Enhancing Customer Support with Retrieval-Augmented Generation (RAG) Techniques

Dr.Varsha Bhole; Rahul Santra; Aditya Redekar; Nishant Rathod
Information Technology, A.C.Patil College of Engineering Kharghar,India

Abstract— A customer support system in the modern digital environment should be able to respond to user inquiries in a timely, accurate, and contextually appropriate manner. The most recent development enhances conversational AI by using the Retrieval-Augmented Generation (RAG) model, which enables companies to easily integrate AI agents into their websites. Accordingly, customers can upload a unique document or URL, which will be processed to create a customized vector store. This system's strength is its ability to retrieve a large number of pertinent documents from the vector store and use a large language model (LLM) or a related idea to produce contextually aware answers. This way gives AI agents the ability to swiftly reply to a vast array of client inquiries while providing extremely precise and pertinent responses. Furthermore, this system adjusts interactions based on the knowledge base of each company it is speaking with; as a result, in addition to being accurate, the responses are appropriately tailored to the user's particular situation. In general, this strategy enables companies to provide more intelligent and skilled customer service, better meet the needs of their clients, and increase customer happiness. As a result, the company will provide a smooth, knowledgeable, and user-responsive intelligent system of customer care.

Keywords— Large language model, retrieval-augmented generation, natural language processing, information retrieval.

I.Introduction

As businesses increasingly adopt artificial intelligence (AI) to enhance customer service, traditional chatbots often fail to meet the dynamic needs of users. These conventional systems are typically limited by their pre-programmed responses, making them less effective at handling complex or domain-specific queries. However, with the advent of Retrieval-Augmented Generation (RAG), a more advanced and adaptable solution is emerging [1, 2, 7]. RAG combines the generative capabilities of large language models (LLMs) with retrieval mechanisms that access external, domain-specific knowledge bases, significantly improving the accuracy and relevance of responses.

This paper introduces an AI-driven conversational agent designed to be integrated into business websites as a customer support tool. By leveraging the RAG framework, the system builds customized knowledge bases from documents and URLs provided by businesses, enabling it to deliver accurate and contextually appropriate responses. Integrating the power of LLMs with a retrieval system allows for addressing complex customer queries, offering a scalable, intelligent solution for modern enterprises. This approach not only enhances the customer experience but also reduces the workload on human support teams, making it a cost-effective method for managing diverse customer interactions.

II.Related Work

It is divided into three sub-sections, though: 2.1 uses of Retrieval-Augmented Generation, discussion of 2.2 effectiveness of RAG in improving response quality, and 2.3 various applications of chatbots in different industries.

A.Applications of Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation is the next big thing. Patrick Lewis et al [1]. proposed it back in 2020 and is the combination of retrieval and generative capabilities to enhance accuracy and relatedness of responses generated. Hybrid, it enables models to be beneficial for vast external knowledge bases that may improve their ability to answer domain-specific queries [3].

DOI: https://doi.org/10.5281/zenodo.18264443

RAG has proved to be a very useful application beyond customer support outside the scope of just customer support as document retrieval, summarization, and question answering applications across several domains such as healthcare and finance.

B.Effectiveness of RAG in Improving the Quality of Response

Now, many works have established the effectiveness of Retrieval-Augmented Generation in improving the quality of response. For example, by Vladimir Karpukhin et al. in 2020, it has already proven that an improvement in the retrieval and generative capabilities can significantly enhance the effectiveness of question-answering systems [2]. Indeed, retrieving contextually relevant information is really more about generating precise answers, and this notion is a direct dictate of conversational agent design.

C.Multifaceted Applications of Chatbots in industrial

Chatbots are predominantly utilized as an interface for FAQ on the websites (Abu Shawar and Atwell, 2015) [3]. Now, they are also employed in mental health care thereby aiding a user in therapy, training, and screening for depressive and autistic conditions (Abd-Alrazaq et al., 2019) [4]. Improved purchases of products and customer service enhancement, along with the prompt response to the users, are some roles of the chatbots (Albayrak et al., 2018) [5]. They respond to the changing expectations of customers in a positive manner and enhance profitability in the banking industry. The following examples demonstrate the impact of such interfaces on the usage of generative language models within customer interaction processes.

III.Methodolgy

This section describes the methodology that is adopted while input data processing with the resolution of customer queries using the RAG approach.

