ACCESIBILIDAD MUSEÍSTICA MEDIANTE INTELIGENCIA ARTIFICIAL: EL MUSEO DE AMÉRICA

MUSEUM ACCESSIBILITY THROUGH ARTIFICIAL INTELLIGENCE: MUSEUM OF AMERICA

TRABAJO FIN DE GRADO
CURSO 2023-2024

MARTA GAGO MACÍAS
AUTOR

DIRECTOR
RUBÉN FUENTES-FERNÁNDEZ

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

ACCESIBILIDAD MUSEÍSTICA MEDIANTE INTELIGENCIA ARTIFICIAL: EL MUSEO DE AMÉRICA

MUSEUM ACCESSIBILITY THROUGH ARTIFICIAL INTELLIGENCE: MUSEUM OF AMERICA

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

MARTA GAGO MACÍAS
AUTOR

DIRECTOR
RUBÉN FUENTES-FERNÁNDEZ

CONVOCATORIA: JUNIO 2024

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

27 DE MAYO DE 2024

RESUMEN

Accesibilidad museística mediante Inteligencia Artificial: El Museo de América.

Los museos se enfrentan en la actualidad a importantes retos y cambios en sus labores de conservación, diseminación e investigación del patrimonio que atesoran. Uno de los principales es cómo adaptar la comunicación relativa a ese patrimonio a públicos que demandan y necesitan experiencias cada vez más particularizadas a sus características y gustos particulares.

Los recientes avances en las Inteligencias Artificiales Generativas (IAG) con sus capacidades para producir a partir de la información de entrada nuevos contenidos, abren un nuevo enfoque para esta adaptación. Este trabajo explora esta aproximación en el marco de una colaboración con el Museo de América de Madrid (España) para el rediseño de varias salas temáticas.

En concreto, este proyecto consiste en crear una aplicación donde el avatar que representa a un personaje histórico interactúa con los usuarios empleando la "personalidad" de dicho personaje (ej. se cree ser el personaje y usa su tipo de lenguaje). Para ello se emplea un Modelo Grande de Lenguaje Grande (LLM por sus siglas en inglés) adaptado a dicho personaje.

Esta aplicación constituye una prueba de concepto del uso de la Inteligencia Artificial para introducir nuevos tipos de exposiciones en los museos que los vuelvan más atractivos y accesibles a su potencial público general actual.

Código disponible: https://github.com/martagago48/tfg.git

Palabras clave

Museo, Exposición, Historia, Accesibilidad, Inteligencia Artificial Generativa, Modelo Grande de Lenguaje, Aplicación

ABSTRACT

Museum Accessibility through artificial intelligence: Museum of America

Museums today face important challenges and changes as researchers and preservationists of the heritage they are responsible for. One of the main challenges is how to adapt the communication relative to that heritage to a public that demands more experiences catered to their personal wants and needs as time goes by.

The recent developments in the field of Generative Artificial Intelligence (GAI), with their capability of creating information have opened a new door for these adaptations. This project seeks to explore this new way approximation in collaboration with the Museum of America in Madrid (Spain).

This project consists of creating an application where an avatar, which represent a historic figure can interact with visitors taking on the personality of said figure (ex. taking on their way of speaking). To personify this character, we will make use of a Large Language Model (LLM).

This application constitutes a conceptual test of the use of Artificial Intelligence to introduce new types of exhibitions in museums that can increase the interest of potential visitors.

Keywords

Museum, Exposition, History, Accessibility, Generative Artificial Intelligence, Large Language Model, Application.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

Capítulo 1 - Introducción

1.1 Motivación

Los museos son una institución cultural, educativa y de investigación que trabajan en torno al patrimonio que preservan. Llevan siglos siendo una herramienta clave de conocimiento, educación y difusión de la cultura en nuestra sociedad .

A pesar de su larga historia e importancia, hoy en día se enfrentan a importantes retos en relación con su público. Por un lado, existen museos convertidos en grandes iconos turísticos (ej. los museos del Prado, el Louvre o el Británico), donde los visitantes, en muchos casos, han perdido parte del interés cultural reemplazado por uno meramente de visita a una gran atracción del lugar. Por otro lado, el COVID-19 causó que gran parte de los museos se viesen obligados cerrar sus puertas al público, mermando sus recursos económicos y poniendo en riesgo su supervivencia (Ibermuseos, 2020).

Los públicos, especialmente los más jóvenes, están cada vez más alejados de clásicas experiencias contemplativas de las piezas de los museos y la lectura de la información asociada, y buscan por el contrario experiencias más interactivas e inmersivas. Además, las nuevas tecnologías y la facilidad para encontrar información por Internet pueden desembocar en que cada vez menos gente elija visitar museos solo para ver en el mundo real información disponible en línea (Morales Carmona & Freitag, 2014).

Conscientes de esta situación, las instituciones y sus expertos están en una continua búsqueda de la transformación de los museos y sus actividades para tener mejores interacciones con su entorno. Las Tecnologías de la Información (TI) juegan un papel clave en estos esfuerzos, por ejemplo, con iniciativas como los sitios web, las redes sociales, las audioguías, las video-instalaciones o las experiencias inmersivas con realidad virtual y aumentada (Boiano, Cuomo, & Gaia, 2016). Aunque éstas tienen un gran éxito cuando se realizan apropiadamente, los museos tienen que lidiar con la

dificultad para aportar novedad e integrar y reutilizar las diferentes intervenciones, además de su elevado coste respecto a sus limitados presupuestos.

Este proyecto busca contribuir a esta línea mediante la creación de experiencias museísticas más accesibles, innovadoras e interesantes para los visitantes, con un desarrollo efectivo en recursos para las instituciones, mediante el uso de nuevas tecnologías, en particular de la Inteligencia Artificial (IA) Generativa (IAG). En nuestro caso, construiremos una aplicación que pueda servir como una instalación museística con la que el usuario pueda interactuar de forma directa.

1.2 Objetivos

La funcionalidad principal de nuestra aplicación es un chat donde el usuario se comunica con la IA en una conversación donde la IA está interpretando a un personaje histórico. La IA se ajusta a este contexto mediante instrucciones del propio desarrollador, técnicas de adaptación y documentos que se le proporcionan.

Dentro del desarrollo de la aplicación podemos ver los objetivos de manera más específica:

- Integrar un modelo de IAG en la aplicación como base, en concreto un Modelo Grande de Lenguaje (LLM, *Large Language Model*).
- Implementar una Interfaz de Usuario (UI por sus siglas en inglés) para la aplicación.
- Uso de distintas técnicas de adaptación de IA, como la Generación Aumentada por Recuperación (RAG por sus siglas en inglés), para mejorar el desempeño de nuestro LLM.
- Desarrollar la caracterización de la IA como personaje histórico.
- Implementar un historial del chat para el mantener un registro de cada conversación de la aplicación.
- Implementar funcionalidades para la aplicación como reconocimiento y síntesis de voz para permitir comunicarse con la IA mediante la voz y así mejorar la accesibilidad de la aplicación.

1.3 Plan de trabajo

Para la consecución de los objetivos establecidos, se plantea un plan de trabajo que consiste en varias etapas sucesivas:

1. **Investigación.**

Se comienza con una fase de investigación sobre los LLM, recursos disponibles para usarlos, el estado actual de esta tecnología y su evolución próxima, requisitos de infraestructura y otros aspectos de su aplicación.

2. **Elección de la arquitectura y establecimiento de la infraestructura.**

Una vez se haya concluido la investigación, pasaremos a elegir la arquitectura que vamos a usar en el proyecto y también estableceremos la infraestructura (ej. librerías, frameworks y herramientas de desarrollo).

