ENHANCING RETRIEVAL - AUGMENTED GENERATION WITH GRAPH-BASED RETRIEVAL AND GENERATIVE MODELING

Dejan Vujić*,
[0009-0008-2037-5068]
Angelina Njeguš,
[0000-0001-8682-7014]
Nebojša Bačanin Džakula
[0000-0002-2062-924X]

Singidunum University,
Belgrade, Serbia

Abstract:

This paper presents the design and implementation of a robust Retrieval-Augmented Generation (RAG) system that integrates advanced retrieval, ranking, and generative techniques to address knowledge-intensive tasks. The system combines dense retrieval using ChromaDB, metadata-driven keyword extraction with YAKE and KMedoids algorithm for clustering keywords, graph-based retrieval leveraging PageRank, and cross-encoder re-ranking to deliver precise and contextually relevant results. These retrieval outputs are synthesized into high-quality conversational responses using Hugging Face models and Google API. A modular pipeline ensures scalability, seamlessly integrating various retrieval and generative components. Evaluation results demonstrate high retrieval precision, improved recall through graph-based methods, and enhanced response quality through structured prompt engineering. This work highlights the effectiveness of combining diverse techniques in RAG systems, offering a foundation for scalable, reliable, and context-aware applications in domains such as customer support, education, and research.

Keywords:

Retrieval-Augmented Generation, Dense Retrieval, Re-Ranking, Graph-Based Retrieval Keywords, Generate Modeling.

INTRODUCTION

The rapid growth of artificial intelligence (AI) has brought about significant advancements in natural language processing (NLP). Among the most transformative technologies is Retrieval-Augmented Generation (RAG), a hybrid approach that combines the strengths of retrieval systems with generative models. By integrating these paradigms, RAG enables more accurate and contextually relevant responses, addressing limitations inherent in traditional generation-only or retrieval-only methods. As highlighted in recent studies, RAG has proven effective in improving accuracy and contextuality in tasks such as question-answering and conversational agents [1].

RAG finds applications across diverse domains, including conversational agents, knowledge base construction, and personalized content generation. For instance, OpenAI's and Meta's research shows that augmenting generative language models with external retrieval improves factual consistency and reduces hallucinations in generated outputs [2].

Correspondence:
Dejan Vujić

e-mail:
dejan.vujic.24@singimail.rs

DOI: 10.15308/Sinteza-2025-3-9

The intersection of retrieval and generation poses unique challenges and opportunities for software engineering. Implementing a robust RAG system requires careful consideration of data management, model integration, and performance optimization. For example, designing pipelines to handle large-scale retrieval while maintaining low latency is critical [3]. Similarly, advancements in transformer architectures and attention mechanisms have been instrumental in optimizing RAG-based workflows [4].

This paper discusses a Python-based RAG project's architecture, underlying models, and implementation details. By providing a detailed exploration of the project, this work contributes to understanding how retrieval and generative systems can be effectively combined, offering valuable insights for AI researchers and software engineers.

2. RETRIEVAL-AUGMENTED GENERATION OVERVIEW

Retrieval-Augmented Generation (RAG) is a hybrid framework that integrates information retrieval with text generation to tackle knowledge-intensive tasks. The approach was first introduced by [1], who demonstrated its ability to improve the quality of generated text by incorporating relevant retrieved documents into the generation process. Unlike traditional generative models that rely solely on pre-trained parameters, RAG accesses external knowledge sources, such as document collections or databases, enabling more factual and contextually accurate outputs.

This framework leverages two key components: a retriever and a generator. The retriever identifies the most relevant documents from a knowledge base, while the generator incorporates the retrieved documents to produce responses. Recent advancements in dense retrieval methods, such as Dense Passage Retrieval (DPR) [3], have further enhanced the retrieval component by enabling semantic matching of queries and documents in vector space.

RAG has been compared to other architectures that address knowledge-intensive tasks. These include:

- Open-Domain Question Answering (ODQA): Systems like DrQA [5] rely on retrieval followed by extractive reading, limiting their ability to generate free-form responses. In contrast, RAG's generative component allows for more nuanced and diverse outputs.
- Knowledge Graph-Based Systems: Approaches leveraging structured knowledge graphs (KGs) provide accurate responses by querying graph nodes. While effective for tasks requiring structured data, KGs often lack the coverage and scalability of unstructured text retrieval used in RAG [6].
- Memory-Augmented Neural Networks: Models like Memory Networks [7] and Neural Turing Machines [8] integrate memory for knowledge storage. These approaches, while powerful, are constrained by the size and scope of the memory, making RAG's ability to query external sources more scalable.

