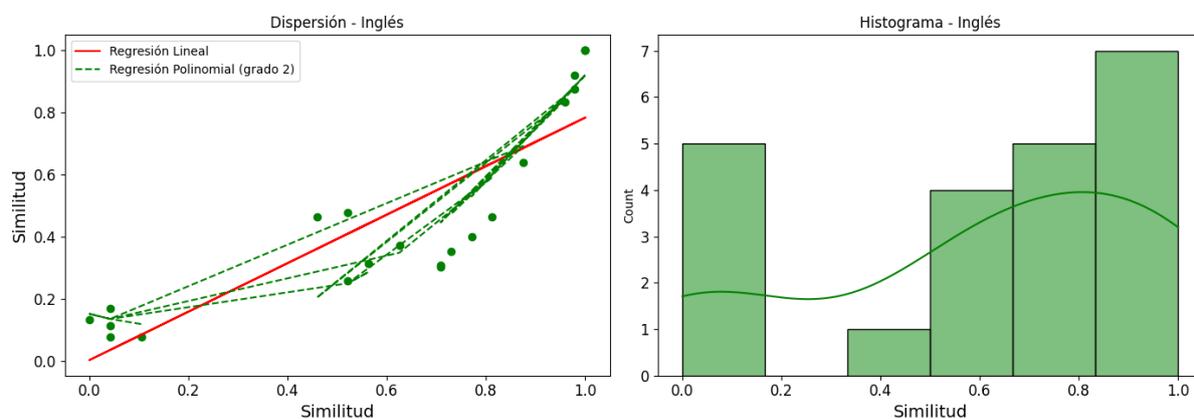


Figura 4.3: Distribución datos inglés.

Una vez escogida la mejor métrica de correlación, se realizó la correlación. Esto se puede ver tanto en la Tabla 4.1 que hace referencia a la correlación en castellano y la tabla 4.2 que muestra los resultados para la correlación en inglés.

Tabla 4.1: Comparación de métricas de similitud y correlación en español.

Métrica	Similitud (Español)	Correlación (Español)
Bert embedding	0.95	0.77
Coseno	0.62	0.92
Tokens	0.73	0.79
Sentence BERT	0.97	0.88
LangSmith	0.95	0.36
BLEU	0.51	0.94
ROUGE	0.63	0.94
METEOR	0.71	0.90

Tabla 4.2: Comparación de métricas de similitud y correlación en inglés.

Métrica	Similitud (Inglés)	Correlación (Inglés)
Bert embedding	0.87	0.90
Coseno	0.50	0.78
Tokens	0.69	0.69
Sentence BERT	0.93	0.80
LangSmith	0.81	0.67
BLEU	0.28	0.81
ROUGE	0.47	0.89
METEOR	0.58	0.89

Hay que destacar que la métrica de ROUGE proporciona varios valores: recall, precisión y F1 Score. En este caso, se escogió el F1 Score ya que es el promedio de la precisión y el recall. Esto permite encontrar un equilibrio entre las palabras generadas relevantes y poder cubrir la mayoría del contenido relevante.

Para interpretar los resultados observables en las tablas 4.1 y 4.2, es fundamental enfocarse en los valores más realistas de correlación y similitud media. Una tasa de similitud buena indica que las respuestas generadas por los modelos son muy parecidas a las esperadas. Si además la correlación es alta, significa que la métrica utilizada es consistente con la evaluación manual.

En la tabla 4.1, que presenta los resultados en español, se observa que las métricas de similitud de Bert embedding y sentence BERT muestran una similitud alta, de 0.95 y 0.97 respectivamente. Sin embargo, estas métricas también presentan una correlación con la evaluación manual de 0.7 y 0.88 respectivamente, lo que sugiere que, aunque correlacionan bastante bien con la evaluación manual, tienden a ser demasiado optimistas y, por tanto, poco realistas.

Un caso a destacar es el de LangSmith, que tiene una similitud alta de 0.95 pero una correlación baja de 0.36, lo que indica que esta métrica no se alinea bien con la evaluación manual y, por tanto, no se considera fiable.

Por otro lado, las métricas METEOR y la basada en tokens destacan positivamente.

METEOR tiene una buena similitud de 0.71 y una alta correlación de 0.9, mientras que la métrica basada en tokens presenta una similitud de 0.75 y una correlación de 0.79. Estas métricas no solo proporcionan resultados consistentes, sino que también son realistas y fiables para la evaluación de los modelos.

Con respecto a los resultados en inglés (ver tabla 4.2), el caso más similar al castellano en el que haya unos valores muy elevados son el caso de Bert emedding con una similitud de 0.87 y una correlación del 0.9 y sentence Bert con una similitud del 0.93 y correlación del 0.8, indicando en ambos casos una posible sobreestimación, ya que se es poco realista con esos valores. Por otro lado, como métricas interesantes al igual que en castellano, se destacan con el modelo basado en tokens con una similitud y correlación del 0.69 y METEOR con una similitud media de 0.58, pero con una correlación alta (0.89).

Con todo ello, las métricas METEOR y el modelo basado en tokens son las que tienen un porcentaje de similitud y de correlación bastante equilibrado y realista al compararlo con la evaluación manual. En base a esto, se va a realizar un análisis sobre cada uno de los cuatro modelos para ver cual de las métricas proporciona mejores resultados. Para ello se aplicará sobre un conjunto de respuestas esperadas vs generadas en los cuatro modelos y se observará la media de acierto/similitud de cada modelo. En la tabla 4.3 se aprecia la similitud media de METEOR por modelo tanto para inglés como para castellano. Del mismo modo, la tabla 4.4 muestra los resultados del análisis para el modelo de tokens.

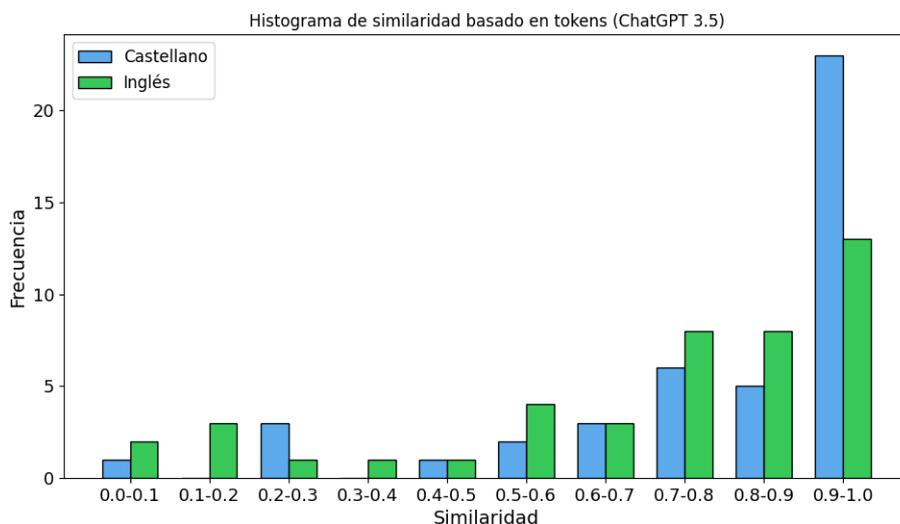
Tabla 4.3: Similitud media METEOR por idioma y LLM.

	GPT4	ChatGPT3.5	PoliGPT	Mistral
Español	0.80	0.74	0.72	0.56
Inglés	0.77	0.58	0.61	0.48

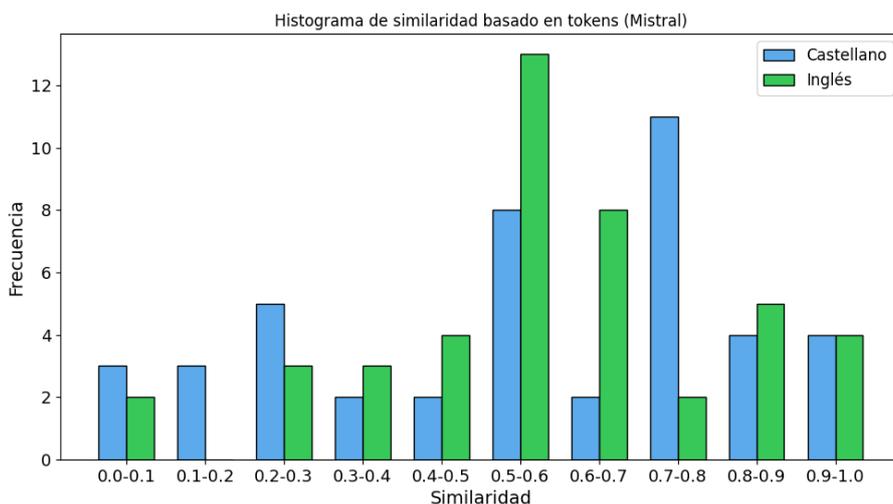
Tabla 4.4: Similitud media tokens por idioma y LLM.

