Innovative Corporate Query handling with Domain-Specific RAG

1st Simran Kumari
Computer Science and Engineering
Sikkim Manipal Institute of Technology, SMU
Gangtok, India
simran141003@gmail.com

2nd Udit Kr. Chakraborty
Computer Science and Engineering
Sikkim Manipal Institute of Technology, SMU
Gangtok, India
udit.c@smit.smu.edu.in

3rd Basab Nath
Computer Science and Engineering
Sikkim Manipal Institute of Technology, SMU
Gangtok, India
basab.n@smit.smu.edu.in

4th Avinash Kumar
Computer Science and Engineering
Sikkim Manipal Institute of Technology, SMU
Gangtok, India
avinash_202100077@smit.smu.edu.in

5th Pratham Srivastava
Computer Science and Engineering
Sikkim Manipal Institute of Technology, SMU
Gangtok, India
pratham_202100066@smit.smu.edu.in

6th Namsani Vivek
Computer Science and Engineering
Sikkim Manipal Institute of Technology, SMU
Gangtok, India
namsani_202100045@smit.smu.edu.in

***Abstract*—Large Language Models (LLMs) have significantly impacted Natural Language Processing, but face limitations in domain-specific query response generation. To overcome these issues, this work proposes a novel method that combines LLMs with Retrieval Augmented Generation (RAG). We have designed a hybrid model to improve the accuracy of responses to domain-specific questions by integrating data from minutes of meetings from an external knowledge store. It involves preprocessing meeting data, developing a retrieval mechanism using similarity search, and implementing a generation model based on the LLaMA-2 13B architecture. We evaluated our model using BLEU and ROUGE scores, demonstrating improved performance in closed-domain question answering. The BLEU score of 0.56037 and ROUGE scores (ROUGE-1: 0.7211, ROUGE-2: 0.6217, ROUGE-L: 0.7211) indicate the model's proficiency in generating accurate responses while reducing hallucinations.**

***Keywords*—RAG, LLM, , GPT3.5, Closed Domain**

I. Introduction

Large Language Models have brought a great impact in the field of Natural Language Processing. Large Language Models(LLMs) are models that are trained on extensively huge datasets because of which they are easily able to understand text and comprehend the context behind them [1]. Because of this capability LLMs can capture the context in text and have great proficiency in text generation and summarization.This is enabled because such language models are trained on billions of parameters.Based on transformer architecture,they consist of many layers of neural networks whose parameters can be fine-tuned and whose performance can be increased further by using the attention mechanism.These models are pre-trained with vast amounts of data and because of this they can generate text without the need to access any kind of external memory source [2].Though this is has brought a significant impact, it has is own disadvantages for instance- a Large Language Model cannot generate text for any specific domain related queries, in addition to this memory revision and expansion is time consuming and computationally tedious and many a times it may generate texts that are not relevant [3].

To cater these issues in our work we have used an additional RAG-Retrieval Augmented Generation Model for information retrieval.It is a framework that enhances the text generation capability of an LLM by integrating an additional external knowledge base.When the generated text is based on an external knowledge source it is ensured that the responses generated are relevant to the query of the user.Along with this the need to update or revise the intrinsic knowledge of the LLM is totally eradicated and is not required to train the model continuously on any new data [4]. RAGs use vector databases to store the external data and use similarity search to retrieve the results.This approach is referred to as a Hybrid Model approach where RAG serves as the retriever component whereas the LLM acts as the generator component that synthesizes the response from the chunks of data retrieved by RAG [11]. Hybrid Models improve the efficiency of the responses generated by the LLMs by mitigating their limitations.

The primary contribution of our work are as follows:

- Incorporating a hybrid model approach, whose external knowledge base consists of the minutes of the meeting data, this leads to the development of a model that can answer domain specific questions based on the minutes

2024 OITS International Conference on Information Technology (OCIT) | 979-8-3315-1040-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/OCIT65031.2024.00058

979-8-3315-1040-4/24/$31.00 ©2024 IEEE
DOI 10.1109/OCIT65031.2024.00058

of the meeting.

