

Comparative Analysis of Industrial SAP Chatbots: RAG-LLM and Cloud-based Approaches

Fatimah Alali¹, Abul Bashar², Haya Aldawsari¹, and Sajjad Mahmood¹

¹College of Computer Science and Information Technology, King Fahd University of Petroleum and Minerals

²College of Computer Engineering and Science, Prince Mohammad Bin Fahd University

{g202423180, g202424400, smahmood}@kfupm.edu.sa, abashar@pmu.edu.sa

Abstract

RAG (Retrieval Augmented Generation) is the process of integrating the output generated from LLM (Large Language Model) and embedding it with external knowledge, without the need to re-train the model. Many studies have been conducted to evaluate the integration of RAG with LLMs. However, they have rarely compared it with cloud-based chatbots that address the industry's requirements, which this research addressed. A systematic literature review was conducted, followed by interviews with industry participants to gather chatbot requirements. Based on the interview analysis, the RAG-LLM chatbot was implemented and a comparative analysis between RAG-LLM and cloud-based (Microsoft Azure) chatbots was performed. The experimental evaluation showed that RAG-chatbot achieved high performance: expert evaluation (96% vs. 80% Azure), LLM-Judgment (94%), BLEU (67.2%), MRR (73.33%), Fuzzy Match (91.33%), Exact Match (67%).

1. Introduction

The cloud-based chatbots are becoming more widely available in the business environment, but their usage is still limited, especially for those sectors that place much importance on data privacy and domain-specific tasks. This research was motivated by the real need identified in the oil& gas industry, where SAP (Systems, Applications, and Products in data processing) users frequently raised repetitive support tickets due to difficulties in navigating complex SAP documentation. The organization required a solution that could operate locally to ensure data privacy, retrieve exact answers from internal manuals, and reduce workload. Retrieval Augmented Generation offers an alternative by enabling businesses to ground answers in their texts without model retraining and full data control.

Based on the study of the literature in this domain, it was found that there is a notable increasing interest in RAG implementation, however, there is a visible gap that most of the research either discusses RAG's architecture or compares it with base LLMs, but few analyze how RAG-based chatbots compared to cloud-based toolkits such as Microsoft Azure in real-world, industry-related applications. To address this gap, an interview was conducted with the industry employees to gather requirements to develop a custom chatbot that suits industrial needs. The objective was to develop a tool that performs better in terms of answer quality, supported with a privacy mechanism, specifically to outperform the existing solution, Microsoft Azure.

The three main operations of any RAG system are indexing, retrieval, and generation. The document will be preprocessed and indexed by transferring it to a vector representation using dense embeddings and storing it in a vector database. The documents are split based on characters, sections, or semantic units. When a user submits a query, it will fetch the top-k documents from the vector database by using a similarity search such as cosine similarity. The final step is the generator, which is usually a pre-trained LLM, that will provide the final retrieved response and integrate it with the LLM. The RAG pipeline is illustrated in Fig 1