	GPT4	ChatGPT3.5	PoliGPT	Mistral
Español	0.85	0.81	0.73	0.54
Inglés	0.80	0.71	0.69	0.57

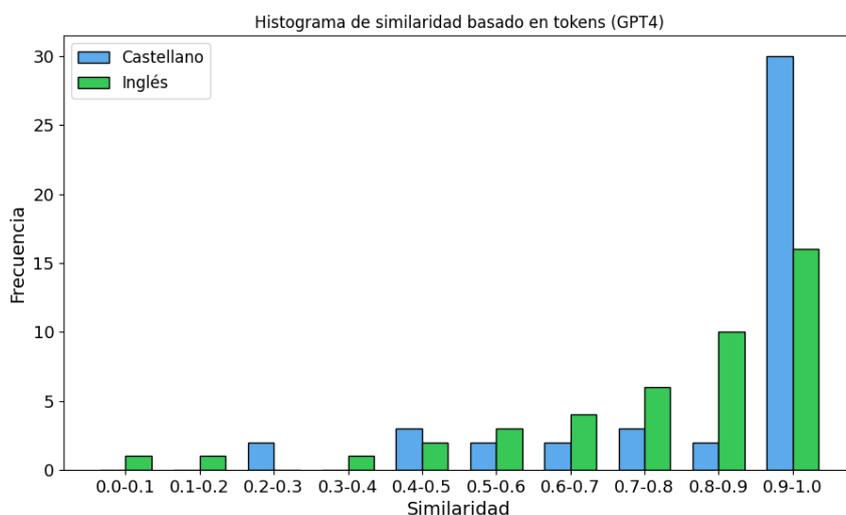
Viendo ambas tablas, se puede observar cómo, en líneas generales, el modelo de tokens proporciona mejores resultados, ligeramente superiores a METEOR, por lo que las valoraciones finales de similitud de los modelos eran en base a este modelo. Esto permite observar cuál de los modelos genera unos mejores resultados y escoger al mejor LLM.



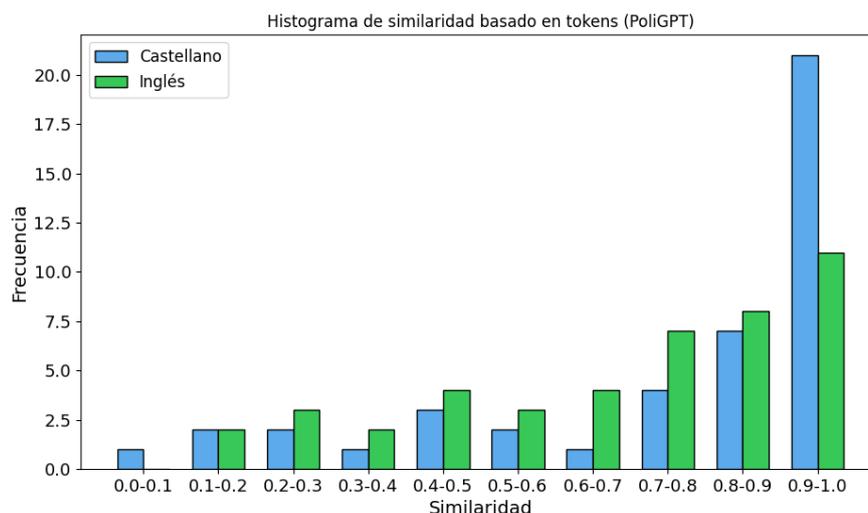
(a) Resultados ChatGPT3.5.



(b) Resultados Mistral.



(c) Resultados GPT4.



(d) Resultados PoliGPT.

Figura 4.4: Resultados comparativos similitudes.

Observando los gráficos, donde se ve la distribución de los valores de la métrica, en la figura 4.4, se puede destacar que PoliGPT, ChatGPT4 y ChatGPT3.5 proporcionan los mejores resultados. En contraste, el gráfico para el modelo Mistral no muestra resultados tan positivos. Estos gráficos son complementarios a la tabla 4.4 y se han realizado para analizar la distribución de la precisión de los modelos utilizando la métrica basada en tokens.

Por lo tanto, con todos estos datos de similitud, se puede observar que el modelo ChatGPT4 en principio es el que mejores resultados proporciona para ambos idiomas. Es notable que, en términos de calificaciones positivas, los tres mejores modelos ofrecen muy buenas calificaciones para el idioma español. Sin embargo, si nos enfocamos en el inglés, ChatGPT3.5 y PoliGPT proporcionan resultados aceptables, pero quedan ligeramente por debajo de GPT4. En contraste, el modelo Mistral ofrece resultados muy pobres para ambos idiomas.

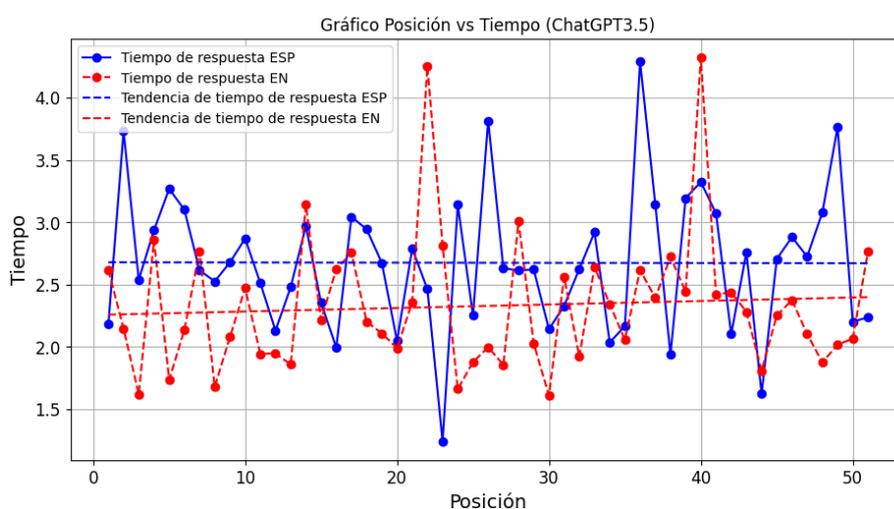
4.2. Evaluación del Tiempo de Respuesta

Por otro lado, se realizó una evaluación de los tiempos de respuesta para cada LLM, donde se observó cómo de rápido responde cada modelo y permitiendo identificar cuál hace que los usuarios tengan un menor tiempo de espera y les cause una menor fatiga a la hora de tratar con él.

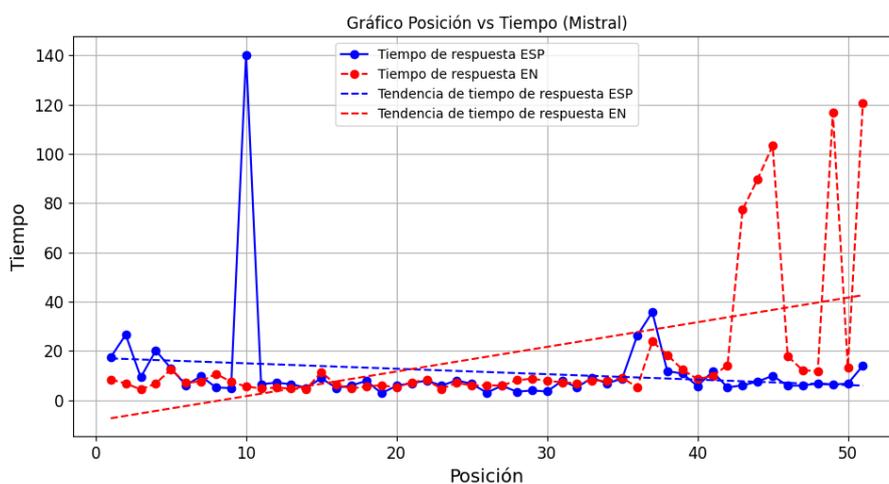
Para la realización del experimento se realizaron en dos ocasiones cincuenta y una consultas a cada modelo, todas estas preguntas estaban almacenadas en una base de datos para así poder tener las mismas preguntas para todos los modelos y que en las dos ejecuciones sean las mismas respuestas esperadas. La doble ejecución se realizó para comprobar que si hubiese datos anómalos o cualquier tipo de patrón extraño que no fuese por una casualidad y así evitar resultados extraños. Una vez se almacenaron los tiempos de cada consulta se guardaron y se analizaron para determinar qué modelo es el mejor.

