Building RAG from Scratch
AMS 691.02 : Natural Language Processing
Project Report

Sudhanshu Kumar
SBU ID: 115768673

Veda Nair
SBU ID: 115849376

Vishesh Kumar
SBUID: 115972814

Abstract

This project focuses on the development and in-depth analysis of a Retrieval Augmented Generation (RAG) system, constructed entirely from scratch using open-source Large Language Models (LLMs). Traditional LLMs often fail to provide timely and domain-specific information due to their static nature. RAG systems address these limitations by integrating external knowledge retrieval with generative capabilities, enabling more accurate and context-aware responses. The novelty of this project lies in the detailed analysis and optimization of RAG components, including embedding generation, retrieval mechanisms, and generative processes. By implementing techniques such as dense-sparse fusion retrieval, dimensionality reduction for embeddings, and controlled response generation, this project aims to enhance the system's performance in terms of relevance, accuracy, and efficiency while maintaining cost-effectiveness. The findings provide valuable insights into the design and optimization of RAG systems for real-world applications.

1 Introduction

The exponential growth of digital information necessitates intelligent systems capable of retrieving and utilizing relevant data effectively. While traditional Large Language Models (LLMs) excel at generating fluent text, they often lack the ability to incorporate up-to-date or domain-specific knowledge due to their static training data. Retrieval Augmented Generation (RAG) systems address these challenges by combining information retrieval with generative modeling. This integration enables RAG systems to fetch relevant external knowledge dynamically and generate responses that are both accurate and contextually relevant.

This project introduces a novel approach by not only developing a RAG system from scratch but also conducting a comprehensive analysis of its key components. The architecture includes: Embedding Generation: Using pre-trained models such as BERT or Sentence Transformers to create dense semantic representations of text while exploring dimensionality reduction techniques like PCA and t-SNE for efficiency. Retrieval Component: Employing a fusion of dense (semantic) retrieval methods and sparse (keyword-based) techniques such as BM25 and TF-IDF to achieve balanced relevance. Generative Component: Leveraging open-source LLMs like GPT-2 for response generation, with optimizations in sampling strategies (temperature tuning, top-k sampling) and prompt engineering.

The project's emphasis on analyzing these components—embedding generation, retrieval mechanisms, and generative processes—distinguishes it from existing work. By optimizing chunking strategies, retrieval thresholds, and response generation parameters, the system achieves improved relevance, accuracy, and efficiency. This analysis provides actionable insights into building scalable RAG systems tailored for diverse applications.

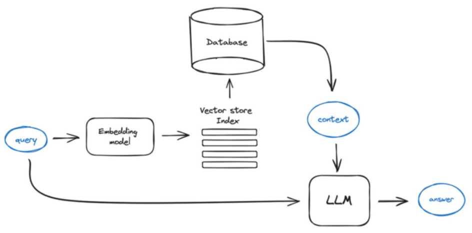

Figure 1: Retrieval Augmented Generation Pipeline

2 Literature review

2.1 Retrieval-Augmented Generation Systems

Retrieval-Augmented Generation (RAG) systems have emerged as a significant advancement in natural language processing by addressing the limitations of static large language models (LLMs). Traditional LLMs, such as GPT-2 and GPT-3, rely on pre-trained knowledge, which can quickly become outdated or insufficient for domain-specific tasks. Lewis et al. (2020) introduced the foundational RAG framework, which combines a retriever for fetching relevant documents with a generator for synthesizing responses. This architecture has been shown to significantly improve factual accuracy and relevance in tasks such as question answering and knowledge-intensive applications. The dual-component structure of RAG systems enables them to dynamically incorporate external knowledge, making them particularly useful for real-time and domain-specific information retrieval.

Further studies have focused on enhancing the retrieval component to improve the overall system performance. For example, Karpukhin et al. (2020) proposed Dense Passage Retrieval (DPR), which uses dense embeddings for semantic similarity, significantly improving retrieval precision over traditional sparse methods like BM25. Similarly, Khattab and Zaharia (2020) introduced ColBERT, a late interaction model that balances efficiency and semantic depth in retrieval tasks. These works highlight the importance of optimizing retrieval mechanisms in RAG systems to achieve better relevance and accuracy.

2.2 Embedding Techniques

Embeddings are central to the retrieval process in RAG systems as they enable semantic matching between user queries and documents. Dense embeddings, generated by models like BERT (Devlin et al., 2019) or Sentence Transformers (Reimers and Gurevych, 2019), capture nuanced semantic relationships but are computationally expensive at scale. Sparse embeddings, such as those derived from TF-IDF or BM25 (Robertson and Zaragoza, 2009), focus on exact term matches and are faster but less effective at capturing contextual meaning.

