AnswerBook

Valadez Gutierrez David, Chávez Velasco Cristian, Ruíz González Mariana,
LopezArce Delgado Jorge Ernesto

CENTRO UNIVERSITARIO DE CIENCIAS

EXACTAS E INGENIERÍAS, (CUCEI, UDG)

david.valadez4501@alumnos.udg.mx

cristian.cvelasco@alumnos.udg.mx

mariana.ruiz7824@alumnos.udg.mx

jorge.lopezarce@academicos.udg.mx

***Abstract*— Lo que realmente diferencia a AnswerBook de otros chatbots es su integración con la vasta biblioteca de Project Gutenberg, ofreciendo acceso directo a miles de libros clásicos. Mientras otros chatbots pueden proporcionar respuestas generales, AnswerBook se especializa en profundizar en textos literarios de forma precisa y específica. Su capacidad para entender y analizar contenido complejo lo hace ideal para estudiantes, investigadores y cualquier persona interesada en obtener información detallada y contextualizada sin perder tiempo. Además, la interfaz de AnswerBook está diseñada para ser intuitiva y fácil de usar, garantizando una experiencia de usuario superior.**

***Palabras claves* – NLP, chatbot, extracción de información, libros.**

Repositorio de código:
https://github.com/davidvaladez09/Red-Neuronal-Generativa-para-Consulta-de-libros
Versión actual del código:
1.0.0
Licencia legal código:
Código abierto copyleft.

I. Introducción

El proyecto "AnswerBook" trata de mejorar la interacción entre humanos y sistemas de inteligencia artificial en la exploración y comprensión de contenidos literarios. Dirigido a usuarios diversos como puede ser el caso de un estudiante o amantes de la lectura, busca desarrollar un chatbot capaz de comprender el lenguaje natural, extraer información relevante y generar respuestas adecuadas a partir de libros.

Su importancia radica en facilitar el acceso a la información, fomentar el aprendizaje continuo y avanzar en el campo de la inteligencia artificial aplicada al procesamiento del lenguaje natural. La implementación de una Red Neuronal Generativa promete soluciones innovadoras al generar respuestas coherentes, potenciando la capacidad del chatbot para comprender consultas complejas y proporcionar respuestas precisas.

II. Trabajos Relacionados

Existen varios proyectos que abordan problemáticas similares. Por ejemplo Humata.io desarrolla una inteligencia artificial especializada en la comprensión y generación de respuestas a partir de texto, pero su enfoque se centra en aplicaciones empresariales y su acceso puede requerir pago ya que es de servicio premium. Por otro lado, está la inteligencia de copilot de Edge, ofrecen soluciones de inteligencia artificial para procesamiento de lenguaje natural, pero su capacidad de manejo de archivos pesados puede ser limitada. También se encuentran modelos de inteligencia artificial como ChatGPT 3.5, que cuenta con capacidades avanzadas de procesamiento del lenguaje natural, pero su acceso gratuito puede tener restricciones en términos de funcionalidad.

No existe una inteligencia gratuita que sea exclusivamente para preguntar cosas específicas de algún libro en cuestión que aparte te permita cargar directamente la información, buscamos con este proyecto ayudar a estudiantes a comprender mejor temas o dudas específicas de libros o algún amante de la literatura que tenga algunas preguntas.

III. Descripción del Desarrollo del Proyecto Modular

El equipo adoptó una metodología de trabajo modular para el desarrollo del proyecto. Se dividió en tres áreas principales: BackEnd, FrontEnd y Documentación, cada una con responsabilidades específicas. Se estableció una comunicación fluida y se asignaron tareas de acuerdo con las habilidades y competencias de cada miembro.

Fig. 1 Cronograma de actividades general

Se realizó una planificación detallada de las tareas, asignando responsabilidades específicas a cada miembro del equipo. Se establecieron plazos y se monitoreó el progreso del proyecto de manera regular.