La tabla 4.5 muestra los tiempos de respuesta promedio para cada idioma, además también se encuentran los gráficos de cada uno de los modelos en la figura 4.5 donde se puede ver cómo evoluciona el tiempo de respuesta a medida que se realizan consultas. En este caso, los gráficos representan la posición vs tiempo puesto que lo que se busca es ver cómo varía el tiempo de respuesta a medida que se va aumentando el historial de la conversación. Conforme se van realizando más interacciones, el número de tokens que se va procesando en las consultas es cada vez mayor y esto puede afectar al rendimiento del sistema.

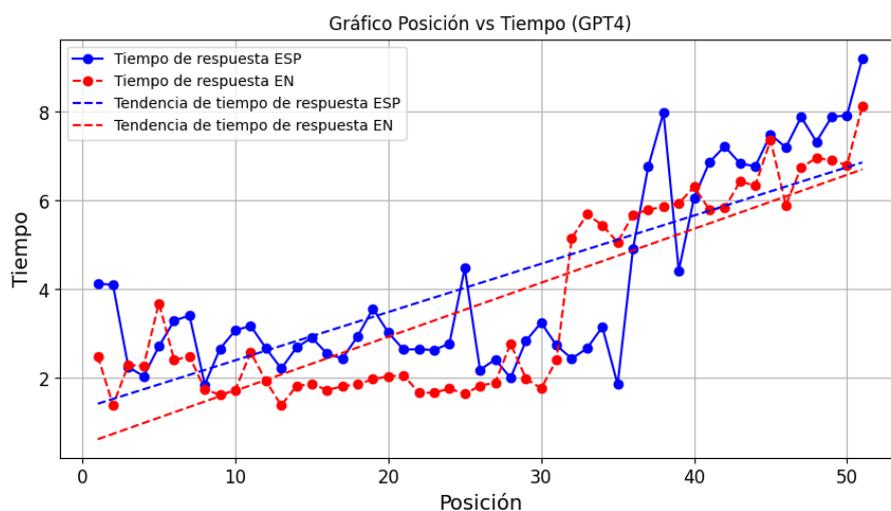
Analizando todos estos datos, tanto los tiempos promedio que se pueden ver en la tabla 4.5, como las tendencias y tiempos de respuesta en los gráficos de la figura 4.5, es evidente que hay una diferencia notable entre un par de modelos y el otro, ya que simplemente observando el eje de la Y ya se puede observar la diferencia de la escala entre los modelos pertenecientes a OpenAI y a Mistral y PoliGPT. Por lo que ya simplemente con los gráficos ya se intuye cuáles son los modelos que mejores resultados pueden proporcionar y cuáles son los que más se ven afectados con el paso de las preguntas.



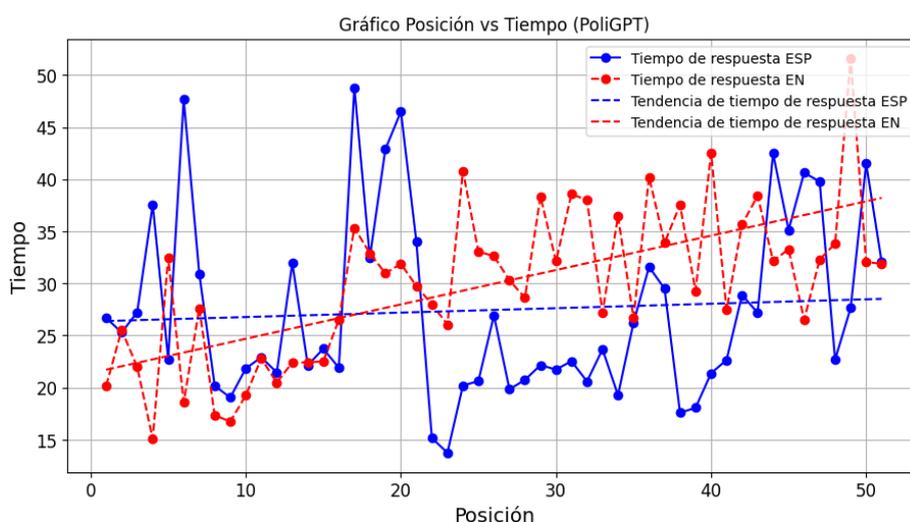
(a) Resultados ChatGPT3.



(b) Resultados Mistral.



(c) Resultados GPT4.



(d) Resultados PoliGPT.

Figura 4.5: Resultados tiempos de respuesta.

Tabla 4.5: Tiempo medio por idioma y LLM.

	ChatGPT4	ChatGPT3.5	PoliGPT	Mistral
Español	4.13	2.68	24.45	11.33
Inglés	3.66	2.33	29.96	17.57

Para tener un mayor peso en las diferencias de tiempos se realizó un análisis de significancia para confirmar si existe una diferencia real entre algunos modelos o como en el caso de ChatGPT3.5 y GPT4 ver sino se considera significativa. Se empezó con un análisis de la significancia [9] de dos factores, puesto que lo que se quería determinar era si los modelos influían en el tiempo, al igual que los idiomas y si hay una relación entre estos dos factores sobre el tiempo. Para ello se establecieron varias hipótesis nulas e hipótesis alternativas.

▪ **Hipótesis nula (H0):**

1. No hay diferencias significativas entre los grupos de la variable independiente Modelo respecto a la variable dependiente Tiempo.
2. No hay diferencias significativas entre los grupos de la variable independiente Idioma respecto a la variable dependiente Tiempo.
3. No hay una interacción significativa entre las dos variables Modelo e Idioma respecto a la variable dependiente Tiempo.

▪ **Hipótesis alternativa (H1):**

1. Hay diferencias significativas entre los grupos de la variable independiente Modelo respecto a la variable dependiente Tiempo.
2. Hay diferencias significativas entre los grupos de la variable independiente Idioma respecto a la variable dependiente Tiempo.
3. Hay una interacción significativa entre las dos variables Modelo e Idioma respecto a la variable dependiente Tiempo.

Para determinar la homogeneidad de las varianzas se utilizó la prueba de Levene (Media) y la prueba de Brown-Forsythe (Mediana). Ambas pruebas demostraron que los tiempos no son homogéneos ($p < 0.001$), sugiriendo así diferencias en la dispersión de los tiempos de respuesta entre los grupos.

Al no seguir una distribución normal, es necesaria la realización de una prueba no paramétrica como Kruskal-Wallis además de ANOVA para poder verificar los resultados, puesto que ANOVA cuando no hay homogeneidad puede proporcionar resultados erróneos.

Tabla 4.6: ANOVA de dos factores sin medidas repetidas.

	df	F	p	Etap2
Modelo	14979.45	89.25	<0.001	0.4
Idioma	397.16	2.37	0.125	0.01
Modelo x Idioma	251.7	1.5	0.214	0.01
Error	400	167.84		

ANOVA proporcionó los resultados de la tabla 4.6, donde se nos indica que respecto a los modelos sí que existen diferencias significativas ($F=89.25$ y p -value prácticamente de 0.00), ya que al tener un p -value inferior a 0.05 se considera significativa la diferencia. En segundo lugar, en la fila del idioma al tener $F=2.37$ y p -value=0.125 nos indica que no existen diferencias significativas entre los idiomas. Y en último lugar la interacción modelo e idioma que también nos indica que no hay una interacción significativa con un p -value de 0.214 y $F=1.5$. Con estos resultados se rechaza la hipótesis nula de los modelos, en cambio se aceptan las hipótesis del idioma y de la interacción modelo-idioma.