3. **Primera fase del desarrollo: Aplicación inicial.**

Comienza el desarrollo de la aplicación. Se crea un primer prototipo partiendo del código fuente disponible. El alcance de este prototipo será poder disponer de una aplicación con una interfaz simple donde el usuario pueda comunicarse con el LLM y tengamos un historial de la conversación que se muestre en pantalla.

4. **Segunda fase de desarrollo: Sistema de Inteligencia Artificial.**

Se implementa el sistema de Inteligencia Artificial por completo: Se eligen los distintos modelos que se van a usar, se realiza la caracterización de la IA al personaje histórico a través de técnicas como por ejemplo la RAG.

4. **Tercera fase de desarrollo: Interfaz de Usuario.**

Se refina la aplicación, mejorando la UI mediante la incorporación y modificación de elementos de UI, incluyendo botones, menús desplegables y áreas de texto. También se implementa el reconocimiento de voz para las peticiones de usuario y la síntesis de voz para las respuestas del sistema.

5. Cierre del desarrollo.

Se revisa que todo lo implementado funcione sin problema, se solucionan bugs, se comprueba el rendimiento, la calidad de las respuestas y más. Se crea un modo administrador/usuario que restringe ciertas opciones de la aplicación dependiendo del nivel de acceso del usuario.

Capítulo 2 - Estado de la cuestión

En este capítulo vamos a hablar de los museos, su importancia como institución educativa y su evolución, y como las nuevas tecnologías llevan ya años usándose en museos para innovarlos y volverlos más accesibles. Después vamos a pasar a hablar de la Inteligencia Artificial, una de estas nuevas tecnologías desde la perspectiva de los museos, que se encuentra en una fase de expansión gigantesca en cuanto a su desarrollo y su alcance. Por último, vamos a juntar todas estas ideas para explorar las alternativas para la creación de una instalación museística desarrollada con IA y como puede ser un activo positivo para un museo.

2.1 Los museos, su evolución y su función

Los museos son una institución fundamental para la educación, la investigación y la preservación de la cultura en nuestras sociedades, y llevan cumpliendo esta función durante siglos. A lo largo de esos siglos, los museos y las sociedades que los albergan han establecido un diálogo donde los cambios en uno afectaban al otro y disparaban ciertas adaptaciones en su relación con las manifestaciones culturales. Recientemente, la globalización y las nuevas tecnologías, entre otros, han creado nuevas vías de circulación de bienes y servicios culturales, que exigen instituciones renovadas para responder a estas nuevas dinámicas culturales (Fiallos, 2023).

El concepto de museo a lo largo de la historia ha evolucionado así en el tiempo, pasando por una constante adaptación conceptual. Esta institución se ha reinventado continuamente para atraer, educar, establecer vínculos y satisfacer las necesidades de los públicos cada vez más exigentes (Fiallos, 2023).

En el año 2022 el Consejo Internacional de Museos (ICOM) define un museo como:

"Una institución sin ánimo de lucro, permanente y al servicio de la sociedad, que investiga, colecciona, conserva, interpreta y exhibe el patrimonio material e inmaterial. Abiertos al público, accesibles e inclusivos, los museos fomentan la diversidad y la sostenibilidad. Con la participación de las comunidades, los museos operan y

comunican ética y profesionalmente, ofreciendo experiencias variadas para la educación, el disfrute, la reflexión y el intercambio de conocimientos." (ICOM, 2022)

Su función no se limita por tanto a la mera exposición de objetos y obras o su investigación. También debe mediar el conocimiento que se construye en la interacción entre el espacio de exposición, las obras y objetos expuestos, y su público. (Morales Carmona & Freitag, 2014)

2.2 Los museos: problemas que afrontan

Por más que se resalte su importancia, es imposible ignorar los diversos retos y dificultades a los que se enfrenta esta institución hoy en día. Su contexto social se ha modificado notablemente y ello afecta a su desempeño desde varios ángulos, de los cuales aquí nos centramos en los que afectan a su relación con el público.

En relación con sus visitantes, los museos de mayor renombre han experimentado un proceso de turistificación. Si bien éste resulta positivo desde el punto de vista de algunos aspectos como el impacto como marca o icono cultural, su alcance en cuanto a diseminación y el económico, también contribuye a la masificación del espacio y a la pérdida del interés cultural y educativo por parte de los visitantes.

Otro problema es el declive en el interés por los museos del público general. Hay diversas razones responsables de este declive, entre ellas la expansión de Internet y el acceso a información de manera instantánea, y el desinterés por el modelo tradicional contemplativo de exposición, sobre todo entre el público más joven.

Todo esto se ha visto exacerbado por la mayor crisis sanitaria que se ha desatado a escala mundial en los últimos tiempos, el COVID-19. Debido a la naturaleza del virus y su método de propagación, se establecieron una serie de medidas para limitar el contagio, que incluyeron el cierre de los museos y la cancelación de todas las actividades de este sector.

Dos informes de la UNESCO y del ICOM confirman que los museos se vieron especialmente afectados por la pandemia. Casi el 90% de los museos, es decir, más de 85.000 instituciones, cerraron sus puertas durante distintos períodos de tiempo durante la

crisis sanitaria, y se estima que casi el 13% de los museos de todo el mundo puede que nunca vuelvan a abrir sus puertas (UNESCO, 2020).

Así, es más importante ahora que nunca que los museos hagan uso de nuevas tecnologías para implementar herramientas, actividades y exposiciones que puedan responder a las nuevas dinámicas culturales, que expandan su capacidad educativa, y que puedan ayudar a recuperarse del declive sufrido atrayendo a visitantes que posean interés cultural y educativo.

Esta reflexión no es novedosa, y desde comienzos del siglo XXI se ha observado como los museos han hecho uso de las nuevas tecnologías para crear nuevas experiencias. Algunos ejemplos son aplicaciones que permiten al usuario participar en una búsqueda del tesoro o en la resolución de enigmas (ej. (Cabrera, y otros, 2005) (Dini, Paternò, & Santoro, 2007)) o juegos cooperativos(ej. (Bonnat, Oliveira, & Sanchez, 2020)).

2.3 Inteligencia Artificial

A continuación, vamos a centrarnos en la Inteligencia Artificial, una disciplina de la computación que está en auge en esta nueva década.

Dentro de los muchos tipos de IA que existen, las que mayor crecimiento, atención e impacto han tenido en estos últimos años han sido probablemente las IA generativas. La Inteligencia Artificial Generativa (IAG) es un tipo de tecnología que crea contenido nuevo a partir de los modelos de aprendizaje profundo que están entrenados con grandes conjuntos grandes de datos. En la actualidad, las aplicaciones con esta tecnología se utilizan para generar textos, imágenes, código y mucho más (RedHat, 2024). Nos vamos a centrar concretamente en los LLM, modelos de lenguaje de aprendizaje profundo y propósito general, entrenados con una gran cantidad de datos (normalmente miles de documentos).

Aunque la IA es una tecnología revolucionaria y el pilar central de nuestro proyecto, no se deben ignorar los potenciales riesgos y problemas que derivan de su uso, como la opacidad en los procesos de toma de decisiones, el sesgo y las discriminaciones de todo tipo. De la misma manera, los artefactos que hacen uso de IA,

a medida que avanza su autonomía, podrían tener comportamientos impredecibles o potencialmente dañinos (Parra Sepúlveda & Concha Machuca, 2021).

En el contexto de nuestro proyecto, los problemas que más nos preocupan son precisamente la opacidad de su funcionamiento interno, que hace que sea mucho más difícil para nosotros poder adaptar y modificar un modelo para que cumpla su tarea específica (Mittal, 2023), y la posibilidad de que el modelo genere información incorrecta o mensajes inapropiados.

Creemos que es positivo usar esta tecnología con fines educativos, siempre que se haga de manera ética y considerando los potenciales riesgos y problemas, y que se debe advocar por una IA transparente, imparcial y responsable para poder afrontar dichas preocupaciones (Cath, 2018).

