RefinedRAG: Precision Retrieval for Improved Responses

Samriddhi Singh
ss5675@drexel.edu

Aditya Sinha
as5869@drexel.edu

***Abstract*—The rise of remote work and online education has led to an increase in virtual meetings and lectures, resulting in professionals and students often missing important sessions. This creates the challenge of manually reviewing lengthy recordings and transcripts, a process that is time-consuming and inefficient. Retrieval-Augmented Generation (RAG) offers a promising solution by integrating large language models (LLMs) with external knowledge sources to enhance information retrieval and summarization. However, existing RAG models struggle with retrieving accurate and contextually relevant documents in response to complex queries. Our project, RefinedRAG, addresses this challenge by implementing and evaluating various retrieval enhancement methods including semantic chunking, similarity thresholds, maximum marginal relevance, and hybrid search approaches. Our results show that hybrid search combining keyword and embedding-based approaches achieves the best performance with a Mean Average Precision of 0.85 and Semantic Similarity of 0.781, significantly outperforming the vanilla RAG baseline. These improvements enable more precise, context-aware answers to specific queries about missed meetings or lectures, enhancing productivity and learning outcomes.**

***Index Terms*—Retrieval-Augmented Generation, Natural Language Processing, Document Retrieval, Vector Similarity, Large Language Models**

I. Background

The transition to remote work and online education has significantly increased the volume of virtual meetings and digital lectures. As a result, professionals and students frequently miss important sessions, creating a need for efficient ways to review and extract key information from recordings and transcripts. Traditional methods of manual review are both time-consuming and inefficient, making them impractical for daily use in environments with frequent virtual interactions.

Professionals and students frequently face the challenge of sifting through lengthy recordings and transcripts to extract critical information, an effort that is both labor-intensive and prone to inefficiencies. Traditional methods of manual review are not scalable in environments where virtual meetings and lectures occur daily. This creates a demand for automated solutions that can efficiently process and summarize meeting content.

Retrieval-Augmented Generation (RAG) offers a promising solution by integrating large language models (LLMs) with external knowledge sources to enhance information retrieval and summarization. RAG models can significantly improve the efficiency of accessing and processing information from lengthy transcripts. However, existing RAG implementations face challenges in retrieving accurate and contextually relevant documents in response to complex queries, which limits their effectiveness for specialized applications like meeting summarization.

This gap underscores the need for systems that can efficiently extract and summarize key points from transcripts, provide relevant answers to specific queries, and reduce the time required to process large volumes of meeting data. By addressing these limitations, such systems could significantly enhance productivity and learning outcomes by making vital information more accessible and actionable.

II. Related Work

Several research efforts have focused on enhancing RAG systems for document understanding and question-answering tasks. Veturi et al. [1] developed a RAG-based question-answering system for contextual response prediction, demonstrating improvements in response relevance through advanced retrieval techniques. Their work emphasizes the importance of context preservation in document chunking, a concept we build upon in our semantic chunking implementation.

Fan et al. [2] conducted a comprehensive survey on RAG systems integrated with LLMs, particularly focusing on meeting contexts. Their work highlights the challenges in processing conversational data and the importance of retrieval mechanism optimization. They identify key limitations in current approaches, including context fragmentation and retrieval relevance, which our work directly addresses.

In the domain of multimedia content understanding, Sajeev et al. [3] introduced VisionVerse, a system leveraging RAG for dynamic video question answering. While their focus was on multimodal content, their retrieval component shares similarities with our approach, particularly in embedding generation and similarity search optimization.

The challenges in existing RAG systems include contextual understanding of unstructured conversations, NLP complexity in handling varied phrasings, accuracy and relevance of extracted information, and efficient processing of large data volumes. Our work addresses these challenges by implementing and systematically evaluating multiple retrieval enhancement methods specifically tailored for meeting transcripts and lecture content.

III. Methodology

A. System Architecture

Our Retrieval-Augmented Generation (RAG) system is designed to enhance the capabilities of large language models (LLMs) by integrating external knowledge sources. It consists of three primary components:

1) **Ingestor:** Processes and embeds external documents into a vector database for efficient retrieval.
2) **Retriever:** Fetches relevant document chunks based on user queries using vector similarity measures.
3) **Generator:** Combines retrieved information with an LLM to produce contextually accurate and coherent responses.

