Memoria Proyecto Capstone (IIA)

Sistema de Recuperación de Información (RAG) para Contact Center

Autor: Felipe González García
Fecha: 23 Abril 2025
Curso: Desarrollador 10x con IA (IIA)

1. Introducción

Este proyecto tiene como objetivo desarrollar un sistema de recuperación de información (Retrieval-Augmented Generation, RAG) para asistir a los agentes de un contact center corporativo.

El sistema se basa en técnicas de Procesamiento de Lenguaje Natural e IA para ayudar a los agentes del contact center a encontrar rápidamente la información que necesitan para responder a las consultas de los clientes, mejorando la eficiencia y la calidad del servicio. Permite a los agentes realizar consultas en lenguaje natural y obtener respuestas relevantes basadas en una base de conocimientos interna, en este caso, un manual de procedimientos almacenado en una hoja de cálculo de Google Sheets.

La necesidad de este tipo de sistema surge de la creciente complejidad y volumen de información que los agentes deben manejar diariamente. Un sistema automatizado de búsqueda y recuperación de información permite reducir el tiempo dedicado a la búsqueda manual, liberando a los agentes para que se concentren en la interacción con el cliente.

2. Descripción del Sistema

El sistema se basa en la arquitectura RAG (Retrieval-Augmented Generation), una técnica que combina la búsqueda de información tradicional con la generación de lenguaje natural. Esta arquitectura permite al sistema no solo encontrar la información relevante, sino también sintetizarla y presentarla de forma clara y concisa.

Módulos

El sistema se compone de los siguientes módulos:

Ingestión de Datos:

- Se utiliza la librería gspread para acceder a la hoja de cálculo de Google Sheets que contiene el manual de procedimientos.
- Los datos se leen y se almacenan en un DataFrame de Pandas para su procesamiento.
- Se define una columna "documento_completo" que combina información de varias columnas para crear el texto que representa cada procedimiento.

Generación de Embeddings:

- Se utiliza la librería langchain_openai y el modelo text-embedding-ada-002 de OpenAI para generar embeddings (representaciones vectoriales) de cada procedimiento.
- Estos embeddings capturan el significado semántico de los procedimientos y permiten realizar búsquedas por similitud.

Búsqueda de Información:

- Se utiliza la similitud del coseno para comparar el embedding de la consulta del usuario con los embeddings de los procedimientos.
- Se devuelven los procedimientos más similares como contexto relevante para la respuesta.

Generación de Respuestas:

- Se utiliza la librería langchain_openai y el modelo gpt-4 de OpenAI para generar la respuesta final.
- Se utiliza una plantilla de prompt que proporciona al modelo el contexto relevante y la pregunta del usuario.
- El modelo genera una respuesta que se basa en el contexto proporcionado.

Interfaz de Chatbot:

- Se implementa un chatbot interactivo que permite a los usuarios realizar consultas en lenguaje natural.
- El chatbot utiliza el sistema RAG para obtener la respuesta y la muestra al usuario.

Flujo de Trabajo

1. **Entrada del agente**: El agente formula una pregunta en lenguaje natural a través de una interfaz de chat.
2. **Preprocesamiento**: La pregunta del agente se procesa para eliminar palabras irrelevantes y estandarizar el formato.
3. **Recuperación de información**: El sistema busca en la base de datos los procedimientos más relevantes a la pregunta utilizando un modelo de embeddings y la similitud coseno.

    - Modelo de embeddings: OpenAI Embeddings se utiliza para convertir los procedimientos y la pregunta en vectores numéricos que representan su significado semántico.
    - Similitud coseno: Se calcula la similitud entre el vector de la pregunta y los vectores de los procedimientos para identificar los más relevantes.
4. **Generación de respuesta**: El LLM (ChatGPT) genera una respuesta final basándose en los procedimientos recuperados y la pregunta del agente, siguiendo una plantilla de prompt específica.
    - Prompt engineering: Se utiliza una plantilla de prompt para guiar al LLM en la generación de la respuesta, especificando el contexto y el formato deseado.
5. **Postprocesamiento**: La respuesta generada se formatea y se le añaden enlaces de interés si es necesario, extraídos de los procedimientos recuperados.
6. **Salida**: El sistema muestra la respuesta al agente a través de la interfaz de chat.