Se realizó la prueba post-hoc Bonferroni para identificar específicamente que grupos proporcionan diferencias entre sí. En primer lugar, se aplicó al idioma, cuya diferencia media entre 'es' (español) y 'en' (inglés) es -1.97 con un p -value= 0.125, lo que indica que muestra que no existe una diferencia significativa.

En segundo lugar, se utilizó la prueba para los modelos, cuyos resultados se pueden ver en la tabla 4.7, donde hay diferencias significativas entre todos los modelos excepto entre ChatGPT3.5 y GPT4.

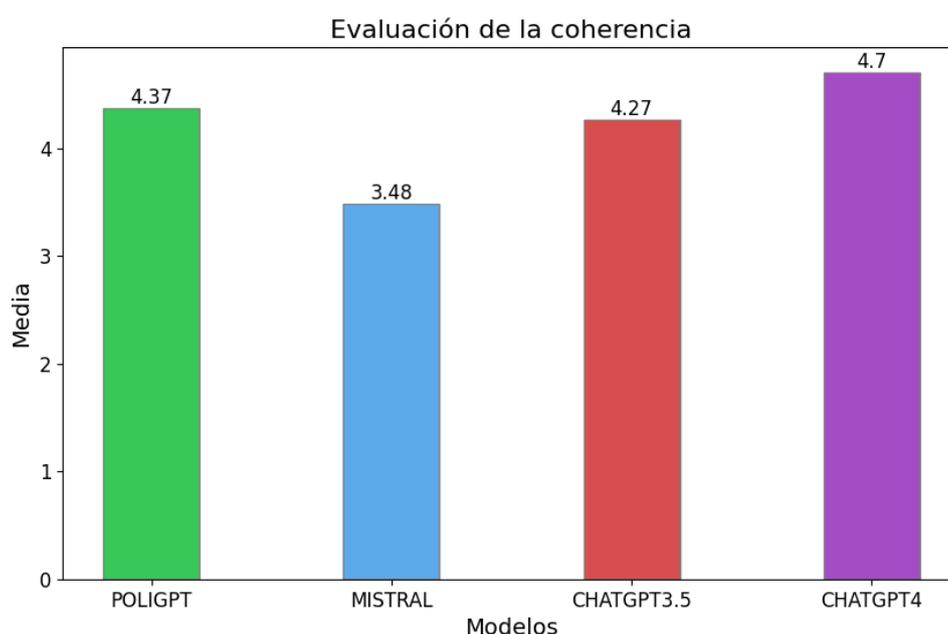
Modelo 1	Modelo 2	p
ChatGPT4	ChatGPT3.5	1
ChatGPT4	PoliGPT	<.001
ChatGPT4	Mistral	<.001
ChatGPT3.5	PoliGPT	<.001
ChatGPT3.5	Mistral	<.001
PoliGPT	Mistral	<.001

Tabla 4.7: Resultados de los modelos Bonferri.

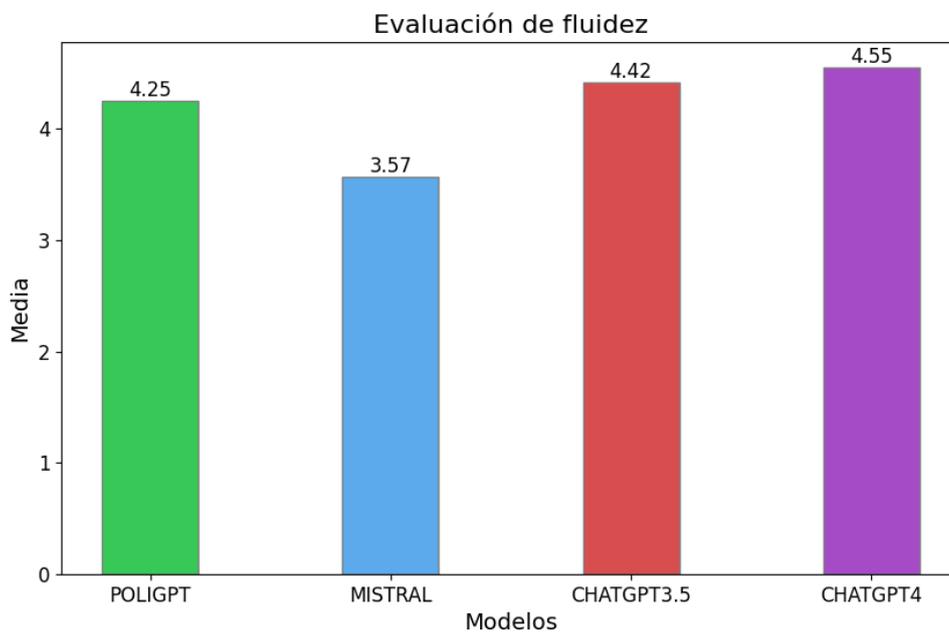
En último lugar, se analizó la interacción modelo-idioma con dicha prueba post-hoc, dando como resultado que las interacciones no muestran diferencias significativas en muchos casos. Existen algunos modelos-idiomas que sí que tiene una interacción significativa, pero son una minoría, lo que refuerza que el idioma no tiene un impacto significativo en el rendimiento de los tiempos de los modelos.

Para complementar los resultados de ANOVA, ya que al no ser datos homogéneos puede no ser tan fiable, se utilizó la prueba de Kruskal-Wallis como alternativa no paramétrica. Kruskal-Wallis proporcionó un valor de p-value = 1.79e-63. Este valor nos acaba indicando que hay diferencias significativas en los tiempos de respuesta entre los diferentes grupos. Concordando así con los resultados del ANOVA, confirmando que hay diferencias significativas entre los modelos y que los tiempos no son atribuibles al puro azar.

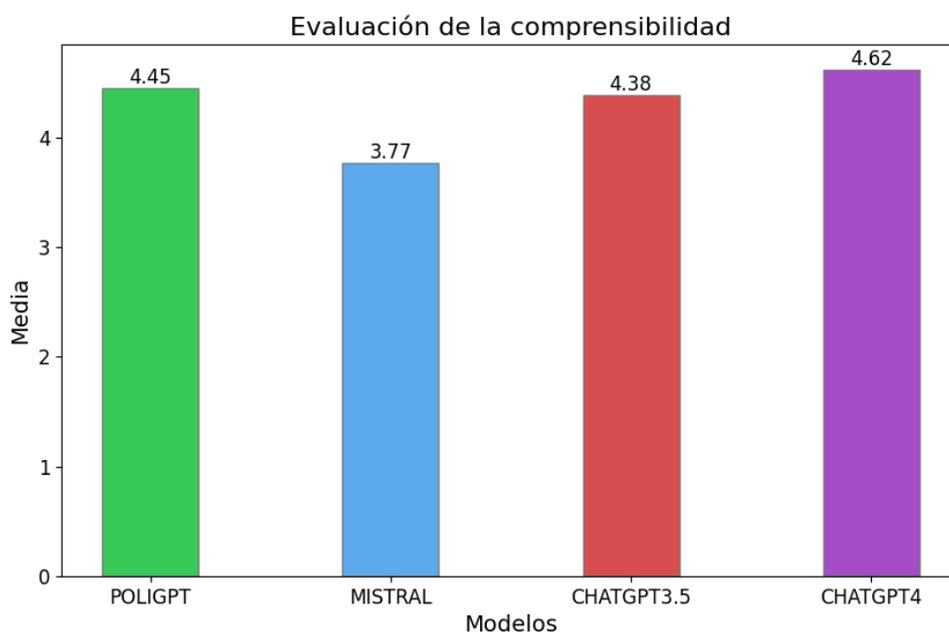
4.3. Evaluación de la Naturalidad de las Respuestas



(a) Resultados Coherencia.



(b) Resultados Fluidez.



(c) Resultados Comprensibilidad.

Figura 4.6: Resultados naturalidad de los LLMs.

