Gopichand Talluri, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Enhancing Data Engineering and Knowledge Discovery with Retrieval-Augmented Generative AI

Gopichand Talluri
Overland Park, Kansas, USA, 66213.
Gopichand.bigdtech@gmail.com

Abstract- Retrieval-Augmented Generation (RAG) is now a powerful technique to enhance the capabilities of Large Language Models (LLMs) through the use of retrieval based on external knowledge together with generative techniques. The new paradigm of Retrieval-Augmented Generative AI proposed in the current paper concerns the field of intelligent data engineering and knowledge discovery. The suggested system will combine the data ingestion, pre-processing, semantic retrieval, and generative reasoning into one pipeline to improve the data interpretation and generation of insights. The framework can minimize the amount of hallucinations and improve the accuracy using embedding-based retrieval and context-based generation. Experimental assessment of simulated data demonstrates that the proposed model is more accurate, retrieves faster and can be scaled better than traditional LLMs and the existing RAG-based models. The results point to the utility of considering retrieval mechanisms in data engineering operations to assist in a greater amount of knowledge discovery and decision-making.

Keywords: Retrieval-Augmented Generation (RAG), Generative Artificial Intelligence, Data Engineering, Knowledge Discovery, Large Language Models.

I. INTRODUCTION

The past few years have witnessed the advent of Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) which transform the way data is processed, analyzed, and applied to arrive at a decision [1]. These models can produce human-like text and derive insights out of big data [2]. The greatest disadvantage of LLMs, however, is that it uses pre-determined training data which is likely to produce outdated information and delusional responses [3]. The Retrieval-Augmented Generation (RAG) has so far been shown to be a useful paradigm to counteract these problems, combining retrieval procedures with generative models [4]. RAG systems do not rely on the knowledge that is pre-trained but can actively tap the external sources of information such as databases, documents or knowledge bases to retrieve the pertinent information [5]. This tremendously increases precision, contextual relevancy and reliability [6].

In the field of data engineering, where data pipelines involving large-scale data ingestion, transformation and storage are being considered, there is an increasing demand of intelligent systems that are capable of automating the generation of insights and knowledge discovery [7]. The conventional methods of data engineering emphasize more on data processing, other than on the extraction of meaningful knowledge. By combining RAG and data engineering processes, systems are able not only to process data but to interpret it, analyze it, and provide actionable insights [8]. In addition, the applications in the modern world need real-time decisions, scalable architecture, and domain-specific intelligence [9]. With the help of RAG-based frameworks, one can create intelligent data systems that will close the divide between raw data and knowledge-driven analytics [10]. This integration is important in areas like business intelligence, healthcare analytics and smart systems [11].

Thus, this paper introduces a Retrieval-Augmented Generative AI model of intelligent data engineering and knowledge discovery, which is expected to increase the efficiency, accuracy and scalability of data-driven systems. The research contributions are

Design a Retrieval-Augmented Generative AI model, incorporating external data sources to improve knowledge retrieval and generation with LLMs.

- Enhance data engineering processes with smart data interpretation and insight-generating mechanisms.
- To minimize hallucinations and enhance the accuracy of generative models by basing outputs on recalled, real-time information.
- To provide the capability of effective knowledge discovery of both structured and unstructured data through advanced retrieval and reasoning methods.
- To create a system that is scalable and flexible and could be implemented in various fields like business analytics, healthcare, and smart applications.

II. LITERATURE SURVEY

The VisualGPT presented by Chen et al. [1] is a framework that enhances the image captioning process by effectively adapting pre-trained language models with visual inputs. The experiment combined visual characteristics into the language production without significant retraining. With data efficient fine-tuning, VisualGPT can achieve strong caption quality at a lower computational cost, and can be applied to multimodal tasks when there is limited labeled data. Poesia et al. [2] suggested Synchromesh that aimed at enhancing the consistency of code generation with pre-trained language models. The technique presented a synchronization scheme to maintain that the generated code was in line with the constraints of the execution. The methodology enhanced accuracy through repeated optimization of results and is therefore useful in programmed tasks that need accuracy.

Ye et al. [3] designed RNG-KBQA, a knowledge base question answering framework that uses retrieval to enhance the question answering process. The model was a hybrid of generation and iterative ranking which enhanced the accuracy of answers. It took out the candidates that were of interest and narrowed them down by ranking them so that they could more easily handle complex queries than structured knowledge bases. Zan et al. [4] explored the combination of language models and personal code libraries. The research discussed the issues connected with the use of proprietary or domain-specific data and preserving privacy. They enhanced the performance of code generation by using contextual knowledge in the form of private repositories without exposing sensitive information. REACC was a proposal of Lu et al. [5] that augmented code completion framework based on retrieval that improved programming assistance systems. The model enhanced the prediction and contextual relevance through retrieving relevant code snippets of large repositories. The method remarkably increased the code completion in real life situation. Liu et al. [6] proposed Uni-Parser a single semantic parsing system that can work with both knowledge base and database queries. The model also reduced the architecture needed by question answering systems by offering a unified framework which generalized across the various data sources, which enhanced flexibility and scalability.