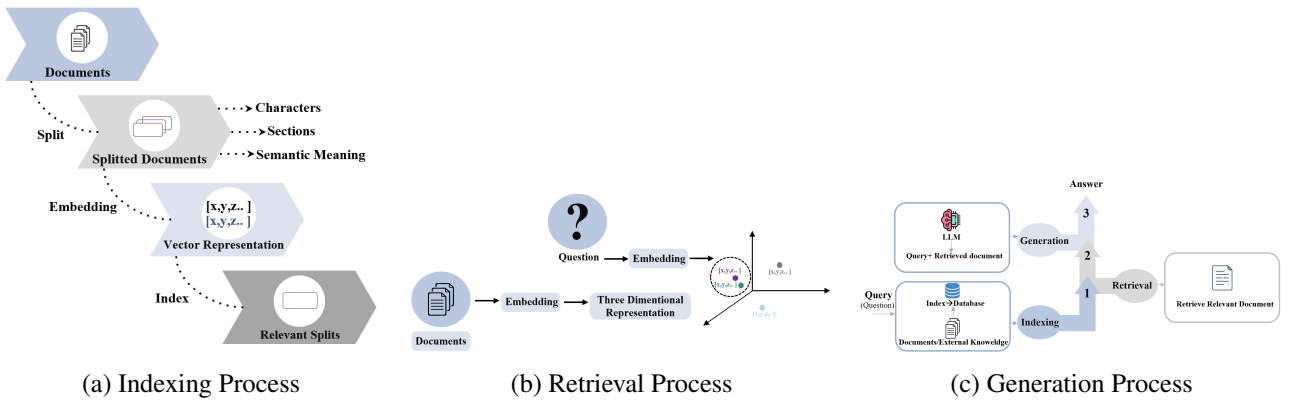


Figure 1: RAG Pipeline: indexing, retrieval, and generation Processes

Table 1: Summary of Key Metrics Findings Across Different Domains

Domain	Key Findings	Metrics Used	Toolkits/Platforms	References
Healthcare	Accuracy and relevance improvement (95% context recall)	SUS, BLEU, ROUGE	GLM-4, DIAG-GPT	[14]
Education	Student Satisfaction Rate: 93.3%	SUS, BLEU	GPT-3.5, GPT-4	[18]
Supply Chain	Operational agility and efficiency	MRR	Azure AI Search	[15]
E-Commerce	User experience satisfaction and reduced response times	MRR, Recall	Falcon-RW-1B, GPT-3.5	[4]

2. Related Work

With the large amount of data in organizations and the challenges associated with manual searching, RAG technology is emerging as a promising solution. However, there is a noticeable lack of studies that address specific industry requirements, which was identified as a research gap. To address this, a systematic literature review was conducted focusing on the implementation of RAG with LLM to develop customized private chatbots suited to industry needs. This review was guided by specific research questions aimed at understanding how RAG-based chatbots can effectively meet organizational requirements, which are: RQ1: How does an integrated LLM with RAG-based chatbot outperform cloud-based toolkits? RQ1: What are the metrics that assess the performance? RQ2: What toolkit will be used in this study? RQ3: How does the RAG-chatbot facilitate industry work? RQ4: Why does the industry not rely on LLM only to load their documents? RQ5: What are industry professionals' views on developing secure, customized chatbots? RQ6: What are the practitioners' requirements to implement an effective RAG chatbot?

This paper categorized the findings based on different metrics to highlight the effectiveness of RAG in different domains. In most studies collected, the RAG outperforms the base LLM model, showing better accuracy, relevance, and contextual understanding. To illustrate, in [9], the improvement of relevance was 13-25% compared to the base LLM. In addition, RAG performs very well in domain-specific applications such as healthcare, education, supply chain management and e-commerce. Table 1 summarizes key metrics findings split by domain.

The studies [14], [18], [15], [17], [8], [12] are showing the RAG chatbot applications in healthcare illustrating improvement after implementing RAG in user satisfaction, response accuracy and relevance, where specifically [18] achieves a 95% context recall and 93.73% faithfulness. Education is another field in which many chatbots are implemented to assist students. In the studies [7], [1] RAG showed high performance, student satisfaction, and accuracy improvement. For instance, in [1], the satisfaction rate was 93.3%. In the supply chain domain, operational agility improved, increased efficiency and reduced disruptions [19]. The research presented in [2], [4] highlights the impact of

using RAG-chatbots in e-Commerce and Customer Support, focusing on two points: prompt engineering and real-time data retrieval requirements. The work in [2], demonstrates that a simple query has a direct impact on retrieval improvement, and [4], shows the RAG performance in terms of user satisfaction and response time reduction.

The key findings reported in the studies are: First, RAG improves response quality, reducing hallucinations and enhancing domain-specific performance. Second, the different LLMs and tools commonly used are: Llama 3.1 and Phi-3 [9], GLM-4 [16], Coze Platform [5], Azure AI Search [3], and retrieval frameworks such as LangChain and vector-based retrieval for optimizing retrieval pipelines. The common metrics found to measure RAG performance are: Mean Reciprocal Rank [2], BLEU and ROUGE Scores [1], System Usability Scale (SUS) [18] and other metrics such as accuracy, Relevance score, Context recall, and Faithfulness.

The studies addressed challenges and limitations such as latency, small dataset, computational cost, and security. In [6], the work revealed that high latency occurred in the RAG system, which lowers the performance. The problem of hallucinations is common in the RAG system, and [10] proposed a new RAG methodology, called ReRAG, to reduce hallucinations by utilizing fragment size and feedback loops. Most of the collected papers highlighted the need for large datasets to enhance the result [19], [7], [3], [2], [18], [15], [1], [4]. For the security metrics, it was tightly addressed, and some studies do not highlight it. However, [6], shows the benefit of using local RAG in security improvement, and [3] focuses on OpenAI's security compliance. In [11], [13], [4], focus is on the need to have a secure framework for data protection and system reliability.

