ISSN: 2754-4737

Journal of Neurology Research Reviews & Reports

SCIENTIFIC Research and Community

Review Article

Open Access

The Wonders of RAG: Streamlining Knowledge with Advanced Techniques Systematic Literature Review Report

Wafaa Bazzi* and Mervat Gaith

Education and Programming, Bint Jbeil, South Lebanon - Nabatieh Governorate, Lebanon

ABSTRACT

The Retrieval-Augmented Generation (RAG) framework enhances Large Language Model (LLM) performance by incorporating external knowledge through information retrieval, addressing inherent limitations in standard LLMs. RAG forces fine-tuning based on relevance, to improve Open Domain Question Answering and dynamically updates external data during model training, specifically within Dense Passage Retrieval (DPR) models. This approach facilitates up to date dialogue generation, personalizes responses with external sources, and employs metrics to evaluate both sources and answers. While RAG offers large benefits in reducing hallucinations and improving answer quality, challenges remain. The quality of external data directly influences response accuracy, and hallucinations can persist due to insufficient input information or evaluation metrics. Future research should prioritize enhancing data integration, refining query prompts, developing real-time correction mechanisms, and adapting RAG for specific domains to fully realize its potential.

Purpose

The aim of this report is to explore and analyze the advanced framework known as Retrieval Augmented Generation (RAG). This framework significantly enhances the abilities of Large Language Models (LLMs) by incorporating external knowledge to refine answers and generate optimal responses.

***Corresponding author**

Wafaa Bazzi, Education and Programming, Bint Jbeil, South Lebanon - Nabatieh Governorate, Lebanon.

Received: March 19, 2025; **Accepted:** March 26, 2025; **Published:** March 31, 2025

Introduction

In the enlarge of need in the domain of Natural Language Processing (NLP), Retrieval Augmented Generation (RAG) appears as solution integrating external data in Language model. RAG represent significant advancement over traditional approaches, enhancing the generation capabilities of Large Language Models (LLMs) and open new opportunities in various NLP applications. The Report delves into the wonders of RAG, exploring its basic concepts, applications, and impact on streaming knowledge with advanced techniques.

The Importance of Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) is an approach that combines retrieval and generation techniques to enhance the performance of Large Language Models (LLMs). There are numerous challenges when working with LLMs such as knowledge gaps in the field, realism issues, and hallucinations. Retrieval Augmented Generation (RAG) provides a solution to reduce some of these problems by augmenting LLMs with external knowledge such as databases. RAG is suitable in knowledge conversational where the knowledge is frequently updating. The benefit of RAG over other approaches is that the LLM does not need to be retrained on task applications. The Report proposes a systematic literature review that summarizes previous researches about RAG for LLMS, explains the advantages of approach over other exiting methods as LLMS alone, also explains, assess and interpret some research evidence about the subject.

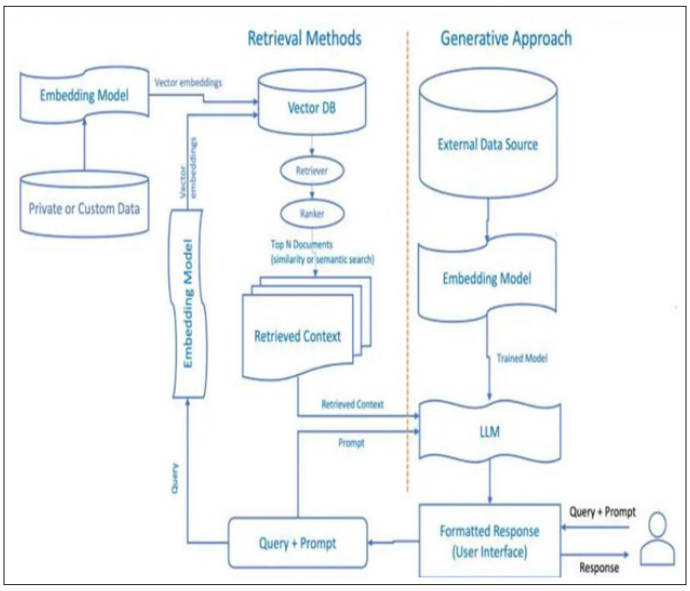

Figure 1

Background

Language Model pre-training has become a foundation in Natural Language Processing, aiming to acquire learning useful representations of language, from unlabeled text corpora. These pre-trained models serve as foundational knowledge bases,