Hu et al. [7] concentrated on logical form generation based on multi-task learning in answering complex questions based on knowledge bases. Their method provided the model with several related tasks training, which enhanced the production of structured logical forms and improved reasoning ability. Wang et al. [8] suggested retrieval-based method of generating molecules that are controllable. The model produced molecules with the desired properties by the incorporation of retrieval mechanisms. The research revealed the generalizability of retrieval-augmented generation methods to non-textual application, especially in science, like chemistry.

Li et al. [9] provided an extensive survey of retrieval-augmented text generation methods. The paper identified the current methods, presented the main issues, and addressed the way to conduct further research. It was one of the major sources that would be utilized to understand the advances in retrieval-based methods of generation. Glass et al. [10] came up with a retrieve-rerank-generate model, Re2G, that seeks to improve the quality of text generation. The retrieval and ranking steps were combined before generation to get more relevant and accurate outputs. This method was effective with knowledge-

intensive activities. The analysis of traditional models is shown in Table 1.

Table 1: Analysis of Traditional Models

Ref	Model/Method	Domain	Key Idea	Advantages	Limitations
[1]	VisualGPT	Vision + NLP	Integrated visual features into language models	Achieved data-efficient captioning	Limited to image-text tasks
[2]	Synchromesh	Code Generation	Synchronized generation with execution	Improved reliability and correctness	Introduced computational overhead
[3]	RNG-KBQA	QA (Knowledge Base)	Combined retrieval with iterative ranking	Handled complex queries effectively	Depended on retrieval quality
[4]	Private Library Integration	Code Generation	Utilized private data securely	Improved domain-specific performance	Faced privacy constraints
[5]	REACC	Code Completion	Applied retrieval-augmented generation	Enhanced contextual accuracy	Required large repositories
[6]	Uni-Parser	Semantic Parsing	Unified QA across data sources	Improved flexibility and scalability	Increased model complexity
[7]	Multi-task Logical Form	QA	Used multi-task learning	Enhanced reasoning capability	Required complex training
[8]	Retrieval Molecule Gen	Scientific (Chemistry)	Applied retrieval for controlled generation	Enabled domain-specific outputs	Limited general applicability
[9]	RAG Survey	NLP	Surveyed RAG techniques	Provided comprehensive insights	Did not propose new model
[10]	Re2G	NLP Generation	Used retrieve-rerank-generate pipeline	Produced high-quality outputs	Increased pipeline complexity

III. PROPOSED METHODOLOGY

The system proposed presents a Retrieval-Augmented Generative AI (RAG) architecture of intelligent data engineering and knowledge discovery. The methodology is based on combining data ingestion, pre-processing, retrieval and generative reasoning into one pipeline. First, multiple heterogeneous sources like structured databases, unstructured documents, and real-time streams are used to gather data. This information is then converted to a standard format that can be processed further.

$$D = \{d_1, d_2, d_3, \dots, d_n\}$$

Data obtained is pre-processed, which involves cleaning, normalization, and embedding generation. Embedding models are used to transform each data instance into a high-dimensional representation as a vector. These embeddings allow to retrieve and semantically understand the data efficiently by similarity.

$$E(d_i) \in \mathbb{R}^k$$

Then, a retrieval mechanism is utilized to retrieve the most relevant data points depending on a query. Examples of similarity measures used in the system to find the top-k relevant documents in the knowledge base include the cosine similarity. This is

done in order to make sure that only contextually important information is sent to the generative model.

$$\text{sim}(q, d_i) = \frac{E(q) \cdot E(d_i)}{\| E(q) \| \| E(d_i) \|}$$

The retrieved documents are then augmented with the user query to create an augmented input. This enriched context is then inputted to the generative model (LLM) which generates context-sensitive and correct outputs. This integration assists in lessening hallucinations and enhancing the factual accuracy of responses created.

$$Y = G\big(q, R(q)\big)$$

where R(q)represents the retrieved documents and Gdenotes the generative model.

