Retrieval-Augmented Generation (RAG) and LLM Integration

Büşra Tural
Research & Development Center, Vakıf Participation
Istanbul, Turkey
busra.tural@vakifkatilim.com.tr

Zeynep Örpek
Research & Development Center, Vakıf Participation
Istanbul, Turkey
zeynep.orpek@vakifkatilim.com.tr

Zeynep Destan
Research & Development Center, Vakıf Participation
Istanbul, Turkey
zeynep.destan@vakifkatilim.com.tr

2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS) | 979-8-3315-4010-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISAS64331.2024.10845308

***Abstract*— Advances in Natural Language Processing (NLP) have led to the emergence of complex structures such as Large Language Models (LLM). LLMs are highly successful in understanding the subtleties of language and processing context by being trained on large datasets. However, the difficulties encountered in Information Retrieval (IR) processes have created an awareness that these models are not sufficient on their own. Traditional IR methods have generally been insufficient in understanding the complexity of natural language in responding to specific queries and retrieving appropriate information from documents or databases. Since this process is based only on keywords, it cannot fully capture the semantic meaning of the language. For this reason, it has been necessary to go beyond traditional IR methods for more precise information creation based on context and meaning. As a result of these requirements, the Retrieval-Augmented Generation (RAG) architecture has come to the fore. RAG offers the ability to create richer and contextually meaningful answers to user queries by integrating LLMs with information retrieval processes. This architecture allows the language model to instantly access external information sources; thus, it generates more accurate and contextual responses armed with existing information. These features of RAG provide appropriate solutions to users' information-based demands by better understanding the complexity of natural language. In this study, it is emphasized that the integration of RAG architecture with information retrieval systems and LLMs provides more sensitive and accurate solutions in information-intensive tasks. This study emphasizes that the RAG architecture's ability to retrieve information by dynamically using the learnings obtained from large datasets of LLMs strengthens applications in the field of NLP.**

Keywords*— Retrieval-Augmented Generation, Large Language Models, Information Retrieval, Natural Language Processing

I. Introduction

In studies conducted on NLP, LLM has demonstrated superior performance compared to other models. LLM has achieved this success with large datasets on which it was trained with larger parameters. Nevertheless, the lack of resources, insufficient current data, and inadequate number of datasets on which LLMs were trained may impede the model's success from reaching the desired level. LLMs are constrained to generating text based on the datasets from which they were trained. Nevertheless, the quantity of data generated on a daily basis, both in the physical world and in the digital domain, is rapidly expanding. The inability of LLMs to adapt to an increase and updates in data has an adverse effect on the success of the model. The content, quantity, and caliber of the data utilized for model training become less significant over time, leading to a decline in model performance. As a preliminary solution, it was proposed that the models be retrained with new data. Nevertheless, retraining the models with each new dataset is not The studies conducted by Gerard Salton in the 1970s constituted the foundation for contemporary IR methodologies. Information retrieval (IR) can be defined as the process of finding keywords within a given text. The process entails identifying the desired message and the sought information through a systematic examination of the text. Salton used TF-IDF to calculate the frequency of terms in the text or document they appear in [1]. This approach has informed the development of new IR models, including the Best Match 25 and Vector Space Model. The objective of IR models is to identify the document that is most relevant to a given query, and this is achieved through the use of techniques such as pre-indexing documents and vectorizing both documents and queries. Despite the efficacy of IR models in identifying the most pertinent document on a given subject, they have not yet reached the desired level of success due to the intricate structure of natural language. It has been demonstrated that IR models are unable to meet the requisite demands in isolation.

The efficacy of generative language models in generating text has declined over time due to the static and limited nature of the datasets. In response to these challenges, the Facebook artificial intelligence research team unveiled an architectural framework, designated as the RAG model, in 2020. The RAG model is based on the integration of text generation capabilities inherent to generative artificial intelligence models with the ability of IR models to identify the most pertinent text. The RAG architecture represents an approach that combines generative models with information retrieval models. The advantageous aspects of the two language models have been combined for a common purpose.