In summary, this paper applied a narrative synthesis to the collected studies, and it was found that RAG is a better choice for domain-specific applications. It outperformed the base LLM in terms of accuracy, relevance, and contextual understanding. Also, RAG has a significant impact on facilitating industry work, especially in real-time data retrieval. However, there are limitations such as computation cost, latency, small datasets, and hallucination. These studies were also limited to focusing on industry requirements, privacy, and comparing the RAG with a cloud-based solution that was available but had limited use in the industry. The deep understanding of industry LLM adoption and requirements to develop a chatbot remains unclear, which requires deep analysis in industrial settings.

3. Methodology

This research follows a qualitative and quantitative approach where interviews are conducted and a system is developed. The first step was conducting a semi-structured interview with nine industry professionals from different domains, ensuring generalization, collecting their insights, requirements, and concerns about using LLM chatbots. The domains were Information Technology, Human Resources, Engineering, Planning, Safety, and Interior Design across seven industries. The feedback provides a deep understanding of enterprise settings, which helps to develop a chatbot based on industry considerations. The interview consists of 13 piloted questions related to RQs. The development of the chatbot was done after the interview analysis to meet the industry requirements. The domain of the developed chatbot is for the SAP system, where it is widely used in the industry, which includes a large documentation requiring automation. Microsoft Azure chatbot is involved in the study as a comparison tool versus the RAG chatbot. As investigated, the industries have wide access to Microsoft tools, and the availability of Chatbot Azure within the kits, but they did not utilize it because of privacy concerns related to cloud-based solutions and the inability to retrieve the expected answer. Six evaluation metrics were used to evaluate the developed RAG-Chatbot (LLM-Judgment, BLEU score, MRR score, Fuzzy Match, Exact Match, and expert evaluation), and one metric was used for the comparison with the cloud-based Microsoft Azure chatbot (expert evaluation).

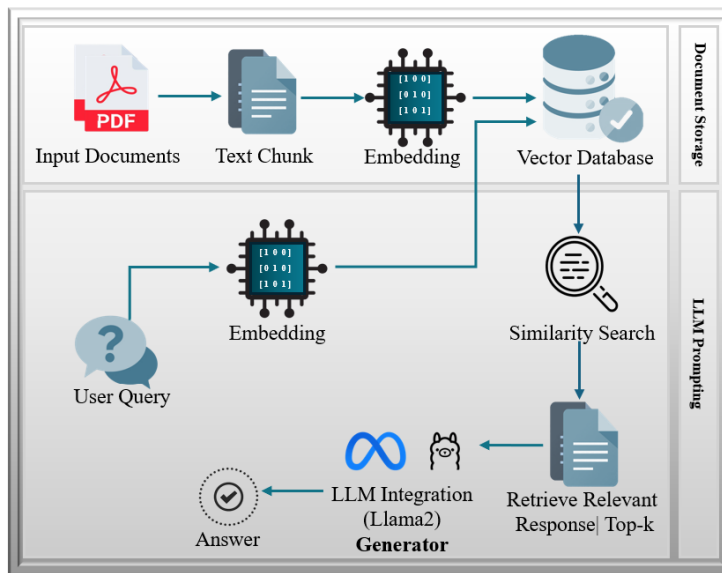


Figure 2: High-level Architecture of RAG-Chatbot

3.1. Interview Analysis

This section aims to analyze the interview responses to build the chatbot and answer related research questions. The analysis is based on four themes: Privacy issues with the data—among all sectors, they focused on data privacy restrictions from companies to use AI-based tools for loading the documents due to industry policy and concern. Difficulties in retrieving and searching for the information—the participants demonstrate frustration in the searching process, which was time-consuming, and it affects key performance indicators, especially in the IT department, where the immediate answer to the user is required. Strong desire for a custom chatbot—the participants acknowledged that it would speed up the work rhythm by eliminating the searching time and relying on human support, which delays the process. Features required to develop the chatbot—the participants have common desired features, which are privacy, accurate answers, operating based on the organizational data, and integrating it with the internal system of the company.