Podemos encontrar estudios sobre el beneficio del uso de la IA como una herramienta educativa (ej. (Taylor, Yeung, & Bashet, 2021)), y específicamente en el contexto museístico: (Boiano, Borda, Gaia, Rossi, & Cuomo, 2018), (Varitimiadis, y otros, 2020). También tenemos ejemplos concretos de la implementación de herramientas que hacen uso de IA en museos como Alfa, un chatbot creado para el Museo de Ciencia de Milán (Boiano, Gaia, & Caldirini, Make Your Museum Talk: Natural Language Interfaces For Cultural Institutions, 2003).

Capítulo 3 - Planteamiento de nuestro proyecto

En este capítulo se va a hablar del proyecto que se ha propuesto desde el Museo de América, y como hemos afrontado y planteado la posible solución.

3.1 Propuesta

Al comienzo de este proyecto, nos pusimos en contacto con el Museo de América. Ellos nos propusieron incorporar una aplicación basada en IA donde los usuarios del museo pudieran mantener conversaciones con un personaje histórico en una exposición del museo. La exposición que tenían en mente era sobre Francisco de Arobe, y por lo tanto este sería el personaje que caracterizar mediante nuestro modelo.

Figura 3-1. Los mulatos de Esmeraldas (1599), Andrés Sánchez Gallque

Para entender quien fue Francisco de Arobe y su importancia, es necesario hablar primero del contexto histórico de la conquista de América.

En 1532, Francisco Pizarro desembarca en algún punto de la costa norte de Ecuador para luego dirigirse al sur, hacia el corazón de Perú. Tras la caída del Imperio inca, esta zona fronteriza empezó a despoblarse debido a enfermedades y traslados

forzosos. Fue bautizada con el nombre de Esmeraldas, y se convirtió en un lugar casi maldito, en cuya costa naufragaban decenas de barcos, trayendo a la costa piratas, desertores y esclavos fugados que se instalaban en la selva y aterrorizaban a las poblaciones cercanas, tanto indias como españolas.

Entre los muchos barcos que naufragaban en Esmeralda, en uno de ellos se encontraba Andrés Mangache, un africano capturado para seguramente acabar siendo vendido como esclavo. Tras el naufragio, por suerte, consiguió escapar junto a su amada hacia la espesura. No tardaron mucho en rodearse de seguidores, plantar cara a los indios y mezclarse con otros naturales, como los de la "tierra de Dobe" que los acogieron y del cual sus descendientes seguramente tomaron el apellido "Arobe".

Los Mangache tuvieron dos hijos, Juan y nuestro don Francisco, que acabó ocupando el puesto de cacique tras la muerte de los dos anteriores.

Aunque nunca se mostraron completamente amistosos con las autoridades quiteñas, los Arobe se ganaron la fama de ser más colaboradores y cercanos que otros cacicazgos, llegando incluso a convertirse al cristianismo.

Se cree que, debido a esta buena disposición por parte de los Arobe, se decidió abordar los problemas de la perpetua resistencia esmeraldeña mediante la negociación. Así, se arregló un encuentro entre los Arobe y el oidor Del Barrio en la Real Audiencia de Quito.

Don Francisco fue subsecuentemente nombrado Gobernador de Esmeraldas a cambio de obediencia a la corona y protección de sus costas. Para conmemorar la ocasión, el rey Felipe III recibió un lienzo de don Francisco y sus dos hijos como regalo. Este cuadro (ver Figura 3-1) se encuentra actualmente en el Museo de América y es la pieza más importante de esta exposición (Rodríguez, 2021).

3.2 Requisitos del sistema

En base a esta propuesta, podemos hacer un primer planteamiento de los requisitos de nuestra aplicación.

La idea principal es crear una instalación en el Museo de América sobre Francisco de Arobe donde los usuarios dispondrán de una aplicación a través de la cual podrán comunicarse con don Francisco.

Usaremos un modelo LLM como base y aplicaremos varias técnicas adaptativas para lograr la caracterización de Francisco de Arobe. Posiblemente será necesario el uso de varios tipos de modelos de IA adicionales para las técnicas adaptativas. Llamaremos a este conjunto nuestro Sistema de Inteligencia Artificial.

Con el objetivo de que los usuarios puedan mantener una conversación fluida se implementará la posibilidad de hacer una consulta y recibir una respuesta tanto de manera escrita como usando la voz: en el caso de hacer una consulta usaremos el reconocimiento de voz mientras que para la respuesta se usará la síntesis de voz.

Los museos acogen a público de todo tipo, así que necesitaremos una interfaz de usuario simple e intuitiva que responda a las necesidades de todos los visitantes.

Todos estos requisitos se recogerán e implementarán mediante una aplicación.

Capítulo 4 - Infraestructura

Antes de comenzar a hablar sobre la arquitectura y la propia aplicación, conviene explicar qué herramientas se han usado para realizar este trabajo.

En primer lugar, el lenguaje de programación **Python** (Van Rossum & Drake, 2009) debido a la gran cantidad de recursos que hay para trabajar con IA en él, además de su facilidad de uso, flexibilidad y cantidad de librerías disponibles.

Junto con Python se ha usado **Anaconda** (Anaconda, 2020), una distribución específica de Python que proporciona la creación y uso de entornos. Los entornos facilitan el uso de versiones específicas de librerías para evitar problemas de dependencias.

Para llevar un control de versiones del código se ha utilizado **GitHub** (github, 2020), que se encarga de rastrear los cambios en el código fuente del proyecto a lo largo del tiempo. También sirve para tener una copia del proyecto en otra fuente que no sea local y permitir el trabajo en distintos computadores.

El editor de código fuente usado ha sido **Visual Studio Code** (Microsoft, 2021), una herramienta de gran utilidad y muy extendida en el mundo de la programación.

También es importante hacer referencia a los repositorios y códigos fuentes que han sido utilizados para la elaboración del proyecto. El más importante es el repositorio de GitHub **retrieval-augmented-generation**[1]. Este repositorio nos ha servido como base para desarrollar nuestra aplicación, ya que consiste en una demo básica de una aplicación con características muy similares.

Dentro del lenguaje de programación Python, cabe mencionar el uso de varias librerías relevantes en este proyecto:

- **Streamlit** (Streamlit, 2024): Se emplea para construir la UI de la aplicación y añade numerosas facilidades para su ejecución.

[1] https://github.com/kesamet/retrieval-augmented-generation.git

- **Langchain** (Langchain, 2024)**:** Es la encargada de realizar múltiples funciones relacionadas con el sistema de IA implementado en nuestra aplicación, entre ellos comunicarse con los modelos o transformar los documentos suministrados a un formato legible para la IA.
- **pyttsx3** (Bhat, 2020): Se usa para realizar una síntesis del habla para convertir las respuestas que dará nuestra LLM en voz. En conjunto usamos el módulo **AudioManager**[2], que nos permite guardar, borrar y reproducir los archivos de audio generados por pyttsx3.
- **Speech_Recognition** (Zhang, 2017): Se usa para reconocer consultas que pueda realizar un usuario en voz alta, y transcribirlas a texto. Debido a la inexactitud de estos servicios, también hemos implementado que una vez se haya reconocido la pregunta del usuario, esta sea transcrita a una ventana de texto editable, de manera que el usuario pueda editar su pregunta si algo no se ha transcrito correctamente.

4.1 Sistema de Inteligencia Artificial

En esta subsección vamos a hablar de, probablemente, la parte más importante de nuestra infraestructura: el Sistema de Inteligencia Artificial. Este sistema se compone de los distintos tipos de IA que se usan en nuestro proyecto, y los modelos específicos elegidos.