enhancing generalization and performance in down- stream tasks through fine-tuning. One of the prevalent variant of pre-training is the masked language model (MLM), popularized by BERT, where the model predicts masked tokens within input text pas- sages, thereby encoding syntactic, semantic, and world knowledge. In the REALM of Natural Language understanding, Open Domain Question Answering (OQA) stands as task for evaluating a model ability to integrate knowledge effectively. unlike traditional SQUAD, Open-QA tasks do not provide a predefined document containing the answer. Instead, models must recollect knowledge from huge corpus of knowledge to generate accurate responses. Retrieval-based approaches, which retrieve relevant documents from knowledge corpus and then extract answers, is widely used in Open-QA systems. In other hand, generation systems generate responses token-by-token using sequence-to-sequence models. In the context, the Retrieval Augmented Generation (RAG) framework represents a significant advancement, joining the gap between Language model pre-training and Open-QA tasks. By incorporating external data through Information Retrieval, RAG enhances the generation ability of Large Language Models (LLMs) and addresses limitations inherent in traditional pre-training methodologies. This integration of retrieval-based techniques with Language model pre-training holds potential for further optimizing Open-QA systems and advancing Natural Language understanding capabilities [1-3].

Steps of RAG

Retrieval Augmented Generation (RAG) combines search techniques for external data with text generation models, enhancing the quality of responses generated by Large Language Models (LLMs).

Here are the fundamental concepts behind RAG:

Tokenization: The input text is split into tokens to create a manageable representation for the model.

Encoding: The tokenized input is encoded into a dense vector representation using neural networks transformers. This encoding captures contextual information and semantic relationships among tokens.

Retrieval: RAG performs a relevancy search to retrieve relevant information from an external knowledge base. This knowledge base could be a specialized domain-specific dataset or a company's internal documents. Decoding: the retrieved information is combined with encoded input to create context-aware representation. The decoder generates a sequence of tokens based on this representation. Attention Mechanism: RAG uses attention mechanism to focus on relevant parts of the input during decoding. This ensures that the generated response aligns with the retrieved information. In Summary RAG powers external knowledge sources to supplement LLMs, resulting in more accurate responses [1,4-6].

Set UP your RAG

To create your own RAG system, you first need to specify the library or search engine that will bring in external resources from web, or alternatively, you can utilize Large Language Models (LLMs). once your library is established, you'll need to create a well-formatted prompt with specific phrases and words. This prompt will guide the system in generating the desired output. Next, you'll need to define metrics or parameters to refine the inputs. These could be labels or other criteria that help the system understand to produce desired output. After the system generates an answer, the system will evaluate it and assign a score based on predefined evaluation rules. If the score is met the specified threshold, the answer is considered as optimal, otherwise, the system will iteratively update the answer based on feedback from the reader, whom can be transformer model, ChatGPT, or a human reviewer [4].

ChatGPT of LLM

ChatGPT is a conversational model developed by OpenAI1. It's a type of Large Language Model (LLM). Which means it's trained on vast amount of data, including code, scientific literature, books, and internet articles. LLMs are capable of understanding and generating, natural language to perform wide range of tasks. Retrieval Augmented Generation (RAG) is a process that optimizes the output of Large Language model by referencing a commanding knowledge base outside of its training data sources before generating a response. It extends the capabilities of LLMs to specific domains or organization's internal knowledge base, all without the need to retrain the model. While ChatGPT is a part of LLMs, it's not essential in RAG. However, it can be used in conjunction with RAG to create more accurate and contextually relevant responses. For example, RAG can be used to retrieve relevant information from a data source and pass that information to ChatGPT alongside the user's prompt. This information is used to improve the model's output by augmenting the model's base knowledge. So, while ChatGPT is not essential for RAG, their combination can lead to improved results [1,4,5,7].