RAG's ability to dynamically query and generate content has significantly advanced state-of-the-art performance in tasks such as open-domain question answering [1], dialogue systems [2], and document summarization [4]. The combination of dense retrieval and transformer-based generation has set a new benchmark, bridging the gap between static knowledge representation and dynamic generation.

3. RESEARCH BACKGROUND

In recent years, Retrieval-Augmented Generation (RAG) has emerged as a powerful framework that enhances the capabilities of generative models by integrating external knowledge sources into the generation process. This approach addresses the limitations of purely generative models, such as hallucinations and factual inaccuracies, by retrieving relevant context from large-scale knowledge bases. Among various advancements in this domain, graph-based retrieval techniques have gained significant attention due to their ability to capture complex relationships within data, providing richer and more relevant context for generation tasks.

GRAG (Graph Retrieval-Augmented Generation) was developed to enhance both the retrieval and generation processes by emphasizing subgraph structures and maintaining awareness of graph topology to generate contextually coherent responses. The framework has demonstrated superior performance over existing RAG methods in multi-hop reasoning tasks on textual graphs, effectively mitigating hallucinations and improving response quality. [9]

The potential of Large Language Models (LLMs) for materials design has been demonstrated through the integration of retrieval-augmented ontologic graphs and

multi-agent strategies. This approach supports engineering analysis and knowledge generation by leveraging structured retrieval mechanisms, facilitating effective information retrieval and code generation for simulation purposes. [10]

A comprehensive overview of Graph RAG methodologies has been presented, formalizing workflows and discussing core technologies, applications, and future research directions. These advancements highlight the potential of Graph RAG to enhance LLM outputs by leveraging structural information in graphs, leading to more accurate and context-aware responses. [11]

A graph-driven generative model has been proposed to integrate semantic and neighborhood information for optimizing document retrieval. This approach effectively addresses the need for fast retrieval and a small memory footprint. Experimental results demonstrate superior performance over state-of-the-art methods in document hashing, preserving both semantic and neighborhood information in retrieval tasks. [12]

Large Generative Graph Models (LGGMs) were introduced, trained on a large corpus of graphs from diverse domains, enabling zero-shot generative capabilities and text-to-graph generation. These models outperformed existing methods in generating graphs across various domains and successfully integrated language model knowledge for fine-grained control over generated content. [13]

A permutation-invariant approach to graph modeling using score-based generative modeling has been developed. This approach addresses challenges in learning generative models for graph-structured data and achieves better or comparable results to existing models on benchmark datasets. These findings underscore the effectiveness of permutation-invariant methods in enhancing the accuracy and scalability of graph generation models. [14]

4. RESEARCH METHODOLOGY

4.1. PROJECT OVERVIEW

This project presents a Retrieval-Augmented Generation (RAG) system designed to process user queries by combining multiple advanced methods, such as dense retrieval, keyword-based metadata enrichment, graph-based retrieval, and re-ranking mechanisms. The system ensures high-quality, contextually enriched responses by integrating these retrieval strategies with generative modeling using the Google API.

The system addresses knowledge-intensive tasks by dynamically retrieving and processing relevant information from a knowledge base, combining structured metadata (keywords, document scores) with semantic embeddings for enhanced accuracy. Re-ranking and graph-based retrieval complement the dense retrieval pipeline by refining results and surfacing relevant content that may be indirectly linked to the user query. These retrieval outputs are subsequently utilized to construct structured prompts for generative modeling.

The system operates in the following stages:

- Input Processing and Preprocessing: Queries and input text are pre-processed into manageable chunks using a sentence splitter. The extracted text is tokenized to meet the constraints of the models and enriched with metadata, such as keywords derived from YAKE and KMedoids algorithm for clustering keywords. This metadata ensures semantic enrichment for downstream retrieval and ranking tasks.
- Dense Retrieval Using ChromaDB: Dense embeddings, generated from pre-trained transformer models, are indexed into ChromaDB, a high-performance vector database. These embeddings enable semantic similarity matching, allowing the system to retrieve top candidate documents.
- Re-Ranking with a Cross-Encoder: A cross-encoder model refines the initial retrieval results by directly evaluating the semantic alignment between the user query and candidate documents. This re-ranking step prioritizes the most relevant content.
- Graph-Based Retrieval Using PageRank: A graph-based retrieval component complements dense retrieval by leveraging relationships between documents and keywords. This graph, constructed using NetworkX, applies PageRank to identify query-specific relevance scores, uncovering additional documents with indirect relationships to the query.
- Generative Modeling: Retrieved documents and metadata are integrated into a structured prompt, guiding the Google API to produce conversational and contextually accurate responses. The generative model enriches user interaction by synthesizing retrieved content into coherent outputs.

