RAG Question-Answer System

Lohith Sasanapuri
lsasanapuri@tulane.edu

Pavan Kumar Sanjay
psanjay@tulane.edu

Department of Computer Science
Tulane University, New Orleans, LA

1 Abstract

Conventional chatbots often face challenges in maintaining a natural conversational flow and delivering precise, factual information. This study investigates chatbots based on retrieval-augmented generation (RAG), which have the potential to be more efficient and user-friendly. By considering the resource demands of large language models (LLMs) within the RAG framework, we have implemented a system based on the Retrieval-Augmented Generation for Knowledge-Intensive Natural Language Processing Tasks[4], which necessitates minimal retraining or fine-tuning of the LLMs utilized. Our objective was to create a chatbot that requires few training resources while ensuring high levels of accuracy. This report outlines our approach, findings, and the efficacy of our RAG-based chatbot.

2 Introduction

Conventional chatbot systems face challenges in delivering a natural conversational experience, particularly in comprehending user input. Operating primarily on static, rule-based frameworks limits their ability to enhance responses and often results in difficulties when providing fact-based answers. This rigidity can lead to a disjointed user experience.

To address these shortcomings, advanced AI-based architectures, such as Retrieval-Augmented Generation (RAG), are emerging as advantageous solutions to develop more capable chatbots. This architecture leverages retrieval capabilities to extract relevant context from prior conversations or associated documents. In addition, it employs the generative qualities of large language models (LLMs) to facilitate more accurate, contextually relevant, and fact-based dialogues, with the aim of a more comprehensive understanding of user needs.

Although various RAG architectures have been proposed recently, this study focuses on a specific architecture from the *RAG Lewis* [4], which is designed for tasks involving natural languages. This architecture requires minimal to zero retraining, yet it strives to deliver competitive performance.

We aim to implement this architecture to create a chatbot and evaluate its effectiveness in providing accurate and contextually relevant responses with minimal to no retraining, ultimately demonstrating its potential as a significant improvement over traditional chatbot systems.

3 Related Work

Previous research has extensively explored how to equip large language models (LLMs) with up-to-date knowledge to address issues like factual accuracy and knowledge base limitations. The*RAG Lewis* [4] in "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," presented a significant advancement in this area. Their proposed RAG system distinguishes itself by requiring minimal to zero task-specific retraining or fine-tuning. It employs a dense retrieval component, trained using a pre-trained frequency model, to identify and extract relevant information from an external

knowledge source. This retrieved information is then used to augment the input to a generative model, enabling the generation of fact-based conversational responses. The RAG architecture's ability to perform effectively on knowledge-intensive tasks without extensive retraining makes it a particularly relevant foundation for our work, where we aim to develop a chatbot that can provide accurate information with minimal adaptation.

Building upon the principles of retrieval-augmented generation, the development of ChatDiet, as described by Yang et al. (2023) in "ChatDiet: Empowering personalized nutrition-oriented food recommender chatbots through an LLM-augmented framework"[6] demonstrates the application of a RAG-like architecture for personalized conversational agents. ChatDiet leverages retrieval technology to provide personalized nutrition-oriented food recommendations to users. The system operates by accessing two primary databases: a personal model database containing user-specific information and a population-based database with nutritional elements. Upon receiving a user query, ChatDiet retrieves relevant information from these databases and passes it as context to a large language model. The LLM then generates a response tailored to the user's nutritional needs and preferences based on the retrieved information. This approach of using specific, retrieved knowledge to personalize chatbot interactions and provide contextually relevant recommendations served as a key inspiration for our own work in developing a chatbot that can leverage relevant information to enhance its responses.

4 Methodology

4.1 Overview of the Proposed Chatbot Architecture

Our goal is to develop a chatbot that leverages the Retrieval-Augmented Generation (RAG) architecture to provide accurate and contextually relevant responses with minimal retraining. The system comprises two main modules: a retrieval module that fetches relevant information from the knowledge base and the large language model that generates the final response based on the user query and the retrieved context (see Figure 1 for a high-level diagram)