The primary characteristic of the RAG architecture is its capacity to draw upon external sources of information in real time, extending beyond the confines of the dataset utilized during the text generation process. In this manner, the model is distinct from traditional, large language models that are based on static datasets, as it is capable of generating texts that are supported by dynamic, up-to-date information. The RAG model employs IR systems to identify the most pertinent documents for a given query and then synthesizes these documents to generate accurate and comprehensive responses. This approach enables the model to obtain more dynamic and precise results by leveraging data that emerges after the training period. Consequently, the RAG model seeks to

979-8-3315-4010-4/24/$31.00 ©2024 IEEE

address the limitation of LLMs being dependent on a static database.

The RAG architecture has enabled LLMs to ingest not only the dataset they are trained on, but also larger and more current data sources, including the internet and archives. Large language models, such as GPT (Generative Pre-trained Transformer) and Bert (Bidirectional Encoder Representations from Transformers), have been supported by IR models, including VSM (Vector Space Model) and DRM (Dense Retrieval Model). This has enabled them to produce more meaningful and accurate results. The RAG architecture has enhanced the capabilities of LLM by integrating external information sources. The RAG architecture is predominantly employed in the context of complex language models. In particular, the RAG architecture is commonly employed in question-answering models to scan documents and identify the pertinent subject matter, thereby facilitating the generation of responses. In addition, the RAG architecture is employed in the generation of meaningful text, such as blog posts and news articles, due to the enhanced efficacy of generative language models when supported by information retrieval models. The structure of the RAG architecture is also employed in dialogic applications, such as chatbots, as it enables the establishment of a continuous dialogue with the user and the scanning of the internet or large archives for up-to-date answers. As can be observed, RAG architecture is a methodology employed primarily in applications that necessitate current and sophisticated information.

II. Related Works

NLP is a subfield of machine learning and deep learning that deals with the processing and interpretation of language. Transformer-based models (e.g. GPT, BERT) have directly influenced the development of architectures such as RAG. These models have formed the basis for systems that can understand and create human language [2].

Transformer-based models such as BERT and GPT can capture the meaning of language in more depth by being trained on large datasets. While BERT can effectively capture the context of words in text thanks to its bidirectional approach, models such as GPT have become more effective in text generation by using forward language modeling (unidirectional). These models have strengthened the language generation and understanding capabilities that form the basis of systems such as RAG [3] [4].

As the complexity of language models increases, the need for knowledge retrieval and access systems to improve the accuracy of models in knowledge-intensive tasks has also increased. Open-Domain Question Answering (ODQA) systems have been developed to meet this need. These systems scan large document collections and retrieve relevant documents to increase the accuracy of the answer created by the language model when answering a user question. In such systems, instead of directly creating knowledge, language models first retrieve the necessary documents and then create an answer using these documents. Important steps have been taken in the development of ODQA systems. First-generation systems, such as DrQA, access large knowledge bases such as Wikipedia, find text fragments that are relevant to a particular question and create an answer using this information [5] .

RAG-like architectures are based on information retrieval systems. These systems aim to retrieve the most relevant information from large datasets (documents, articles, web pages, etc.) based on a specific query. Traditional information retrieval systems (such as search engines) are based on keywords. However, with the development of artificial intelligence and deep learning methods, more sophisticated systems have emerged [6].

Lewis et al. (2020) introduced their pioneering work, the RAG architecture, to develop a model that improves the performance of LLMs using external data sources in knowledge-intensive natural language processing tasks. This work demonstrates how RAG creates more accurate results by accessing external databases without relying solely on the parameters of language models [7].

In this context, the RAG architecture emerges as a remarkable innovation in natural language processing. This architecture has the capacity to provide more contextual and meaningful answers to users by integrating LLM and information retrieval processes. When the literature is examined, various studies are conducted to understand the effectiveness and application areas of RAG.