3.2. Experimental Setup & Procedure

After performing and analyzing the interview responses, the settings to implement both chatbots were addressed as follows:

- **Implementing RAG-chatbot**

Model & Libraries: The RAG system is implemented using Llama2, which is known for high-quality response generation. LangChain is the library used to connect LLM with the external database. For example, data that will be retrieved from PDFs.

Embedding model: OllamaEmbeddings with nomic-embed-text

Database: The database used is a vector database called Chroma DB. It provides a fast, high-quality response and is easy to integrate with LangChain.

Documents: PDF files for SAP user manual, error resolution, transaction codes, and FAQs.

Dataset: The dataset is based on an actual enterprise scenario, user manuals, and FAQs (Frequently Asked Questions) for the SAP system in the oil and gas industry, where all the required documents for SAP to be used in the company were loaded. It varies from complex documents where detailed technical guidance is required to simple documents that include a list of email support. The format of the documents loaded is PDF, and the processing is done using LangChain and Chroma.

Front-end&Back-end Development: The framework is Flask combined with JavaScript

and HTML. The back-end is Python 3.12.

Run time: Locally using Ollama to achieve the privacy requirement.

- **Procedure**

The user will interact with the developed system through an interactive chatbot interface(Flask UI). Once the query is submitted, in the backend, it will be embedded using Ollama Embeddings with the embed-text model, the retrieval specified for top k=5 using vector similarity (cosine similarity). The stored query in Chroma DB will be compared with the embedding of the generating query. A template used for prompt construction, so the retrieved chunks will be fed into it. The retrieved prompt feeds into LLM (Llama2) via Ollama, which will add quality to the response. The response will be retrieved with metadata (source of the document), to ensure that the answer only comes from the data that exists in the document, because of the customization required in the chatbot. Fig 2 illustrates the high-level architecture.

- **Azure Chatbot Creation**

Building an interactive chatbot with Microsoft Azure services is a simple process where a non-Technical user can create a fully functional chatbot with their knowledge base data through the Microsoft Azure portal. The first step is AI model selection; the model chosen in this experiment is OpenAI. Two models will be created, the first one is GPT-35-Turbo, and the second one is for vector searching purposes, called text-embedding-ada-002. To train the model on the required data, all the files were uploaded. Next, selecting the searching mechanism where its two types: keywords and semantic. Lastly, the chatbot deployment will be as a Web App deployment and the user can start to chat against the deployed chatbot.

Table 2: Comparison between RAG-based LLM Chatbot and Microsoft Azure Chatbot

Aspect	RAG with LLM	Microsoft Azure
Privacy	Private Environment where the data will not be breached outside, the chatbot runs locally.	The data goes through an API, which is cloud-based, so the data can be breached.
Ease of Development	Requires programming skills with RAG implementation and LLMs knowledge, which makes it more complex than using the built-in chatbot kit.	The availability of the Microsoft SDK and a user-friendly interface facilitates its use.
Evaluation	Allow a variety of use case testing, such as semantic and lexical evaluation using different metrics.	Cannot reach the underlying phase. The testing is only via the interface (expert evaluation).
Performance	High-quality and detailed answer. The performance can be slightly decreased depending on the time spent retrieving data.	High performance in general, but the answer is in a general form.
Use Cases	Best for domain-specific tasks, when responses are based on specific documents or databases, and details are crucial factors of the response	Best for business organizations, especially organizations using Microsoft Services.

4. Results & Discussion

This section demonstrates the results of RAG Chatbot implementation and Microsoft Azure cloud-based chatbot. The analysis will be provided based on research questions and the inference from the interview.

- **Performance Evaluation**

Both chatbots, cloud and RAG-based, show a high performance in terms of relevance and accurate response, but each one has its use case. According to the expert evaluation, when the queries require detailed guidance answers with deep information, the RAG chatbot demonstrates a higher accuracy performance. It is also suitable for data-intensive tasks and specialized domains where the answer is required to be specific to the organization based on its documents. In addition, the RAG chatbot performs well when the data is retrieved from multiple resources, such as data scattered in different PDF files. Microsoft Azure chatbot is working at a higher