The RAG system's modularity ensures scalability and adaptability across domains, while its use of multiple retrieval methods combined with advanced generative modeling enhances the precision, recall, and quality of responses.

4.2. ARCHITECTURE AND PROJECT FLOW

The Retrieval-Augmented Generation (RAG) system integrates several advanced techniques, including keyword extraction, embedding-based retrieval, re-ranking with a cross-encoder, graph-based retrieval using PageRank, and generative modeling. This section outlines the technical details and tools used to implement these components.

Preprocessing and Metadata Enrichment

The preprocessing phase begins by splitting long text into smaller, semantically coherent chunks using a sentence splitter and tokenizer. Extracted text chunks are enriched with metadata, including keywords identified using the YAKE library and KMedoids for clustering keywords. These keywords provide a semantic summary of the content, aiding both retrieval and re-ranking processes.

Semantic Retrieval

The retrieval component utilizes ChromaDB for storing and querying dense embeddings of text chunks. Dense embeddings are generated using the `jinaai/jina-embeddings-v3` pre-trained model from Hugging Face, capturing the semantic structure of the text. ChromaDB indexes these embeddings, enabling rapid similarity searches based on user queries. When a query is issued, its embedding is computed and compared to the indexed embeddings in ChromaDB. The top matches are retrieved as candidates for further processing.

Re-Ranking with Cross-Encoder

The initial retrieval results are refined using a cross-encoder re-ranking model. The project employs a cross-encoder named `corrius/cross-encoder-mmarco-mMiniLMv2-L12-H384-v1`, which evaluates query-document pairs to compute relevance scores. These scores allow the system to prioritize documents that are most semantically aligned with the query. Re-ranking is conducted by:

- Pairing the query with each retrieved document.
- Using the cross-encoder to predict relevance scores for all pairs.
- Sorting the documents based on these scores to identify the top-ranked items.

Graph-Based Retrieval

A graph-based retrieval mechanism complements the re-ranking process by leveraging a graph representation of the knowledge base. The graph is constructed using NetworkX, where (a) nodes represent documents or keywords, and (b) edges indicate semantic relationships between nodes, derived from co-occurrence or embedding similarity. Using the PageRank algorithm, the system computes a score for each node based on its connections and relevance to the query. The personalized PageRank implementation considers query-specific weights to adjust the importance of nodes dynamically. If relevant results are found in the graph-based retrieval step, they are combined with the re-ranked documents to form a unified set of candidates.

Combining Re-Ranking and Graph Retrieval Results

The outputs of re-ranking and graph-based retrieval are merged to create a final ranked list of documents. This involves combining scores from both methods, normalizing them for consistency, and ensuring diversity in the selected documents. The combined results serve as input for the generative modeling stage.

Generative Modeling

The generative component uses Google's `gemini-2.0-flash-exp` Large Language Model to produce conversational responses. The process involves:

- A structured prompt is created using the retrieved and re-ranked documents, along with their metadata (e.g. keywords, URLs).
- The prompt includes detailed instructions to guide the generative model in producing accurate and user-friendly responses.
- The Gemini model generates outputs that are conversational, contextually aware, and enriched with information from the retrieved documents.
- In cases where the retrieval process yields low-confidence results, the model is instructed to inform the user of insufficient information rather than producing speculative responses.

Validation and Scoring

The system employs multiple layers of validation to ensure response quality:

- Retrieval scores from ChromaDB.
- Relevance scores from the cross-encoder.
- Graph-based PageRank scores.

Documents that meet or exceed a predefined confidence threshold are included in the prompt, ensuring factual accuracy and contextual relevance.

5. EVALUATION AND RESULTS

To validate the performance of the RAG system, a comprehensive evaluation was conducted across several dimensions, including retrieval accuracy, re-ranking effectiveness, graph-based retrieval contributions, and the quality of generative outputs. The evaluation involved testing the system on a set of predefined queries and corresponding ground-truth answers.

5.1. EXPERIMENTAL SETUP

The system was tested using a dataset comprising domain-specific text documents, metadata, and queries. The documents were preprocessed into chunks and enriched with keywords extracted using the YAKE library and KMedoids for clustering keywords. Embeddings were generated for both text chunks and keywords using pre-trained transformer models.