RAG architecture integrates with LLM's information retrieval systems. Before answering a question, these systems retrieve relevant documents and use this information to create answers. This provides a great advantage, especially in providing accurate and up-to-date information. RAG is an advanced system with models such as Google's T5 and offers more efficient information creation processes [8].

While RAG combines knowledge retrieval and generation, similar approaches have also been developed. For example, the FiD (Fusion-in-Decoder) architecture combines retrieved documents to create an answer. RePAQ (Retrieval-enhanced Pretrained Autoregressive Query) offers a more compact structure, allowing faster knowledge retrieval and response generation [8] [9].

In his study, Reimers (2019) developed sentence-level embedding techniques to make retrieval-based systems such as RAG work more efficiently. This method is frequently used in the retrieval phase of RAG [10].

The ColBERT system developed by Khattab and Zaharia (2020) has made the document retrieval process more efficient with a BERT-based bidirectional attention mechanism. This model has greatly contributed to the development of systems where retrieval and generative models work together, such as RAG [11].

Nogueira et al. (2020) studied pre-trained sequence-to-sequence models for ranking retrieved documents. This work plays a critical role in the ranking and evaluation processes of retrieved information in RAG systems [12].

Guu, K. et al. (2020) Analyzing the effects of RAG on information retrieval and language modeling, this study provides important findings on real-time information integration [13].

In their work, Karpukhin et al. (2020) examine the transitive document retrieval methods based on the RAG architecture [14].

In their work, Xiong et al. (2021) examine the effects of transformer-based models on information sorting and discuss its relationship with the RAG architecture [15].

Gao et al. (2023) examine the development of the RAG paradigm in their work, addressing the Naive RAG, Advanced

RAG, and Modular RAG models, and analyzing in detail the three core components of RAG systems: retrieval, generation, and augmentation techniques. They also introduce current evaluation frameworks and benchmarks, highlighting the current challenges and future research areas of RAG [16].

In their study, Fan et al. (2024) systematically examine the ability of LLMs integrated with RAG to improve content quality by leveraging external knowledge sources. The research reviews the existing RAG and LLM literature from three main technical perspectives, evaluating the advantages offered by RAG to overcome the models' inherent knowledge constraints and ensure knowledge timeliness; it also discusses current challenges and potential directions for future research [17].

Salemi et al. (2024) propose a new method for evaluating RAG systems, called eRAG, introducing an approach where each retrieved document is individually used by the large language model. eRAG provides more accurate evaluations at the document level while providing higher correlation and significantly less computational resource consumption compared to traditional methods [18].

There are a significant number of studies in the literature on the applications and impacts of RAG architecture. These studies deeply examine the advantages and solutions provided by the integration of RAG with knowledge retrieval processes in natural language processing tasks. For example, the research conducted by Lewis et al. (2020) reveals how the RAG architecture creates more effective answers in knowledge-intensive NLP tasks [7].

Studies conducted on the RAG architecture in the literature reveal the potential and effectiveness of this system in information-intensive natural language processing tasks. RAG's ability to create more contextual and accurate answers by combining large language models with information retrieval processes is supported by various studies. In the future, further development of this architecture will contribute to the further strengthening of applications in the field of information retrieval systems and natural language processing. Therefore, the RAG architecture stands out as an important innovation in terms of accelerating and increasing the accuracy of information-based processes, and research in this area will enable the development of more reliable and effective systems. Such studies are critical to understanding the interaction between information retrieval and language creation in RAG, and research conducted in this context plays an important role in the future development of the field.

III. Language Models and RAG Architecture

The intense interest in NLP and the artificial intelligence ecosystem, which includes a large number of developers, has provided the basis for the spread of new technologies and architectures. Language models trained on large datasets stand out with their human-like abilities, such as creating text, editing, answering questions, summarizing, and translating by learning the mathematical structure of the language.

