Hierarchical RAG Framework with Multi-Level Granularity

Ran Arino, Nipun Ravindu Fernando Warnakulasuriya, Umer Aftab

rarino@myseneca.ca, nwarnakulasuriya1@myseneca.ca, uaftab@myseneca.ca

School of Software Design and Data Science, Seneca Polytechnic

Abstract

Traditional **Retrieval-Augmented Generation (RAG) systems** often struggle with fixed retrieval granularity and lack hierarchical context understanding.

This project introduces a **hierarchical RAG framework leveraging multi-level granularity** (documents, clusters, chunks).

Our approach involves structured indexing via **semantic clustering and multi-step retrieval** (cluster-to-chunk search), aiming to enhance both efficiency and context-awareness.

Objective

1. Develop a hierarchical RAG architecture utilizing **multi-level granularity**.
2. Implement structured indexing using **semantic clustering (document/chunk)**.
3. Design **an multi-step retrieval process**.
4. Evaluate the framework's effectiveness in retrieval accuracy and generation quality on diverse QA tasks.

Challenges

- **Suboptimal Granularity**:
  Fixed-size chunk struggles with balance contextual completeness and noise inclusion.
- **Lack of Hierarchy**:
  Flat retrieval ignores document structure, hindering context synthesis.
- **Efficiency vs. Effectiveness**:
  Balancing retrieval speed, accuracy, and contextual relevance remains difficult.

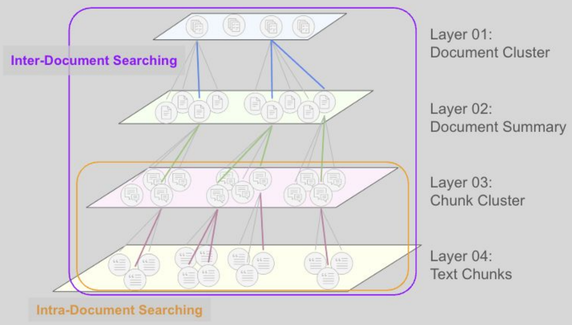
Framework

Indexing Phase:

1. Summarization (Inter-Doc Only):
   Generate & embed document summaries.
2. Segmentation:
   Divide documents into overlapping, semantically coherent chunks (~100 tokens).
3. Embedding:
   Generate vector representations for all chunks.
4. Clustering:
   Group related chunk embeddings by K-means/GMM; store cluster centroids. Creates a hierarchical structure (Doc -> Cluster -> Chunk or Cluster -> Chunk).

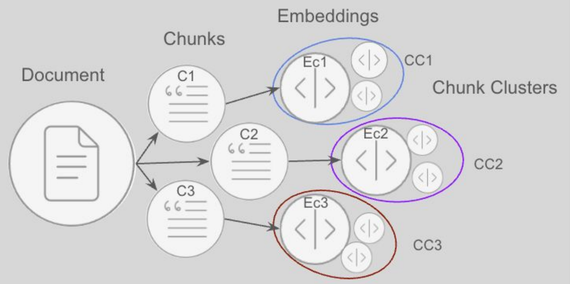
Retrieval Phase:

1. Top-Level Search:
   Match query against highest-level centroids (document or chunk clusters) to find relevant clusters.
2. Query Expansion (Optional) :
   Refine query based on retrieved cluster context.
3. Second-Level Search:
   Search within selected clusters using the (expanded) query to retrieve relevant chunks.
4. Generation:
   Feed retrieved chunks context to LLM for answer synthesis.

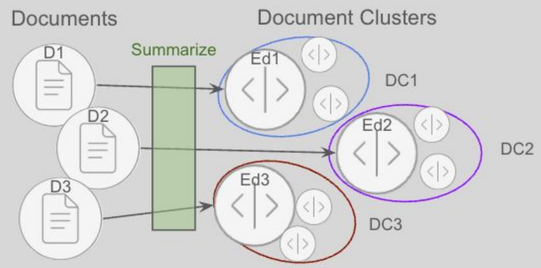


Examples

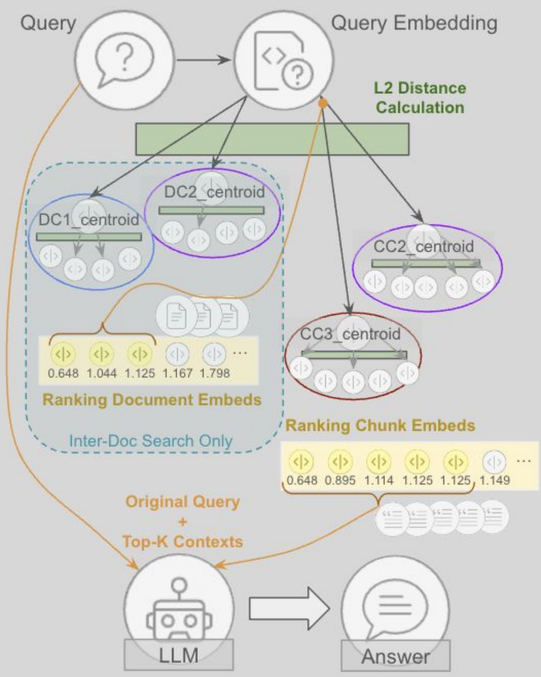
Document => Chunks => Chunk Clusters



Documents => Clusters (only inter-doc search)



How do Retrieval & Generation Work?



Evaluations

Datasets:

- Intra-Document:
  Qasper, NarrativeQA
- Inter-Document:
  MultiHop-RAG, Frames

Retrieval Metrics:

- MAP@K, MRR@K, Hit@K (Relevance & Ranking).
- LLM-as-a-Judge (Contextual Sufficiency).

Generation Metrics:

- ROUGE (Precision/Recall/F1; 1-, 2-, N-gram overlap).
- BLEU (Precision/Fluency).
- Semantic Similarity (Sentence-BERT) vs. Ground Truth.
- LLM-as-a-Judge (Factuality, Relevance, Coherence).

Conclusions

This project proposes a hierarchical RAG framework to overcome limitations of flat retrieval and fixed granularity.

By structuring information hierarchically and employing multi-step retrieval, we aim for improvements in retrieval efficiency and contextual quality of generated responses, especially for complex information needs across single or multiple documents.

Reference

Arino, R., Warnakulasuriya, N.R.F., & Aftab, U. Hierarchical RAG Framework with Multi-Level Granularity.