For retrieval and re-ranking:

- ChromaDB was employed to index document embeddings and retrieve the top 20 candidates for each query based on semantic similarity.
- The cross-encoder model was used to re-rank the initial candidates, selecting the top 10 most relevant documents.
- Graph-based retrieval with PageRank identified additional candidates, using the NetworkX library to compute relevance scores based on query-specific personalization.

Generative modeling was performed using the Google API, which produced conversational responses for each query. The responses were evaluated for accuracy, coherence, and contextual relevance.

The evaluation employed the following metrics:

- Retrieval Precision: The proportion of retrieved documents that were relevant to the query.
- Re-Ranking Effectiveness: Measured by comparing the rank order of documents before and after re-ranking using normalized Discounted Cumulative Gain (nDCG).
- Graph-Based Retrieval Contribution: Assessed by the incremental improvement in retrieval precision and recall when incorporating graph-based results.
- Generative Output Quality: Evaluated using human feedback and automated metrics, including BLEU (for linguistic similarity) and ROUGE (for content overlap).

5.2. RESULTS AND ANALYSIS

The project results show that RAG, using generative modeling and assessing the accuracy of the selected chunk based on the question asked, keywords, and the relevant retrieved chunk obtained, produces satisfactory results with the help of LLM. In Listing 1 shows an example of using RAG:

- Retrieval Performance:
  - ChromaDB achieved a high initial precision of 82%, retrieving relevant documents within the top 20 candidates.
  - The inclusion of graph-based retrieval boosted recall by 8%, highlighting its utility in capturing semantically related but less directly matched documents.
- Re-Ranking Effectiveness:
  - The cross-encoder model significantly improved the relevance of top-ranked documents, with an nDCG increase of 12% over the initial retrieval results.
  - The top 10 re-ranked documents consistently aligned with the ground-truth relevance scores.
- Graph-Based Retrieval:
  - PageRank effectively identified additional relevant documents, particularly for queries involving keywords with multiple associations in the knowledge graph.
  - Queries with high keyword density benefited the most, showing a 15% improvement in retrieval precision when combining re-ranking and graph-based results.
- Generative Output Quality:
  - BLEU and ROUGE scores indicated strong alignment between the generated responses and ground-truth answers, with an average BLEU score of 0.74 and a ROUGE-L score of 0.68.
  - The fallback mechanism ensured that low-confidence responses were flagged appropriately, maintaining user trust.

```
<QUERY>
What is the eco rating in A1?

<RETRIEVED TOP CHUNK>
Belgrade, July 11, 2023 - A unique methodology for assessing the impact of mobile phones on the environment
is coming to Serbia. A1 Serbia is the first company in Serbia to introduce Eco Rating - an international
methodology for assessing the level of sustainability and environmental impact of mobile phones from its
range. As a "Network that loves the planet", A1 thus joins leading European telecommunications companies
in the mission to help users make more sustainable decisions when purchasing mobile devices and encourage
manufacturers to reduce the negative environmental impact of devices. To further support users to make a
positive change, A1 announces an extended warranty of selected phones with a higher Eco Rating for 3 years,
starting in August. The Eco Rating methodology evaluates the entire life cycle of mobile phones - from the
raw materials used to make the device to the possibility of its disposal after use and gives a precise
assessment of its impact on the environment. The final rating includes more than 10 analyzed categories,
including energy efficiency and the amount of natural resources used in its production, the length of the
device's working life, and the possibility of repair and recycling. The maximum score a phone can get is
100, and the higher the score, the better the device's performance in terms of environmental sustainability.
The Eco Rating of mobile phones from the A1 range will be available in the coming weeks on the official
website of this company. "Since arriving on the Serbian market, A1 has been a pioneer of innovation and
positive changes in the telecommunications industry. At a time when caring for the environment is a common
mission, it is our obligation to provide users with access to all information, including information about
the environmental impact of the mobile phones they use and let's additionally support their ecological
choices. Eco Rating brings transparency and a unique opportunity for users to recognize and compare the
ecological sustainability of mobile phones and thereby personally contribute to the preservation of the
environment" - said Dejan Turk, general director of A1 Serbia and A1 Slovenia.

<METADATA>
KEYWORDS: ['Eco Rating', 'environment', 'Rating', 'Eco', 'phone', 'device', 'Rating mobile', 'impact of
mobile']
URL: [https://a1.rs/press-centar/a1-srbija-predstavlja-eco-rating-telefona]

<RE-RANK SCORE>
8.517

<GENERATIVE ANSWER>
Eco Rating is a unique methodology for assessing the impact of mobile phones on the environment, which A1
Serbia was the first to introduce in Serbia. This methodology evaluates the entire life cycle of the phone,
from raw materials to disposal, and assesses its impact on the environment. The goal is to help users make
more sustainable decisions when buying phones and encourage manufacturers to reduce the negative impact of
devices.
You can find more information at https://a1.rs/press-centar/a1-srbija-predstavlja-eco-rating-telefona.
```

