

Advancing Retrieval-Augmented Generation for Structured Enterprise and Internal Data

Chandana Cheerla
IIT Roorkee
chandana_c@mfs.iitr.ac.in

July 17, 2025

Abstract

Organizations increasingly rely on proprietary enterprise data—including HR records, structured reports, and tabular documents—for critical decision-making. While Large Language Models (LLMs) exhibit strong generative capabilities, they remain constrained by static pretraining, limited context windows, and challenges in processing heterogeneous data formats. Although conventional Retrieval-Augmented Generation (RAG) frameworks solve some of these constraints, they often fall short in handling structured and semi-structured data effectively. This work introduces an advanced RAG framework that combines hybrid retrieval strategies leveraging dense embeddings (allmpnet-base-v2) and BM25, enhanced through metadata-aware filtering using SpaCy NER and cross-encoder reranking for improved relevance. The framework employs semantic chunking to preserve textual coherence and explicitly retains the structure of tabular data, ensuring the integrity of row-column relationships. Quantized indexing techniques are integrated to optimize efficiency, while a human-in-the-loop feedback mechanism and conversation memory enhance the system’s adaptability over time. Evaluations on proprietary enterprise datasets demonstrate significant improvements over baseline RAG approaches, with a 15% increase in Precision@5 (90% vs. 75%), a 13% gain in Recall@5 (87% vs. 74%), and a 16% improvement in Mean Reciprocal Rank (0.85 vs. 0.69). Qualitative assessments further validate the framework’s ef-

fectiveness, showing higher scores in Faithfulness (4.6 vs. 3.0), Completeness (4.2 vs. 2.5), and Relevance (4.5 vs. 3.2) on a 5-point Likert scale. These findings underscore the framework’s capability to deliver accurate, comprehensive, and contextually relevant responses for enterprise knowledge tasks. Future work will focus on extending the framework to support multimodal data and integrating agent-based retrieval systems. The source code will be progressively released at: [GitHub Repository](#).

Keywords: Retrieval-Augmented Generation (RAG), Structured Data Retrieval, Hybrid Retrieval, Tabular Data Processing, Cross-Encoder Reranking, Enterprise Knowledge Augmentation, Metadata Filtering, Semantic Chunking, Query Reformulation, Feedback-Driven Retrieval

1 Introduction

Large Language Models (LLMs) such as GPT-4 [11], PaLM [3], and LLaMA [18] have significantly advanced capabilities in natural language understanding and generation, excelling in tasks like question answering, summarization, and knowledge retrieval. Despite these achievements, LLMs remain inherently constrained by static pretraining on fixed corpora and limited context windows [1, 13], which bounds their adaptability to dynamic or proprietary enterprise data. In domains like corporate governance, human resources, and finance, critical information is often encapsulated in structured

records, policy documents, and tabular formats that LLMs cannot naturally ingest or reason over post-deployment. Retrieval-Augmented Generation (RAG) frameworks have emerged to address this gap by integrating retrieval mechanisms with LLMs, enabling models to fetch external knowledge at inference time [9, 7]. This paradigm enhances response relevance and factuality by grounding generation in up-to-date, domain-specific data. However, conventional or baseline RAG methods, optimized primarily for unstructured textual data, face notable challenges when applied to enterprise datasets comprising structured, semi-structured, and tabular information [5]. Key limitations include:

- **Fragmented Contextual Representation:** Baseline chunking strategies, typically using fixed token-length splits, often fracture meaningful contexts, especially in complex documents like policies or manuals [12].
- **Inadequate Handling of Tabular Data:** Flattening tables into linear text formats destroys the intrinsic row-column relationships necessary for precise data retrieval within tables [20].
- **Limited Retrieval Completeness:** Exclusive reliance on dense embeddings or sparse keyword methods like BM25 [15] restricts the model’s ability to balance semantic understanding with exact matching.
- **Absence of Relevance Reordering:** Without reranking mechanisms, initial retrieval results may not adequately prioritize the most relevant information [8].
- **Static Query Interpretation:** These systems lack the capability to refine or disambiguate queries dynamically, resulting in suboptimal retrieval for vague or incomplete user prompts.

1.1 Contributions

To address these failings, we introduce an enterprise-focused RAG framework that systematically enhances the retrieval and generation

process across diverse data modalities. The contributions of our system are as follows:

1. **Hybrid Retrieval Mechanism:** We combine dense semantic retrieval using all-mpnet-base-v2 [14] with BM25-based sparse retrieval, weighted at a 0.6 to 0.4 ratio, to synergize semantic depth with lexical accuracy.
2. **Structure-Aware Chunking:** For text, we adopt a recursive character-based chunking strategy with empirically tuned chunk sizes (700 tokens) that balance coherence and granularity. For tabular data, we employ Camelot [2] and Azure Document Intelligence to extract and store tables as structured JSON with metadata capturing file name, row identifiers, and column headers, enabling row-level indexing via FAISS for fine-grained retrieval.
3. **Metadata-Driven Filtering:** Named Entity Recognition (NER) via spaCy [17] enriches document metadata, facilitating entity-aligned filtering that sharpens retrieval relevance.
4. **Contextual Re-Ranking:** We integrate a cross-encoder reranking layer using MS MARCO fine-tuned models [10], which reorders retrieved candidates based on contextual alignment with the query.
5. **Interactive Query Refinement:** Leveraging LLaMA and Mistral models on ChatGroq, our system supports query expansion and rephrasing, guided by user feedback and conversational memory to iteratively enhance retrieval and response quality.