TABLA I
Entregables David

Responsable	Objetivo	Resultado	Entregable	Fecha
David	Objetivo del proyecto, Delimitaciones y herramientas	Envio de correo electronico	Correo electronico	30/04/2024
	Objetivos individuales	Tareas para cada integrante	Objetivos en cronograma	02/05/2024
	Implementacion de modelo	Commit de modelo en repositorio	Modelo funcionando	07/05/2024
	Resolver problemas con back y front	Commit final de proyecto terminado	Correcciones de codigo	09/05/2024
	Proyecto terminado, Presentacion de proyecto terminada	Entrega final, Presentacion de proyecto	Proyecto final, Presentacion del proyecto	14/05/2024

TABLA II
Entregables Mariana

Responsable	Objetivo	Resultado	Entregable	Fecha
Mariana	Objetivo del proyecto, Delimitaciones y herramientas	Envio de correo electronico	Correo electronico	30/04/2024
	Objetivos individuales	Tareas para cada integrante	Objetivos en cronograma	02/05/2024
	Implementacion una ventana con tkinter	Commit de ventana en repositorio	Codigo ventana con tkinter	07/05/2024
	Ventana tkinter junto con backend	Commit final back y front unidos	Codigo ventana tkinter junto con back	09/05/2024
	Proyecto terminado, Presentacion de proyecto terminada	Entrega final, Presentacion de proyecto	Proyecto final, Presentacion del proyecto	14/05/2024

TABLA III
Entregables Cristian

Responsable	Objetivo	Resultado	Entregable	Fecha
Cristian	Objetivo del proyecto, Delimitaciones y herramientas	Envio de correo electronico	Correo electronico	30/04/2024
	Objetivos individuales	Tareas para cada integrante	Objetivos en cronograma	02/05/2024
	Implementar interaccion con usuario	Commit de modelo en repositorio	Modelo funcionando con usuario	07/05/2024
	Probar funcionamiento de la ventana junto con bakcend	Correo electronico de errores	Reporte en correo sobre errores	09/05/2024
	Proyecto terminado, Presentacion de proyecto terminada	Entrega final, Presentacion de proyecto	Proyecto final, Presentacion del proyecto	14/05/2024

Los principales requerimientos del proyecto incluyen la implementación de un modelo de red neuronal para el procesamiento del lenguaje natural, el desarrollo de una interfaz de usuario intuitiva y funcional, y la generación de documentación detallada sobre las herramientas utilizadas.

1. *Manejo de Consultas:* El usuario envía una consulta a través de la interfaz de usuario. La consulta se procesa mediante las técnicas de NLP (análisis gramatical, reconocimiento de entidades) para entender su significado.

2. *Generación de Respuestas:* El modelo de generación de respuestas utiliza la información extraída de la base de datos de libros para generar una respuesta adecuada. La respuesta se verifica para asegurar que sea coherente y relevante al contexto de la consulta.

3. *Entrega de Respuestas:* La respuesta generada se envía de vuelta al usuario a través de la interfaz de usuario. La interfaz de usuario presenta la respuesta de manera clara y comprensible.

4. *Optimización y Eficiencia:* Se realizan pruebas y ajustes continuos para mejorar la precisión y relevancia de las respuestas del chatbot.

Se emplearon diversas tecnologías, como Large Language Models (LLM), LangChain, Albert Small 2, Transformer, entre otras. Estas herramientas permitieron manejar grandes volúmenes de datos de texto, generar contenido coherente y relevante, y procesar el lenguaje natural de manera efectiva.

A. Large Language Models (LLM)

Modelos de lenguaje estadísticos o basados en aprendizaje automático que están diseñados para manejar grandes cantidades de datos de texto y generar contenido coherente y relevante.

B. LangChain

Framework que nos permite interactuar con LLMs de manera fácil y rápida. El cual cuenta con dos funcionalidades principales: Integración de datos (llamado data-aware): conecta un LLM con otra fuente de datos, por ejemplo, todo el texto que tiene un PDF Es posible que queramos que nuestra aplicación tenga un contexto determinado, o responda de una manera específica y solo cuando tenga respuestas específicas.

C. ConversationChain

Es una cadena más versátil diseñada para gestionar conversaciones. Genera respuestas basadas en el contexto de la conversación y no necesariamente depende de la recuperación de documentos.

D. ConversationalRetrievalChain

Está diseñado específicamente para responder preguntas basadas en documentos. Útil cuando tiene documentos específicos que desea utilizar para generar respuestas.

E. Embeddings

Técnica de procesamiento de lenguaje natural que convierte el lenguaje humano en vectores matemáticos. Estos vectores son una representación del significado subyacente de las palabras, lo que permite que las computadoras procesen el lenguaje de manera más efectiva. Permiten que las palabras sean tratadas como datos y manipuladas matemáticamente.

