An Open-Source RAG Architecture for LLMs

Lakshmi Boppana, Manav Bhadoria and Ravi Kishore Kodali
Department of Electronics and Communications Engineering
National Institute of Technology, Warangal
WARANGAL, INDIA
ravikkodali@gmail.com

***Abstract*—Accurate product classification in e-Commerce and supply chain management is essential to smooth operations and enhance the customer experience. While Large Language Models (LLMs) perform exceptionally in natural language processing, they encounter issues like model hallucination and dependence on outdated information. Furthermore, LLMs often rely on outdated data. This paper introduces an open source cloud-based RAG model, using Amazon Web Services (AWS) and vector databases to address these issues. The RAG architecture combines retrieval-based and generation-based methods, allowing them to supplement responses with up-to-date information from external sources, thus reducing the risk of model hallucination. The project employs a Vector DB deployed in EC2 to improve contextual understanding and retrieval capabilities of these large language models. Through comprehensive experimentation and AWS deployment, the RAG system improved contextual comprehension and increased the accuracy of the generated output. Semantic similarity search results significantly improve retrieval performance.**

***Index Terms*—Large language models, recovered generation, natural language processing, machine learning.**

I. Introduction

Large Language Models (LLMs) have transformed the realm of machine learning, showcasing exceptional abilities in handling natural language processing tasks. However, despite their success, they encounter several shortcomings. These include model hallucination, lack of a domain-specific knowledge repository, and dependence on outdated information. To overcome these issues and boost performance, the adoption of Retrieval-Augmented Generation (RAG) techniques is suggested. RAG synergizes the strengths of LLMs with the retrieval of pertinent information from external data sources. RAG is promising for improving the effectiveness of LLMs in various fields. Efficient product categorization in e-Commerce and supply chain management is vital to operations and customer experience. This work explores the use of vector databases and RAG architecture to develop a scalable merchant classification system. The methods use vector-based embeddings to identify semantic similarity between products and leverage RAG for contextual understanding and text generation, which can also enhance recommendation systems. The goal is to advance automated classification systems and recommendation engines. This paper provides a detailed analysis of vector databases and RAG techniques, especially in the areas of data extraction and recommendation systems.

II. Related Work

The advent of retrieval-augmented generation (RAG) models marks a notable progress in natural language processing, offering an innovative method that combines retrieval and generation techniques [1]. The RAG model framework consists of two primary components: the retriever and the generator. The retriever component fetches relevant documents from a dense vector index according to the input query or context, while the generator component generates text based on internal knowledge within the pre-trained language model and external information retrieved by the retriever.

In addition, incorporating vector database solutions within the RAG architecture significantly improves the capability of these models to manage and utilize external knowledge sources. The study on Vector Database Management Systems Survey [2] presents innovative techniques for effective vector data management, tackling issues such as semantic similarity ambiguity, vector dimensions, and the computational burden associated with similarity comparisons. By seamlessly integrating VectorDB solutions into the RAG framework, researchers can adopt enhanced storage and indexing methods, facilitating more efficient retrieval of pertinent documents during the RAG models' retrieval phase.

Vector Database Management Systems (VDBMS) have become essential tools for the efficient management of high-dimensional vector data across various applications [3]. These systems are specifically designed to store and retrieve vectors that represent complex data entities, providing unique indexing and querying methods that cater to the specific properties of vector data. In the field of VDBMSs, several prominent systems have been recognized, such as Pinecone, Chroma, Milvus, Weaviate, Qdrant, and Deep Lake.

A detailed examination of the YouTube recommendation algorithm highlights significant improvements facilitated by deep learning techniques [4]. The recommendation system operates in two stages: candidate generation and ranking.

TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON) | 979-8-3503-5082-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/TENCON61640.2024.10903064

During candidate generation, the model extracts a subset of videos from the vast collection, ensuring that they are broadly relevant to the user. The ranking model then evaluates these candidates using a variety of features that describe both the video and the user. The highest-ranking videos are finally presented to the user, personalized, and ordered by relevance. In candidate generation and ranking, deep neural networks are essential, as they learn embeddings for videos, users, and other contextual attributes to effectively distinguish between videos. These models are designed with multiple layers of fully connected Rectified Linear Units (ReLU), utilizing features such as video and search history, as well as demographic data.