REALM

REALM pre-training is a method that enhances the capabilities of LLMs by allowing them to retrieve and join documents from large corpus during pre-training, fine-tuning and inference [1].

Wonder of RAG (ODQA)

The wonder of Retrieval Augmented Generation (RAG) lies in its recent advancement in Open-Domain Question Answering (ODQA). It combines the benefits of ODQA with additional advantages, making it a powerful tool for information retrieval and generation. This leads the building of robust and efficient question-answering systems, that adapt to specific domains by updating all components during training. Furthermore, it generates optimal responses by adding external data sources for information retrieval. This contributes to creating a dynamic and up-to-date conversational AI system. Moreover, it helps reduce the hallucination of Language Model (LM) [1,2,4,5,8].

Wonder of RAG (Uni-MS-RAG Framework)

The Uni-MS-RAG Framework is a new technology that integrates planning, retrieval, and generation tasks using large language models in sequence-to-sequence manner. The methodology of this framework consists of three main stages:

Planning: this stage involves converting original query into independent tokens and formulating the independencies to create sequence-to-sequence generation. It also involves gathering the start and the end positions for the encoded query and configuring the roles of tokens.

Retrieval: this stage retrieves the top-n results from external databases according to the decisions made in the planning stage. It introduces a unified training approach for retriever and user queries by searching tasks and generating similarity tokens to predict relevance scores. The Uni-MS-RAG systems is then trained to evaluate the relevance of dialogue context and evidence, serving as a retriever during inference. Generation this is the process in a dialogue system where all previous results, such as retrieved

evidence, and similarity scores, are combined with dialogue context to produce a response. The input is structured in a way that helps the language model identify and focus on relevant evidence, while filtering out unrelated information. In the training stage, the focus is on acquiring relevance scores to determines the relationship between dialogue context and evidence. It details two methods for obtaining these scores, the first use fine-tuned DPR model and prompting large language models like ChatGPT. The DPR method involves creating a dataset with positive and negative evidence, or prompt the LLMs used to predict similarity scores. The Inference Stage is a method for refining system responses by evaluating the quality of evidence using similarity and consistency scores. The process involves calculating a consistency score between a piece of evidence and the generated response, determining an overall score by combining the similarity and consistency scores, adjusting the evidence list based on these scores and re-generating responses using refined evidence, and iteratively applying this refinement process to improve response quality. This innovative technology is a significant advancement in the field of AI and machine learning, providing more accurate and contextually relevant responses in dialogue systems [1, 3-5].

RAG-end 2end Extension of RAG

The RAG-end2end model is extended version of RAG model with several components. The retriever is a dual encoder retrieval model that uses BERT to encode questions and passages. The similarity between questions and passages is determined by of their embeddings. The generator is BART sequence-to-sequence language model that is trained to generate responses with the retrieved passages serving as latent variables. The training process involves a modified cross-entropy loss for the probability of document selection given a context. All passages are encoded using the Passage Encoder before training. To retrieve similar passages, a dot product calculation is performed between the question embeddings from the Question Encoder and encoded passages. Due to the large volume passages, this retrieval could slow down training. To improve efficiency, RAG uses the FAISS indexing method, which reduces redundant computation and speeds up the retrieval process. The RAG-End2end model proposes training the components by updating the passage encoder and the knowledge base index. The process involves iterative re-encoding and re-indexing of the knowledge base, which can be time consuming but is essential for model to adapt to new domains. The goal of statement reconstruction is to enhance the model's domain-specific knowledge. The process involves encoding the input statements, retrieving similar passages, and reconstructing the input statements. A special token is used to distinguish reconstruction tasks from QA tasks, and this approach helps model better learn and adapt to domain-specific information during RAG training [2,5].