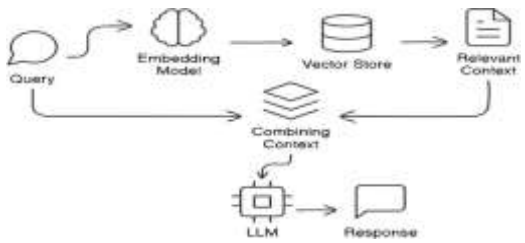

Fig. 1. Workflow of Retrieval-Augmented Generation

This methodology has a number of key features that work in tandem to efficiently process customer queries. When a customer submits a query, the system's processing begins with converting the input into a query embedding using an advanced embedding model. This transformation is critical in allowing the system to understand the semantic meaning of the query in numerical format. The system can compare the query to other stored embeddings in the database by representing the query as a vector, therefore, allowing for a more subtle comprehension of the inquiry. The representation of the query allows for the numerical capture of those attributes that identify intent and key terms for retrieval of more relevant information in further processes.

A.Proposed Systems

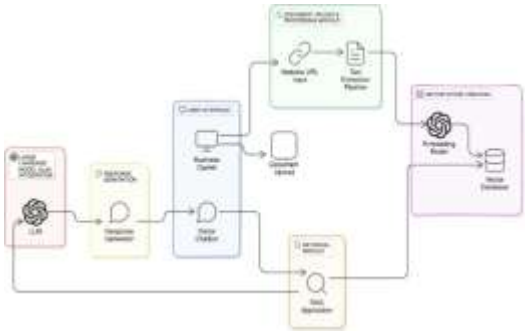

Fig. 2. Diagram of proposed system

1)The architecture is designed to facilitate efficient interaction between customers and the system while leveraging the Retrieval-Augmented Generation (RAG) framework. This section breaks down the core components of the system architecture, highlighting how each module contributes to the overall functionality.

2)Data Ingestion: Initially, the data ingestion process involves the ingestion of documents and URLs streamed directly from businesses for transformation into a structured format that can be properly utilized in customer interactions. In this stage, the accumulation of different data sources-product descriptions, user manuals, FAQs, and similar web pages-can be assured of a profound knowledgebas

DOI: https://doi.org/10.5281/zenodo.18264443

Once it enters the body, this information then gets embedded using sophisticated techniques such as Sentence Transformers or Universal Sentence Encoder. These embeddings transform the text into numerical representations, carrying semantic meaning in the content, hence allowing the system to capture the context behind information and the relationships involved within it.

This transformation is pivotal since it will enable the system to index the data in an efficient manner; thus, quick retrieval and accurate matching during query processing will be achieved. Embeddings capture not only textual information but all those nuances and subtleties associated with understanding user inquires. When such tools are taken, like Langchain, then this architecture ensures the entire process related to ingestion and indexing is streamlined to set up the screen for efficient processing of queries and to generate responses, finally enhancing overall performance in the customer support system.

Vector Store : Right at the heart of the system is what is called the vector store, forming the repository of knowledge. It stores the embeddings in the documents and URLs uploaded and organizes them appropriately. As such, it uses a vector database like Qdrant, Chroma, LanceDB, Vespa and Milvus to index and retrieve high-dimensional data at blazing speeds [12].

In this process, to obtain the documents, the business serves to create embeddings from it, capturing its semantic content. This vector store maintains it in a structured way so that
similarity searches can be carried out efficiently. This design ensures the system can glance through the information pretty quickly based on the queries of the customers, and that's critical to return responses to customers in time, with an accuracy that is precise.

3)Query Processing: When a query from a customer is input to the system, it will be processed in determining what information best corresponds to the one retrieved from the vector store. The query is also represented as an embedding, and this enables the system to perform a semantic match against the stored embeddings. It will then query for similarity, normally through cosine similarity or other distance measures, for any relevant documents or snippets up to the top-k result. The retrieval module facilitates easy and fast access to relevant information, important to provide timely responses.

4)Retrieval Module: The retrieval module forms an integral part of the system's high-level architecture and is involved in the process of finding pertinent information when the user submits a query. With the query received from the customer, the system translates this into an embedding, much like it does with the documents that are being processed [6, 7]. The system then uses a similarity search mechanism to compare the embeddings available in the vector store against the input embedding.

The retrieval module uses distance metrics such as cosine similarity or Euclidean distance for returning the top-k most relevant documents or snippets. At this point, the system can efficiently filter out the copious amounts of information to only relevant content, ready for utilization in the process of response generation.

5)Response Generation*:* Once the retrieval module determines the information as relevant, it passes this to an LLM to produce the final response. The LLM is built on an idea of forming coherent answers; the pre-trained knowledge is synthesized with retrieved data so that both are relevant to the context.

The state-of-the-art LLMs, GPT-3, or BERT, for example, can understand and write text almost indistinguishably from a human [11]. Therefore, with the retrieval module coupled with the LLM, the system successfully adapts to utilize the superior precision provided by the knowledge base as well as the generative capability imparted by the language model in making for a very effective conversational agent.