Finalmente, para poder tener un poco más de métricas y evaluaciones de los modelos se realizó otro cuestionario para evaluar la naturalidad de las respuestas que los modelos pudieron proporcionar. Este cuestionario solo se realizó en castellano, a diferencia del primero, ya que podían ser irreales los resultados en inglés puesto que

la gente que los realizó no eran ingleses nativos y se corría el riesgo de que su nivel de inglés no fuera el suficiente como para medir la naturalidad de las respuestas. El cuestionario se estableció para medir la fluidez, comprensibilidad y coherencia de las respuestas utilizando 5 preguntas y respuestas por modelo. Esto se realizó también para tener un mayor muestreo y por escoger el mejor modelo con una mejor cantidad de información. Como ejemplo de una de las preguntas y su forma de evaluación se puede observar en la figura 4.7.

Dada la siguiente pregunta y su respuesta evalúala:

Pregunta: Que me recomiendas ver en Valencia

Respuesta generada: Valencia es una ciudad llena de historia, arte y naturaleza. Considera visitar la Ciudad de las Artes y las Ciencias, un complejo arquitectónico que combina elementos futuristas con jardines y lagos. También deberías explorar el centro histórico de Valencia, donde puedes encontrar la catedral de Santa María y la plaza de la Reina. Si eres amante de la naturaleza, no te pierdas el Jardín del Turia, un parque que recorre el río Turia y ofrece áreas verdes y tranquilas para relajarte.

	1	2	3	4	5
Coherencia	<input type="radio"/>				
Fluidez	<input type="radio"/>				
Comprensibilidad	<input type="radio"/>				

Figura 4.7: Cuestionario naturalidad.

Los resultados del cuestionario se se pueden ver en las tres gráficas de la figura 4.6 donde cada una de las gráficas representa una métrica de evaluación y en ellas están los modelos evaluados.

Capítulo 5

Discusión

Para la sección de discusión se interpretan y analizan en profundidad los resultados presentados en el capítulo 4. Comentando lo que significan estos resultados, su relevancia, sus implicaciones, y cualquier limitación que pueda haber en los experimentos realizados.

5.1. Discusión de la Similitud y Correlación

En la evaluación de la similitud y correlación de los modelos PoliGPT basado en Llama3, se realizó un análisis exhaustivo de las respuestas generadas tanto en español como en inglés. Los resultados del cuestionario, con su evaluación manual, mostraron que la similitud media fue de 0.75 para el castellano y 0.69 para el inglés, lo que sugiere que el modelo es más eficaz en la generación de respuestas en español. Esto, en líneas generales, es algo sorprendente puesto que la mayoría, sino la totalidad, de estos modelos son influenciados tanto en el volumen como en la calidad de datos de entrenamiento en inglés. Por ello, lo lógico sería que los resultados indicaran que el modelo responde mejor a las preguntas con un contexto y pregunta en inglés. Uno de los inconvenientes del cuestionario era la limitación de cantidad de personas que pudieron realizar el cuestionario puesto que este también requería de unos quince minutos para poder realizarlo. En este caso, participaron catorce personas en el cuestionario, una muestra pequeña pero suficiente para poder ver cuál es la valoración en líneas generales de las respuestas.

Para comprobar si el resto de los modelos también tenían unos resultados en los que el castellano tuviese una mejor valoración que el inglés, se buscó una correlación para determinar cuál de todos los modelos de evaluación automático era el mejor y más parecido a la evaluación manual. El resultado se extrapoló al resto de LLMs y, de ese modo, se pudo obtener un resultado parecido a la evaluación manual. Para obtener la correlación, se estudió la distribución de los datos para extraer la mejor correlación posible, en este caso Spearman capaz de captar relaciones monotónicas.

Los resultados en las tablas 4.1 y 4.2 mostraron que métricas como METEOR y la basada en tokens proporcionaron valores de similitud y correlación más realistas y consistentes con la evaluación humana. Existían modelos con unos mejores resultados a nivel de similitud y correlación, como Bert embedding y sentence BERT que mostraron altas similitudes, pero sus correlaciones con la evaluación humana indican

una posible sobreestimación de su efectividad, ya que es prácticamente imposible acertar en tan alto porcentaje. Una vez aplicados METEOR y el modelo basado en tokens a los LLMs, los resultados indicaron unos mejores resultados de similitud con el modelo basado en tokens.

Estos resultados indican que, aunque hay variaciones en las métricas de similitud, la correlación con la evaluación humana sugiere que algunas métricas son más fiables que otras. En contraste, métricas como METEOR y la basada en tokens muestran un equilibrio más realista y preciso entre similitud y correlación, lo que las hace más adecuadas para evaluar la calidad de las respuestas generadas por los LLMs.

Con el modelo basado en tokens, una pequeña anomalía a destacar es en el gráfico de ChatGPT3.5 de la figura 4.4 es el pequeño valle entre los valores más bajos y los más altos justo en las valoraciones medias. Este comportamiento lo diferencia del resto, ya que en el resto de los casos suele tener un incremento desde el valor mínimo yendo en aumento, hasta alcanzar los valores más altos de similitud.

Una vez analizados todos los modelos con sus métricas de evaluación, se puede destacar que el que mejores resultados proporciona es el LLM de ChatGPT4. Esto se puede observar en la distribución de los valores de similitud y la similitud media en la tabla 4.4. A este le sigue ChatGPT3.5, con un valor bastante próximo en castellano y uno un poco más lejano en inglés.

5.2. Evaluación del Tiempo de Respuesta

En la aplicación que se plantea en este proyecto, el tiempo de respuesta es crucial para la usabilidad de los modelos conversacionales, especialmente cuando se consideran aplicaciones en tiempo real. Los resultados de los tiempos de respuesta medios de la tabla 4.5 mostraron diferencias significativas entre los modelos evaluados. ChatGPT4 y ChatGPT3.5 de OpenAI demostraron tiempos de respuesta mucho más rápidos en comparación con PoliGPT y Mistral. En español, ChatGPT4 tuvo un tiempo de respuesta promedio de 4.13 segundos y ChatGPT3.5 de 2.68 segundos, mientras que PoliGPT y Mistral tuvieron tiempos de 24.45 y 11.33 segundos respectivamente.

Estos tiempos de respuesta se vuelven aún más importantes al considerar que, en conversaciones prolongadas, el número de tokens procesados aumenta hasta llegar al límite de la memoria de K interacciones, lo que puede afectar a la latencia. Los gráficos de la figura 4.5 muestran cómo los tiempos de respuesta evolucionan con el aumento de la longitud de la conversación, haciendo evidente que los modelos de OpenAI mantienen tiempos de respuesta más consistentes. En el caso de ChatGPT3.5 los tiempos de respuesta son más rápidos y consistentes a lo largo del tiempo, teniendo siempre un rango de tiempos bueno, sin ningún tiempo anómalo. En cuanto al modelo GPT4 los tiempos siguen siendo muy buenos, sin embargo, a partir de la consulta número treinta el tiempo empieza a aumentar llegando a ser prácticamente del doble del tiempo inicial. En un principio se podría pensar que es culpa de la memoria que se va aumentando, no obstante, esto no puede ser el causante puesto que al tener un límite de K interacciones a almacenar, en este caso doce, debería verse afectado desde la consulta inicial hasta alcanzar la consulta número doce. Como hipótesis está en la posible sobrecarga de la API a la hora de realizar tantas consultas para el modelo GPT4o puesto que el incremento es en ambos idiomas y no es algo excepcional, ya que estos experimentos de tiempo de respuesta se realizaron dos veces para todos los

modelos poder ver si los resultados eran similares o por cualquier motivo se había producido una anomalía en los tiempos.

En otro lugar, se encuentran los modelos PoliGPT y Mistral, ambos presentan incrementos significativos en el tiempo de respuesta a medida que la conversación progresa. Respecto a Mistral, este aumento con el paso del tiempo para el modelo en inglés se debe a una serie de datos anómalos que producen un incremento muy acelerado de la línea de tendencia, ya que se pasa de unos tiempos de ocho segundos a más de ochenta segundos para responder a una pregunta. Para el modelo en castellano también existe un dato anómalo, pero al inicio, de ciento cuarenta segundos para responder, en cambio, con el paso del tiempo va reduciéndose muy poco el tiempo de respuesta. PoliGPT tiene unos tiempos que se mueven dentro de unos rangos constantes en castellano y esto se puede ver en la línea de tendencia que es una línea recta, sin embargo, el modelo en inglés va aumentando el tiempo con el paso de las cuestiones hasta mantener un rango consistente a partir de la decimosexta interacción.