1. Modelo Principal

Este es nuestro LLM, el eje principal de toda nuestra aplicación. Esta IAG está pensada para mantener un diálogo con un usuario. Para adaptarlo a la imitación de un personaje histórico, usamos varias técnicas:

- **System prompt:** Se define como un conjunto inicial de instrucciones que sirven como punto de partida para el modelo cuando comenzamos una nueva conversación. Ayuda al modelo a concentrarse en la tarea que va a realizar

[2] https://github.com/DougDougGithub/Babagaboosh.git

y mejorar sus respuestas. En nuestro caso el system prompt incluye "Vas a actuar como Francisco de Arobe…"

- **Parámetros de configuración:** Los parámetros son una parte de la configuración del modelo que ajustan distintas características de la IA. Por ejemplo, la temperatura es un parámetro con valores entre 0.0 y 1.0 determina la aleatoriedad de los resultados. Cuanto más alta sea la temperatura, más impredecibles y creativos serán los resultados (Mao, 2024). En nuestro proyecto estos parámetros están definidos en el archivo config.yaml.
- **La Generación Mejorada por Recuperación (RAG):** Una técnica de IA que permite optimizar la salida de un modelo LLM mediante el acceso a una base de conocimiento más pequeña separada de sus datos de entrenamiento originales. Ayuda a extender las capacidades de los LLM a dominios específicos sin la necesidad de reentrenar el modelo (Amazon, 2023). En nuestra aplicación, esta base de conocimiento adicional será un VectorDB, que detallaremos en el siguiente apartado.

Aunque al comienzo del proyecto vimos que había muchos posibles modelos para usar, tras un cuidadoso estudio decidimos escoger Mistral, que es una LLM francesa de código abierto y en concreto el modelo **mistral-7b-instruct-v0.2**[3]**.** Como indica el propio nombre este modelo tiene 7 billones de parámetros y es una versión instruct fine-tuned del modelo original. Hemos escogido este modelo por su alta comprensión del español, mejor que otros modelos con mayor soporte como puede ser Llama2, la disponibilidad pública del modelo, su compatibilidad con librerías de inteligencia artificial como Langchain y su bajo uso de RAM, de manera que se pueda ejecutar en dispositivos que no dispongan de mucha potencia de hardware.

Nuestra LLM se almacena de manera local como un solo archivo **mistral-7b-instruct-v0.2.Q5_K_M.gguf**[4]. GGUF es un formato de archivo diseñado específicamente para facilitar la inferencia de modelos LLM en hardware comunitario. El modelo original

[3] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[4] https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF

es transformado en un solo archivo mediante el uso de llama.cpp, y así el modelo es fácilmente almacenado y compartido.

2. *Modelo Embeddings*

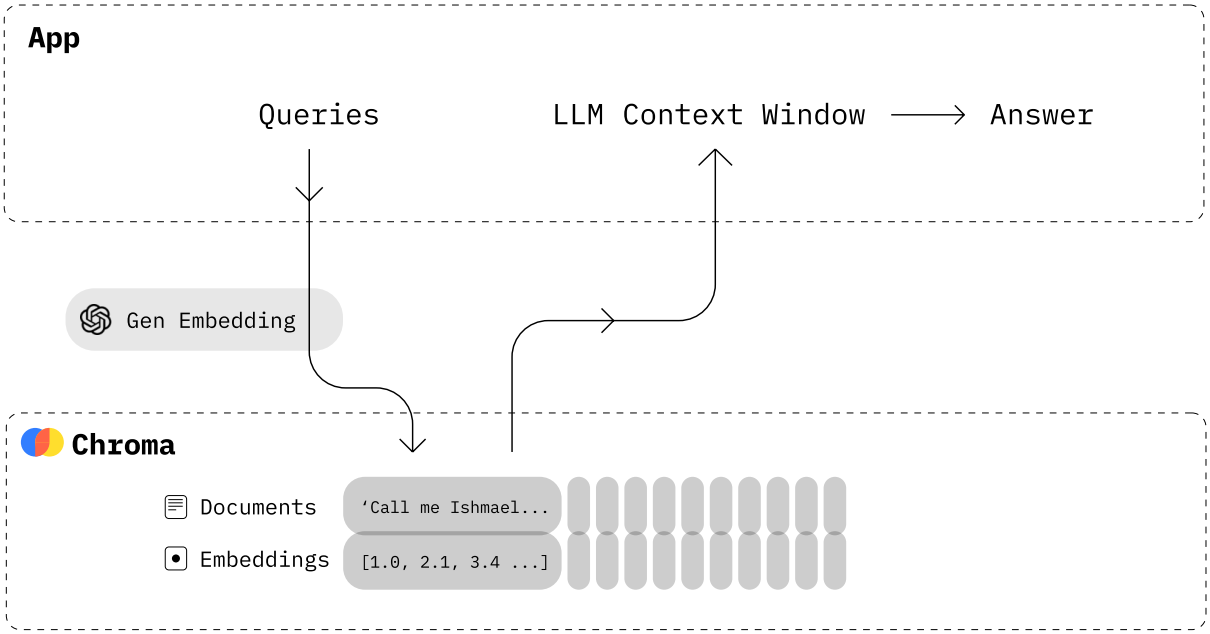

Figura 4-1. Modelo de embeddings, extraído de (Chroma, 2023)

El modelo embeddings es un tipo particular de modelo de IA que se encarga de mapear frases y párrafos a un vector espacial de 384 dimensiones que se usa para hacer clustering o búsqueda semántica. En nuestro proyecto el modelo específico que se usa es **all-MiniLM-L6-v2**[5] (Reimers & Gurevych, 2019).

Este modelo recibe documentos ya divididos en fragmentos de menor tamaño y junto a **Chroma** (Chroma, 2023), una base de datos vectorial, crea un vector que contiene los embeddings de los documentos originales. Este vector recibe el nombre de **VectorDB**, y es una de las piezas claves para nuestra aplicación, especialmente para la RAG. Otra utilidad del VectorDB es que es persistente, y por lo tanto una vez creado se

[5] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

puede acceder a él en sucesivas ejecuciones de la aplicación. Podemos ver ilustrado el funcionamiento de este sistema en la Figura 4-1.

Si la aplicación la usa un usuario con permisos de administrador, se podrá crear un vectorDB nuevo desde la propia interfaz de la aplicación con documentos proporcionados por este usuario.

3. Modelo Reranker

Como antes hemos explicado, nuestro sistema cuenta con documentos que son transformados en embeddings, y almacenados en un vectorDB. Una vez recibida una query de un usuario, este modelo reranker se encarga de computar una puntuación de relevancia a los distintos nodos del vectorDB, que en realidad son fragmentos del texto de los documentos que hemos procesado usando el modelo de embeddings, y se seleccionan los N nodos con la mejor puntuación para devolverlos.

Este mecanismo es muy importante para nuestra aplicación ya que usamos un sistema RAG, y la calidad de nuestra generación va a depender en parte de la calidad de la recuperación de documentos.

En nuestro proyecto, usamos el modelo reranker **bge-reranker-base**[6] (Xiao, Liu, Zhang, & Muennighoff, 2023) que se encarga de hacer la compresión y el re-ranking de los documentos almacenados en el VectorDB.

4. Cadena de recuperación

Una vez hayamos cargado en nuestra aplicación el modelo LLM, el modelo Reranker y el VectorDB, crearemos una cadena de recuperación conversacional (Conversational Retrieval Chain), que incorpora todos estos elementos.