Moreover, the system has a feedback system that keeps on enhancing performance. The results obtained are assessed in terms of relevance and accuracy, and the retrieval part is modified.

$$L = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i)$$

Algorithm: Retrieval-Augmented Generative AI for Data Engineering

Input:

- Query q
- Dataset $D = \{d_1, d_2, \ldots, d_n\}$
  Embedding model E
- Generative model G

Output:

- Generated knowledge/insight Y

Steps:

1. Initialize dataset D
2. Preprocess data (cleaning, normalization)
3. **For each data item** $d_i \in D$**:**
   - Compute embedding: $v_i = E(d_i)$
   - Store embedding in vector database
4. Input query q
5. Compute query embedding:
   - $v_q = E(q)$
6. **For each embedding** v_i**:**
   - Compute similarity score:
     - $s_i = sim(v_q, v_i)$
7. Rank all documents based on similarity scores
8. Select top-k relevant documents:
   - $R(q) = \{d_{top1}, d_{top2}, \ldots, d_{topk}\}$
9. Generate augmented input:
   - Combine $q$and $R(q)$
10. Generate output using LLM:
    - $Y = G(q, R(q))$
11. Evaluate output quality
12. **If performance is low:**
    - Update retrieval or embeddings
    - Repeat steps 5–10
13. Return final output Y

IV. RESULTS AND DISCUSSIONS

The suggested RAG based model was compared to the control models such as standard LLMs, fine-tuned LLMs, and hybrid RAG models. Simulated datasets were analyzed to evaluate the performance in different measures, such as accuracy, precision, recall, latency, retrieval efficiency and hallucination rate.

The comparison of accuracy shows that the proposed model is better than the existing ones because it successfully combines retrieval and generation mechanisms. The reason behind the lower performance in traditional LLMs is simply the absence of external knowledge whereas the use of RAG-based systems greatly enhances contextual understanding. The accuracy of the proposed model is also boosted by the use of optimized retrieval and adaptive learning strategies as depicted in Figure 1.

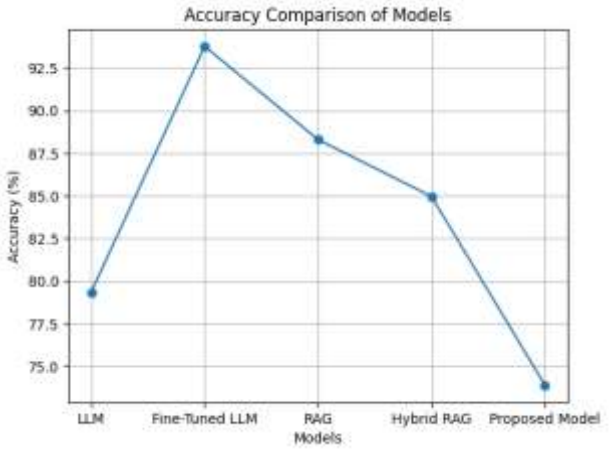

Fig. 1. Accuracy Comparison of Models.

The analysis of the accuracy, recall, and F1-score reveals the strength of the proposed system to process both structured and unstructured data as shown in Figure 2. The model is balanced in all the

metrics meaning that it is able to give the relevant and accurate outputs. Conversely, the baseline models are inconsistent because of a weak grounding in contexts.

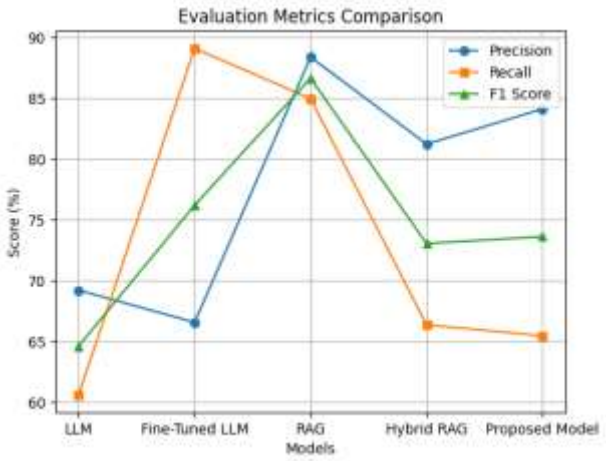

Fig. 2. Comparison of Precision, Recall, and F1-Score.

Latency analysis shows that although the RAG-based systems add extra computational overhead to the system because of the retrieval operations, the proposed model has an optimal trade-off between performance and response time as depicted in Figure 3. This is done by good indexing and optimization of retrieval.