- Manually creating a custom dataset in tabular format consisting of query and responses to train the model with.
- Setting up the RAG pipeline in a meticulous manner such that an efficient mechanism can be established to answer domain specific user related queries with efficiency and precision.(in our case questions related to the minutes of the meeting).

II. Literature Review

LLMs have had a large impact in the domain of Natural Language Processing (NLP). But still they have their own limitations. These limitations are mitigated by models like RAG that include an external knowledge base to improve the relevancy and precision in the responses generated. The following review explores important developments in the domain of using RAG to improve the efficiency of LLMs.

P. Lewis et al. [1] used (RAG) -Retrieval Augmented Generation models to increase the performance of knowledge-intensive tasks by combining both parametric and non-parametric memory for text generation. This helps to generate relevant and precise responses in comparison with other language models that only utilize the parametric memory. Hugo Touvron et al. [2] proposed that the performance of a LLM increases when it is trained on more tokens.It also mentioned how small amounts of fine-tuning results in improvement in the performance of the LLM and their ability to understand and process the instructions. GPT-4 Technical Report [3] shed light on GPT-4 a transformer based model that was trained on data from third parties as well as publicly available data,which was later fine-tuned using Reinforcement Learning.It was able to secure a score among top 10 % of the test makers in the bar exam which proves its efficiency in response generation. Yunfan Gao et al. [4] developed models like Advanced RAG that uses pre-retrieval and post-retrieval processes to mitigate the issues faced by Naive RAG. The Pre-Retrieval process includes optimizing the indexing of data that helps improve the performance of RAG, which removes irrelevancy and irregularities in the data.The post-Retrieval process includes re-ranking of the retrieved data to identify the most relevant information. P. Omrani et al. [5] demonstrated a new hybrid RAG framework combining the Sentence-Window and Parent-Child processes to improve the response generation capability of LLMs.The hybrid model outperformed various other RAG techniques on many metrics stating their proficiency in generating precise and contextually relevant output. M. Besta et al. [6] introduced Multi-Head RAG (MRAG) to address the need of queries that require retrieval of various documents with different contexts to answer them, this becomes an issue for existing RAG models as the retrieval of documents can pose complexity given their far apart embeddings in space. The MRAG model increased the accuracy of the retrieved content given the complex nature of the query. O. Ovadia et al. [7] compared fine-tuning and RAG and stated that RAG outperforms fine-tuning approach for new knowledge as well as existing knowledge.It also mentions about the inefficiency of unsupervised fine-tuning on the responses generated by the LLM. S. Siriwardhana et al. [8] assessed the improvement in the model performance by training both the components of RAG together for the purpose of adapting to a given domain in Question-Answering.They introduced RAG-end2end a model that updates all the components of the extrinsic data during the training phase. J. Torres et al. [9] examined an evaluation technique to assess the responses generated by a RAG-based chatbot and utilized a process to cross-validate the questions that were not answered correctly.Their technique was able to examine LLMs namely Llama-2-Dutch-13B or GPT-3.5 Turbo. A. Thakur et al. [10] introduced Super Retrieval Augmented Generation (Super RAGs) to increase the accuracy of responses generated by LLM.This approach consists of incorporating an extrinsic database having very less structural modifications that improves the relevancy as well as the speed.of the model.The integration of Super RAGs with LLMs has led to AI systems being more trustworthy.

Despite several researches in the field of Large Language Models (LLMs) , several gaps still remain. LLMs still are incapable of generating precise, accurate and relevant responses to user queries. Along with it, the need to continuously update the knowledge base which is computationally expensive along with it requires a substantially large amount of time to train.LLMs also are unable to answer domain specific questions. Hence through our work we address these challenges by incorporating a RAG framework that includes an external knowledge base, consisting of custom minutes of the meeting data that is used to train the model. The custom dataset is created manually in a tabular format consisting of query-response columns which are used for training .This accelerates the retrieval of contextually relevant chunks of data to answer domain specific queries (in this case queries related to the minutes of the meeting).This increases the precision and accuracy of the responses generated as well as significantly reduces the computational and time overhead.