La cadena de recuperación funciona de la siguiente manera (Diaz, 2023):

[6] https://huggingface.co/BAAI/bge-reranker-base

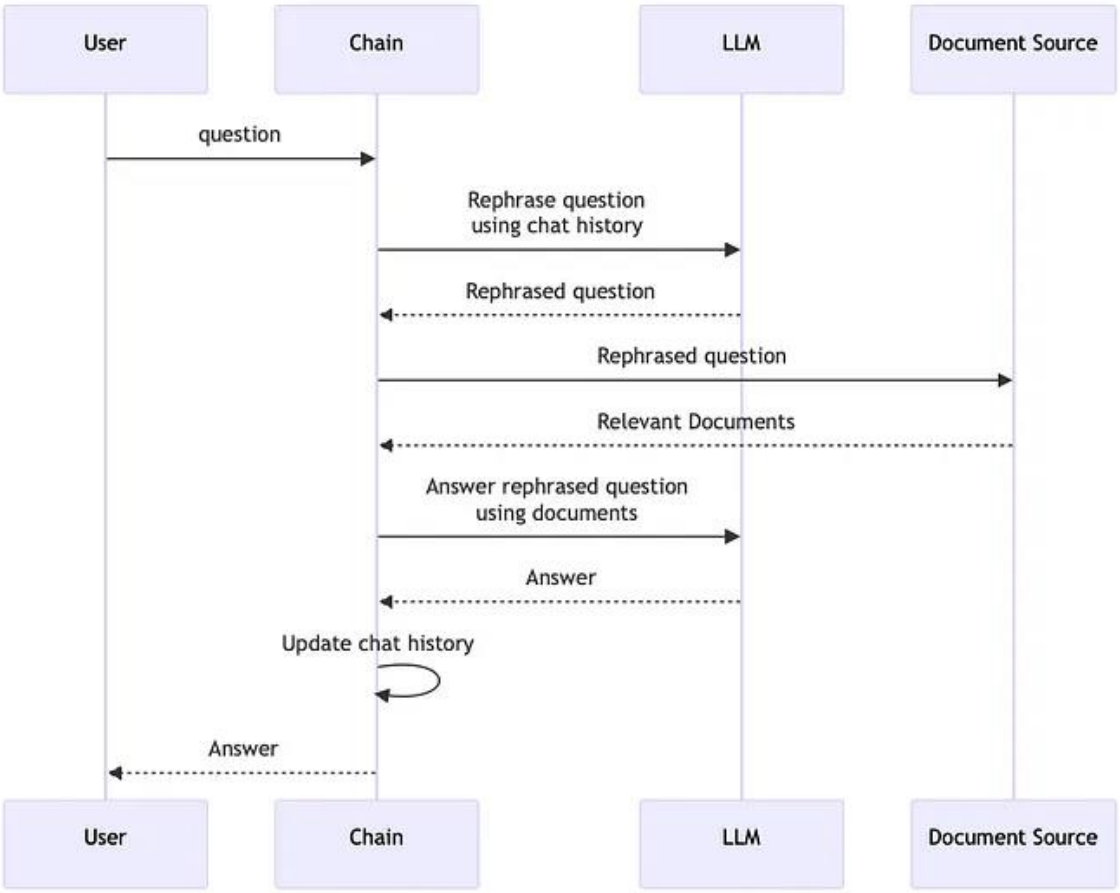

Figura 4-2. Cadena de Recuperación Conversacional, extraído de (Diaz, 2023)

1. Recibe una pregunta del usuario.

2. Reformula la pregunta para que el modelo pueda recordar el contexto de la conversación.

   Esto solo ocurre si ya se ha realizado al menos una pregunta anteriormente. Se manda al LLM el historial del chat y la pregunta para que pueda contextualizarla. El siguiente ejemplo ilustra cómo funciona este proceso:

   P1: ¿Quién fue el primer presidente de Estados Unidos?

   R1: El primer presidente de Estados Unidos fue George Washington.

P2: ¿Cuándo fue elegido presidente?

R2: George Washington fue elegido presidente el 30 de abril de 1789.

P2 solo tiene sentido teniendo en cuenta la pregunta anterior. Así podemos intuir que dada una pregunta como P2, el modelo LLM la reformulará a algo similar a: ¿Cuándo fue George Washington elegido presidente?

3. Recupera los fragmentos de documentos más relevantes con respecto a la pregunta.

   Los documentos están fragmentados y guardados como embeddings en el VectorDB. Usando el modelo reranker se seleccionan los N fragmentos más relevantes, y estos son los que recuperan.

4. Elabora la respuesta a la pregunta reformulada haciendo uso de los fragmentos de documentos más relevantes.

5. El usuario recibe la respuesta y se actualiza el historial del chat.

Capítulo 5 - Arquitectura de la aplicación.

En este capítulo, hablaremos de la arquitectura de la aplicación. Podemos dividir nuestra aplicación en dos partes principales: Interfaz de usuario, la parte visible con la que interactúan los usuarios, y Servicios, que representa la parte de implementación de la funcionalidad incluyendo toda la gestión de información, la IA y los diferentes flujos de datos.

5.1 Interfaz de Usuario

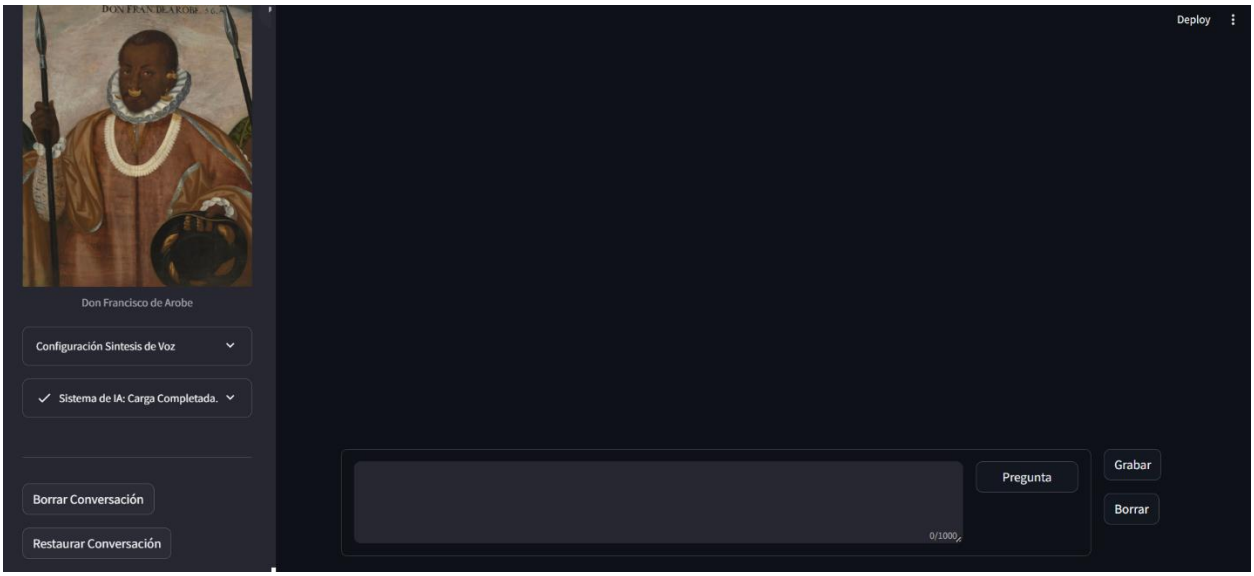

Figura 5-1. Interfaz de Usuario

Nuestra interfaz de usuario (ver Figura 5-1) es la parte visible de la aplicación con la que los usuarios interactúan de manera directa. Podemos dividirla en dos bloques funcionales:

- **Barra lateral:** Incluye el control para activar y desactivar la síntesis de voz. También incluye la funcionalidad de borrar y restaurar el historial del chat que se muestra en la ventana principal.
- **Ventana principal:** Se ocupa de la comunicación entre el usuario y la IA. El usuario puede hacer una consulta mediante texto o reconocimiento de voz. También se ocupa de mostrar el historial del chat a medida que se van haciendo consultas.