The research explores the obstacles posed by the rapid growth of the COVID-19 literature and the development of geospatial and semantic mapping platforms to address these challenges [5]. With more than 23,000 articles released in a brief time frame, conventional search engines and databases do not offer the sophisticated filtering and semantic analysis tools required to efficiently navigate this large amount of data. Numerous existing platforms lack geospatial analysis features, limiting their ability to map the geographical spread of COVID-19 research. To address these shortcomings, researchers have developed an all-encompassing knowledge discovery system specifically for COVID-19 publications. This system combines advanced natural language processing algorithms for semantic analysis with geospatial mapping methods, offering multi-dimensional search options and customizable visualizations. The paper carefully investigates the usage of natural language processing tools, specifically semantic similarity searches, to enhance the speed and effectiveness of reviewing large volumes of information.

III. Retrieval-Augmented Generation Architecture

Retrieval-Augmented Generation utilizes retrieval methods to collect pertinent data from vast knowledge bases, enhancing the generation procedure. This method improves the contextual understanding and the informativeness of the responses produced.

A. Retrieval Component

The retrieval component is tasked with fetching relevant information from diverse knowledge sources, including databases, or the internet. Various methods such as keyword matching, semantic similarity, or dense vector retrieval can be employed for this purpose. In the methodology, the knowledge source comprises embeddings of domain-specific documents. Retrieval can be conducted at different granularities, spanning from entire documents to specific text segments.

B. Augmented Component

Once relevant information is retrieved, it is used to enhance the generation process of a large language model. This is achieved by incorporating the retrieved information as an additional context during the generation phase. The retrieved information serves as a form of external knowledge that guides the generation model in producing more contextually relevant responses. The retrieved information can be concatenated with the input text to provide domain-specific knowledge.

C. Integration of Retrieval and Generation

One of the key challenges in RAG is the effective integration of the retrieval and generation components to ensure seamless interaction between the two. Techniques such as multitask learning, reinforcement learning, or adversarial training can be used to jointly train the retrieval and generation components to optimize their performance. In this report, AWS services are utilized to make the augmented generation retrieval architecture scalable and globally accessible.

IV. Design and Implementation

In this paper, a cloud-based Retrieval-Augmented Generation Model is proposed to enhance the capabilities of Large Language Models. By combining the power of LLMs with external knowledge sources, such as AWS Textract and ChromaDB, the model addresses challenges such as model hallucination and reliance on outdated data. This approach opens up new possibilities for natural language processing tasks by leveraging both internal and external knowledge. Embeddings, essentially compressed vector representations of objects such as text or images, facilitate efficient storage and retrieval within ChromaDB. ChromaDB can integrate with various pre-trained models like all-MiniLM-L6-v2. These models are trained on massive datasets and excel at capturing semantic relationships between words.

ChromaDB is a vector database that stores data extracted from documents processed by AWS Textract. ChromaDB has been deployed as a Docker Image on an AWS EC2 instance, leveraging the flexibility and scalability of cloud computing infrastructure. Running ChromaDB within a Docker container on AWS EC2 ensures seamless integration with AWS Textract and other components of the pipeline. With an indexing time complexity of O(n log n) and a search time of O(log n), ChromaDB ensures rapid data processing while maintaining precision. Among the plethora of available LMs, Llama-2 emerges as a prominent open-source model for its efficacy in understanding and generating text. This study focuses on employing the quantized iteration of Llama-2, with a primary emphasis on optimizing resource utilization without compromising performance. Quantization, a technique involving the compression of model parameters, presents an opportunity to

reduce memory usage and computational complexity while maintaining accuracy.

V. Problem Statement

Efficient product categorization in e-Commerce and supply chain management is essential. This research explores using vector databases and Retrieval-Augmented Generation (RAG) to develop a scalable Merchant classification system. Vector representations capture semantic similarities, and RAG provides contextual understanding and generation. These techniques also extend to recommendation systems. The study aims to improve automated classification systems and recommendation engines.