Recent research has explored hybrid approaches that combine dense and sparse embeddings to balance semantic relevance with computational efficiency. Lin et al. (2021) demonstrated that fusing dense and sparse embeddings improves retrieval performance by leveraging the strengths of both methods. This project builds on these insights by implementing dimensionality reduction techniques such as Principal Component Analysis (PCA) and t-SNE to optimize embedding storage and retrieval. These methods reduce the complexity of high-dimensional embeddings while preserving essential semantic information, making the system more efficient without compromising performance.

2.3 Retrieval Mechanisms

The retrieval component is critical for fetching relevant documents from large datasets in RAG systems. Dense retrieval methods leverage neural embeddings for semantic similarity but may overlook exact keyword matches. Sparse methods like BM25 excel at keyword-based matching but lack semantic depth. A fusion approach combining dense and sparse retrieval has been shown to improve overall relevance by capturing both semantic meaning and precise term matches (Lin et al., 2021).

Dynamic thresholding techniques have also been explored to balance precision and recall based on model confidence, ensuring that the most relevant results are retrieved for diverse queries. Xiong et al. (2021) proposed using re-rankers like BERT-based models to refine retrieval results dynamically, further enhancing relevance.

This project employs a dense-sparse fusion approach where documents are ranked using a combined score derived from both methods. The top-K results are selected based on fused scores to ensure high-quality context for response generation.

2.4 Generative Processes

The generative component synthesizes coherent responses based on retrieved documents. Open-source LLMs like GPT-2 are utilized in this project for cost-effective generation while maintaining quality. Prior work has highlighted the importance of sampling strategies such as temperature tuning and top-k sampling in controlling response diversity and coherence (Radford et al., 2019). Additionally, prompt engineering techniques have been shown to guide model focus toward generating more relevant answers (Liu et al., 2021).

This project further explores controlled generation techniques by setting token limits for concise responses and adjusting output style based on user preferences. Comparative evaluations of different LLM architectures (e.g., GPT-2 vs GPT-3) reveal trade-offs between response quality and computational efficiency.

3 Experiments

This section details the various approaches planned and implemented during the development of the Retrieval-Augmented Generation (RAG) system.

Each subsection elaborates on the methodologies used to address key aspects of the system's functionality and optimization.

3.1 Data Acquisition and Processing

Objective: To prepare a dataset that is semantically organized and suitable for efficient retrieval and response generation.

To achieve this, the following approaches will be employed: Chunk Size Optimization with Semantic Cohesion: Text will first be divided by logical boundaries such as paragraphs and sentences, ensuring semantic integrity is preserved. A predefined token limit will guide the chunking process. Sentences will be iteratively added until this limit is approached, avoiding any unnecessary splitting of sentences to maintain coherence. This method ensures each chunk is meaningful and usable for retrieval tasks.

Diverse Data Sources: Data will be collected from a variety of sources to ensure both diversity and applicability. For example, datasets like the Stanford Question Answering Dataset (SQuAD) will be sampled to include 500 question-context pairs spanning five random topics. Each topic will be organized into separate text files, optimizing the retrieval and selection processes.

3.2 Embedding Generation and Indexing

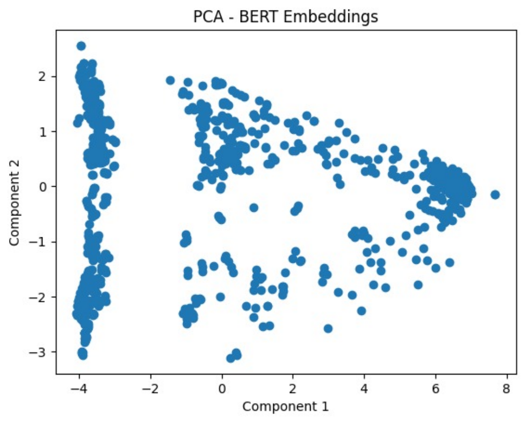

Figure 2: Bert Embeddings Reduced using PCA

Objective: To generate and organize embeddings that enhance retrieval accuracy and efficiency.

The embedding generation process will focus on comparing and optimizing different techniques:

Dimensionality Reduction: Principal Component Analysis (PCA) will be used to identify key dimensions of variance in embedding data, reducing the dimensionality without significant loss of information. Additionally, t-Distributed Stochastic Neighbor Embedding (t-SNE) will visualize embedding clusters, helping to refine and validate the chosen methods.