Nuestra aplicación consta de dos modos de ejecución: administrador (dev) y usuario (user). Las funcionalidades especificadas en este apartado son las disponibles en el modo usuario. Si iniciamos la aplicación en modo administrador, la barra lateral incluirá información sobre el estado de la aplicación, y la opción de cargar nuevos documentos para construir un vectorDB.

5.2 Servicios

Llamamos servicios al resto de nuestra aplicación, que incluye el código y los datos y modelos de los que hacemos uso o con los que nos comunicamos.

Nuestro código está compuesto por un archivo principal: **app.py**, que se encarga de permitir la comunicación entre todos los elementos y sistemas de la aplicación, contiene el código que genera la interfaz e incorpora todas las funcionalidades en esta, y de los distintos módulos en los que apoya.

Muchos de los **módulos** son auto explicativos: llms.py carga el modelo LLM, vectordb.py crea y/o carga el VectorDB, prompt_templates.py guarda los diferentes tipos de prompts que se pueden usar. En general, estos módulos se usan para propósitos muy específicos y para comunicarnos con elementos externos a la aplicación, por ejemplo, los modelos de IA, o librerías externas, como pyttsx3.

Otro archivo que nos parece relevante mencionar es **config.yaml**. YAML es un lenguaje de serialización de datos interpretable por personas que suele utilizarse en el diseño de archivos de configuración. En nuestra aplicación su función es precisamente esa. Este archivo agrupa todas las configuraciones necesarias para el correcto funcionamiento de la aplicación, incluyendo el modo administrador o usuario, los modelos de IA que vamos a usar o los parámetros de dichos modelos. Para permitir el acceso a la información que contiene lo transformamos en una estructura llamada CFG (objeto Box).

Los modelos de IA ya han sido detallados en el capítulo de Infraestructura, así que solo queda mencionar que los modelos y el vectorDB son elementos persistentes a los que accedemos localmente.

5.3 Flujo de información

El flujo de información entre las distintas partes de la aplicación se maneja mediante varios mecanismos. Primero usamos elementos propios de Streamlit que nos facilitan esta comunicación: Por ejemplo, si en nuestra UI hay un botón y lo pulsamos, se activará su respectiva función callback en el código. Otra herramienta para este flujo es el **Logger,** que nos da información sobre el estado interno del programa a través de la consola desde donde se ejecuta la aplicación.

Capítulo 6 - Conclusiones y trabajo futuro

6.1 Conclusiones

Los museos son una institución cultural y educativa cuya importancia es innegable. Para poder seguir cumpliendo su función como institución es necesario que evolucionen paralelamente al mundo que los rodea.

En nuestra sociedad contemporánea, esto implica crear herramientas y recursos haciendo uso de nuevas tecnologías. En nuestro caso, nos hemos centrado en la Inteligencia Artificial, cuyo desarrollo y alcance ha crecido de manera exponencial en la última década.

El Museo de América nos hizo una propuesta: Crear una instalación donde un visitante del museo pueda mantener una conversación con Francisco de Arobe, un personaje histórico del Siglo de Oro. Para cumplir con la propuesta y sus objetivos relacionados se ha desarrollado una aplicación donde un usuario puede comunicarse con un avatar que representa a dicho personaje.

La aplicación cuenta con una interfaz de usuario que incluye la posibilidad de usar tanto escritura como reconocimiento de voz para formular una pregunta y editarla a gusto del usuario. El sistema devuelve una respuesta que se muestra por pantalla y puede ser reproducida a través de la síntesis de voz.

La caracterización del avatar se ha realizado mediante un sistema de Inteligencia Artificial. Este sistema se compone de un modelo principal LLM como base, que hace uso de otros tipos de modelos de IA para que la adaptación sea lo mejor posible.

Para probar la aplicación se han realizado una gran cantidad de consultas variando distintos parámetros, ajustes e incluso modelos para la IA, con el propósito de encontrar la mejor configuración posible. La aplicación desarrollada ha mostrado ser correcta y funcional, pero su alcance e impacto podrían llegar a ser mucho mayores con el trabajo futuro considerado (ver la siguiente sección).

Las respuestas normalmente son apropiadas en cuanto a lo que podría esperar un usuario no experto en el Siglo de Oro del personaje, y el uso de RAG ayuda a mejorar su calidad. Son particularmente apropiadas cuando las consultas tratan sobre aspectos personales de la vida de don Francisco. Sin embargo, a veces las respuestas pueden ser impredecibles o carecer de sentido, especialmente cuanto más se prolongue una conversación.

La aplicación incluye una interfaz del usuario simple e intuitiva, con un funcionamiento fácil de comprender, e incluye todas las funcionalidades necesarias para la aplicación en una sola página. También cabe destacar que al depender de un servidor web local, se elimina la necesidad de una conexión a Internet una vez desplegada de manera local.

6.2 Trabajo futuro

De cara al trabajo futuro, es importante considerar las limitaciones que hemos tenido, y como superarlas para mejorar la aplicación:

Empezando por el hardware, nuestra aplicación ha sido desarrollada en una máquina con una RAM de 16GB. Creemos que ha sido una buena elección porque cualquier máquina con características similares puede ejecutar nuestra aplicación con un rendimiento aceptable, pero es cierto que tener una máquina con más memoria sería potencialmente más beneficioso.

Ese beneficio sería especialmente relevante en el Sistema de Inteligencia Artificial. Los modelos de IA, en particular los LLM, requieren una gran cantidad de RAM para poder ejecutarse de manera local, es decir, que hay mejores modelos que podrían usarse si se dispusiera de un hardware más potente.

Los modelos de IA también son responsables de otra limitación importante: la mayoría de los modelos destacables en cuanto a rendimiento y alcance tienen una comprensión más limitada del español, y consecuentemente la calidad de las respuestas no va a ser la misma que si el personaje histórico fuese anglosajón.

El alcance de las técnicas de adaptación de IA también aumentaría en conjunción con el hardware. El mejor ejemplo es el fine-tuning: una técnica muy potente de reentrenamiento que permite especializar un modelo existente, pero que tiene unos requisitos muy altos de hardware (se necesita disponer de GPU) y de datos (se necesita crear un dataset específico para el reentrenamiento).

En cuanto a software, hemos encontrado ciertas limitaciones al haber optado por el uso de servicios y programas gratuitos. Destaca la síntesis de voz, que está limitada a los servicios gratuitos que posee la máquina. Sería interesante disponer de un programa con una mayor librería de voces (ej. Google Cloud[7]), o incluso usar un programa que pueda crear voces específicas y use IA para la síntesis (ej. ElevenLabs[8]).

La incorporación de un modelo animado de Francisco de Arobe podría ser un gran añadido para la aplicación, mejorando la inmersión del usuario.

Para terminar, creemos que de cara al futuro lo más importante sería colaborar más activamente con el personal del Museo de América para lograr mejores resultados en la caracterización y tener varios ciclos de desarrollo, seguidos de pruebas en el museo para obtener feedback por parte de visitantes.

[7] https://cloud.google.com/?hl=es

[8] https://elevenlabs.io/

Introduction

Motivation

Museums are a cultural and educational institution whose work revolves around the heritage they preserve. For centuries, they have been a key instrument in the expansion of culture, spread of knowledge and education in our society.

Even with their long history and relevance, nowadays museums face important challenges related to their public. On one hand, some museums have been turned into massive tourist spots (ex. Louvre, British Museum, Museo del Prado) where many visitors have had their cultural interest partly replaced by plain sightseeing interest. On the other hand, COVID-19 forced many museums to close their doors for a prolonged period of time, putting a lot of them in danger of never opening their doors again (Ibermuseos, 2020).

The general public, especially the younger generations, stray further away from classic museum experiences where you observe the pieces and read text associated to them. Instead, they look for more immersive and interactive experiences. Also, new technologies and instant access to information may deter people from going to museums, as they can opt for just looking up the information online (Morales Carmona & Freitag, 2014).