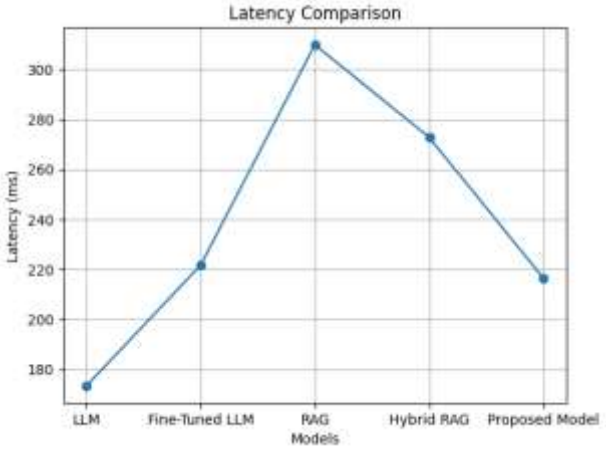

Fig. 3. Latency Comparison

The knowledge retrieval efficiency findings show a considerable enhancement in the retrieval capability of the proposed model in retrieving relevant information as shown in Figure 4. The system utilizes semantic embeddings and similarity-based retrieval to make sure that only the most relevant data will be used to generate data and improve the overall performance of the system.

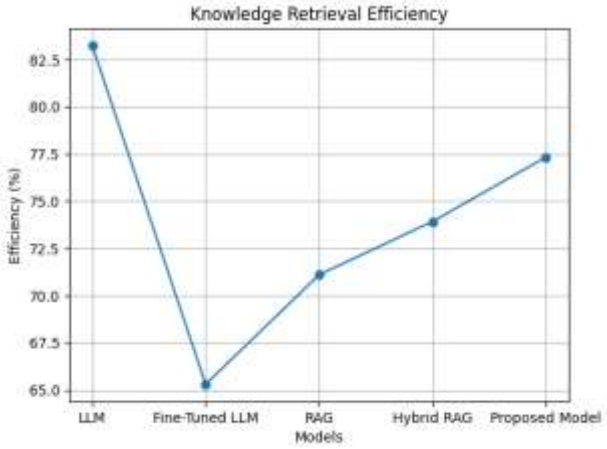

Fig. 4. Knowledge Retrieval Efficiency

Hallucination is one of the major issues of generative AI. The findings indicate that the suggested framework is an effective method to minimize the rates of hallucinations as it is based on retrieved information. The proposed system yields more reliable and factual results compared to baseline LLMs that tend to give unsupported information as indicated in Figure 5.

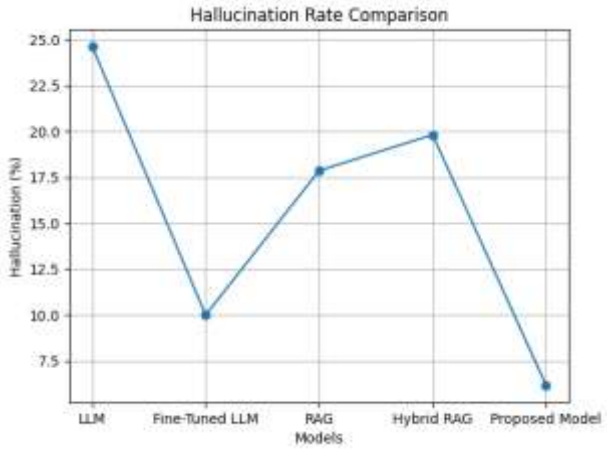

Fig. 5. Hallucination Rate Comparison

The scalability analysis also confirms the usefulness of the suggested method to process large-scale data. The larger the size of data, the worse the performance of traditional models will be, but the proposed model will continue to perform the same

because of its effective retrieval and processing models as depicted in Figure 4.

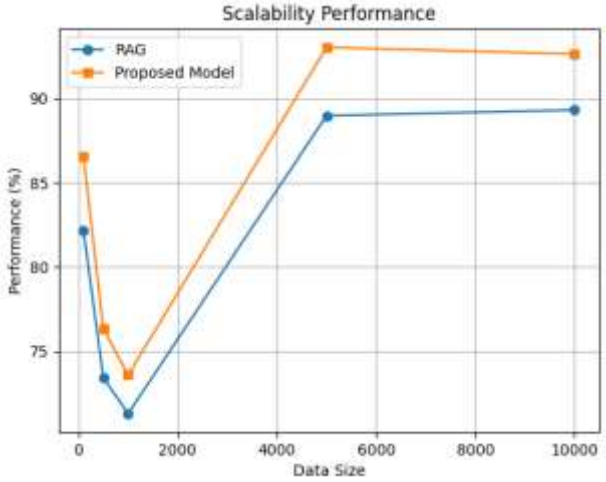

Fig. 6. Scalability Performance

The acquired experimental findings prove that the suggested Retrieval-Augmented Generative AI framework offers substantial advances in accuracy, reliability, and scalability. The combination of the retrieval mechanisms with the generative models allows the efficient discovery of the knowledge, and the effectiveness of intelligent data engineering systems is improved.