Using REALM: Retrieval-Augmented Language Model Pre-Training in ODQA

Natural Language (NLP) tasks such as question answering have seen significant advancements with the introduction of language model pretraining. Once such advancement is the Retrieval Augmented Large Model (REALM) pre-training, which augments the language model with a knowledge retriever. This retriever can access a large corpus like Wikipedia during pre-training, fine-tuning and inference, allowing the model to capture knowledge in a more interpretable and modular way. For the first time, this knowledge retriever has been pretrained in unsupervised manner, using masked language modeling as the learning signal and backpropagation through a retrieval step that considers millions of documents. This approach exposes the role of world knowledge by asking the model to decide what knowledge to retrieve and use during inference. The effectiveness of REALM pre-training has been demonstrated by fine-tuning on the challenging task of Open-Domain Question Answering (Open- QA). When compared against state-of-the-art models for both external and implicit knowledge storage, REALM outperforms all previous methods by significant margin, while also providing qualitive benefits such as interpretability and modularity. Recent advances in language model pre-training have shown that models such BERT and ROBERTA stores a surprising amount of world knowledge, acquired from the massive text corpora they are trained on. However, to capture more world knowledge implicitly, one must train ever-larger networks, which can be slow or expensive. In contrast, REALM's approach explicitly exposes the role of world knowledge by asking the model to decide what knowledge to retrieve and use during inference. Before making each prediction, the language model uses the retriever documents from large corpus such Wikipedia, and then attends over those documents to help inform its prediction. The key insight of REALM is to train the retriever using a performance-based signal from unsupervised text, and this approach constitutes a significant computational challenge, since the retriever must consider millions of candidate documents for each pre-training step, and the system must backpropagate through the decisions. The selected documents can be formulated as Maximum Inner Product Search (MIPS) [1,3].

RAG in NLP

The advancement in natural language understanding of Retrieval-Augmented Generation (RAG) has significantly improved the generation of conversational responses and the contextual coherence in language generation. RAG leverages external knowledge sources to augment the capabilities of language models, enabling them to generate more accurate and contextually coherent responses. In the con- text of conversation generation, RAG has been instrumental in enhancing the quality of generated responses. By retrieving relevant information from an external knowledge source and conditioning the response generation on this retrieved information, RAG allows for the generation of responses that are not only contextually relevant but also rich in factual information. This has led to a significant improvement in the quality of conversational agents, making them more useful and engaging for users. Furthermore, RAG has also contributed to improving the contextual coherence in language generation. Traditional language models often struggle with maintaining coherence over long passages of text. However, by conditioning the generation process on retrieved information, RAG helps maintain a consistent theme throughout the generated text, thereby improving its contextual coherence [2-5].

Mitigating Hallucinations

The concept of hallucination in the context of Large Language Models (LLMs) and Retrieval Augmented Generation refers to instances where the model generates incorrect, or inconsistent information that deviates from factual foundation. This can occur due the limitations of LLM's training data or when the model fails to properly relate the query with the data required to generate a meaningful response. To resolve hallucinations, the RAG framework is used to fill LLMs with up-to-date and trusted data from a company's internal sources. This allows the LLM to ground its answers in real, internal data, thereby improving accuracy and reducing the occurrence of hallucinations. The process involves retrieving information relevant to a user's query from internal sources, then combining that information with the user's query to create enhanced prompt for the LLM. This helps the LLM generates responses that are more accurate and relevant [2,5,8].

Evaluation of the Prediction of RAG

The evaluation of the prediction of Retrieval Augmented Generation (RAG) model involves assessing the performance of Language models using various methods such as ROUGE-L, cosine similarity, and Language Model Likelihood (LLM) evaluation. ROUGE-L is used to check how closely the model's responses match human answers, providing a measure of the overlap in content. COSINE similarity, on the other hand, checks how well the model's output associates with human references, measuring the cosine of the angle between two vectors to determine their similarity. For LLM evaluation, a binary scoring system is employed by GPT3.5, where score of 0 or 1 is given based on whether the model response and the reference answer conveyed the same idea. this helps to examine how well the model understands and responds to prompts. Also, there ability to use dataset created by human experts to test the models' understanding and response capabilities. Additionally, the fine-tuning methods used for each model, including few shots prompting, Open AI API-based fine-tuning, or using the Hugging- Face Transformer's trainer class. This comprehensive evaluation process ensures the robustness and reliability of the RAG models in various applications [2-5].