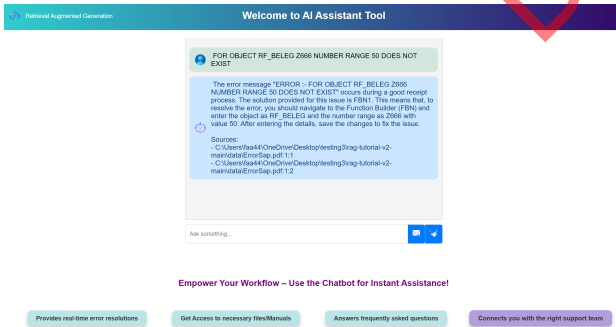
speed, and RAG is showing a latency in some query retrieval, especially the complex one. However, it has a lower accuracy rate compared to the RAG chatbot due to general answer retrieval. It is suitable for cases requiring fast answers, not detailed ones, such as customer support. Table 2 compares the different aspects of the chatbot’s performance. For better result demonstration and comparison, 50 sample queries with the same input were fed into both chatbots. The result is based on expert evaluation, where the successful cases are counted and divided by the sample size, and the result is presented in Table 3. Fig 3a & Fig 3b show one of the scenarios out of 50 where the user asks the chatbot a query requesting to resolve the error in the SAP system. In this case, both chatbots deliver the answer, but the RAG-LLM chatbot provides a more detailed answer that the user can resolve the error easily by following the chatbot’s instructions, while the Microsoft Azure chatbot provides short answers, which the user may need human expert knowledge and support. As a result, RAG chatbot is suitable for the SAP application since it provides a better guide in the process, with details that will eliminate the need to request extra support, such as the IT department in which will save department and user time, and thus increase the efficiency.

• **Result based on Evaluation Metrics**

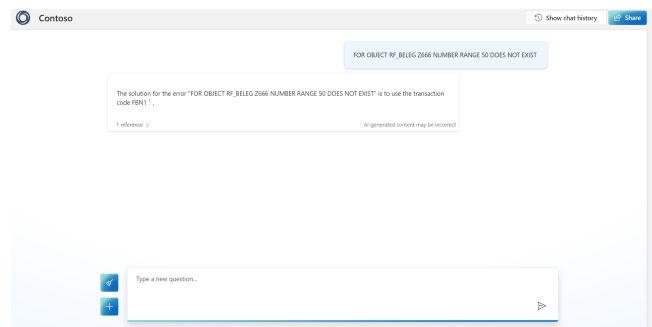
To accurately assess the performance of the RAG-LLM chatbot, six evaluation metrics were implemented, each with its own evaluation purpose. For the Microsoft Azure chatbot, the valid metric is limited to accuracy based on expert evaluation, as it cannot be customized to implement other metrics. Table 4 discusses each metric’s result.

Table 3: System Expert Evaluation

Chatbot	No. of Tests	Functional Responses	Accuracy
RAG with LLM	50	48	96%
Microsoft Azure	50	40	80%



(a) Answer Generated by RAG-LLM Chatbot



(b) Answer Generated by Microsoft Azure Chatbot

Figure 3: Comparison of Chatbot Answers from RAG-LLM and Microsoft Azure

Based on the metrics evaluation, the results indicate that the RAG-LLM chatbot generates high-quality, reliable answers that are relevant to the SAP documents, which are suitable for industries. The answer generated matched the expectation in structure, and meaning can be inferred from the Fuzzy score and Expert Evaluation. The highly relevant context retrieval is reflected by the MRR score. In some cases, exact words are required, and the Exact Match metric extracted these cases correctly. Lastly, the LLM judgment of answer correctness verifies that RAG-LLM has strong semantic correctness.