This architecture facilitates a seamless flow from document ingestion to response generation, enabling the system to deliver accurate and context-specific outputs.

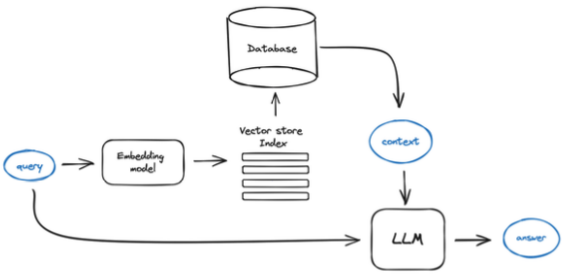

Fig. 1: Retrieval Augmented Generation Pipeline showing the workflow from user query through embedding model to vector store, then to LLM for response generation.

B. Document Processing Pipeline

1) Document Ingestion and Embedding: The document ingestion process is critical to ensure that the RAG system can retrieve relevant and meaningful data. Key components include:

- **Text Extraction:** Raw documents and meeting audio files are processed using tools like PyPDFium2Loader and OpenAI's Whisper model to extract text content.
- **Semantic Chunking:** Unlike naive character-based splitting, we implement semantic chunking that segments documents into coherent sections based on logical boundaries such as sentences, paragraphs, or topics. This ensures that chunks retain meaningful context and improves retrieval accuracy.
- **Vector Embedding:** The extracted chunks are embedded using FastEmbedEmbeddings, which represent the semantic meaning of the text in a high-dimensional vector space.
- **Storage:** The resulting embeddings are stored in a Qdrant vector database, which supports efficient similarity search operations for subsequent retrieval.

Semantic chunking significantly improves over recursive character-based splitting by ensuring text chunks maintain contextual integrity. This approach avoids splitting related content across multiple chunks, enhancing retrieval accuracy and reducing hallucinations in LLM-generated responses.

C. Retrieval Enhancement Methods

We implemented and evaluated several retrieval mechanisms to improve document retrieval precision:

1) Vanilla RAG (Top-k Retrieval): The baseline approach retrieves the k-most similar document chunks based on cosine similarity between the query embedding and document embeddings. While straightforward, this method can retrieve irrelevant documents when similarity scores are low across all documents.

2) Similarity Score Threshold Retrieval: This method only retrieves documents whose similarity scores exceed a defined threshold, improving precision by filtering out less relevant results. The threshold can be dynamically adjusted based on domain-specific requirements.

3) Maximum Marginal Relevance (MMR): MMR balances relevance with diversity by selecting documents that add new information rather than redundant content. The algorithm uses the following formula to score documents:

$$\text{MMR} = \lambda \cdot \text{sim}(d_i, q) - (1 - \lambda) \cdot \max_{d_j \in S} \text{sim}(d_i, d_j) \quad (1)$$

where λ is a tunable parameter controlling the trade-off between relevance and diversity, q is the query, d_i is a candidate document, S is the set of already selected documents, and sim is the similarity function.

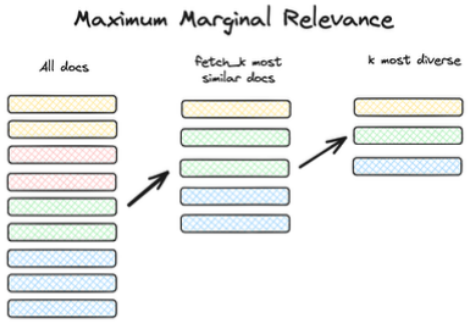

Fig. 2: Maximum Marginal Relevance illustration showing how the algorithm balances between selecting the most similar documents to the query while introducing diversity in the results to avoid redundancy.

4) Approximate Nearest Neighbor (ANN) Search: We implemented the Annoy (Approximate Nearest Neighbors Oh Yeah) algorithm to optimize retrieval speed for large datasets. ANN constructs a tree-based index that allows for efficient approximate similarity search, trading minimal accuracy for significant speed improvements.