F. ChromaDB

Base de datos especializada en el almacenamiento y recuperación eficiente de información lingüística, incluyendo datos de texto, anotaciones semánticas y sintácticas. Es particularmente útil para el almacenamiento y la gestión de grandes cantidades de datos de lenguaje natural, lo que permite a los desarrolladores aprovechar al máximo los avances en algoritmos de aprendizaje automático y análisis de texto.

G. RetrievalQA

Herramienta que combina técnicas de recuperación de información con procesamiento del lenguaje natural para responder preguntas formuladas en lenguaje natural sobre textos largos o documentos extensos. La idea detrás de este enfoque es que, en lugar de analizar todo el documento cada vez que se hace una pregunta, la herramienta primero busca en la base de datos o corpus relevante para encontrar los fragmentos más prometedores que podrían contener la respuesta.

El código del proyecto se encuentra disponible en el repositorio: [https://github.com/davidvaladez09/Red-Neuronal-Generativa-para-Consulta-de-libros].

Se llevaron a cabo pruebas para garantizar el funcionamiento correcto de cada componente del sistema. Las pruebas se enfocaron en validar funciones individuales y asegurar la unión entre los diferentes módulos.

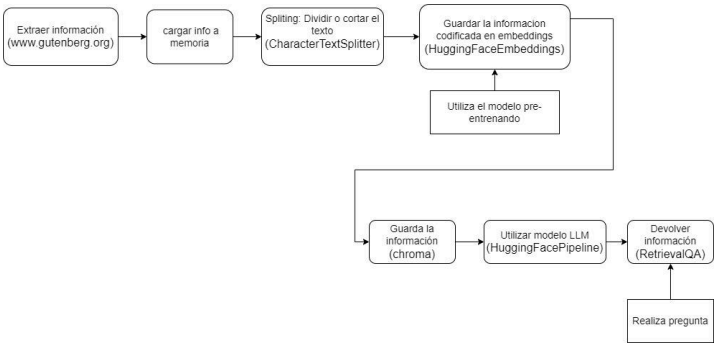

Fig. 2 Diagrama de funcionamiento

Durante el desarrollo del proyecto, se enfrentaron diversos problemas, como incompatibilidad entre versiones de librerías de Python y dificultades en la implementación de funciones

actualizadas. Se buscaron soluciones efectivas, como adaptar una interfaz gráfica en Google Colab y utilizar versiones específicas de las librerías para mantener la estabilidad del entorno de desarrollo.

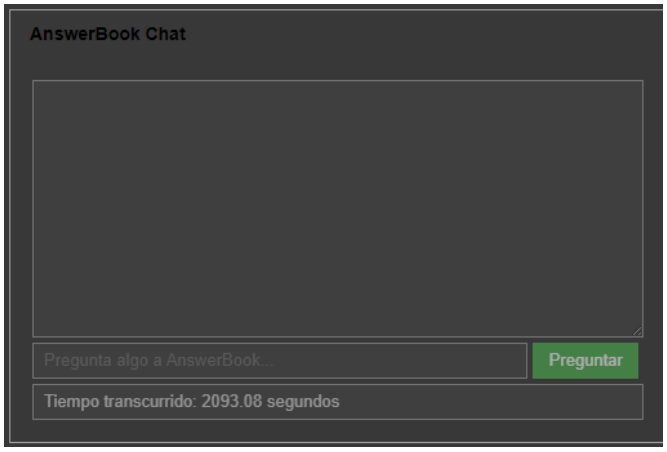

Fig. 3 Vista previa de la ventana

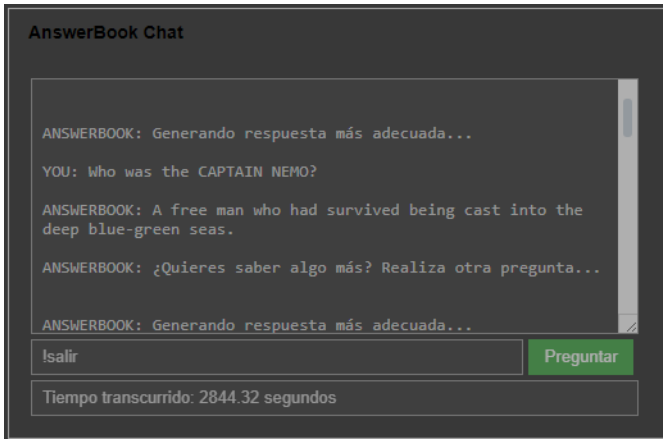

Fig. 4 Prueba de funcionamiento

TABLA IV
Pruebas y porcentaje de error

Respuesta correcta	Tiempo de respuesta
0	36.45
0	93.44
1	38.76
1	39.88
0	41.09
1	32.07
1	34.58
1	48.12
1	39.08
0	59.23
1	34.51
0	53.53
1	93.00
1	25.18
0	75.89
0	30.37
0	34.48
0	93.70
1	28.25
1	49.54
0	49
1	62