6)Fine-Tuning: The system goes through a full fine-tuning process optimized specifically for the domain and context within which it will serve the business [13]. This is essential because a large language model is considered not just to be operating at a general level but knowing the industry-specific nuances for which it is trained to respond. Fine-tuning may incorporate training the LLM on a domain-specific curated dataset that would include domain-specific language, terminology, and scenarios related to the business sector which may be healthcare, finance, or retail, or any other field where the responses had to be customized.

DOI: https://doi.org/10.5281/zenodo.18264443

In this way, by refining the model through all these processes, the system will achieve higher accuracy and relevance in responses. The system can consequently be so effective in each unique business need, ensuring a timely and contextually appropriate answer to the needs of the customers. Gradually, this fine-tuning does not only contribute towards enriching the overall quality of customer interaction but also results in improved customer satisfaction and loyalty since users feel understood and valued concerning their engagement with a business.

IV.Result and Discussion

We compare the performance of different language models (GPT-4o-mini, Llama 3-1 Instruct 8B, Claude 3.5 Sonnet, GPT- 3.5 Turbo, Mistral 7B Instruct) integrated with the Qdrant Vector Database. The evaluation considers key factors like retrieval accuracy, response quality, latency, scalability, and cost to determine the best model for specific business needs.

A. **Communication effectiveness**

The LMSys Chatbot Arena ELO Score [15] measures communication effectiveness, especially how well each model generates coherent, relevant, and technically sound responses.

GPT-4o-mini ranks highest in communication quality, followed closely by Claude 3.5 Sonnet (June). This means that GPT-4o-mini is particularly effective for applications requiring strong communication capabilities, such as research writing where coherence and technical accuracy are crucial. Llama 3-1 Instruction 8B and GPT-3.5 Turbo are pretty good as well and good for general technical text. Mistral 7B has decent performance but might not be ideal for sensitive technical documents.

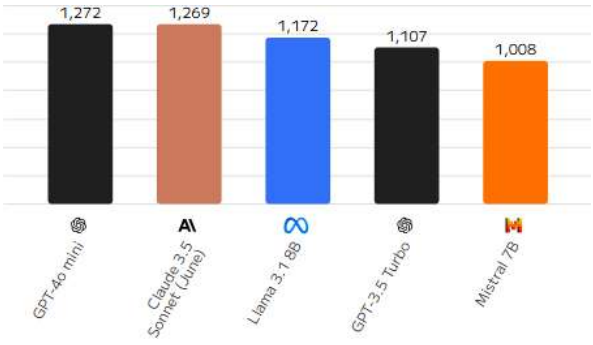

Fig. 3. Communication (LMSys Chatbot Arena ELO Score)

B. **Quality vs. Price Evaluation**

This analysis shows the quality versus price: Quality vs. Price shows the relative trade-offs each model balances in quality of output and operating cost in USD per million tokens [15]. It's particularly helpful to users concerned about costs and want very high-quality output.

Fig. 4. Quality vs Price

GPT-4o-mini is the least expensive to obtain per million tokens at current prices. This is the most feasible option for technical writing in budget and high-volume applications, where cost efficiency is a concern. Though Claude 3.5 Sonnet (June) makes the best quality, it is too expensive and, hence, may only be valuable for niche high-quality need. Llama 3-1 Instruct 8B and GPT-3.5 Turbo are generally a middle-of-the- road mix between cost and quality and should be used in general applications for writing. Mistral 7B might be adequate for less critical and detailed work where quality could be less stringent.

C.Latency Analysis

Latency is the delay experienced by a model before outputting the first token from receiving an API request. Latency is very crucial for applications involving real-time delivery such as customer support because every moment counts, as minimal delay is critical for smooth operation. Users would then enjoy nearly instant response time.

Fig. 5. Latency chart

DOI: https://doi.org/10.5281/zenodo.18264443

As can be seen from the latency graph, GPT-4o mini is well-performing at 0.46 seconds, thereby putting it in a competitive position [15]. Though Mistral 7B Instruct has the lowest latency of 0.36 seconds, this is countered by other performance issues that have been discussed below. Contrasted with this, the other model, Claude 3.5 Sonnet which was released in June '24, shows the highest latency at 0.93 seconds, thus becomes less suitable for those particular scenarios where latency is also critical. Models such as GPT-3.5 Turbo and Llama 3.1 Instruct 8B, display similar latency at 0.47 and 0.39
respectively, do not perform uniformly over some
of the other key metrics.
GPT-4o mini in customer support systems offers a good balance of speed and reliability for sustaining user engagement with rapid response times. Mistral 7B Instruct may be too fast but also lags elsewhere in performance. For example, the application of which is discussed later in this paper has very poor latency. Although it may be very accurate for generalized tasks, such as completing a paragraph that a user begins, this does not make it very useful for real-time applications.