Language models work by probabilistically modeling the distribution of words in sentences and their use together, and thus can predict the next word.

Language models are classified according to the dataset they are trained on and the parameter size used. While LLMs contain more than 100 million parameters, small language models (SLM) contain less than 100 million parameters.

LLMs face difficulties in accessing accurate and up-to-date information during the real-time usage processes of the model when they are trained on huge datasets. Language models can create answers based on the dataset on which they are trained. This requires the model training process to be run again after a data update in the dataset. This process is insufficient in terms of time benefit and cost when the LLM training process and costs are taken into consideration.

The training process of LLMs raises serious concerns about sustainability and a green environment. The equipment used in model training are devices with high energy requirements and operate with high energy consumption for long periods. This situation creates a significant environmental impact in terms of sustainability and causes excessive consumption of energy resources. The necessity of retraining the model with each new data update turns into a process that harms the environment and has negative effects on energy resources over time. Alternative and more sustainable solutions have been sought for this problem.

RAG architecture has become widespread as a flexible and robust architecture against changes in the information in the dataset that is effective in the IR processes of LLMs. An update on the data in the dataset provides access to up-to-date data in real-time without the need to retrain the language model. RAG architecture prevents the model from being dependent on fixed data with the IR layer and enables the creation of responses based on dynamic data. With these features, RAG architecture has a flexible and robust structure against changing data.

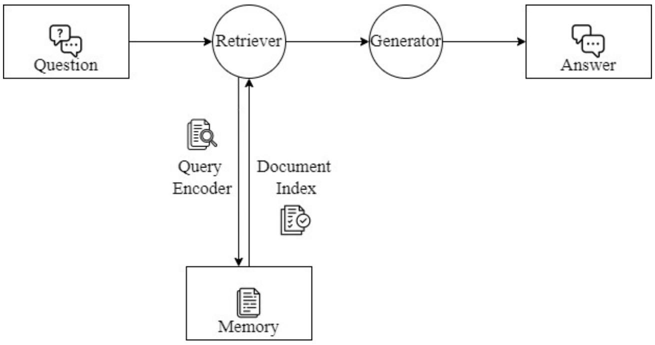

Fig. 1. RAG Architecture.

RAG architecture consists of Retrieval Document Search, Augmentation, and Generation stages. These stages are as follows.

- Retrieval Document Search: This stage includes the process of finding and retrieving documents related to the question asked by the user from the dataset.
- Augmentation: This stage includes the processes of making sense of the documents returned regarding the question, increasing the value of the data by adding additional data and documents and updating the data. In this way, the answer is created by adding not only the trained field but also the current data.
- Generation: This stage covers the process of producing the final answer to the question asked by the user. Answers to the relevant question are

created through data collected from different channels.

The RAG architecture is used as the basis for many architectures customized to different needs and domains. Its main variations are Standard RAG, Corrective RAG, Speculative RAG, Fusion RAG, Agentic RAG, Self RAG, Graph RAG, Modular RAG, and RadioRAG.

Standard RAG, the basis for other variations of this architecture, is highly successful in question-and-answer systems, and in summarizing large texts. Despite its widespread use, it may fail to retrieve data related to the user's question in the IR step. This may cause the answer created to be incorrect or insufficient.

Corrective RAG proposes to add an additional verification layer to check and correct the accuracy of the generated answer after the IR and answer generation stages. In this way, in cases where the answer created by standard RAG is incomplete or insufficient, it can create more successful answers because it includes a re-improvement and answer generation phase [19].

Speculative RAG offers a solution to provide correct answers in cases where the returned data is insufficient. It involves the process of the model predicting the answer based on the information in the returned data and other information in the language model. However, the answer created as a result of these studies may not be correct [20].

Fusion RAG is an architecture that allows a holistic response to be created from data obtained by collecting data from different sources. In particular, it aims to create successful outputs by combining relevant data in cases where data from different data sources contradict each other. However, one of the biggest challenges to be encountered is when there is a lot of data that contradict each other, in which case it becomes difficult to ensure accuracy in the response [21].