III. Preliminaries

A. Large Language Models (LLMs)

Large Language Models have brought a great impact in the domain of Natural Language Processing.They are models trained on extensively huge datasets due to which they are easily able to understand and generate text like humans. Fine-tuning them with domain-specific datasets enables the LLMs to generate specialized responses. LLM's architecture consists of deep neural networks, specifically models based on transformers. Such models have many layers of self-attention mechanisms as well as feed forward neural networks.The self-attention mechanism enables the LLM to understand the different levels of gravity of different tokens during processing of data thus leading to better contextual understanding. Transformer based architectures are known to capture long-range dependencies thus making them efficient for natural language processing jobs.[12] For tasks like translation and text generation encoder-decoder architecture is utilized where in the encoder processes and creates representations based on

the input and the decoder determines the output based on these representations. By incorporating such architectures LLMs are able to determine their dexterity in tasks involving context understanding. Figure 1 depicts Large Language Model.

B. Retrieval Augmented Generation Model -(RAG)

The architecture of the RAG -Retrieval Augmented Generation Model consists of various components that enhance the response generation abilities of a LLM.It consists of three major steps - segmenting the data into smaller chunks, converting the chunks into vectors and indexing them and then retrieving data based on semantic similarity.[4] The chunking process converts the data into segments so that it can be retrieved and processed efficiently.The data chunks are then encoded to their vector representations using an encoder.In the retrieval step the data is retrieved based on similarity between the user entered query vector and the vector encoded data chunks.On the basis of the evaluated similarity scores, the retriever retrieves top-k chunks of contextually relevant data.These chunks are then concatenated with the user query and then provided to the LLM or the generator model, that using its text generation capabilities generates a response. This enables the model to generate accurate and relevant responses to the user query.Thus, a RAG reduces the probability of a LLM to generate incorrect or irrelevant responses by providing an additional knowledge source. Figure 2 depicts Retrieval Augmented Generation Model.

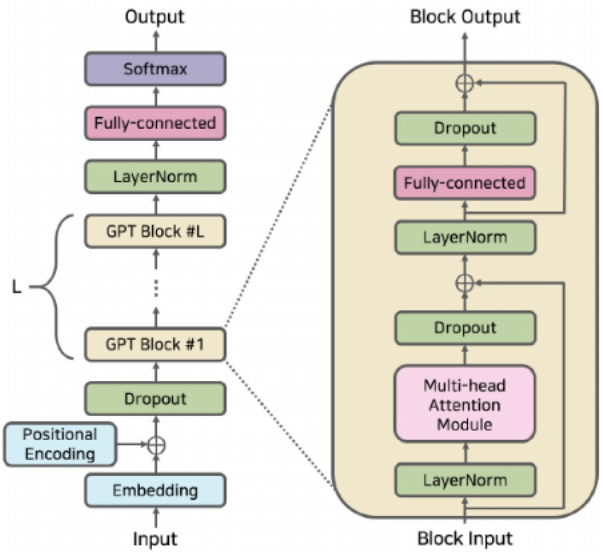

Fig. 1. GPT Models conceptual framework [13]

IV. Methodology

The proposed methodology talks about the implementation of the retriever component as well as the generator component for enhancing the responses of LLM using RAG-

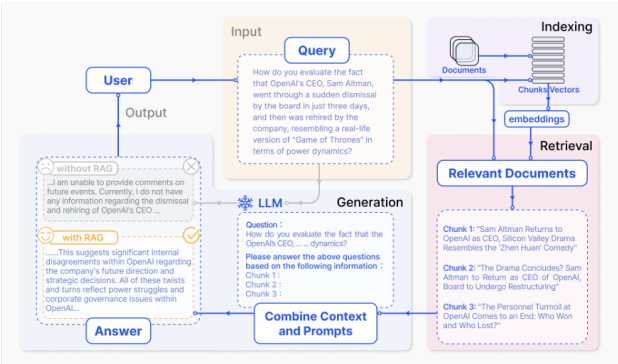

Fig. 2. Illustration of the RAG model used for query-response generation [4]