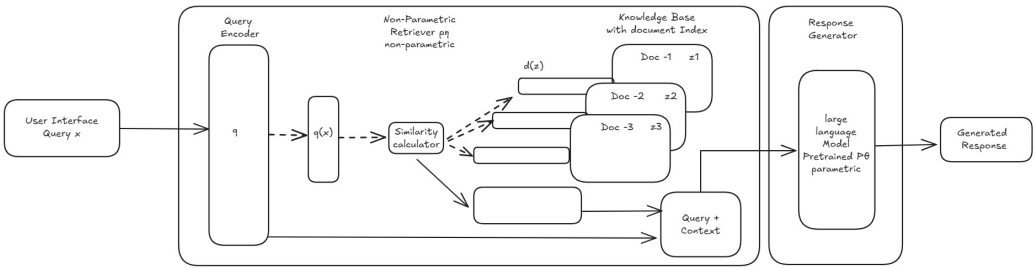

Figure 1: Proposed Chatbot Architecture

As depicted in Figure 1, the system first takes a user query and uses the retrieval module to identify the top five most relevant documents from the knowledge base. These retrieved documents, along with the original user query, are then fed into the large language model to generate the response.

4.2 Retrieval Module

The retrieval component of our chatbot operates as follows:

- **Knowledge Base:** Our knowledge base is the MS MARCO[2] dataset, an open-source, human-generated machine reading comprehension dataset curated for question answering.
- **Embedding Generation:** To represent both user queries and documents within the MS MARCO dataset as dense vectors, we utilize the **Sentence Transformer all-MiniLM-L6-v2**[5] model. This model is known for generating effective sentence embeddings.
- **Indexing and Similarity Search:** For efficient storage and retrieval of these vector embeddings, we employ the **FAISS**. Specifically, we use the `IndexFlatL2`[3] index, which performs a flat (brute-force) k-nearest neighbors search based on the Euclidean distance (L_2 norm) between the query vector and the document vectors.

- **Retrieval Process:** When a user inputs a query (X_q), it is first encoded into an embedding $Q(X_q)$ using the Sentence Transformer model. We then calculate the Euclidean distance between this query embedding and all document embeddings within our FAISS index. The top five documents with the smallest Euclidean distances (i.e., the most similar) are retrieved as context.

4.3 Large Language Model for Generation

The generative component of our chatbot is powered by the `microsoft/phi-3-mini-instruct`[1] model. This instruction-tuned LLM is designed for high-quality reasoning and was trained on a publicly available dataset. The retrieved top five documents, along with the original user query, are provided as context to this parameterized LLM (P_θ) to generate a relevant and informative response. The LLM processes both the non-parametric memory (retrieved documents) and its internal parametric knowledge to produce the final output.

4.4 Implementation Details

While we initially planned to deploy this chatbot as a Flask application for broader accessibility, time constraints led us to implement it as a Jupyter Notebook for the present study. We intend to transition to a Flask-based deployment in future work.

In summary, our system takes a user query, encodes it into a vector embedding, retrieves the top five most similar documents from our pre-processed MS MARCO [2] knowledge base using Euclidean distance via FAISS[3], and then provides the query and retrieved documents as context to the `microsoft/phi-3-mini-instruct`[1] model to generate a response within a Jupyter Notebook environment for this study.

4.5 Experimental Setup

4.5.1 Retrieval Parameter Exploration

While our current implementation uses the Euclidean distance metric, future experiments could explore the impact of more advanced, attention-based, and neural network-based similarity metrics on retrieval performance.

4.5.2 Document Chunking Strategy Evaluation

We conducted experiments by randomly sampling 50 questions from the dataset to evaluate the impact of different document chunking strategies on the quality of the generated responses. We compared two main approaches: fixed-size chunking and recursive character chunking, further exploring the effect of overlap within recursive chunking.

- **Fixed-Size Chunking:** We experimented with a fixed chunk length of 200 tokens as one of our baseline strategies.
- **Recursive Character Chunking with Overlap:** We investigated the impact of introducing overlap between consecutive chunks for the recursive character chunking strategy. Overlap refers to the number of common words shared between two adjacent chunks. We experimented with three different overlap percentages: 0% overlap, 20% overlap, and 50% overlap. The rationale behind exploring overlap is to provide the language model with more contextual continuity across retrieved segments, potentially leading to improved coherence and more informative generated answers.

.

These experiments aimed to determine the optimal chunking strategy and overlap configuration for providing sufficient context to the language model and improving the coherence and informativeness of the generated answers.

4.6 Evaluation Metric

We evaluated the performance of our question answering system using the ROUGE-L metric. This metric measures the length of the longest common subsequence (LCS) between the generated answer and the human-generated reference answer in the MS MARCO[2] dataset, providing an indication of the factual overlap and alignment. Specifically, the ROUGE-L Recall is calculated as:

$$\text{ROUGE-L Recall} = \frac{\text{Length of LCS}}{\text{Total number of words in the reference text}}$$

This recall value indicates the proportion of words in the reference answer that are also present in the generated answer, considering their sequential order.