Agentic RAG enables the model to decide independently which type of data it needs. In this way, it adds decision-making ability to the model, allowing the model to be used by prioritizing the most appropriate data in case of different types of data. However, an error or failure in the prediction mechanism can directly affect the answer to be created and may cause incorrect outputs to be created [22].

Self RAG stands out with its feature of evaluating the model's performance. The model contributes to the model's consistency with the dataset by evaluating the quality of the answer it creates while producing an answer to the relevant question. However, since the model's performance evaluation depends on the accuracy of the data, it will create wrong answers if the data is incorrect, incomplete, or insufficient [23].

Graph RAG enables understanding and organizing information through relationships by incorporating graph-based data structures into IR processes. It is suitable for use in areas that require relational understanding, such as biological research in understanding the relationships between genes, proteins, and diseases. However, outdated, incorrect, or incomplete graphs affect the accuracy of the answer to be created. Therefore, graph structures must be kept correct and up-to-date [24].

Modular RAG is an approach to optimizing all components separately and independently. The process of separating them into modules makes the system more flexible and customizable. In this way, improvements and fine-tuning can be done only to a certain module. However, it can be difficult to ensure that different modules work seamlessly and are fully compatible with each other [25].

Radio RAG was developed to integrate real-time and radiology information into LLM. It was tested using a dataset called RadioQA and strengthened LLM’s ability to diagnose diseases with real-time radiological information. According to the results of the tests conducted, it was observed that some models increased the diagnostic accuracy by up to 54%. The test results demonstrate the potential of Radio RAG to improve and change disease diagnosis processes [26].

Each variation of the RAG architecture has been shaped according to different needs and challenges in different areas. The standard RAG is the basis for most architectures. On the other hand, widely used and known variations such as Corrective RAG, Speculative RAG, Fusion RAG, Agentic RAG, Self RAG, Graph RAG, Modular RAG, and RadioRAG have been developed to provide architecture suitable for the main requirements. As these models develop and prove their success more accurately and contextually, their developer base and usage areas will increase day by day.

IV. Result

The RAG architecture is an advanced solution that overcomes the current limitations of LLMs, offering significant advantages in effects-based missions. The shortcomings of LLMs, such as the inability to pass training data and the inaccessibility of external data sources, are effectively addressed by RAG's IR data forwarding. Thanks to this architecture's ability to pull information from external data sources, more contextual, up-to-date, and highly accurate solutions can be created. Especially for applications with large datasets and updated data, RAG revolutionizes the field of NLP by providing more efficient and flexible partitioning of language models. RAG has provided a solution to LLM's concerns about maintainability. The model of ensuring that datasets are updated has been an alternative solution to the problem of re-training process. Architectures like RAG are everywhere, with rich contributions from the accessibility of IR and language partitions, sustainability information, and smarter and more dynamic systems.

Acknowledgment

This research was made possible thanks to the support and infrastructure provided by Vakıf Participation Bank R&D Center. The valuable contributions of Vakıf Participation Bank R&D Center were effective in the successful completion of this study. We would like to express our sincere gratitude to Vakıf Participation Bank R&D Center for their support.

References

[1] K. Spärck Jones, " IDF term weighting and IR research lessons," *Journal of documentation,* pp. 60(5), 521-523., 2004.

[2] A. Vaswani, "Attention is all you need.," *Advances in Neural Information Processing Systems,* 2017.

[3] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805.,* 2018.

[4] J. &. R. S. Howard, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146.,* 2018.

[5] D. Chen, "Reading Wikipedia to answer open-domain questions.," *arXiv preprint arXiv:1704.00051.,* 2017.

[6] C. D. Manning, "Introduction to information retrieval.," 2008.

[7] P. P. E. P. A. P. F. K. V. G. N. .. &. K. D. Lewis, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems,* pp. 33, 9459-9474., 2020.