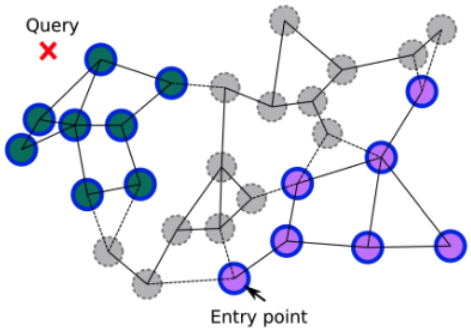

Fig. 3: Approximate Nearest Neighbor search visualization showing how the algorithm efficiently partitions the vector space using tree-based indexing to quickly find similar items without exhaustive comparison.

5) Hybrid Search: Our most effective enhancement combines keyword-based and embedding-based search methods. The retriever processes queries through both channels:

- **Keyword matching:** Identifies documents containing explicit query terms
- **Embedding similarity:** Finds semantically related documents using vector similarity

Results from both approaches are merged and ranked using a weighted scoring system, capturing both explicit and contextual relevance.

6) Vector Similarity Measures: For comparing document embeddings, we primarily use cosine similarity, which measures the cosine of the angle between two vectors, effectively evaluating their directional similarity regardless of magnitude.

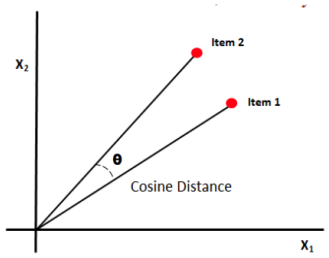

Fig. 4: Cosine similarity visualization showing how the measure computes the angle between two vectors to determine their semantic similarity, independent of vector magnitude.

This measure is particularly effective for high-dimensional embeddings, as it focuses on the orientation rather than magnitude, making it robust for semantic comparisons.

D. Sequential Document Processing

To maintain up-to-date information, we implemented sequential document processing that processes new documents in chronological order. This approach ensures that:

- Earlier data is replaced or augmented without creating conflicting or outdated entries
- Answers generated always rely on the most up-to-date context
- Updates can be streamlined without re-ingesting the entire database

This methodology is particularly important for meeting contexts where information evolves over time.

E. Response Generation

Once the most relevant document chunks are retrieved, the generative model takes over:

- **Input Format:** The retrieved chunks and user query are concatenated to provide comprehensive context.
- **Generative Model:** The input is fed into ChatGroq (leveraging Llama LLM) via API integration.
- **Token Management:** The system ensures inputs stay within token limits using efficient chunking strategies.

F. User Interface

We developed a Streamlit-based web application that serves as the frontend for end-users. Key features include:

- Simple query input for user questions
- Real-time response display
- Document status tracking
- Adjustable retrieval parameters

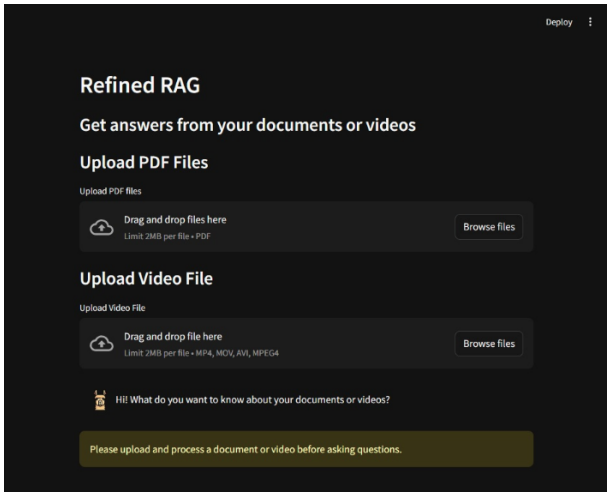

Fig. 5: RefinedRAG user interface showing document upload functionality, query input field, and response display panel in a clean, intuitive layout.