0	37
1	60
0	59
1	74
0	60
1	58
1	80
1	50
0	42
1	120
1	59
1	45
0	77
0	70
0	57
1	75
0	66
1	80
1	27.84

1	53.22
0	15.2
0	26.32
0	12.03
0	57.11
1	58.1
1	46.23
1	13.02
1	83
1	30
1	56.32
1	15
0	60
0	103
RESPUESTAS CORRECTAS: 31 / 55	PROMEDIO DE TIEMPO:
PORCENTAJE DE ACIERTOS: 56%	48.65.

Módulo I Arquitectura y programación de sistemas

En el desarrollo de AnswerBook, se decidió utilizar Python como el lenguaje de programación principal debido a su versatilidad y amplia adopción en el procesamiento del lenguaje natural y el desarrollo de inteligencia artificial. Para la gestión de datos, se usó una base de datos vectorial utilizando ChromaDB, la cual permite almacenar y recuperar

eficientemente información lingüística. Se realizó un modelado detallado del sistema, abarcando desde la interfaz de usuario hasta el backend, asegurando coherencia y funcionalidad a lo largo de todo el proyecto.

Módulo II Sistemas inteligentes

AnswerBook se apoya en varias ramas de la inteligencia artificial, específicamente redes neuronales y aprendizaje automático. Se implementaron modelos preentrenados de Large Language Models (LLMs) y técnicas de embeddings para procesar y comprender el lenguaje natural. El modelo matemático subyacente se basa en la transformación de texto a vectores matemáticos, permitiendo la comparación y recuperación eficiente de información. La selección de algoritmos, como los utilizados en LangChain y los embeddings de HuggingFace, se justificó por su capacidad para manejar grandes volúmenes de datos textuales y generar respuestas coherentes y relevantes, cumpliendo así con los objetivos del proyecto de ofrecer una herramienta interactiva y precisa para la exploración de textos literarios.

Módulo III Sistemas distribuidos

El sistema de AnswerBook se diseñó para ser distribuido, permitiendo el procesamiento concurrente de consultas y la gestión de datos a través de Colab, que básicamente es una plataforma colaborativa en la nube. La base de datos vectorial ChromaDB está distribuida para manejar eficientemente la gran cantidad de datos textuales provenientes de Project Gutenberg. El procesamiento de cálculos y generación de respuestas se reparte a través de la infraestructura de Google Colab, asegurando disponibilidad y rendimiento. El sistema establece comunicación entre múltiples componentes distribuidos, facilitando la integración y el procesamiento en tiempo real. Esta arquitectura distribuida mejora la tolerancia a fallos y garantiza que los usuarios puedan interactuar con el chatbot de manera fluida y eficiente.

IV. Resultados Obtenidos del Proyecto

Se lograron los objetivos establecidos para el proyecto AnswerBook al término de su desarrollo. Los resultados obtenidos demuestran la efectividad del chatbot para la consulta de libros y la mejora en la interacción entre humanos y sistemas de inteligencia artificial. Los logros significativos alcanzados en el desarrollo de AnswerBook reflejan un importante avance en la implementación de un chatbot capaz de comprender el lenguaje natural, extraer información relevante y generar respuestas contextualmente adecuadas a partir de libros.

1) Se logró implementar con éxito un chatbot que permite a los usuarios realizar consultas específicas en libros, facilitando la interacción entre humanos y sistemas de inteligencia artificial en el contexto de la exploración literaria.

2) La funcionalidad de categorización y organización de temas dentro de la plataforma garantiza una estructura efectiva para la exploración y discusión de contenidos literarios, cumpliendo con el objetivo de mejorar la experiencia de usuario.

3) La integración de tecnologías como LangChain potencia la capacidad del chatbot para comprender consultas complejas y proporcionar respuestas precisas, destacando el éxito en la implementación de soluciones innovadoras para abordar el problema planteado.