[8] G. &. G. E. Izacard, "Leveraging passage retrieval with generative models for open domain question answering," *arXiv preprint arXiv:2007.01282.,* 2020.

[9] P. W. Y. L. L. M. P. K. H. P. A. .. &. R. S. Lewis, "PAQ: 65 million probably-asked questions and what you can do with them," *Transactions of the Association for Computational Linguistics,* pp. 9, 1098-1115., 2021.

[10] N. (. Reimers, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.," *arXiv preprint arXiv:1908.10084.,* 2019.

[11] O. &. Z. M. Khattab, "Colbert: Efficient and effective passage search via contextualized late interaction over bert.," *In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information,* pp. Retrieval (pp.39-48), 2020, July.

[12] F. N. R. &. L. R. (. .. B. 2. R. G. B. ,. P. .. S. Souza, "BERTimbau: pretrained BERT models for Brazilian Portuguese. In Intelligent Systems: 9th Brazilian Conference,," *Springer International Publishing.,* pp. Part I 9 (pp. 403-417), October 20–23, 2020.

[13] K. L. K. T. Z. P. P. &. C. M. Guu, "Retrieval augmented language model pre-training.," *In International conference on machine learning,* vol. PMLR, pp. pp. 3929-3938, 2020, November.

[14] V. O. B. M. S. L. P. W. L. E. S. .. &. Y. W. T. Karpukhin, "Dense passage retrieval for open-domain question answering.," *arXiv preprint arXiv:2004.04906.,* 2020.

[15] A. N. R. &. L. J. Yates, "Pretrained transformers for text ranking: BERT and beyond.," *In Proceedings of the 14th ACM International Conference on web search and data mining,* pp. pp. 1154-1156, 2021, March.

[16] Y. X. Y. G. X. J. K. P. J. B. Y. .. &. W. H. Gao, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997.,* 2023.

[17] W. D. Y. N. L. W. S. L. H. Y. D. .. &. L. Q. Fan, "A survey on rag meeting llms: Towards retrieval-augmented large language models.," *In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* pp. pp. 6491-6501, 2024, August.

[18] A. &. Z. H. Salemi, "Evaluating retrieval quality in retrieval-augmented generation," *In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval ,* pp. pp. 2395-2400, 2024, July.

[19] S. Q. G. J. C. Z. Y. &. L. Z. H. Yan, "Corrective retrieval augmented generation.," *arXiv preprint arXiv:2401.15884.,* 2024.

[20] Z. W. Z. L. L. Z. H. S. M. S. P. V. .. &. P. T. Wang, "Speculative rag: Enhancing retrieval augmented generation through drafting.," *arXiv preprint arXiv:2407.08223.,* 2024.

[21] Z. Rackauckas, "Rag-fusion: a new take on retrieval-augmented generation.," *arXiv preprint arXiv:2402.03367.,* 2024.

[22] C. S. S. S. &. R. V. Ravuru, "Agentic Retrieval-Augmented Generation for Time Series Analysis.," *arXiv preprint arXiv:2408.14484.,* 2024.

[23] A. W. Z. W. Y. S. A. &. H. H. Asai, "Self-rag: Learning to retrieve, generate, and critique through self-reflection.," *arXiv preprint arXiv:2310.11511.,* 2023.

[24] B. Z. Y. L. Y. B. X. S. H. H. C. .. &. T. S. Peng, "Graph retrieval-augmented generation: A survey.," *arXiv preprint arXiv:2408.08921.,* 2024.

[25] Y. X. Y. W. M. &. W. H. Gao, "Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks.," *arXiv preprint arXiv:2407.21059.,* 2024.

[26] S. T. L. M. B. K. S. R. F. D. K. C. .. &. T. D. Arasteh, " RadioRAG: Factual Large Language Models for Enhanced Diagnostics in Radiology Using Dynamic Retrieval Augmented Generation.," *arXiv preprint arXiv:2407.15,* 2024.