- Data Preprocessing: We performed data preprocessing by initially extracting data from the minutes of the meeting and then converting it into tabular format for storing in the vector database.
- Developing a Retrieval Mechanism:We developed a retrieval mechanism based on similarity search that retrieves the top similar chunks of data.
- Developing a Generation Model: We developed the generational model where a LLM processes the retrieved chunks of data and converts it into a response that is synchronous and relevant.
- Integration of Retrieval and Generation Components: Finally, we integrated the retrieval and generation components to make a model that incorporates the strengths of both RAG and LLM.

A. Developing a Retrieval Mechanism:

In order to build a retrieval mechanism initially the extracted text was embedded in 32 item batches.Each batch was given an id which was unique,using a MD5 hash function.These were converted into embedding vectors using an embedding model. The embedding vectors and its respective ids were upserted in the vector database (pinecone in our case). In the next step, in order to retrieve the most relevant chunks of data similarity search is conducted.During this similarity search the embeddings that were created during data preprocessing are compared with the query vectors, and the most relevant chunks of data are retrieved. Through the integration of the above components , a retrieval mechanism is developed to proficiently retrieve information.

B. Developing a Generation Model:

In order to develop a generation model, Meta LLaMA-2 13B [14] chat model was incorporated that was utilized using the Transformers library.A quantization method was configured to make sure GPU is used effectively.In the next step a text generation pipeline was set up, again using the Transformers library. This pipeline generates relevant output based on the given inputs.Various parameters were configured to enhance the performance of the pipeline for instance-parameters like

temperature were kept low to make the generated output more consistent, maximum token limit was defined to prevent the output from being too long. Thus the setting up of the above pipeline optimizes the text generation process, ensuring the text generated is relevant and contextually accurate.

TABLE I
KEY PARAMETERS TO FINE-TUNE THE GENERATION MODEL

Parameter	Description	Value
Temperature	To control the randomness of the output and ensure consistency.	0.0
Maximum Token Limit	To restrict the length of the output generated.	30
Repetition Penalty	To restrict the probability of generation of repetitive text.	1.1

C. Integration of Retrieval and Generation Components:

Integration allows us to develop a model that incorporates the strengths of both the components.This integrated model generates relevant outputs based upon domain specific queries (minutes of the meeting) in our case.The retriever component retrieves content by employing similarity search between the query and the chunks of data stored in the database.After the retrieval of the text, the generation model based on LLaMA-2 13B uses this text as input and generates output based on them.The retriever is integrated by utilizing the Langchain library to form a text generation pipeline. Through the integration of the retrieval and generation components, the model's text generation ability can be enhanced as well as its limitations can be reduced.

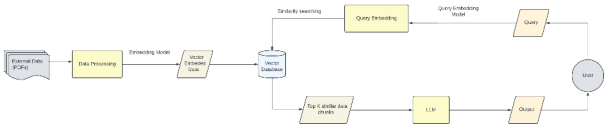

Fig. 3. The Proposed Model Architecture

V. RESULTS AND ANALYSIS

A. Data Preprocessing

The data used as the non-parametric memory in the model was collected manually from the Computer Science and Technology, Department at SMIT(Sikkim Manipal Institute of Technology).It consists of various PDFs of the minutes of the meetings held at the department.