VI. Proposed Work

The initial stage of the study consisted of a thorough evaluation of the feasibility of using Generative AI, such as OpenAI ChatGPT, for image processing to extract relevant data from images. Despite rigorous testing, it was found that while Generative AI could effectively identify details in store images, the extracted data were minimal and inadequate for precise Merchant Identification. Several limitations, including poor image quality and missing information, made this approach unsuitable for further use. During the study, an extensive investigation of machine learning algorithms customized to meet the project's needs was carried out. This investigation covered a wide array of algorithms, such as clustering, classification, and regression techniques. Particular focus was given to addressing similarity search challenges, which led to exploring the generation of embeddings using large-language models (LLMs). These models were found promising for improving the accuracy and effectiveness of similarity analysis in classifying merchant data. To assist in the categorization process, the researchers applied Beautiful Soup, a Python library, to extract all relevant merchant categories from government-approved documents. The extracted data were meticulously cleaned for precision. In addition, government databases were utilized to retrieve essential merchant information, such as GSTINs, to collect merchant details for further categorization. Embeddings for both categories and merchant information were created utilizing a variety of pre-trained large language models (LLMs), including the all-MiniLM-L6-v2. This particular model was identified as the ideal balance between speed and precision in the study. These embeddings effectively encapsulated the goods and services provided by each merchant. To assess the distance between embeddings and quantify the similarity between category descriptions and merchant offerings, several similarity algorithms were used, such as cosine similarity, Euclidean distance, and L2 norm distance. Recognizing the limitations of traditional storage systems, the research incorporated vector databases such as ChromaDB. By employing vector databases, the study achieved enhanced query processing speeds and better scalability for storing and handling high-dimensional embeddings.

To address challenges with conflicting goods and services sold by merchants, the research proposed the implementation of Retrieval Augmented Generation. This system combines retrieval-based techniques with generation-based methods to enhance the accuracy and relevance of the results. Through extensive experimentation and deployment on AWS, the RAG system enriched contextual understanding and improved the precision of generated output. The LLM adopted for research purpose was the open source Llama-2. RAG, with its retrieval-based techniques and generation-based methods, offers a unique opportunity to enhance recommendation systems. By leveraging contextual information from a large repository of information. RAG reliant on both its retrieval based technique and generation based technique forms a large knowledge base reconciling discrepancies and inconsistencies in the data.

VII. Cloud Architecture

A cloud-based, open-source Retrieval-Augmented Generation Model is proposed, utilizing Amazon Web Services (AWS) and vector databases to meet this demand. Amazon

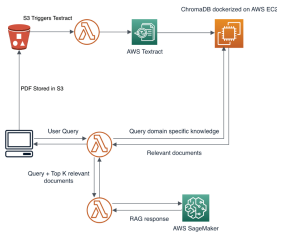

Fig. 1. Overview of cloud architecture

Web Services (AWS) Elastic Compute Cloud (EC2) is a cloud computing service that provides flexible computing resources in the cloud. With EC2, users can rent virtual servers, known as instances, and deploy their applications on them. EC2 offers a wide range of instance types tailored to different use cases, offering options for performance, storage, and computing needs. In the research, EC2 serves as a core component for deploying the Docker image of a vector database. This vector database plays a crucial role in the Retrieval Augmented Generation setup as it stores document embeddings. Using EC2, limitations often associated with traditional infrastructure setups can be overcome. EC2's scalability and agility enable adjustment of computing resources as needed, ensuring the ability to meet evolving demands efficiently. This flexibility facilitates the effective management of computational requirements while maximizing performance and resource utilization. This work uses AWS sagemaker, a fully managed service