Componentes

- **Google Sheets**: Base de datos de procedimientos del contact center, almacenada en una hoja de cálculo de Google para facilitar su acceso y actualización.
- **Google Colab**: Entorno de desarrollo y ejecución del sistema, aprovechando sus recursos computacionales y la integración con otras herramientas de Google.
- **Pandas & Numpy**: Librerías de Python para el manejo y procesamiento de datos, utilizadas para cargar y manipular la base de datos de procedimientos.
- **Scikit-learn**: Librería de Python para aprendizaje automático, utilizada para calcular la similitud coseno entre los vectores de texto.
- **LangChain**: Framework para el desarrollo de aplicaciones de IA que simplifica la integración de diferentes componentes como modelos de lenguaje, embeddings y bases de datos.
- **OpenAI Embeddings**: Modelo de embeddings preentrenado que proporciona una representación vectorial del texto, permitiendo la comparación semántica entre documentos.
- **ChatGPT**: Modelo de lenguaje de OpenAI que se utiliza para generar respuestas coherentes y relevantes basándose en la información recuperada.

3. Resultados y Conclusiones

El sistema desarrollado ha demostrado su capacidad para recuperar información relevante y generar respuestas útiles para las preguntas de los agentes.

Evaluación del sistema

Se realizaron pruebas con un conjunto de preguntas de prueba, comparando las respuestas generadas por el sistema con las respuestas esperadas. Se observó una alta precisión en la recuperación de información y una buena calidad en la generación de respuestas.

Beneficios del sistema

- **Reducción del tiempo de búsqueda de información**: El sistema automatiza el proceso de búsqueda, permitiendo a los agentes encontrar la información que necesitan de forma rápida y eficiente.
- **Mejora de la calidad de las respuestas**: El sistema proporciona respuestas precisas y completas, basadas en la información más actualizada disponible.
- **Mayor satisfacción del cliente**: Al reducir los tiempos de espera y mejorar la calidad de las respuestas, el sistema contribuye a una mejor experiencia para el cliente.

Conclusiones

El proyecto ha demostrado el potencial de las tecnologías de IA en la automatización y mejora de los procesos en entornos de atención al cliente. El sistema desarrollado ofrece una solución viable para mejorar la eficiencia y la calidad del servicio en un contact center corporativo.

4. Limitaciones y Trabajo Futuro

A pesar de los resultados positivos, este trabajo presentado como proyecto Capstone no deja de ser una POC, y el sistema presenta algunas limitaciones:

- **Dependencia de la calidad de la base de datos**: El sistema se basa en la información contenida en la base de datos de procedimientos. Si la base de datos es incompleta o contiene errores, la calidad de las respuestas se verá afectada.
- **Necesidad de actualización constante**: La base de datos debe actualizarse regularmente para reflejar los cambios en los procedimientos y políticas del contact center.
- **Limitaciones del modelo de lenguaje**: Aunque ChatGPT es un modelo de lenguaje muy potente, puede generar respuestas incorrectas o irrelevantes en algunos casos.

Como trabajo futuro para aterrizar esta POC, pueden ser buenas ideas:

- Implementar un sistema de actualización automática de la base de datos.
- Experimentar con diferentes modelos de embeddings y LLMs para mejorar la precisión y la calidad de las respuestas.
- Integrar el sistema con una interfaz de usuario más amigable e intuitiva (canales de Google Chat p.ej)
- Realizar una evaluación más exhaustiva del sistema con un conjunto de datos más amplio y métricas de desempeño más robustas.
- Explorar la posibilidad de incorporar otras fuentes de información, como bases de conocimiento externas o documentos internos de la empresa.