IV. Experiments and Results

A. Evaluation Metrics

We assessed the performance of different retrieval methods using five key metrics:

- **Mean Average Precision (MAP):** Measures the ranking quality of retrieved documents
- **Mean Reciprocal Rank (MRR):** Focuses on the position of the first relevant result
- **Semantic Similarity Score:** Measures contextual relevance using cosine similarity

- **Response Time:** Records the time taken for retrieval operations
- **Manual Response Accuracy Check:** Human evaluation of response quality

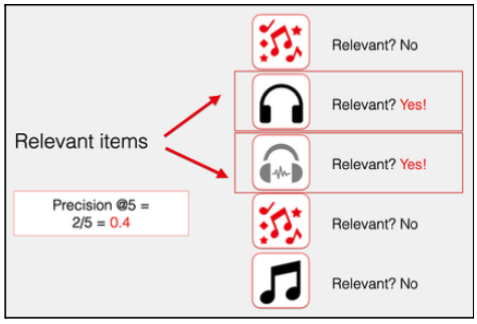

Fig. 6: Mean Average Precision calculation illustration showing how precision is calculated at each relevant result position.

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

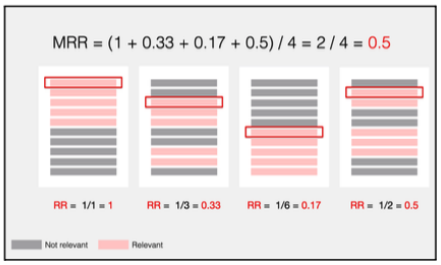

Fig. 7: Mean Reciprocal Rank visualization demonstrating how MRR calculates the average of inverse positions of the first relevant results across multiple queries.

B. Experimental Setup

Our experiments used a dataset consisting of meeting transcripts derived from audio files transcribed using OpenAI's Whisper model. We implemented various retrieval methods on this dataset and evaluated their performance.

C. Retrieval Process Implementation

The retrieval component is central to our system's performance, as shown in this execution flow:

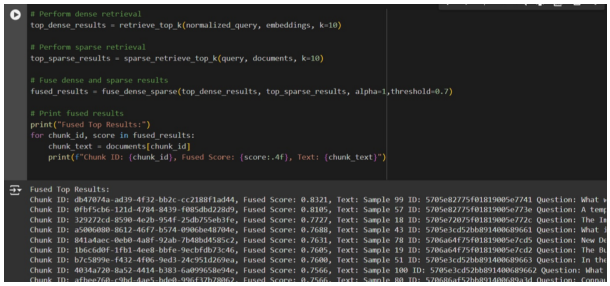

Fig. 8: Code execution flow for the retrieval process showing the query transformation, similarity search, and results filtering steps.

D. Results and Analysis

The performance comparison of different retrieval methods is summarized in the following table:

Improvement Method	Mean Average Precision	Mean Reciprocal Rank	Semantic Similarity	Response Time (seconds)	Manual Accuracy Check (topmost similarity result)
Specifying top k (Vanilla RAG)	0.65	0.62	0.649	7.99	Yes
Similarity Score threshold retrieval	0.82	0.78	0.752	9.78	Yes
Maximum Marginal Relevance retrieval	0.72	0.68	0.608	7.97	Sometimes
Approximate Nearest Neighbour Search	0.60	0.55	0.632	5.28	Sometimes
Hybrid Search	0.85	0.81	0.781	8.95	Yes

Fig. 9: Performance comparison of different retrieval methods showing MAP, MRR, Semantic Similarity, Response Time, and Manual Accuracy metrics.

1) Vanilla RAG (Top-k):

- MAP: 0.65 (moderate precision)
- Semantic Similarity: 0.649 (decent relevance)
- Response Time: 7.99s
- Best suited for general-purpose applications

2) Similarity Score Threshold:

- MAP: 0.82 (high precision)
- Semantic Similarity: 0.752 (superior relevance)
- Response Time: 9.78s (slowest method)
- Ideal for precision-critical applications

3) Maximum Marginal Relevance (MMR):

- MAP: 0.72
- Semantic Similarity: 0.608 (lowest due to diversity focus)
- Response Time: 7.97s
- Well-suited for exploratory research or summarization

4) Approximate Nearest Neighbor (ANN):

- MAP: 0.60 (lowest precision)
- Semantic Similarity: 0.632
- Response Time: 5.28s (fastest method)
- Best for real-time or high-traffic systems

5) Hybrid Search:

- MAP: 0.85 (highest precision)
- Semantic Similarity: 0.781 (best relevance)
- Response Time: 8.95s
- Best overall performer, balancing precision and speed

6) Key Findings:

1) MMR retrieval has the lowest semantic similarity score due to its focus on diversity.
2) ANN offers the fastest response time but sacrifices retrieval quality.
3) Hybrid search performs best overall by combining keyword and semantic approaches.
4) There is a clear trade-off between retrieval accuracy and response speed.