V. Conclusiones y Trabajo a Futuro

El proyecto AnswerBook ha logrado desarrollar un chatbot efectivo para la consulta de libros, demostrando su capacidad para comprender el lenguaje natural y generar respuestas contextualmente adecuadas. Con la implementación de tecnologías como LangChain ha contribuido significativamente a mejorar la capacidad del chatbot para manejar consultas complejas y proporcionar respuestas precisas. La estructuración eficiente de la plataforma, incluyendo la categorización de temas y la organización de la información, ha mejorado la experiencia del usuario en la exploración de contenidos literarios.

El proyecto ha demostrado ser una solución innovadora para abordar el problema de la interacción entre humanos y sistemas de inteligencia artificial en el contexto de la exploración literaria.

Trabajo a Futuro:

1) Explorar métodos alternativos para mejorar la precisión y relevancia de las respuestas del chatbot, como la implementación de modelos de lenguaje más avanzados o la integración de técnicas de aprendizaje automático adicionales.

2) Realizar pruebas exhaustivas para identificar posibles áreas de mejora en la interfaz de usuario y en la estructura de la plataforma, con el objetivo de optimizar la experiencia del usuario y aumentar la usabilidad del sistema.

3) Investigar la posibilidad de expandir la base de datos de libros disponibles para consulta, así como explorar la inclusión de otras fuentes de información relevantes para enriquecer el contenido ofrecido por el chatbot.

Reconocimientos

Queremos manifestar nuestro reconocimiento a la Universidad de Guadalajara por el conocimiento que nos otorga.

REFERENCIAS

[1] Yannicksteph. (2024, January 2). | NLP | LLM | LangChain RAG | QA Data https://www.kaggle.com/code/yannicksteph/nlp-llm-langchain-rag-qa-data

[2] Alonso, F. (2023, August 29). ¿Qué son y como funcionan los Large Language Models? Future Space S.A. https://www.futurespace.es/large-language-models/

[3] ¿Qué es LangChain?: Explicación sobre LangChain: AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/langchain/

[4] Hernández, B. (2024, February 14). ¿Qué es LangChain y cómo crear aplicaciones Python con esta librería? Paradigma Digital. https://www.paradigmadigital.com/dev/que-es-langchain-como-crear-aplicaciones-python-libreria/

[5] Difference between ConversationChain and ConversationalRetrievalChain · Issue #7885 · langchain-ai/langchain. (n.d.). GitHub. https://github.com/langchain-ai/langchain/issues/7885

[6] Local Embeddings with HuggingFace - LlamaIndex. (n.d.). https://docs.llamaindex.ai/en/stable/examples/embeddings/huggingface/

[7] Split by character | 🦜🔗 LangChain. (n.d.). https://python.langchain.com/v0.1/docs/modules/data_connection/document_transformers/character_text_splitter/

[8] Casano, A. B. (2024, March 17). Bases de Datos Vectoriales: Teoría y Práctica con ChromaDB y LangChain | Medium. Medium. https://medium.com/@aberrospic1/bases-de-datos-vectoriales-teor%C3%ADa-y-pr%C3%A1ctica-con-chromadb-y-langchain-0140ab635a24

[9] Jorcan. (2023, April 26). ChromaDB en el Procesamiento del Lenguaje Natural (NLP): Una guía completa. BrainQ. https://brainq.ai/chromadb/

[10] Espíndola, G. (2023, March 3). ¿Qué son los embeddings y cómo se utilizan en la inteligencia artificial con python? Medium. https://gustavo-espindola.medium.com/qu%C3%A9-son-los-embeddings-y-c%C3%B3mo-se-utilizan-en-la-inteligencia-artificial-con-python-45b751ed86a5

[11] Pretrained Models — Sentence-Transformers documentation. (n.d.). https://www.sbert.net/docs/pretrained_models.html#sentence-embedding-models/

[12] Naka, M. (2024, January 10). Enhancing LangChain's RetrievalQA for real source links. Medium. https://nakamasato.medium.com/enhancing-langchains-retrievalqa-for-real-source-links-53713c7d802a

[13] sentence-transformers/paraphrase-multilingual-mpnet-base-v2 at main. (n.d.). https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2/tree/main

[14] Ignosblog. (2024, April 17). LangChain y Hugging Face. Introducción. Ignos Blog. https://ignos.blog/langchain-y-hugging-face-introduccion

[15] Evaluar modelos. (n.d.). Google Cloud. https://cloud.google.com/translate/automl/docs/evaluate?hl=es-419

[16] Evaluar modelos. (n.d.). Google Cloud. https://cloud.google.com/translate/automl/docs/evaluate?hl=es-419