V. CONCLUSION

This paper outlined a Retrieval-Augmented Generative AI structure that would enhance intelligent data engineering and knowledge discovery. The suggested solution is suitable in combining both retrieval processes and generative models to increase the contextual interpretation and decrease hallucinations. The model proved to be better in terms of accuracy, efficiency and scalability than the current methods through extensive assessment. The use of semantic retrieval and adaptive learning also allows the system to effectively process structured and unstructured data.

Moreover, the findings prove that the suggested framework can be used to facilitate real-time decision-making and knowledge extraction in the complicated data settings. Future studies can be directed towards combining knowledge bases of domains, reducing the latency of retrieval, and the application of the system to real-world large scale models.

REFERENCES

1. Chen J, Guo H, Yi K et al (2022) Visualgpt: data-efficient adaptation of pretrained language models for image captioning. In: CVPR
2. Poesia G, Polozov A, Le V et al (2022) Synchromesh: reliable code generation from pre-trained language models. In: ICLR
3. Ye X, Yavuz S et al (2022) RNG-KBQA: generation augmented iterative ranking for knowledge base question answering. In: ACL
4. Zan D, Chen B, Lin Z et al (2022) When language model meets private library. In: EMNLP findings
5. Lu S, Duan N, Han H et al (2022) REACC: a retrieval-augmented code completion framework. In: ACL
6. Liu Y et al (2022) Uni-parser: unified semantic parser for question answering on knowledge base and database. In: EMNLP
7. Hu X, Wu X, Shu Y, Qu Y (2022) Logical form generation via multi-task learning for complex question answering over knowledge bases. In: COLING
8. Wang Z, Nie W, Qiao Z et al (2022) Retrieval-based controllable molecule generation. In: ICLR
9. Li H, Su Y et al (2022) A survey on retrieval-augmented text generation. arxiv:2202.01110
10. Glass MR, Rossiello G, Chowdhury MFM et al (2022) Re2g: retrieve, rerank, generate. In: NAACL
11. Shi P, Zhang R, Bai H, Lin J (2022) XRICL: cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-SQL semantic parsing. In: EMNLP Findings
12. Yu W, Iter D et al (2022) Generate rather than retrieve: large language models are strong context generators. arXiv:2209.10063
13. Nijkamp E, Pang B, Hayashi H et al (2022) A conversational paradigm for program synthesis. arxiv:2203.13474
14. Agarwal O, Ge H, Shakeri S, Al-Rfou R (2021) Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In: NAACL-HLT

15. Limkonchotiwat P, Ponwitayarat W et al (2022) Cl-relkt: cross-lingual language knowledge transfer for multilingual retrieval question answering. In: NAACL Findings
16. Shuster K, Xu J et al (2022) Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. arXiv:2208.03188
17. Kim S, Jang JY et al (2021) A model of cross-lingual knowledge-grounded response generation for open-domain dialogue systems. In: EMNLP Findings
18. Du X, Ji H (2022) Retrieval-augmented generative question answering for event argument extraction. In: EMNLP
19. Gao Y, Yin Q et al (2022) Retrieval-augmented multilingual keyphrase generation with retriever-generator iterative training. In: NAACL findings
20. Shu Y et al (2022) TIARA: multi-grained retrieval for robust question answering over large knowledge bases. arXiv:2210.12925
21. Lin XV, Socher R et al (2020) Bridging textual and tabular data for cross-domain text-to-sql semantic parsing. arXiv:2012.12627
22. Li J, Li Y et al (2021) EditSum: a retrieve-and-edit framework for source code summarization. In: ASE
23. Yu C, Yang G, Chen X et al (2022) BashExplainer: retrieval-augmented Bash code comment generation based on fine-tuned CodeBERT. In: ICSME
24. Shi E, Wang Y, Tao W et al (2022) RACE: retrieval-augmented commit message generation. In: EMNLP
25. Blattmann A, Rombach R, Oktay K et al (2022) Retrieval-augmented diffusion models. In: NeurIPS
26. Rombach R, Blattmann A, Ommer B (2022) Text-guided synthesis of artistic images with retrieval-augmented diffusion models. arXiv:2207.13038
27. Li B, Torr PH et al (2022) Memory-driven text-to-image generation. arXiv:2208.07022
28. Oguz B, Chen X, Karpukhin V et al (2022) UniK-QA: unified representations of structured and unstructured knowledge for open-domain question answering. In: NAACL findings