designed to streamline the development, training, and deployment of machine learning models on a large scale. It serves as the foundation for hosting large language models, such as Llama 2, which constitute the core of the retrieval augmented architecture. By integrating a vector database deployed on EC2, the project improves contextual understanding and retrieval capabilities of these large language models. This leads to enriching the LLM's performance enabling efficient storage and retrieval of document embeddings. AWS S3 serves as an infrastructure solution in the augmented retrieval generation (RAG) architecture, serving as a repository for all required documents in the natural language processing pipeline. By putting document storage in S3, a robust pipeline is fundamentally created, facilitating an efficient data processing workflow. S3 was seamlessly integrated with the host hosting an EC2 instance of ChromaDB, the vector database solution. Here, the extracted text using AWS Textract continues to be processed and converted for embedding. This project utilizes lambda functions to trigger various workflows. These lambda functions invoke the SageMaker endpoint and interact with the vector database. API gateways are also utilized to serve as the primary interface for HTTP requests, providing external applications with access to the Retrieval Augmented Generation (RAG) system. These gateways are configured to expose endpoints to trigger lambda functions.

VIII. Result

In the initial analysis using only the vector database, Result 1 exhibited alignment with the Merchant Category in 62 instances, while Result 2 aligned in 34 instances. However, a significant portion of the results did not align directly with any Merchant Category, with 48 instances showing no alignment. Moreover, instances where multiple results only partially aligned with the Merchant Category posed a challenge, accounting for 22 occurrences. Result 3 demonstrated alignment in only 10 instances, highlighting the limitation of relying solely on the vector database for accurate categorization. Upon implementing the Retrieval-Augmented Generation

Description	Count
Result_1 aligns with the Merchant Category	98
No result aligns with the Merchant Category	5
Result_2 aligns with the Merchant Category	50
Result_1 and Result_2 partly aligns with the Merchant Category	26
Result_3 aligns with the Merchant Category	2
All the results partly align with the Merchant Category; challenge to determine a single Merchant Category	4
Not Sufficient amount of Merchant Data	8
Result_1 and Result_3 partly aligns with Merchant Category	4
Result_2 and Result_3 aligns with Merchant Category	3

Fig. 2. Categorization of Merchants based on Vector Databases

(RAG) model, notable improvements were observed across various metrics. Result1 (Fig.2) displayed alignment with the Merchant Category in 98 instances, showcasing a considerable improvement from the initial count. Similarly, Result 2 (Fig.3) showed alignment in 50 instances, indicating a significant improvement in categorization accuracy. The number of instances where multiple results only partially aligned with the Merchant Category decreased to 26, reflecting the efficacy of using external knowledge sources.

Description	Count
Result_1 aligns with the Merchant Category	62
No result aligns with the Merchant Category	48
Result_2 aligns with the Merchant Category	34
Result_1 and Result_2 partly aligns with the Merchant Category	22
Result_3 aligns with the Merchant Category	10
All the results partly align with the Merchant Category; challenge to determine a single Merchant Category	8
Not Sufficient amount of Merchant Data	8
Result_1 and Result_3 partly aligns with Merchant Category	5
Result_2 and Result_3 aligns with Merchant Category	3

Fig. 3. Categorization of Merchants based on RAG

Fig. 4. CSV Results of top three recommendations

IX. Conclusion

A cloud-based RAG model aimed at enhancing the functionality of LLMs has been explored. By integrating LLMs with external systems such as AWS Textract and ChromaDB, the model addresses key issues like model hallucination and dependency on outdated data. Utilizing a comprehensive knowledge base, the amalgamation of retrieved data and a pre-trained LLM within the RAG framework substantially improves the outcomes of semantic similarity searches. This cutting-edge method not only advances natural language processing tasks but also expands the scope of exploiting both internal and external knowledge in text generation activities.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021.

[2] J. J. Pan, J. Wang, and G. Li, "Survey of vector database management systems," 2023.

[3] T. Taipalus, "Vector database management systems: Fundamental concepts, use-cases, and current challenges," *Cognitive Systems Research*, vol. 85, p. 101216, 2024.

[4] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016.

[5] X. Ye, J. Du, X. Gong, and et al., "Geospatial and semantic mapping platform for massive covid-19 scientific publication search," *Journal of Geovisualization and Spatial Analysis*, vol. 5, no. 5, 2021.