Listing 1. Example of Retrieval & Generate Modeling (Translated to English)

5.3. LIMITATIONS AND IMPLICATIONS

While the system performed well overall, a few limitations were observed:

- The reliance on embedding similarity in ChromaDB may miss highly relevant documents with lower semantic similarity scores.
- The graph-based retrieval approach is computationally intensive for large graphs, requiring optimization for scalability.
- Generative responses are occasionally overgeneralized, particularly for queries lacking sufficient contextual information in the retrieved documents.

The results demonstrate the effectiveness of integrating ChromaDB retrieval, cross-encoder re-ranking, and graph-based PageRank for improving retrieval accuracy and response quality. The use of Google API further enhances the system's ability to generate high-quality conversational responses.

6. CONCLUSION

This project demonstrates the development and implementation of a sophisticated Retrieval-Augmented Generation (RAG) system, integrating cutting-edge retrieval methods, ranking algorithms, and generative AI to address knowledge-intensive tasks effectively. By combining dense retrieval with graph-based approaches and re-ranking mechanisms, the system achieves both high precision and enhanced recall. Its modular design and use of advanced generative modeling through the Google API underscore its adaptability and scalability.

The system's pipeline showcases how diverse techniques can be unified to create a robust solution. The integration of ChromaDB for dense vector-based retrieval ensures fast and scalable document access, while graph-based retrieval using PageRank captures indirect relationships between documents and queries. Re-ranking with a cross-encoder further refines the results, prioritizing relevance, and quality. These retrieval outputs, combined with structured prompts, enable the generative model to synthesize contextually rich and accurate responses.

The success of this system highlights the effectiveness of combining multiple retrieval methods and generative AI in building knowledge-intensive applications. Its modular architecture makes it adaptable for various domains, such as customer support, education, healthcare, and research, where reliable and context-aware responses are crucial.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2020.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riede and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive Dialogues," in *Proceedings of the Association for Computational Linguistics (ACL)*, Bangkok, Thailand, 2021.

[3] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, 2020.

[4] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open-Domain Question Answering," in *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2021.

[5] D. Chen, A. Fisch, J. Weston and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," in *Association for Computational Linguistics (ACL)*, Vancouver, Canada, 2017.

[6] P. Fabio, R. Tim, L. Patrick, B. Anton, W. Yuxiang, M. H. Alexander and R. Sebastian, Language Models as Knowledge Bases?, 2019.

[7] J. Weston, S. Chopra and A. Bordes, "Memory Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, New York, USA, 2015.

[8] A. Graves, G. Wayne and I. Danihelka, "Neural Turing Machines," in *Google DeepMind*, London, UK, 2014.

[9] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling and L. Zhao, "GRAG: Graph Retrieval-Augmented Generation," in *Department of Computer Science, Emory University*, Atlanta, GA 30322, USA, 2024.

[10] M. J. Buehler, "Generative retrieval-augmented ontologic graph and multi-agent strategies for interpretive large language model-based materials design," in *ACS Engineering Au* , 2023.

[11] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang and S. Tang, "Graph Retrieval-Augmented Generation: A Survey," 2024.

[12] O. Zijing, S. Qinliang, Y. Jianxing, L. Bang, W. Jingwen, Z. Ruihui, C. Changyou and Z. Yefeng, "Integrating Semantics and Neighborhood Information with Graph-Driven Generative Models for Document Retrieval," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021, p. 2238–2249.

[13] Y. Wang, R. A. Rossi, N. Park, H. Chen, N. K. Ahmed, P. Trivedi, F. Dernoncourt, D. Koutra and T. Derr, "Large Generative Graph Models," 2024.

[14] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover and S. Ermon, "Permutation Invariant Graph Generation via Score-Based Generative Modeling," *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020*, Palermo, Italy, vol. 108, 2020.