For the data preprocessing, initially data was manually scanned from the PDF documents. As these documents were not machine readable, pytesseract's Optical Character Recognition(OCR) was utilized to convert the images to text. The pages in each of the PDFs were iterated using PyPDF2.The found images in the PDF's were then changed into PIL Image object and henceforth processed using pytesseract's Optical Character Recognition(OCR) and then subsequently converted into machine readable text. This process was applied to all the PDF's and then all the retrieved text was appended to a list. In order to create a structured PDF, each entry in the list was converted back into PDF format.This was done to convert machine unreadable PDF files to consistent machine-readable PDFs. For the purpose of batch processing, a library called hashlib was utilized to create batches of data of unique ids.For each batch metadata is created which consists of two fields - instruction and response. This metadata helps in better understanding of the context of each entry. These entries are then upserted in the Pinecone database.

Fig. 4. The data upserted in the Pinecone Database.

B. Closed Domain Question Answering

The integrated RAG model has shown a massive improvement in generating responses related to a given domain also known as closed domain question answering.The domain in our case consists of the meetings conducted in CSE Department of SMIT and the queries related to it.Incorporation of both the models allows the generation of contextually relevant responses while ensuring that the responses are consistent.

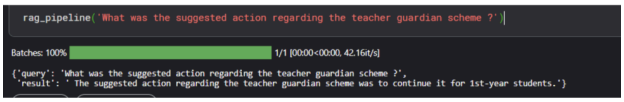

Fig. 5. Query Response Generation by the Model

VI. EVALUATION METRICS

The following metrics were evaluated to assess our model :

A. BLEU (Bilingual Evaluation Understudy) [15]

The BLEU score [15] for the actual versus the predicted outputs turned out to be - 0.56037. BLEU score is used in NLP to assess the quality of the output generated by measuring the similarity between the predicted response versus the actual response.In our case a BLEU score of 0.56037 suggests that the predicted outputs are moderately similar to the actual outputs.

B. ROUGE Score :(Recall-Oriented Understudy for Gisting Evaluation) [16]

ROUGE [16] is a metric used in NLP(Natural Language Processing) to determine the quality of the responses generated by the model. It is calculated by assessing the n-grams that overlap between the actual and the predicted output.The more the overlap-the better is the quality of the output generated

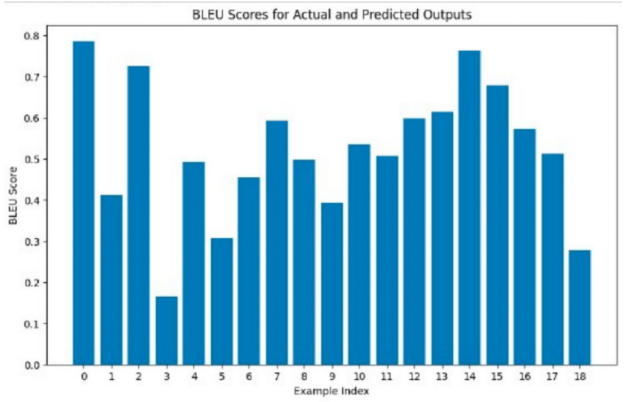

Fig. 6. BLEU Score

.For the given model calculated ROUGE Scores are - Rouge score 1 (unigram - model) , Rouge score 2 (bigram - model) and Rouge - L (average of unigram and bigram) of the actual and predicted output.On calculating the obtained scores were:

TABLE II
THE DIFFERENT KINDS OF ROUGE SCORES OF THE MODEL

Type	Value
ROUGE Score 1	0.7211
ROUGE Score 2	0.6217
ROUGE L	0.7211

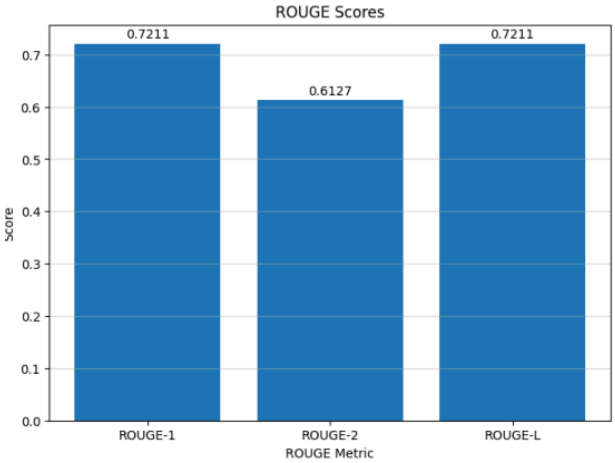

Fig. 7. ROUGE Score

The figure shown below indicates the set of questions that were fed into our model and the set of output it predicted versus the actual output.