Por lo que con todo esto, más la confirmación del análisis de significancia en la que se rechaza la hipótesis nula de los modelos, en la que ChatGPT3.5 y GPT4 no tienen diferencias entre sí, pero sí con el resto de los modelos. Nos permite corroborar que ambos son los mejores y que cualquiera de los dos modelos es totalmente válido para su utilización.

En conclusión, el modelo que proporciona un excelente tiempo de respuesta y que se mantiene constante, sin causar fatiga al usuario por la espera, es ChatGPT3.5, el modelo GPT4 se podría considerar también una muy buena opción puesto que su tiempo medio es inferior, pero al final es una diferencia de un segundo, siendo algo poco perceptible y sin ningún tipo de importancia como se ha demostrado con el análisis de significancia. Siendo ambos modelos totalmente válidos. Sin embargo, PoliGPT y Mistral quedan totalmente descartados en este aspecto puesto que quedan muy lejos. Hay que destacar que, al estar accediendo a la nube con estos modelos, pueden tener fallos de disponibilidad o de rendimiento haciendo que aumente su tiempo considerablemente. Y para terminar destacar que la diferencia entre las variaciones de los modelos OpenAPI con el resto pueden estar relacionadas con la arquitectura interna de los modelos y la eficiencia de su procesamiento.

5.3. Evaluación de la Naturalidad de las Respuestas

Antes de continuar con la discusión sobre la naturalidad de los modelos, es importante destacar que actualmente los modelos que ofrecen mejores resultados son GPT-4 en términos de similitud en castellano y ChatGPT-3.5 en cuanto a tiempos de respuesta. Dado que evaluar la naturalidad en la lengua materna ya presenta desafíos significativos, hacerlo en otra lengua resultaba aún más complejo y requiere un alto nivel de dominio de ese idioma. Por esta razón, el cuestionario sobre la naturalidad del modelo se llevó a cabo únicamente en castellano y contó con la participación de trece personas. Aunque no haber evaluado en inglés podría considerarse una limitación, hacerlo podría introducir muchos sesgos, lo que justificó la decisión de realizar la evaluación solo en castellano.

La naturalidad de las respuestas es una métrica crítica, especialmente en aplicaciones destinadas a interactuar con personas mayores, donde la comprensibilidad

y coherencia son esenciales. El cuestionario realizado en castellano evaluó la fluidez, comprensibilidad y coherencia de las respuestas generadas por cada uno de los modelos.

Estos resultados son cruciales ya que la naturalidad de las respuestas impacta directamente en la experiencia del usuario. Un modelo que genera respuestas más coherentes y comprensibles es más probable que sea adoptado y utilizado de manera efectiva en aplicaciones prácticas. Los resultados indican que ChatGPT4, ChatGPT3.5 y PoliGPT proporcionan respuestas más naturales en comparación con Mistral, donde este último es el peor. ChatGPT4 se consolida en esta métrica como el número uno, seguido de ChatGPT3.5 y PoliGPT ambos con una evaluación muy similar, siendo esta no diferenciable prácticamente.

5.4. Conclusiones Discusión

Para concluir con la discusión, una vez se han visto todos los resultados y analizado cuál es la mejor opción en cada evaluación, destacar que ChatGPT4 es el modelo más equilibrado y efectivo tanto en términos de naturalidad de las respuestas como en similitud. Sin embargo, ChatGPT3.5 proporciona un mejor equilibrio en los tiempos y la diferencia con el resto de las evaluaciones no es muy grande, por lo que la opción de ChatGPT3.5 sería interesante tenerla en cuenta. Por otra parte, el modelo ChatGPT4 ha sido escogido el mejor al nivel de métricas, pero tiene un gran inconveniente y es el coste de ejecución de cada llamada a la API, siendo diez veces superior de media que su modelo anterior ChatGPT3.5. Al tener unos resultados más o menos similares y existir una gran diferencia por el coste de uso, el modelo ChatGPT3.5 sería el más interesante de los 4 para implantarlo y en un futuro continuar desarrollando el sistema para ir perfeccionándolo.

Además, cabe mencionar que, como se discutió en secciones anteriores, ha sido sorprendente obtener mejores resultados con los modelos en castellano en comparación con inglés. Ya que lo lógico hubiera sido esperar mejores resultados en inglés, dado que los modelos fueron entrenados con un conjunto de datos mayor en ese idioma. Sin embargo, los resultados obtenidos contradicen esta hipótesis.

Capítulo 6

Conclusiones y trabajos futuros

Tras la realización del Trabajo de Fin de Máster del Máster Universitario en Ingeniería y Tecnología de Sistemas Software se puede concluir que se han completado los objetivos planteados al comienzo del mismo y se pueden extraer una serie de conclusiones del estudio realizado. Se han cumplido superado el desarrollo de una aplicación para todos los modelos de lenguaje, a excepción de Mistral al desarrollarse en Google Colab, con la que las personas de edad avanzada puedan ser capaces de interactuar con ella de manera sencilla y tengan la capacidad también de utilizar sistemas de audio si tiene problemas para leer. Así pues, cumpliendo así otro de los objetivos al buscar mejorar la calidad de vida de este grupo demográfico a través de un bot con el que puedan interactuar y conversar sobre su vida y sus experiencias. En tercer lugar, se han superado los desafíos técnicos y personales que podía tener al principio del proyecto, al no tener ningún tipo de conocimiento en los ámbitos tratados, permitiéndome así enriquecer mis conocimientos en nuevos sectores y tecnologías como la IA Generativa. Por último, se ha podido evaluar e identificar que LLMs eran mejor o peor en cada una de las métricas establecidas para este trabajo. Destacar que la idea previa de que los tiempos de respuestas en los modelos locales era inferior a los modelos en la nube, puesto que no había que acceder a un servidor externo, ha quedado descartada. Ya que, como se ha visto en los resultados, el modelo local se posiciona en tercer lugar respecto de los tiempos ya que, a pesar de tenerlo en local, depende también de la capacidad del hardware en el que se ejecute.

Como mejor modelo a nivel puramente estadístico, como se ha comentado en la sección anterior, el ganador era ChatGPT4, seguido de ChatGPT3.5, Llama3 (PoliGPT) y en último lugar Mistral7B. Si estos resultados se comparan con la tabla 3.1 de ELO Rating los modelos que se consideran tanto el mejor como el peor, mis resultados han coincidido. Sin embargo, los modelos ChatGPT3.5 y Llama3 se han intercambiado posiciones, pero con unos resultados en las métricas muy próximas.

6.1. Relación con las asignaturas

En esta sección se analizan las competencias y conocimientos obtenidos durante el Máster Universitario en Ingeniería y Tecnología de Sistemas Software. A continuación, se destacan las asignaturas que han tenido un mayor impacto en el desarrollo del proyecto:

- **Desarrollo de Interfaces Multimodales Avanzadas (DIM):** En el desarrollo de mi TFM, he utilizado los conocimientos adquiridos en esta asignatura para implementar interfaces que permiten la interacción natural entre el usuario y el sistema conversacional. Sobre todo, fue muy útil para tener una base en el ámbito de la voz y el audio.
- **Ingeniería del Lenguaje Natural (ILN):** Probablemente la más relacionada con el TFM, puesto que se han utilizado algunos conocimientos en paralelo con la asignatura, es decir, se empezó a trabajar con Python y con tecnologías NLP simultáneamente en la asignatura y el TFM. Como conocimientos a destacar serían la creación de sistemas y aplicaciones que pueden entender y procesar el lenguaje humano, así como el uso de herramientas y técnicas para la extracción de información, generación de texto, etc.
- **Extracción de Información Desde Redes Sociales (ISC):** Esta asignatura me permitió ampliar más los conocimientos vistos en ILN y desarrollar ciertas capacidades que se han visto plasmadas en el proyecto como el desarrollo de la web.

Además de los conocimientos del máster que he podido aplicar en mi TFM, destacar también la oportunidad de aplicar algunas de las competencias transversales vistas a lo largo del curso. Como son el caso de la responsabilidad y toma de decisiones, compromiso social o innovación y creatividad. Por lo que gracias a los conocimientos de las asignaturas y las competencias he sido capaz de llevar a cabo el desarrollo del proyecto.