Table 4: Evaluation Metrics Result, Analysis, and Working Mechanism

Metric	Result	Analysis	Working Mechanism
LLM Automatic Judgment	94%	In most cases, the model returns true, indicating a high semantic performance.	Llama2 checks the generated answer against the expected output by returning a binary decision.
BLEU Score	67.23%	Strong lexical performance, especially for short answers such as transaction code retrieval as "ME53N".	The generated and expected answers will be measured to evaluate how close they are.
MRR	73.33%	Strong context retrieval where the documents contain the answer ranked in the top 1 or top 2 position.	Assess if the targeted document ranked high, best case to be at the first rank. Calculated as: $MRR = \frac{1}{N} \sum \frac{1}{Rank\ of\ correct\ docs}$
Fuzzy Match	91.33%	High performance for paraphrased text.	It uses fuzzy string similarity, which allows for similarity measurement even if words are different or paraphrased. (fuzzy-wuzzy.partial_ratio())/100
Expert Evaluation	RAG-LLM: 96% Microsoft Azure: 80%	This result indicates a high performance for both chatbots, where RAG-LLM outperformed Microsoft Azure in retrieving detailed, relevant answers that are suitable for domain-specific tasks.	Test the system by providing queries and using human evaluation to evaluate successful and failed cases. The accuracy based on human evaluation is calculated as: $Accuracy = \left(\frac{Number\ of\ correct\ precise\ responses}{Sample\ Size} \right) \times 100$
Exact Match	67%	This metric only measures the exact match of words where the test has mixed cases, resulting in 67%, which is working well with the matching.	Check if the exact word appears in the response, return true or false. It is critical when an exact answer is required, such as the SAP transaction code.

5. Limitations

While the paper shows the practical development and evaluation of an industry SAP chatbot using RAG and Microsoft Azure chatbots, there are some limitations, which are a small dataset & participants, limited Microsoft Azure chatbot evaluation. The dataset used is based on the real oil & gas industry SAP documentation; however, it can be increased by embedding extra SAP modules, enhancing the chatbot to answer more queries related to diverse SAP modules. The interview participants were only nine from different sectors, which will provide more insights into the chatbots development and discovering a wide range of organizational contexts if the participants were increased. The evaluation of Azure chatbot is limited to expert evaluation due to restrictions on reaching the underlying infrastructure of the Azure chatbot, resulting in minimizing the depth of comparison in terms of evaluation metrics. The prototype was developed and tested by expert evaluation and evaluation metrics; however, to improve the analysis, the chatbot needs to be tested with the interview participants for feedback collection and improvement.

6. Conclusion & Future Work

This paper developed and evaluated the RAG-Chatbot based on the industry environment and compared it with Microsoft Azure chatbot in terms of answer correctness, privacy, and overall response quality. Interviews were conducted with the professionals to explore industry requirements for developing a custom chatbot. The participants' responses were analyzed based on four themes. The main features highly required by the industries were privacy, accuracy, and sourced data. The chatbot was developed by integrating RAG with LLM using Ollama to achieve privacy, where the chatbot was hosted locally, providing accurate and high-quality answers using Llama2, and for sourced data, all retrieved answers were indexed with the source of the document. The RAG chatbot was tested using six evaluation metrics, which are expert evaluation (96% vs. 80% Azure), LLM-Judgment (94%), BLEU (67.2%), MRR (73.33%), Fuzzy (91.33%), Exact Match (67%). The result showed that the RAG chatbot provided accurate and high-quality responses for various queries, making it more suitable for the SAP domain than Microsoft Azure, which provides more general answers. Microsoft Azure Chatbot has simple and user-friendly kits that allow users with no technical background to build a custom chat-

bot by simply loading the documents into the chatbot, while RAG chatbot requires a technical background to develop. Both approaches achieved high accuracy based on expert evaluation, although each has its best case for implementation. The research directions for the future would include the integration of Multimodal Retrieval-Augmented Generation since the chatbot is designed for SAP applications, which often include user guide screenshots, and workflow figures.

7. Acknowledgement

The authors would like to express their thanks to King Fahd University of Petroleum and Minerals for providing the required research support.