Aware of this situation, both institutions and experts are constantly researching ways of transforming museums to have better interactions with their surroundings. Information Technologies (ITs) play a crucial role in this research, for example, with initiatives such as websites, social media, audio guides or augmented reality (Boiano, Cuomo, & Gaia, 2016). Even though they can be incredibly successful, museums have to deal with the difficulties of innovating, integrating a reusing these experiences, and also their high cost.

This project wants to contribute to this line of development through the creation of more accessible, innovative, and interesting experiences for visitors that make use of new technologies, in particular Artificial Intelligence (AI). In our case, we will develop an

application that can work as a complement to a museum installation and that users can interact with.

Goals

The main functionality of our app is a chat where a user can interact with an AI model in a conversation where the AI is playing the part of a historic figure. The AI will adjust to this context with instructions from the developers, adaptation techniques and procured documents.

Inside general development of our application, we can see more clearly the objectives we have to adhere to:

- Integrate into the app a GAI model as our base, specifically an LLM.
- Implement a User Interface (UI).
- Use of various AI adaptation techniques, such as Retrieval Augmented Generation (RAG), to improve our model's performance.
- Develop the characterization of the LLM as a historical figure.
- Incorporate a chat history to maintain a registry of each conversation that takes place.
- Implement other functionalities such as voice recognition and voice synthesis to allow for spoken communication and conversation, and also to improve the accessibility of our application.

Work plan

To complete the objectives previously stated, we establish a work plan consisting of various successive stages:

1. **Investigation.**

We start our work plan with a researching stage into LLMs, the necessary resources to be able to utilize them, the current state of this technology and its evolution in the foreseeable future, infrastructure requirements and other aspects needed for the project.

2. **Choosing our architecture and infrastructure.**

Once we have concluded our investigation, we will decide what architecture and infrastructure we are going to make use of (ex. Libraries, frameworks, and development tools).

3. **First development phase: Initial Application.**

Development of the application commences. We start by creating a prototype based on our available source code. The scope of this prototype includes an application with a simple interface where the user can communicate with the LLM, and the chat history is shown on screen.

4. **Second development phase: Artificial Intelligence System.**

The Artificial Intelligence System is implemented in its entirety: We select which models we are going to use, and we characterize the LLM as the chosen historical figure through various AI adaptations techniques such as RAG.

5. **Third development phase: User Interface.**

We refine our app, improving our UI by incorporating and modifying UI elements such as buttons, display menus and text areas. We also implement both a voice recognition system and a voice synthesis system.

6. **Development closure.**

We conduct a revision of the entire implementation to make sure everything is functioning correctly: this includes bug fixes, performance checks, quality of response checks and more. We also create an admin/user mode that restricts certain functions depending on the access level of the user.

Conclusions and future work

Museums are a cultural and educational institutions of undeniable importance. To make sure that they can keep fulfilling their role as an institution it is necessary for them to evolve as so does the world that surrounds them.

In our current society, this implies the creation of resources and tools with the use of modern technologies. We have chosen to focus specifically on Artificial Intelligence, whose development and capabilities have grown exponentially in the past decade.

The Museum of America made the following proposal: To create an installation where a visitor can converse with Francisco de Arobe, a historical figure from the 16th century. To complete this proposal, we developed an app where a user can communicate with an avatar that represents said figure.

The app includes a user interface where you can formulate a query both by writing or by speaking, thanks to the voice recognition system. The app will generate a response that will appear on screen as text and can also be synthesized and played as audio.

The characterization of the avatar is managed by an Artificial Intelligence System. This system consists of a LLM as the base, and other types of AI models that help it make the characterization as accurate as possible.

To evaluate the application, a large number of queries have been summited while varying parameters, settings and even AI models, with the intention of finding the best possible configuration. The final product is a correct and functional application, but we consider its scope and performance could be much better (see next section: Future Work).

The responses are usually adequate for what could be expected of a user with no expertise in the character or the 16th century, and the use of AI adaptation techniques such as RAG help improve their quality. These responses are particularly correct when the queries' topics revolve around the character's personal life. However, the responses can sometimes be unpredictable or lack meaning, especially the longer the conversation lasts.

The app's user interface is simple and intuitive, with an operation design easy to understand, and it includes all the different implemented functionalities in a single webpage. Also, as the app is dependent on a local web server, it eliminates the need for an internet connection once deployed locally.

Future Work

As for future work, it is important to consider the limitations we have face, and how to overcome them to improve the application:

First, we should discuss hardware. Our hardware environment consists of a 16GB RAM computer. We chose this option because any machine with equivalent properties will be able deploy the app with acceptable performance, but certainly having access to a machine with more memory would be potentially beneficial.

This benefit would be particularly relevant for the Artificial Intelligence System. AI models, especially LLMs, require a large amount of RAM to be deployed locally, meaning that with more hardware power we would have access better models.

AI models are also responsible for another relevant limitation: most models with remarkable performance and scope have a limited comprehension on the Spanish language, and consequently the quality of the responses will not be as good as if the historic figure selected was Anglo-Saxon.

The scope of AI adaptation techniques would also improve in conjunction with hardware, The best example is fine-tuning: a powerful retraining technique that allows an existing model to be specialized, but that has an enormous cost in hardware (GPU machine necessary) and in data (specific dataset created just for retraining necessary).

Now, moving on to software, the limitations faced have been primarily due to opting on using free services and programs. Notably the synthesis voice system is limited to the free services the machine offers. It would be interesting to have access to a program which includes a bigger repertoire of voices (ex. Google Cloud[9]), or even a

[9] https://cloud.google.com/?hl=es

program that can create personalized voices and use AI to synthesize them (ex. ElevenLabs[10]).

The implementation of an animated model of Francisco de Arobe could be a great addition to the application, improving the users' immersion.

To finalize our thoughts, we think that looking to the future, it would be essential to work more closely with the Museum of America to achieve a better characterization and to incorporate multiple development cycles into the project, followed by test sessions in the museum to obtain feedback from the potential users.

[10] https://elevenlabs.io/

Bibliografía

Amazon. (2023). Amazon Web Services: Machine Learning e IA. *Amazon Web Services: Machine Learning e IA*. Retrieved from https://aws.amazon.com/es/what-is/retrieval-augmented-generation/

Anaconda. (2020). Anaconda Software Distribution. *Anaconda Software Distribution*. Anaconda Inc. Retrieved from https://docs.anaconda.com/

Bhat, N. M. (2020, Julio). pyttsx3. *pyttsx3*. Retrieved from https://pypi.org/project/pyttsx3/

Boiano, S., Borda, A., Gaia, G., Rossi, S., & Cuomo, P. (2018, Enero). Chatbots and New Audience Opportunities for Museums and Heritage Organisations. *Electronic workshops in computing*. doi:10.14236/ewic/EVA2018.33

Boiano, S., Cuomo, P., & Gaia, G. (2016, Enero). Real-time Messaging Platforms for Storytelling and Gamification in Museums: A case history in Milan. *Electronic workshops in computing*. doi:10.14236/ewic/eva2016.60

Boiano, S., Gaia, G., & Caldirini, M. (2003). Make Your Museum Talk: Natural Language Interfaces For Cultural Institutions. *Make Your Museum Talk: Natural Language Interfaces For Cultural Institutions*. Retrieved from https://www.museumsandtheweb.com/mw2003/papers/gaia/gaia.html

Bonnat, C., Oliveira, G., & Sanchez, E. (2020). "Geome", un juego para comprender el Antropoceno durante las visitas escolares a un museo. *Enseñanza de las Ciencias de la Tierra, 28*(1), 89–98.