6.2. Trabajos futuros

En cuanto a trabajos futuros para este proyecto, se sugieren los siguientes aspectos para continuar mejorando la asistencia a personas mayores con el uso de sistemas conversacionales avanzados:

1. Ampliación de la Muestra:

- Ampliar el tamaño de las muestras para todos los tipos de evaluación, involucrando a más participantes y obtener una mayor cantidad de textos generados por los LLMs para poder tener una cantidad de datos mayor, más representativos y reducir el margen de error.

2. Optimización de Modelos:

- Investigar y desarrollar optimizaciones específicas para mejorar los tiempos de respuesta y la eficacia en distintos idiomas, especialmente para modelos en local como Mistral que tiene una mayor flexibilidad a la hora de modificar sus parámetros y ajustes.

3. Integración de Nuevas Tecnologías:

- Explorar la integración de tecnologías de generación de voz más avanzadas y que tengan una mayor flexibilidad con las emociones del texto, así como interfaces de usuario mejoradas para facilitar la interacción con los modelos por parte de las personas mayores.

- Desarrollar métodos avanzados de personalización de modelos que permitan a los sistemas conversacionales aprender y adaptarse a las preferencias y necesidades individuales de los usuarios, como pueda ser ir modificando el estilo de las respuestas del modelo según la actitud que vaya teniendo el usuario.
- Implementar un servidor para almacenar la información en la base de datos sin necesidad de tener que guardarlo en local, enviando la información a través de la API y evitando la sobrecarga de la aplicación.

4. Estudio de Alucinaciones:

- Continuar investigando y realizando estudios sobre las alucinaciones generadas por los modelos para identificar posibles patrones y desarrollar estrategias de mitigación.

5. Aplicación móvil:

- Desarrollar una aplicación para dispositivos móviles o tablets, permitiendo así que los usuarios puedan tener un acceso más sencillo e intuitivo.

6. Aplicaciones Prácticas:

- Probar el proyecto en un entorno real con personas mayores, evaluando la interacción con el sistema y evaluar si esto genera un impacto positivo en su calidad de vida, ajustando las soluciones según vayan proporcionando feedback.

7. Colaboración Interdisciplinaria:

- Fomentar la colaboración entre especialistas en inteligencia artificial, gerontología, psicología y diseño de interfaces para desarrollar soluciones más integrales y holísticas que aborden las diversas necesidades de las personas mayores.

Bibliografía

- [1] Rameel Ahmad. «Enhancing Multilingual Information Retrieval in Mixed Human Resources Environments: A RAG Model Implementation for Multicultural Enterprise». En: *arXiv:2401.01511v1 [cs.IR]* (2024).
- [2] Ishitva Awasthi et al. «Natural Language Processing (NLP) based Text Summarization - A Survey». En: *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. 2021, págs. 1310-1317. DOI: 10.1109/ICICT50816.2021.9358703.
- [3] Satanjeev Banerjee y Alon Lavie. «METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments». En: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, jun. de 2005, págs. 65-72. URL: <https://www.aclweb.org/anthology/W05-0909>.
- [4] P. Bhattacharyya. «Natural Language Processing: A Perspective from Computation in Presence of Ambiguity, Resource Constraint and Multilinguality». En: *Proceedings of the ACM*. ACM, 2012, págs. 312-320.
- [5] X. Cai y W. Li. «Ranking Through Clustering: An Integrated Approach to Multi-Document Summarization». En: *IEEE Transactions on Audio, Speech, and Language Processing* 21.7 (2013), págs. 1424-1433.
- [6] Franz Louis Cesista et al. «Retrieval Augmented Structured Generation: Business Document Information Extraction As Tool Use». En: (2024).
- [7] Wei-Lin Chiang et al. *Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference*. 2024. arXiv: 2403.04132 [cs.AI].
- [8] Nathan Corecco et al. «An LLM-based Recommender System Environment». En: *arXiv:2406.01631v1 [cs.IR]* (2024).
- [9] DATAtab Team. *ANOVA de dos factores o vías (sin medidas repetidas)*. DATAtab e.U. Graz, Austria. 2024. URL: <https://datatab.es/tutorial/two-factorial-anova-without-repeated-measures>.
- [10] Guy Dewsbury et al. «Designing technology with older people». En: *Universal Access in the Information Society* 6 (2007), págs. 207-217. DOI: 10.1007/s10209-007-0079-7.
- [11] Zijian Ding et al. «TalkTive: A Conversational Agent Using Backchannels to Engage Older Adults in Neurocognitive Disorders Screening». En: *CHI '22: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022, págs. 1-15. ISBN: 978-1-4503-9157-3. DOI: 10.1145/3491102.3502005.

- [12] Lucas Douma et al. «Older adults' acceptance of technology-based mental health interventions: A systematic review». En: *International Journal of Geriatric Psychiatry* 32.12 (2017), págs. 1315-1330. DOI: 10.1002/gps.4613.
- [13] Stefan Feuerriegel, Jens Hartmann, Christian Janiesch et al. «Generative AI». En: *Business & Information Systems Engineering* 66 (feb. de 2024), págs. 111-126. DOI: 10.1007/s12599-023-00834-7.
- [14] Dhir Gala y Amgad N. Makaryus. «The Utility of Language Models in Cardiology: A Narrative Review of the Benefits and Concerns of ChatGPT-4». En: (2024).
- [15] Gihan Gamage y Nishan Mills. «Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems». En: *2024 IEEE International Conference on Industrial Technology (ICIT)*. 2024.
- [16] Kaiyuan Gao et al. *Examining User-Friendly and Open-Sourced Large GPT Models: A Survey on Language, Multimodal, and Scientific GPT Models*. Ago. de 2023.
- [17] Yunfan Gao et al. «Retrieval-Augmented Generation for Large Language Models: A Survey». En: (2024).
- [18] Ioannis Giachos et al. «Inquiring Natural Language Processing Capabilities on Robotic Systems through Virtual Assistants: A Systemic Approach». En: *Journal of Computer Science Research* 5.2 (abr. de 2023), págs. 28-36. DOI: 10.30564/jcsr.v5i2.5537. URL: <https://journals.bilpubgroup.com/index.php/jcsr/article/view/5537>.
- [19] Prompting Guide. *Mistral 7B*. Accesado: 28-06-2024. 2024. URL: <https://www.promptingguide.ai/models/mistral-7b>.
- [20] Jeremy Howard y Sebastian Ruder. «Universal Language Model Fine-tuning for Text Classification». En: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. 2018, págs. 328-339.
- [21] Albert Q. Jiang et al. «Mistral 7B». En: *arXiv:2310.06825v1 [cs.CL]* (2023).
- [22] Pratibodh - Journal Editor et al. «Natural Language Processing: A Human Computer Interaction Perspective». En: *PRATIBODH NCDSNS* (feb. de 2024). URL: <https://pratibodh.org/index.php/pratibodh/article/view/150>.
- [23] Bran Knowles y Vicki L. Hanson. «Older Adults' Deployment of 'Distrust'». En: *ACM Trans. Comput.-Hum.* (2018).
- [24] UKP Lab. *Pretrained Models - Multilingual Models*. Accessed: 2024-06-29. 2024. URL: https://www.sbert.net/docs/sentence_transformer/pretrained_models.html#multilingual-models.
- [25] Patrick S. H. Lewis et al. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». En: *CoRR abs/2005.11401* (2020). arXiv: 2005.11401. URL: <https://arxiv.org/abs/2005.11401>.
- [26] Chin-Yew Lin. «ROUGE: A Package for Automatic Evaluation of Summaries». En: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, jul. de 2004, págs. 74-81. URL: <https://www.aclweb.org/anthology/W04-1013>.
- [27] Management Solutions. *El auge de los large language models: de los fundamentos a la aplicación*. <https://www.managementsolutions.com/sites/default/files/minisite/static/72b0015f-39c9-4a52-ba63-872c115bfbfd0/llm/pdf/auge-de-los-llm-08.pdf>. España: Management Solutions, 2024.
- [28] Jing Miao et al. «Integrating Retrieval-Augmented Generation with Large Language Models in Nephrology: Advancing Practical Applications». En: (2024).