D.Discussion

GPT-4o mini is still the best model for real-time conversational systems if we consider the three critical factors discussed here: latency, total response time, and output speed vs price. While Llama 3.1 Instruct 8B outpaces others in terms of output speed, the improved cost that would be incurred might not have a significant gain in its performance regarding most scenarios. Mistral 7B is economically viable in the output but loses consistency of overall response time, which reduces the reliability towards time-sensitive use cases. In comparison, Claude 3.5 Sonnet (June '24), although fully capable, is the slowest and most expensive model, thus least favorable to deploy in cost-conscious real-time applications.
GPT-4o mini offers the best trade-off between performance and cost, which makes it the best option for the target customer support system.

V.Conclusion

GPT-4o mini stands out to be the best solution because it balances out the critical metrics of latency, response time, and cost performance. Fast, reliable, and cost-efficient responses are very critical for high levels of user satisfaction and engagement in customer support systems.
GPT-4o mini meets these requirements because of its competitive latency and consistent total response times.
This means that real-time interactions will be kept with the expectations of the customer. It
also comes in at a reasonable price for large-scale deployments where controlling cost is important. While other models-such as Mistral 7B Instruct, Llama 3.1 Instruct 8B, or others might have some performance benefits for either speed or cost, each of them has some inherent drawbacks-so whether in terms of an inconsistent performance or a significantly higher operational cost-they pose limitations on suitability for continuously high-volume customer interactions. GPT-4o mini strikes the absolute balance between speed, reliability, and affordability, and among all AI models, makes it the best one to use at scale for efficiency improvements and responsiveness of customer-support systems.

Reference

1.Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, & Douwe Kiela. (2021). Retrieval- Augmented Generation for Knowledge-Intensive NLP Tasks. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.)
CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg
(2016).
2.Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6769–6781, Online. Association for Computational Linguistics.
3.Abu Shawar, B. and Atwell, E. (2015) ALICE Chatbot: Trials and Outputs. Computacion y Sistemas, 19, 625-632.
4.Abd-Alrazaq AA, Alajlani M, Alalwan AA, Bewick BM, Gardner P, Househ M. An overview of the features of chatbots in mental health: A scoping review. Int J Med Inform. 2019 Dec;132:103978.doi: 10.1016/j.ijmedinf.2019.103978. Epub 2019 Sep 25. PMID: 31622850.

DOI: https://doi.org/10.5281/zenodo.18264443

5.Albayrak, T., Moutinho, L., & Herstein, R. (2011). The influence of skepticism on green purchase behavior. International Journal of Business and Social Science, 2(13), 1-10.
6.Zhang, Y., Sun, S., Galley, M., Chen, Y. C., Brockett, C., Gao, X., & Dolan, B. (2020). DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. ACL.
7.Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). REALM: Retrieval-Augmented Language Model Pre-Training. Proceedings of the 37th International Conference on Machine Learning (ICML).
8.Vaidyam, A. N., Wisniewski, H., Halamka, J. D., Kashavan, M. S., & Torous, J. B. (2019). Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape. JMIR Mental Health, 6(11), e15001.
9.Boudra, A., & Ferreira, A. (2021). The Impact of Artificial Intelligence in the Banking Industry: Evidence from Chatbot Implementation. Journal of Financial Transformation, 53, 47-54.
10.Kumar, A., & Tripathi, R. C. (2020). Chatbot Adoption in E-commerce: A Review of Literature and Future Research Directions. Journal of Retailing and Consumer Services, 52, 101876.
11.Nogueira, R., Lin, J., & Epure, E. V. (2020). Passage Re-ranking with BERT. arXiv preprint arXiv:1901.04085.
12.Johnson, J., Douze, M., & Jegou, H. (2019). Billion-scale Similarity Search with GPUs. IEEE Transactions on Big Data.
13.Sun, T., Qin, Y., Hu, Q., & Liu, T. (2022). Improving the Performance of Fine-Tuning Large Language Models in Specialized Domains. arXiv preprint arXiv:2201.08231.
14.Thakur, N., Reimers, N., & Gurevych, I. (2021). BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. NeurIPS.
15.Artificial Analysis. (n.d.). LLM Performance Leaderboard and AI Model Comparisons. Retrieved October 27, 2024, from https://artificialanalysis.ai

DOI: https://doi.org/10.5281/zenodo.18264443