Cabrera, J., Frutos, H., Stoica, A., Avouris, N., Dimitriadis, Y., Fiotakis, G., & Liveri, K. (2005). Mystery in the museum: Collaborative learning activities using handheld devices., *111*, pp. 315-318. doi:10.1145/1085777.1085843

Cath, C. (2018). Governing artificial intelligence: ethical, legal and technical opportunities and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376*, 20180080. doi:10.1098/rsta.2018.0080

Chroma. (2023). Chroma: the AI-native open-source embedding database. *Chroma: the AI-native open-source embedding database*. Retrieved from https://www.trychroma.com/

Diaz, J. (2023, Diciembre). Conversational Retrieval Chain, how does it work? *Conversational Retrieval Chain, how does it work?* Retrieved from https://medium.com/@jerome.o.diaz/langchain-conversational-retrieval-chain-how-does-it-work-bb2d71cbb665

Dini, R., Paternò, F., & Santoro, C. (2007). An environment to support multi-user interaction and cooperation for improving museum visits through games., (pp. 515-521). doi:10.1145/1377999.1378062

Fiallos, B. (2023). La función educativa de los museos en las sociedades contemporáneas. *Aula Virtual, 4*, 163-171. doi:10.5281/zenodo.7600940

github. (2020). GitHub. *GitHub*. Retrieved from https://github.com/

Ibermuseos. (2020). Informes de impacto del COVID-19 en el ecosistema del museo. *Informes de impacto del COVID-19 en el ecosistema del museo*. Retrieved from https://www.ibermuseos.org/informes-de-impacto-del-covid-19-en-el-ecosistema-del-museo/

ICOM. (2022). Definición de museo. *Normas y Directrices de ICOM*. Retrieved from https://icom.museum/es/recursos/normas-y-directrices/definicion-del-museo/

Langchain. (2024). Langchain. *Langchain*. Retrieved from https://python.langchain.com

Mao, A. (2024, Febrero). Large Language Model Settings: Temperature, Top P and Max Tokens. *Large Language Model Settings: Temperature, Top P and Max Tokens*. Retrieved from https://www.linkedin.com/pulse/large-language-model-settings-temperature-top-p-max-tokens-albert-mao-0c6ie

Microsoft. (2021, Noviembre). Visual Studio Code. *Visual Studio Code*. Microsoft. Retrieved from https://code.visualstudio.com/

Mittal, A. (2023, Diciembre). El problema de la caja negra en los LLM. *El problema de la caja negra en los LLM: desafíos y soluciones emergentes*. Retrieved from

https://www.unite.ai/es/the-black-box-problem-in-llms-challenges-and-emerging-solutions/

Morales Carmona, I., & Freitag, V. (2014). Los Museos en el Siglo XXI: nuevos retos, nuevas oportunidades. *Revista Digital do LAV, 7*, 30-49. Obtenido de https://www.redalyc.org/articulo.oa?id=337030167004

Parra Sepúlveda, D., & Concha Machuca, R. (2021, Octubre). Inteligencia artificial y derecho. Problemas, desafíos y oportunidades. *Vniversitas, 70*, 1–25. doi:10.11144/Javeriana.vj70.iadp

RedHat. (2024). ¿Qué es la inteligencia artificial generativa? . *¿Qué es la inteligencia artificial generativa? Ejemplos y riesgos*. Retrieved from https://www.redhat.com/es/topics/ai/what-is-generative-ai

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics. Retrieved from http://arxiv.org/abs/1908.10084

Rodríguez, I. (Agosto de 2021). *Los caciques negros de Esmeraldas. La historia detrás de un cuadro*. Obtenido de Euxinos: https://www.euxinos.es/2021/08/09/los-caciques-negros-de-esmeraldas-la-historia-detras-de-un-cuadro/

Streamlit. (2024). Streamlit: The fastest way to build and share data apps. *Streamlit: The fastest way to build and share data apps*. Snowflake Inc. Retrieved from https://streamlit.io/

Taylor, D., Yeung, M., & Bashet, A. Z. (2021). Personalized and Adaptive Learning. doi:10.1007/978-3-030-58948-6_2

UNESCO. (2020, Mayo). La UNESCO y el ICOM preocupados por la situación de los museos del mundo. *La UNESCO y el ICOM preocupados por la situación de los museos del mundo*. Retrieved from https://www.unesco.org/es/articles/la-unesco-y-el-icom-preocupados-por-la-situacion-de-los-museos-del-mundo

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual.* Scotts, Valley, CA: CreateSpace.

Varitimiadis, S., Kotis, K., Skamagis, A., Tzortzakakis, A., Tsekouras, G., & Spiliotopoulos, D. (2020). Towards implementing an AI chatbot platform for museums. *International conference on cultural informatics, communication & media studies*, *1*. doi:https://doi.org/10.12681/cicms.2732

Xiao, S., Liu, Z., Zhang, P., & Muennighoff, N. (2023). C-Pack: Packaged Resources To Advance General Chinese Embedding. *C-Pack: Packaged Resources To Advance General Chinese Embedding*.

Zhang, A. (2017). Speech Recognition (Version 3.8) [Software]. *Speech Recognition (Version 3.8) [Software]*. Retrieved from https://github.com/Uberi/speech_recognition#readme

APÉNDICES

Apéndice A - Glosario de términos

- IA Inteligencia Artificial
- IAG Inteligencia Artificial Generativa
- ICOM Consejo Internacional de Museos (inglés, *International Council of Museums*)
- LLM Modelo de Lenguaje Grande (inglés, *Large Language Model*)
- RAG Generación Mejorada por Recuperación (inglés, *Retrieval Augmented Generation*)
- TI Tecnologías de la Información
- UI Interfaz de Usuario (inglés, *User Interface*)

Apéndice B - Manual de instalación

Para poder ejecutar la aplicación, es importante seguir los siguientes pasos.

1. Descargar el repositorio de GitHub del enlace (https://github.com/martagago48/tfg.git).

2. Extraer el repositorio en algún directorio local, para este ejemplo vamos a suponer C:\hlocal, de manera que la dirección completa del repositorio será C:\hlocal\tfg

3. Si conda no está instalado, es necesario instalarlo primero (https://docs.conda.io/projects/conda/en/latest/user-guide/install/index.html).

4. Descargar los modelos de inteligencia artificial:

   - LLM: (https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF/tree/main). Hay que seleccionar el modelo **mistral-7b-instruct-v0.2.Q5_K_M.gguf.**
   - Modelo Reranker: (https://huggingface.co/BAAI/bge-reranker-base). Para descargarlo es necesario clonar el repositorio.
   - Modelo Embeddings: (https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2). Para descargarlo es necesario clonar el repositorio.

   Todos deben ser almacenados en el directorio **tfg/models/**

5. Abrir el programa Anaconda Prompt (viene incluido con la instalación de conda), y ejecutar los siguientes comandos:
   - Navegar hasta el directorio local del repositorio. En el caso del ejemplo:

     cd C:\hlocal\tfg
   - Crear nuestro env, para evitar dependencias conflictivas:

     conda env create --name tfg -f environment.yaml --force

- Una vez creado, debemos activarlo:

  conda activate tfg

  Sabremos si se ha activado correctamente si los primeros caracteres de la línea de comando pasan de (base) a (tfg).

- Por último, ejecutar la aplicación:

  streamlit run app.py

6. La app se ejecuta en una ventana de navegador, y podemos interactuar con el personaje.

Los anteriores pasos describen el proceso de instalación de requisitos y la primera ejecución. Si queremos volver a ejecutar la aplicación en algún otro momento, hay que seguir los siguientes pasos:

Abrir Anaconda Prompt, y ejecutar los siguientes comandos, asumiendo que la dirección local del repositorio es **C:\hlocal\tfg**:

cd C:\hlocal\tfg

conda activate tfg

streamlit run app.py