RAG-Based Electronic Design Assistant for Renesas Electronics

Monica Meduri
Github Link
Live Server Link

Abstract

The exponential growth of online data has posed significant challenges in retrieving accurate and relevant information. Traditional search engines often generate broad results that require extensive manual filtering, while most existing Retrieval-Augmented Generation (RAG) systems rely heavily on static knowledge bases, limiting their ability to retrieve real-time information. This report presents an advanced LLM-based RAG Fusion system designed to enhance information retrieval and response generation by integrating web scraping, ChromaDB, and an AI agent. The proposed system employs a hybrid approach that combines retrieval-based and generative AI models, along with dynamic external searches, to ensure that the most recent and accurate information is provided. Unlike traditional solutions that depend solely on pre-indexed data, our system can dynamically fetch up-to-date information from online sources when necessary. This report outlines the architecture, implementation, and methodology of the RAG-based electronic design assistant developed for Renesas Electronics, designed to consume and process information from the Renesas Knowledge Base. It also discusses the system's advantages, limitations, and potential future enhancements.

1. Introduction

With the rapid increase in web-based data, obtaining accurate and contextually relevant information has become essential across various domains such as education, business, healthcare, and scientific research. Traditional search engines like Google and Bing, although effective in indexing large volumes of information, often require users to manually browse through multiple sources to extract useful insights. This process is not only time-consuming but also places a cognitive burden on users, who must evaluate the credibility and relevance of the obtained information.

The advent of Large Language Models (LLMs), such as OpenAI's ChatGPT, has introduced a paradigm shift in information retrieval by offering coherent, human-like responses to user queries.However, a major limitation of LLMs is that their knowledge is confined to pre-trained datasets, making them ineffective for delivering up-to-date information. This limitation is particularly critical in fields that require real-time updates such as stock markets and dynamic websites.

To address this challenge, Retrieval-Augmented Generation (RAG) systems have emerged as a promising solution. RAG combines the capabilities of language models with external knowledge bases, enabling the model to fetch relevant data from a pre-indexed vector database and generate more accurate and informed responses. However, conventional RAG systems have their limitations. They lack dynamic search capabilities, resulting in knowledge gaps when new or updated information is not present in the database.

This project introduces an advanced LLM RAG Fusion system designed to overcome these limitations by incorporating dynamic web scraping, vector database retrieval, and real-time external searches through an AI agent. The core functionality of the proposed system includes:

- **Knowledge-Based Question Answering:**
  The bot is designed to answer user queries based on the information available in the Renesas Knowledge Base

- **Image Integration in Responses:**
  If the information in the Knowledge Base contains relevant images, the bot is capable of including those images in its response

- **Page Reference Linking:**
  The bot provides direct references to the pages from which the information was retrieved, allowing users to explore further.

- **Image-Based Query Resolution:**
  The bot supports user-uploaded hand-drawn images, or other relevant visuals. It can analyze the image content and provide contextually accurate answers based on the visual input.

- **AI Agent:**
  In cases where the Knowledge Base does not contain satisfactory information or the confidence score is low, the bot can automatically trigger an external search agent that scans across **renesas.com** for potential answers and provide them to the user.

- **Cloud-Based Deployment:**
  The bot is deployed on a AWS cloud server using a **Streamlit** interface, ensuring easy accessibility from any device or location without requiring local installation.

By integrating these advanced technologies, the LLM RAG Fusion system provides a dynamic, real-time knowledge retrieval mechanism, ensuring that responses remain accurate and contextually relevant. This report further delves into existing solutions, the proposed system architecture, implementation details, sample outputs, limitations, and potential future enhancements.

2. Methodology

2.1. Data Processing

The section explains the data processing technique used for Renesas.com

- **Web Scraping and API Retrieval:**
  BeautifulSoup and Selenium are utilized to extract both structured and unstructured text from relevant online sources and stored as pdfs.

- **Data Cleaning and Structuring:**
  The data is extractedusing PymuPDF and extracted data is cleaned by removing redundant content, handling missing values, and normalizing text to improve search accuracy and retrieval precision.

- **Storing Data in JSON Format:**
  The cleaned and structured data is stored in a JSON format. This format ensures efficient storage and retrieval of the processed information. **Figure 1** illustrates a sample structure of the JSON data.

- **Chunking and Embedding Generation:**
  The next step involves breaking the JSON data into smaller, meaningful chunks to preserve semantic integrity. Each text entry, such as questions, answers, and contextual descriptions, is split into

manageable pieces. This chunking process ensures that significant information is preserved while maintaining coherence.

- **Embedding Conversion:**
  After chunking, the text is converted into high-dimensional embeddings using hugging face all-MiniLM-L6-v2 embedding models. These embeddings enable efficient similarity-based retrieval by transforming the text into vector representations.

- **Storing in ChromaDB:**
  The generated embeddings, along with relevant metadata such as URLs, images, and source references, are stored in ChromaDB.

Workflow

The electronic design assistant is built using a Retrieval-Augmented Generation (RAG) architecture that enables users to query a comprehensive knowledge base and receive context-rich, relevant answers. The system's backend leverages AWS EC2 for deployment, supporting critical components such as PDF extraction (via extractpdf.py), JSON extraction, and embedding-based document retrieval. It processes documents by scraping technical content from various sources, which are then formatted into a structured JSON format containing key attributes like questions, URLs, answers, and associated images. These documents are embedded using an embedding model, which creates vector representations that capture the semantic essence of the content. The embeddings are stored in ChromaDB, a vector database optimized for semantic search, ensuring efficient document retrieval. Images associated with the documents are also linked, allowing for the retrieval of both textual and visual data during query resolution.

The workflow begins when a user submits a query, either in text or image form. The system processes both types of input, converting them into embedding representations to facilitate comparison. For text-based queries, the RAG model uses these embeddings to search ChromaDB for semantically similar documents. If matching documents are found and their similarity score exceeds a predefined threshold (e.g., 0.5), the system combines the documents and the user query to generate a response using a Large Language Model (LLM) such as GPT-4. If no suitable matches are found, the system activates an AI agent to perform a real-time web search across predefined external sources, integrating the newly retrieved information to generate a response. The system also processes image inputs using vision models, combining visual and textual data for more comprehensive answers. As part of this multi-modal approach, the assistant can interpret hand-drawn sketches and integrate them with document-based retrieval, providing highly relevant and accurate responses. By ensuring smooth integration of both text and image retrieval, the system guarantees a robust solution for electronics design queries. The deployment on AWS EC2 ensures scalability, while the Streamlit interface offers an intuitive platform for users to interact with the assistant.

Results

Limitations

The proposed system faces several limitations, including latency challenges due to real-time web scraping, which can slow down response times. Scalability may be affected under high query loads, requiring efficient resource management. Additionally, web scraping restrictions, such as CAPTCHA and bot detection, can limit the system's ability to retrieve up-to-date information.

Future scope

Future improvements include integrating real-time APIs for faster, more reliable data retrieval and reducing reliance on web scraping. Enhancements in query optimization through reinforcement learning and personalized responses based on user interactions will improve efficiency and accuracy. These upgrades aim to make the system more adaptable and effective in real-world applications.