- [29] Kishore Papineni et al. «BLEU: a Method for Automatic Evaluation of Machine Translation». En: 2002, págs. 311-318.
- [30] Aravind Sasidharan Pillai. «Advancements in Natural Language Processing for Automotive Virtual Assistants Enhancing User Experience and Safety». En: *Journal of Computational Intelligence and Robotics* 3.1 (2023), págs. 27-36.
- [31] G. Rodeschini. «Gerotechnology: A new kind of care for aging? An analysis of the relationship between older people and technology». En: *Nursing & Health Sciences* 13.4 (2011), págs. 521-528. DOI: 10.1111/j.1442-2018.2011.00634.x.
- [32] Chat Room. «Llama (LLM)». En: *Algorithms* 5.12 (2024), pág. 40.
- [33] Konstantinos I. Roumeliotis y Nikolaos D. Tselikas. «ChatGPT and Open-AI Models: A Preliminary Review». En: (2024).
- [34] Serbatic. *Small y Large Language Models: ¿Dónde está el futuro de la IA?* [Accesed: 2024-05-05]. 2024. URL: <https://www.serbatic.es/insights/small-y-large-language-models-donde-esta-el-futuro-de-la-ia/>.
- [35] M. Tan et al. «A Scalable Distributed Syntactic, Semantic, and Lexical Language Model». En: *Proceedings of the ACM*. ACM, 2012, págs. 631-662.
- [36] Pinecone Team. *Langchain Conversational Memory*. <https://www.pinecone.io/learn/series/langchain/langchain-conversational-memory/>. Consultado el 29 de junio de 2024.
- [37] Oguzhan Topsakal y Tahir Cetin Akinici. «Creating large language model applications utilizing langchain: A primer on developing llm apps fast». En: *International Conference on Applied Engineering and Natural Sciences*. Vol. 1. 1. 2023, págs. 1050-1056.
- [38] Ashish Vaswani et al. «Attention is All you Need». En: *Advances in Neural Information Processing Systems*. Ed. por I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [39] Salifu Yusif, Jeffrey Soar y Abdul Hafeez-Baig. «Older people, assistive technologies, and the barriers to adoption: A systematic review». En: *International Journal of Medical Informatics* 94 (2016), págs. 112-116. DOI: 10.1016/j.ijmedinf.2016.07.004.

Anexo I

Esta sección describe el acceso al modelo localmente.

```
#####
# Tokenizer
#####

model_name='mistralai/Mistral-7B-Instruct-v0.2'

tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token
tokenizer.padding_side = "right"

#####
# bitsandbytes parameters
#####

# Activate 4-bit precision base model loading
use_4bit = True

# Compute dtype for 4-bit base models
bnb_4bit_compute_dtype = "float16"

# Quantization type (fp4 or nf4)
bnb_4bit_quant_type = "nf4"

# Activate nested quantization for 4-bit base models (double quantization)
use_nested_quant = False

#####
# Set up quantization config
#####
compute_dtype = getattr(torch, bnb_4bit_compute_dtype)

bnb_config = BitsAndBytesConfig(
    load_in_4bit=use_4bit,
    bnb_4bit_quant_type=bnb_4bit_quant_type,
    bnb_4bit_compute_dtype=compute_dtype,
    bnb_4bit_use_double_quant=use_nested_quant,
)

# Check GPU compatibility with bfloat16
if compute_dtype == torch.float16 and use_4bit:
    major, _ = torch.cuda.get_device_capability()
    if major >= 8:
        print("-" * 80)
        print("Your GPU supports bfloat16: accelerate training with bf16=True")
        print("-" * 80)

#####
# Load pre-trained config
#####
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,
)
```

```
standalone_pipe = pipeline(  
    "text-generation",  
    model=model,  
    tokenizer=tokenizer,  
    max_new_tokens=1000,  
    do_sample=False,  
    temperature=0.0,  
    top_p=0.95,  
    top_k=40,  
    repetition_penalty=1.1  
)  
  
response_pipe = pipeline(  
    "text-generation",  
    model=model,  
    tokenizer=tokenizer,  
    max_new_tokens=2000,  
    do_sample=False,  
    temperature=0.2,  
    top_p=0.95,  
    top_k=40,  
    repetition_penalty=1.1  
)
```

```
mistral_standalone_llm = HuggingFacePipeline(pipeline=standalone_pipe)  
mistral_response_llm = HuggingFacePipeline(pipeline=response_pipe)
```

```
# Create prompt from prompt template  
prompt = PromptTemplate(  
    input_variables=["chat_history", "question"],  
    template=PROMPT_TEMPLATE,  
)  
  
# Create llm chain  
llm_chain = LLMChain(llm=mistral_standalone_llm, prompt=prompt)
```

Anexo II

Esta sección describe el acceso al modelo

```
st.session_state['client'] = openai.OpenAI(
    base_url='https://poligpt.upv.es/api/RAG_Pro/ollama',
    api_key='GZFWxZNSDzNFPuG3DWhI4vZU3weRIwwD2utVCEmSC5IRpmmGuE1qVkuU0Ewfd1K9',
)

chat_completion = st.session_state['client'].chat.completions.create(
    messages=messages,
    model='poligpt'
)

sentence = chat_completion.choices[0].message.content
```


Anexo III

Documento para la recopilación de información para personalizar modelo conversacional para personas mayores

1. Información básica:

- Nombre completo
- Edad
- Fecha de nacimiento
- Género
- Lugar de residencia
- Lugar de nacimiento
- Estado civil
- Número de hijos/nietos (si aplica)

2. Antecedentes familiares:

- ¿Qué papel juegan tus familiares en tu vida?
- ¿Tienes alguna anécdota especial sobre tu familia que te gustaría compartir?

3. Intereses y pasatiempos:

- ¿Qué te gusta hacer en tu tiempo libre?
- ¿Tienes algún hobby o pasatiempo favorito?
- ¿Cuál ha sido tu viaje o experiencia más memorable?
- ¿Qué viaje te gustaría hacer en el futuro?

4. Historia laboral y educativa:

- ¿Cuál ha sido tu profesión?
- En caso de haberte dedicado a ser ama de casa, ¿qué ha sido lo mejor de esa labor?
- ¿Cuál ha sido tu trabajo más significativo?
- ¿Qué te gustaba de ese trabajo?

- ¿Tienes algún recuerdo o historia interesante de tu época estudiantil o laboral que te gustaría compartir?
- ¿Qué estudiaste?

5. Experiencias de vida:

- ¿Hay algún evento importante en tu vida que haya tenido un impacto significativo en ti?
- ¿Qué experiencias de tu vida han sido las más importantes para ti?
- ¿Cuáles crees que han sido tus mayores logros?

6. Gustos y preferencias:

- ¿Cuál es tu comida favorita?
- ¿Qué tipo de películas te gustan?
- Si tienes alguna película favorita dímela.
- ¿Puedes decirme un libro que te haya gustado?
- Dime una canción que te guste
- ¿Cuál es tu lugar del mundo favorito?
- ¿Tocas algún instrumento musical?

7. Salud y bienestar:

- ¿Tienes alguna preocupación de salud que desees compartir?
- ¿Prácticas ejercicio de manera habitual?
- ¿Qué deporte te gusta más?
- ¿Cómo manejas el estrés o las preocupaciones?
- ¿Cómo cuidas tu bienestar emocional?
- ¿Haces actividades con otras personas habitualmente, como ir al cine, teatro, excursiones, viajes, etc.?
- ¿Consideras que llevas una dieta saludable?
- ¿Consideras que duermes bien habitualmente?
- ¿Haces algún tipo de actividad que mantenga estimulada tu mente de manera habitual, como puzles, crucigramas, lectura, talleres, etc.)?

8. Aspiraciones y metas:

- ¿Qué te gustaría lograr en el futuro?
- ¿Hay algo que aún no hayas hecho pero que siempre hayas querido hacer?

9. Recuerdos y nostalgia:

- ¿Cuál es tu recuerdo más preciado?
- ¿Tienes alguna historia o anécdota de tu infancia que te gustaría compartir?

10. Opiniones y valores:

- ¿Hay algún tema actual que te preocupe especialmente?
- ¿Qué opinas sobre ese tema?
- ¿Cuáles son los valores más importantes para ti?
- ¿Cuáles de esos valores crees que posees?

11. Autoestima y percepción de sí mismo/a:

- ¿Qué adjetivos positivos crees que te definen?
- ¿Qué cosas se te da bien hacer?