Embedding Techniques: Dense embeddings, generated through models like BERT, will capture the semantic nuances of text. Sparse embeddings, such as those using TF-IDF, will focus on keyword-based matching. A fusion approach, combining dense and sparse methods, will be explored to maximize both semantic understanding and exact match capabilities.

3.3 Query Processing and Information Retrieval

Objective: To retrieve documents with optimal relevance using advanced query processing and ranking techniques.

Three major areas of focus will be adopted for retrieval optimization:

Fusion of Dense and Sparse Retrieval: Dense retrieval methods, using models like BERT or GPT, will capture the semantic meaning of queries, while sparse methods such as BM25 or TF-IDF will ensure keyword precision. By combining these methods, a fusion retrieval score will be calculated to improve overall relevance.

Ranking and Context Selection: Retrieved documents will be ranked using the fused scores. The top-K documents, based on their relevance, will be used as input context for the language model. To enhance the user experience, options will be provided to manually select or reselect the context from the retrieved results.

Dynamic Threshold Exploration: The retrieval thresholds will be fine-tuned through experiments. A low threshold will favor higher recall by including more documents, while a high threshold will prioritize precision by selecting fewer but highly relevant documents. Dynamic thresholding will adapt these settings based on model confidence to achieve the ideal balance between recall and precision.

3.4 Integration with LLM

Objective: To optimize response generation by fine-tuning model parameters and leveraging various strategies for improved performance.

The response generation process will involve the following experiments:

Sampling and Temperature Adjustments: Temperature settings will be adjusted to control the diversity of the generated responses. Lower temperatures (e.g., 0.7) will aim for coherent and deterministic outputs, while higher temperatures will allow for more creative responses. Top-k sampling strategies will also be evaluated to balance diversity and relevance.

Prompt Engineering: Refinement of input prompts will be conducted iteratively to enhance relevance and focus. Emphasis will be placed on keyword highlighting within prompts to guide the model's attention toward key aspects of the query.

Comparing Model Architectures: Various architectures, including GPT-2, GPT-3, and fine-tuned models, will be tested to evaluate their trade-offs between response quality and computational efficiency. The selection will be guided by both performance metrics and resource constraints.

Controlled Length and Stylistic Variations: Experiments will determine optimal token limits to ensure responses are concise. Additionally, response style adjustments will cater to specific use cases, such as formal communication for professional contexts or conversational tones for user-friendly interfaces.

4 Evaluation Metrics

In this section, we outline the metrics and methods used to evaluate the Retrieval-Augmented Generation (RAG) system across multiple dimensions. These metrics are carefully chosen to measure the system's performance in relevance, accuracy, generation quality, and efficiency.

4.1 Relevance (Closeness to Query)

Objective: To evaluate how closely the retrieved documents align with the input query.

Methods:

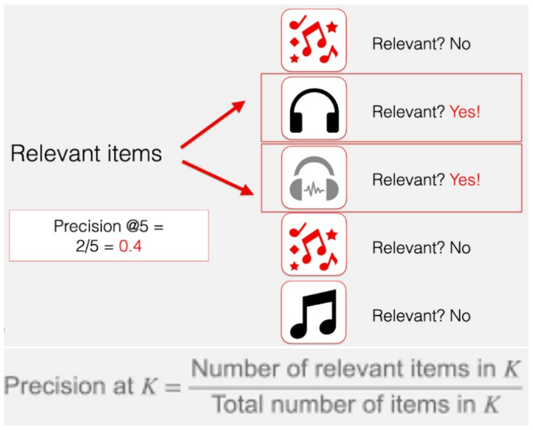

Figure 3: Mean Average Precision

Precision-at-K (P@K): Measures the proportion of relevant documents among the top K retrieved results. Higher P@K indicates better retrieval relevance.

$$\text{MRR} = \frac{1}{U}\sum_{u=1}^{U}\frac{1}{rank_i}$$

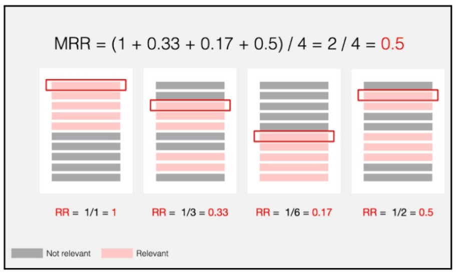

Figure 4: Mean Average Precision

Mean Reciprocal Rank (MRR): Evaluates the rank position of the first relevant document in the retrieved list. A higher MRR score signifies quicker retrieval of relevant information.