CONCLUSION

Through this paper a model has been devised to mitigate the issues faced by a LLM, thus enabling the model to generate

Fig. 8. Data on which the evaluation metrics were calculated

accurate and precise output to domain specific queries of an user.This approach incorporates both the retriever model as well as the generator model to increase the relevancy of the outputs generated.Through the integration of an extrinsic knowledge source containing the minutes of the meeting at the CSE Department SMIT, it is displayed how a LLM can solve domain-specific queries. Using the evaluated BLEU and the ROUGE scores it can be assessed how the model helps in generating better results with less hallucinations thus showing great proficiency in answering closed-domain queries. Future work could include developing a retriever and generator based hybrid models for other domains as well.

REFERENCES

[1] P. Lewis, E. Perez, S. Piktus, M. Lewis, A. Khashabi, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge Intensive NLP Tasks," arXiv preprint arXiv:2005.11401v4 [cs.CL], 12 Apr. 2021.

[2] H. Touvron, L. M. Zaccarelli, A. Vazquez, M. Dymetman, G. Synnaeve, and P. Labatut, "LLaMA: Open and Efficient Foundation Language Models," Meta AI. [Online]. Available: https://arxiv.org/abs/2302.13971. [Accessed: 19-Jun-2024].

[3] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774v4 [cs.CL], 19 Dec. 2023.

[4] Y. Gao, C. Wei, Y. Xiong, L. Yang, and M. Shou, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997v4 [cs.CL], 5 Jan. 2024.

[5] P. Omrani, A. Hosseini, K. Hooshanfar, Z. Ebrahimian, R. Toosi, and M. A. Akhaee, "Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement," in *2024 10th International Conference on Web Research (ICWR)*, Tehran, Iran, 2024, pp. 22-26, doi: 10.1109/ICWR61162.2024.10533345.

[6] M. Besta, K. Zetter, T. Gheiratmand, M. Mrozek, T. Hoefler, and A. Jaggi, "Multi-Head RAG: Solving Multi-Aspect Problems with LLMs," arXiv preprint arXiv:2406.05085v1 [cs.CL], 7 Jun. 2024.

[7] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs," 2024.

[8] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, "Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1-17, 2023, doi: 10.1162/tacl_a_00530.

[9] J. J. G. Torres, M.-B. Bîndilă, S. Hofstee, D. Szondy, Q.-H. Nguyen, S. Wang, and G. Englebienne, "Automated Question-Answer Generation for Evaluating RAG-based Chatbots," University of Twente, Enschede, The Netherlands.

[10] A. Thakur and R. Gupta, "Introducing Super RAGs in Mistral 8x7B-v1," 2024.

[11] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, "Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1-17, 2023, doi: 10.1162/tacl_a_00530.

[12] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A Comprehensive Overview of Large Language Models," arXiv preprint arXiv:2307.06435, Jul. 2023. [Online]. Available: https://arxiv.org/abs/2307.06435.

[13] M. Lee, "A Mathematical Investigation of Hallucination and Creativity in GPT Models," *Mathematics*, vol. 11, no. 2320, 2023, doi: 10.3390/math11102320.

[14] Meta AI, "LLaMA: Large Language Model Meta AI," 2023. [Online]. Available: https://github.com/facebookresearch/llama.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, USA, Jul. 2002, pp. 311-318. doi:10.3115/1073083.1073135.

[16] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Proceedings of the ACL-04 Workshop: Text Summarization Branches Out*, Barcelona, Spain, Jul. 2004, pp. 74-81. doi:10.3115/1073083.1073135. "'