References

- [1] Carlos Alario-Hoyos, Rebiha Kemcha, Carlos Delgado Kloos, Patricia Callejo, Iria Estévez-Ayres, David Santín-Cristóbal, Francisco Cruz-Argudo, and José Luis López-Sánchez. Tailoring your code companion: Leveraging llms and rag to develop a chatbot to support students in a programming course. In *2024 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 1–8. IEEE, 2024.
- [2] Data Analytics and Dishant Sukhwai. Retrieval augmented generation: An evaluation of rag-based chatbot for customer support. *Retrieval Augmented Generation: An Evaluation of RAG-based Chatbot for Customer Support*, 2024.
- [3] Henrik Andersson. Retrieval-augmented generation with azure open ai, 2024.
- [4] J Benita, Kosireddy Vivek Charan Tej, E Vinay Kumar, G Venkata Subbarao, and CH Venkatesh. Implementation of retrieval-augmented generation (rag) in chatbot systems for enhanced real-time customer support in e-commerce. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 1381–1388. IEEE, 2024.
- [5] Chen-Chi Chang, Han-Pi Chang, and Hung-Shin Lee. Leveraging retrieval-augmented generation for culturally inclusive hakka chatbots: Design insights and user perceptions. In *2024 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE)*, pages 1–6. IEEE, 2024.
- [6] Andre Chen and Sieu Tran. Supercharging document composition with generative ai: A secure, custom retrieval-augmented generation approach. In *2024 11th IEEE Swiss Conference on Data Science (SDS)*, pages 123–130. IEEE, 2024.
- [7] Sagnik Dakshit. Faculty perspectives on the potential of rag in computer science higher education. In *Proceedings of the 25th Annual Conference on Information Technology Education*, pages 19–24, 2024.
- [8] A. Fink, J. Nattenmüller, S. Rau, et al. Retrieval-augmented generation improves precision and trust of a gpt-4 model for emergency radiology diagnosis and classification: a proof-of-concept study. In *Proceedings of European Radiology*. Springer, 2025.
- [9] Khant Ko, Thwet Yin Nyein, Khine Khine Oo, Thant Zin Oo, and Thet Thet Zin. Retrieval augmented generation for document query automation using open source llms. In *2024 5th International Conference on Advanced Information Technologies (ICAIT)*, pages 1–6. IEEE, 2024.
- [10] Robin Ko, Mustafa Kağan Gürkan, and Fatoş T Yarman Vural. Rerag: A new architecture for reducing the hallucination by retrieval-augmented generation. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pages 961–965. IEEE, 2024.
- [11] Fei Liu, Zejun Kang, and Xing Han. Optimizing rag techniques for automotive industry pdf chatbots: A case study with locally deployed ollama models. In *Proceedings of the 2024 3rd International Conference on Artificial Intelligence and Intelligent Information Processing*, pages 152–159, 2024.
- [12] L. Masanneck, S. G. Meuth, and M. Pawlitzki. Evaluating base and retrieval augmented llms with document or online support for evidence based neurology. In *Proceedings of NPJ Digital Medicine*, volume 8, page 137. Springer, 2025.
- [13] Mayank P Muthyala, Claire Lauer, and Stephen Carradini. So you want to build a chatbot?: A systematic case study comparing the design and development of two water chatbots. In *Proceedings of the 42nd ACM International Conference on Design of Communication*, pages 128–137, 2024.
- [14] Y Bhanu Sree, Addicharla Sathvik, Damarla Sai Hema Akshit, Omrender Kumar, and Bandaru Sai Pranav Rao. Retrieval-augmented generation based large language model chatbot for improving diagnosis for physical and mental health. In *2024 6th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pages 1–8. IEEE, 2024.
- [15] David Steybe, Philipp Poxleitner, Suad Aljohani, Bente Brokstad Herlofson, Ourania Nicolatou-Galitis, Vinod Patel, Stefano Fedele, Tae-Geon Kwon, Vittorio Fusco, Sarina EC Pichardo, et al. Evaluation of a context-aware chatbot using retrieval-augmented generation for answering clinical questions on medication-related osteonecrosis of the jaw. *Journal of Cranio-Maxillofacial Surgery*, 2025.
- [16] Liwei Xu and Jiarui Liu. A chat bot for enrollment of xi’an jiaotong-liverpool university based on rag. In *2024 8th International Workshop on Control Engineering and Advanced Algorithms (IWCEAA)*, pages 125–129. IEEE, 2024.
- [17] R. Xu, Y. Hong, F. Zhang, and H. Xu. Evaluation of the integration of retrieval-augmented generation in large language model for breast cancer nursing care responses. In *Proceedings of Scientific Reports*, volume 14. Springer, 2024.
- [18] Qingqing Zhou, Can Liu, Yuchen Duan, Kaijie Sun, Yu Li, Hongxing Kan, Zongyun Gu, Jianhua Shu, and Jili Hu. Gastrobot: a chinese gastrointestinal disease chatbot based on the retrieval-augmented generation. *Frontiers in Medicine*, 11:1392555, 2024.
- [19] Beilei Zhu and Chandrasekar Vuppapalapati. Enhancing supply chain efficiency through retrieve-augmented generation approach in large language models. In *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, pages 117–121. IEEE, 2024.