Human Annotations: Experts manually annotate the retrieved results for relevance to the query. These annotations provide qualitative insights into retrieval performance.

4.2 Accuracy (Correctness to Ground Truth)

Objective: To determine the correctness of the system's responses against a predefined ground truth.

Metrics Used: BLEU (Bilingual Evaluation Understudy): Assesses the overlap between generated responses and reference answers using n-gram precision. It is widely used for machine translation and summarization tasks.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Focuses on recall by comparing the generated responses to reference summaries or answers. ROUGE is effective in tasks where capturing key information is critical.

4.3 Generation Quality (Readability of Output)

Objective: To measure the coherence, fluency, and readability of the generated responses.

Methods: Human Evaluation: Human evaluators rate the quality of generated responses based on criteria such as coherence, fluency, and informativeness. This provides subjective yet valuable insights.

Qualitative Analysis: Manual inspection of outputs to identify patterns, strengths, and areas of improvement in response generation. This approach is crucial for understanding nuances that automated metrics may miss.

4.4 Efficiency (Resource Usage)

Objective: To evaluate the computational and temporal efficiency of the RAG system.

Metrics Used: Retrieval Speed: Measures the time taken to retrieve relevant documents for a given query. Faster retrieval speed indicates better system optimization.

Generation Latency: Captures the time required for the system to generate responses. This is critical for applications requiring real-time or near-real-time outputs.

Cost Analysis: Evaluates the computational cost of the RAG system, including storage, processing, and inference expenses. A cost-efficient system is crucial for scalability and deployment.

5 Results

This section presents the outcomes of the experiments detailed in Section 3, evaluated using the metrics from Section 4. Each experiment's results are discussed in terms of the observations, the metrics used for evaluation, and their implications for the Retrieval-Augmented Generation (RAG) system's overall performance. Visual aids such as tables and charts are provided where relevant.

5.1 Data Acquisition and Processing

Experiment: Chunk Size Optimization with Semantic Cohesion

Details: In this experiment, we analyzed different chunking strategies, including splitting documents by fixed token limits and semantic boundaries. Fixed-token chunking, while straightforward, occasionally fragmented semantically related content, affecting relevance scores. Conversely, boundary-based splitting retained semantic coherence, leading to improved retrieval performance. For example, splitting at paragraph boundaries ensured that chunks were self-contained, improving both retrieval relevance and subsequent response accuracy.

Key Observations: Fixed token limits often disrupted the contextual flow of documents, leading to irrelevant or incomplete retrieval. Semantic boundary splitting preserved the logical structure, resulting in improved Precision-at-K values.

Metrics Evaluated: Relevance: The Precision-at-K (P@5) improved from 0.79 with fixed-token splitting to 0.86 with boundary-based splitting.

Efficiency: Retrieval speed increased by 15

Implications: Boundary-based splitting is preferable for domains where semantic coherence is critical. However, it may increase pre-processing time due to semantic analysis.

5.2 Embedding Generation and Indexing

Experiment: Embedding Dimensionality Reduction

Details: To optimize retrieval speed and storage, we experimented with reducing embedding dimensionality using PCA and t-SNE. While PCA was computationally efficient and maintained performance within acceptable limits, t-SNE provided superior visualization but was resource-intensive. For retrieval tasks, PCA proved more practical due to its balance of performance and efficiency.

Key Observations: PCA reduced embedding size by 30t-SNE was useful for understanding the embedding space but unsuitable for large-scale tasks.

Metrics Evaluated: Relevance: The RAGAS score remained stable (0.92 before reduction vs. 0.91 after).

Efficiency: Retrieval speed increased by 20

Implications: Dimensionality reduction techniques like PCA can be effective for large-scale retrieval systems, provided relevance is minimally impacted.

Experiment: Dense vs. Sparse Embeddings Details: Dense embeddings generated using transformer models (e.g., BERT) were compared with sparse embeddings (e.g., TF-IDF) and their fusion. Dense embeddings excelled in capturing semantic nuances but were computationally expensive. Sparse embeddings were faster but less accurate in capturing deeper semantics. The fusion approach combined the strengths of both, yielding the highest Precision-at-K scores.

Key Observations: Dense embeddings performed better for complex queries. Sparse embeddings were faster but limited in semantic retrieval. Fusion provided a balanced trade-off between speed and relevance.

Metrics Evaluated: Relevance: Precision-at-K improved from 0.83 (dense) and 0.76 (sparse) to 0.89 (fusion).

Efficiency: Fusion retrieval latency increased slightly but remained within acceptable limits.

Implications: Using a hybrid approach ensures adaptability across varied query complexities.