V. Conclusions

Our RefinedRAG system demonstrates significant improvements over vanilla RAG implementations for meeting transcript analysis and question answering. The key conclusions from our work include:

1) Semantic chunking substantially improves retrieval quality by maintaining contextual integrity of document segments, addressing a fundamental limitation of traditional RAG systems.
2) Hybrid search combining keyword-based and embedding-based retrieval achieves the best overall performance, particularly for applications requiring both precision and contextual understanding.
3) There is an inherent trade-off between retrieval precision and speed, with different retrieval methods offering optimal performance for specific application requirements.
4) Sequential document processing ensures information currency, which is critical for meeting contexts where topics evolve over time.

These improvements enable more accurate, contextually relevant responses to user queries about missed meetings or lectures, enhancing productivity and learning outcomes. By optimizing the retrieval component of RAG, our system addresses the core challenges of meeting transcript analysis: contextual understanding, retrieval accuracy, and computational efficiency.

VI. Future Work

Building on our current implementation, several promising directions for future work include:

1) **Adaptive Retrieval Method Selection:** Developing an intelligent system that automatically selects the optimal retrieval method based on query characteristics and performance requirements.
2) **Multi-modal RAG:** Extending the system to incorporate visual elements from presentations or whiteboard content alongside transcript text for more comprehensive context understanding.
3) **Personalized Retrieval:** Implementing user preference modeling to tailor retrieval based on individual interests, prior knowledge, and information needs.
4) **Dynamic Thresholding:** Creating self-adjusting similarity thresholds that adapt to corpus characteristics and query complexity, optimizing the precision-recall balance.
5) **Cross-meeting Context Awareness:** Enhancing the system to understand and link related discussions across multiple meetings, providing more comprehensive responses spanning temporal boundaries.
6) **Improved Evaluation Framework:** Developing standardized benchmarks specifically for meeting transcript RAG to facilitate comparative analysis across different implementation approaches.

Additionally, exploring more efficient vector indexing methods could further improve retrieval speed without sacrificing accuracy, potentially resolving the current precision-speed trade-off observed in our experiments.

References

[1] S. Veturi, S. Vaichal, R. L. Jagadheesh, N. I. Tripto, and N. Yan, "RAG based Question-Answering for Contextual Response Prediction System," arXiv preprint arXiv:2409.03708, 2024.

[2] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T. Chua, and Q. Li, "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, 2024, pp. 6491-6501.

[3] A. S. Sajeev, A. S. Joseph, A. Madhav T et al., "VisionVerse: Dynamic Video Question Answering Through Retrieval-Augmented Generation," Research Square, 2024. [Online]. Available: https://doi.org/10.21203/rs.3.rs-4372886/v1

[4] "Vector Stores and Retrievers," LangChain Documentation, 2024. [Online]. Available: https://python.langchain.com/docs/integrations/vectorstores/

[5] "Vector Similarity Explained," Pinecone Learning Center, 2024. [Online]. Available: https://www.pinecone.io/learn/vector-similarity/

[6] "Recommender Systems: Machine Learning Metrics and Business Metrics," Neptune.ai, 2024. [Online]. Available: https://neptune.ai/blog/recommender-systems-metrics

[7] E. Bernhardsson, "Approximate nearest neighbors oh yeah," GitHub repository, 2023. [Online]. Available: https://github.com/spotify/annoy

[8] "Qdrant Vector Database," Qdrant Documentation, 2024. [Online]. Available: https://qdrant.tech/documentation/

[9] "Whisper Model: Robust Speech Recognition," OpenAI, 2023. [Online]. Available: https://openai.com/research/whisper

[10] "Groq LLM API Documentation," Groq, 2024. [Online]. Available: https://groq.com/