Conclusion

The Retrieval Augmented Generation (RAG) represents a significant advancement in the field of Natural Language Processing (NLP). It integrates external data into Language Models, enhancing the generative capabilities of Large Language Models (LLMs) and opening new opportunities in various NLP applications. RAG bridges the gap between Language Model pre-training and Open-QA tasks by incorporating external data through Information Retrieval. This enhances the generative ability of LLMs and addresses limitations inherent in traditional pre-training methodologies. This combination of retrieval-based techniques with Language Model pre-training holds potential for further optimizing Open-QA systems and advancing Natural Language understanding capabilities. The report mentions that the use of pretraining knowledge sources is unsupervised. RAG introduces a knowledge retriever to Language Models, enabling access to a large corpus like Wikipedia, and backpropagating through millions of documents, highlighting the importance of world knowledge. REALM is one of the state- of-the-art models in Open-Domain Question Answering (Open-QA) and offers significant benefits like interpretability and modularity. The report also highlights the essential concepts of RAG, including tokenization, encoding, retrieval, decoding, and the attention mechanism. It emphasizes that RAG powers external knowledge sources to supplement LLMs, resulting in more accurate responses. The report offers a guide on how to create your own RAG system, starting from specifying the library or search engine that will bring in external resources, to defining metrics or parameters to refine the inputs, and finally evaluating the system's answer based on predefined evaluation rules. The report also discusses the role of ChatGPT, a type of LLM, in RAG. While ChatGPT is not essential for RAG, it can be used in conjunction with RAG to create more accurate and contextually relevant responses. The report concludes by discussing the recent advancements in Open-Domain Question Answering (ODQA) brought about by RAG. It highlights the Uni-MS-RAG Framework, a new technology that integrates planning, retrieval, and generation tasks using Large Language Models in a sequence-to-sequence manner. This leads to the building of robust and efficient Question-Answering systems that adapt to specific domains by updating all components during training, thus creating a dynamic and up-to-date conversational AI system. Furthermore, it helps reduce the hallucination of Language Models (LMs). RAG Evaluation employs methods like ROUGE-L, Cosine Similarity, and Language Model likelihood to assess Language Model performance, ensuring robustness and reliability. In summary, these instances highlight the power of RAG and its potential applications for optimizing language models. After a deep study, it can be concluded that RAG is just a framework and the wonder techniques are the LLMs. However, RAG has some limitations. The quality of the generated responses may be impacted by the quality of the incorporated external data. If the data is inaccurate or biased, this could negatively affect the responses. Furthermore, hallucinations remain a challenge because inaccuracies can arise if the input does not contain sufficient information or metrics for evaluation. Future work should focus on enhancing data integration by putting some metrics, educating the prompt query, developing real-time correction mechanisms, and adapting RAG for specific domains [1-5,8-10].

References

1. Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang (2020) REALM: Retrieval-Augmented Language Model Pre-Training. Computer Science https://arxiv.org/abs/2002.08909.
2. Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, et al. (2023) Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. Association for Computational Linguistics 11: 1-17.
3. Liang Zhang, Katherine Jijo, Spurthi Setty, Eden Chung, Fatima Javid, et al. (2024) Enhancing Large Language Model Performance to Answer Questions and Extract Information More Accurately. Computer Science https://arxiv.org/abs/2402.01722.
4. (2024) How to set up Retrieval Augmented Generation (demo) by Don Woodlock post on.
5. Hongru Wang, Wenyu Huang, Yang Deng, Rui Wang, Zezhong Wang, et al. (2024) Uni-MS-RAG: A Unified Multi-source Retrieval-Augmented Generation for Personalized Dia- logue Systems. Computer Science https://arxiv.org/abs/2401.13256.
6. Copilot of Microsoft Bing.
7. How ChatGPT and our language models are developed from www.help.openai.com.
8. SM Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, et al. (2024) A comprehensive survey of hallucination mitigation techniques in large language models. Computer Science https://arxiv.org/abs/2401.01313.
9. Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, et al. (2024) The power of noise: Redefining retrieval for RAG systems. Computer Science https://arxiv.org/abs/2401.14887
10. CHATGPT3.5 developed by OpenAI.