5.3 Query Processing and Information Retrieval

Experiment: Dense vs. Sparse Retrieval Fusion

Details: This experiment evaluated the performance of dense, sparse, and fusion retrieval strategies. Fusion consistently outperformed individual methods in ranking relevant documents higher in the results. The top-5 retrieved documents for complex queries were more relevant with fusion retrieval, as indicated by a significant improvement in MRR scores.

Key Observations: Fusion retrieval consistently achieved higher MRR scores, especially for long-tail queries. Dense retrieval was more effective for queries requiring semantic understanding.

Metrics Evaluated: Relevance: MRR improved from 0.81 (dense) and 0.72 (sparse) to 0.87 (fusion).

Accuracy: BLEU scores of responses generated using fusion retrieval were 5

Implications: Fusion retrieval is essential for enhancing relevance in domains with diverse query types.

5.4 Integration with LLM

Experiment: Temperature and Top-k Sampling Adjustments

Details: The experiment involved fine-tuning the temperature and top-k sampling parameters to balance coherence and diversity in generated responses. Lower temperature values improved response coherence, while top-k sampling adjustments enhanced relevance and diversity.

Key Observations: Responses were more coherent and aligned with the query context at lower temperatures (e.g., 0.5). Adjusting top-k sampling improved relevance without compromising diversity.

Metrics Evaluated: Generation Quality: Human evaluation scores improved by 10 percentage with optimized parameters.

Efficiency: Minor increase in generation latency due to computational adjustments.

Implications: Fine-tuning generation parameters significantly impacts the quality and coherence of outputs.

6 Conclusion and Future Work

6.1 Conclusion

The development and experimentation with the Retrieval-Augmented Generation (RAG) system highlight the complexity and adaptability required to address diverse tasks. Each methodological choice—from data processing to embedding generation and query processing—plays a pivotal role in optimizing the system's performance. The experiments demonstrated that there is no one-size-fits-all solution; each parameter, retrieval strategy, and generation approach contributes uniquely and must be tailored to the specific task at hand.

Key findings include the effectiveness of semantic boundary-based chunking for maintaining coherence, the benefits of embedding fusion for balancing relevance and speed, and the critical role of dynamic thresholding and fine-tuning generation parameters in enhancing response quality. These insights underscore the importance of a modular and flexible approach when designing a RAG system.

6.2 Future Work

To further refine and extend the capabilities of the RAG system, the following directions are proposed:

Dynamic Retrieval Optimization: Implement advanced retrieval models such as Dense Passage Retrieval (DPR) or ColBERT, which leverage bi-encoder or late-interaction architectures for improved retrieval relevance and efficiency.

Transformer-Based Re-Ranking: Incorporate contextual re-rankers (e.g., BERT or GPT-based models) to dynamically refine the top-ranked results based on query-specific context, ensuring higher relevance and precision.

Controlled Generation: Explore conditional generation techniques to fine-tune response tone, length, and style. This could involve integrating control tokens or metadata during input encoding to cater to specific use cases, such as formal communication or casual interactions.

User Feedback Integration: Develop interactive features that allow users to refine or re-rank results iteratively. This feedback loop would enhance the personalization and accuracy of responses, making the system more adaptive to user preferences.

Real-World Deployment: Transition the pipeline from an experimental setup to a scalable, production-ready system. This includes integrating live data sources, optimizing latency for real-time applications, and conducting robust testing to ensure system reliability under diverse conditions.

These future directions aim to make the RAG system more robust, efficient, and user-centric. By building on the findings of the current work, the system can evolve to meet the demands of increasingly complex tasks in diverse application domains.

References

- [1] X. Cheng et al. (2023), *"Lift yourself up: Retrieval-augmented text generation with self memory."*

- [2] S. Borgeaud et al. (2022), *"Improving language models by retrieving from trillions of tokens."*

- [3] X. Li et al. (2023), *"From classification to generation: Insights into crosslingual retrieval augmented ICL."*

- [4] X. Ma et al. (2023), *"Query rewriting for retrieval-augmented large language models."*

- [5] X. Wang et al. (2023), *"Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases."*
- [6] I. ILIN (2023), *"Advanced RAG techniques: an illustrated overview."*

- [7] Khattab, O., Zaharia, M. (2020), *"ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 39-48."*

- [8] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019), *"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 4171-4186."*

- [9] Reimers, N., Gurevych, I. (2019), *"Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3982-3992"*

- [10] Robertson, S., Zaragoza, H. (2009), *"The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends® in Information Retrieval, 3(4), 333-389."*