5 Results

5.1 Quantitative Evaluation of Chunking Strategies

To quantitatively evaluate the impact of different document chunking strategies on the performance of our RAG-based question answering system, we calculated the ROUGE-L scores for 50 randomly sampled questions from the MS MARCO dataset. The descriptive statistics for each chunking strategy are summarized in Table 2 and visualized in Figure 3.

	Fixed Lenght	Recursive	Recursive overlapping 20%	Recursive overlapping 50%
count	50.000000	50.000000	50.000000	50.000000
mean	0.554282	0.516347	0.553697	0.557429
std	0.267143	0.285961	0.297970	0.306160
min	0.000000	0.000000	0.000000	0.000000
25%	0.337500	0.322818	0.289286	0.286654
50%	0.500000	0.486842	0.506091	0.533597
75%	0.794444	0.666667	0.800000	0.800000
max	1.000000	1.000000	1.000000	1.000000

Figure 2: Descriptive Statistics of ROUGE-L Scores

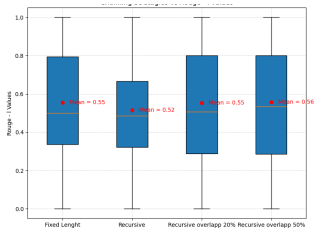

Figure 3: Box Plot Comparison of ROUGE-L Scores

The box plot (Figure 3) visually compares the distribution of ROUGE-L scores across the four chunking strategies. The numerical statistics in Table 2 provide a more detailed breakdown of these distributions, including the mean, standard deviation, quartiles, and range.

5.2 Visual Comparison of Performance Distribution

As illustrated in Figure 3, the spread of ROUGE-L scores varies across the different chunking methods. The Recursive strategy exhibits a slightly lower median and a narrower upper quartile compared to the other methods. Conversely, the Recursive strategies with 20% and 50% overlap show comparable or slightly better median and upper quartile values than the Fixed Length approach. The lower whiskers suggest that the overlapping recursive strategies might have fewer instances of extremely low performance compared to the standard Recursive method.

6 Conclusion and Future Directions

Based on our observations, the choice of document chunking strategy significantly impacts the performance of the RAG-based question answering system, as measured by the ROUGE-L score. While the average performance is relatively close, the introduction of overlap within the recursive chunking method demonstrates a promising avenue for improvement. Specifically, the recursive strategy with a 50% overlap yielded the highest mean and median ROUGE-L scores across our evaluation set. This suggests that providing more continuous contextual information to the language model can lead to better alignment with the reference answers.

Furthermore, it is important to highlight that our RAG-based chatbot, even with minimal training resources, demonstrates the capability to produce quality-based output, achieving encouraging

ROUGE-L scores. This underscores the potential of the architecture to leverage readily available pre-trained models and external knowledge effectively, reducing the need for extensive task-specific training while still delivering high-quality results.

However, the substantial variability in scores across all strategies indicates that the optimal chunking approach might be context-dependent. For future work, we recommend conducting a more in-depth analysis of the questions and documents where different chunking strategies perform particularly well or poorly. Furthermore, exploring a wider range of overlap percentages and fixed chunk sizes, as well as investigating more sophisticated, semantic-aware chunking techniques, could potentially lead to further performance enhancements and a more robust and consistent question answering system. Additionally, conducting statistical significance tests to validate the observed trends and performing qualitative evaluations of the generated responses would provide a more comprehensive understanding of the impact of different chunking strategies. Finally, exploring the trade-offs between computational resources, training requirements, and the quality of the generated output remains a crucial direction for future research in this area.

References

[1] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, and X. Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

[2] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang. Ms marco: A human generated machine reading comprehension dataset, 2018. URL https://arxiv.org/abs/1611.09268.

[3] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library, 2025. URL https://arxiv.org/abs/2401.08281.

[4] 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. URL https://arxiv.org/abs/2005.11401.

[5] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. URL https://arxiv.org/abs/2002.10957.

[6] Z. Yang, E. Khatibi, N. Nagesh, M. Abbasian, I. Azimi, R. Jain, and A. M. Rahmani. Chatdiet: Empowering personalized nutrition-oriented food recommender chatbots through an llm-augmented framework. *Smart Health*, 32:100465, 2024. ISSN 2352-6483. doi: https://doi.org/10.1016/j.smhl.2024.100465. URL https://www.sciencedirect.com/science/article/pii/S2352648324000217.