We evaluated our framework on a corpus of publicly available enterprise policy documents, including the HR Policy Dataset and analogous corporate materials. For tabular data, we experimented with both full-table chunking and row-level indexing, finding the latter significantly enhances precision in row-specific queries while the former suffices for small tables.

1.2 Results

Our approach delivers substantial improvements over baseline RAG systems:

- Precision@5: 90% vs. 75%
- Recall@5: 87% vs. 74%
- Mean Reciprocal Rank (MRR): 0.85 vs. 0.69

These results underscore the robustness and applicability of our framework in real-world enterprise contexts, delivering superior retrieval accuracy, comprehensive responses, and higher contextual relevance.

We envision extending this framework towards agentic RAG systems [4, 16], where intelligent agents autonomously select retrieval strategies, adaptively reformulate queries, and integrate multimodal data sources—including images, scanned documents, and audio—thereby broadening the landscape of enterprise knowledge augmentation.

1.3 Paper Organization

The rest of the paper is organized as follows:

- Section 2: Elaborates on the related work.
- Section 3: Presents the methodology.
- Section 4: Discusses experiments and results.
- Section 5: Concludes with directions for future work, including potential extensions towards agentic retrieval systems.

2 Related Work

Retrieval-Augmented Generation (RAG) has emerged as a pivotal paradigm to bridge the limitations of static LLMs by dynamically fetching relevant information at inference time. Lewis et al. [9] introduced the foundational RAG architecture by coupling dense retrievers with generative models, improving factual grounding in open-domain question answering. Building on

this, Izacard and Grave [7] proposed Fusion-in-Decoder (FiD), integrating multiple retrieved contexts directly into the decoding process, significantly enhancing response accuracy. Dense retrieval techniques such as Dense Passage Retrieval (DPR) [8] rely on bi-encoder models to map queries and documents into the same vector space, enabling efficient retrieval via vector similarity. However, DPR and similar dense retrievers can struggle with lexical precision, especially when queries are domain-specific or contain rare terminology. Sparse retrieval methods like BM25 [15] compensate for this by focusing on exact term matching, which, while precise, often lacks semantic understanding. To overcome the trade-offs between dense and sparse retrieval, hybrid strategies have been explored. Mialon et al. [21] surveyed augmented language models and emphasized that combining dense and sparse signals leads to better retrieval performance across varied data types. Yet, most hybrid retrieval approaches have predominantly targeted unstructured textual data, with limited exploration into structured or semi-structured formats such as tables and structured records. Handling structured data directly has been tackled through models like TURL (Table Understanding through Representation Learning) by Zhang et al. [20], which encodes tabular semantics for downstream tasks, and TAPAS [6], which leverages transformers to answer questions over tables. However, these models are optimized for direct table question answering rather than integrating into broader retrieval-generation pipelines that handle mixed data formats. Reranking retrieved candidates post-retrieval further enhances relevance. Nogueira and Cho [10] introduced BERT-based passage reranking, showing that cross-encoders can significantly refine retrieval results by evaluating query-document pairs holistically. Similarly, ColBERT [22] employs a late interaction mechanism, balancing between efficiency and precision, though its scalability remains an area of active research.

2.1 Advantages of Prior Approaches

- **Semantic and Lexical Balance:** Hybrid retrieval effectively combines semantic depth with keyword accuracy.
- **Improved Relevance:** Reranking mechanisms like cross-encoders enhance the contextual fit of retrieved results.
- **Handling Structured Data:** Specialized models like TAPAS [6] and TURL [20] address the nuances of tabular understanding.

2.2 Gap Addressed by Our Work

While these advancements have laid a robust foundation, there remains a lack of unified frameworks that seamlessly integrate hybrid retrieval, structured data handling, metadata filtering, reranking, and interactive refinement specifically tailored for enterprise environments. Our work addresses this comprehensive need, extending RAG capabilities to structured, semi-structured, and unstructured enterprise data, setting a precedent for future multimodal and agent-driven retrieval systems.

3 Methodology

Our proposed advanced RAG framework is designed to effectively retrieve and generate responses from heterogeneous enterprise data, including text, structured documents, and tabular records. The architecture addresses limitations in naive RAG pipelines through a combination of optimized document preprocessing, hybrid retrieval strategies, advanced ranking mechanisms, and feedback-driven refinement. This section details each stage of our system.

3.1 Document Preprocessing and Chunking

3.1.1 Text Extraction and Semantic Chunking

All textual documents, primarily sourced from publicly available HR policies (e.g., NASSCOM datasets), are extracted using pdfplumber. The

extracted text is segmented using a Recursive Character Text Splitter with a chunk size of 2000 characters and a 500-character overlap, ensuring semantic coherence while maintaining the token constraints of LLMs.

3.1.2 Table Extraction and Representation

To accurately handle tabular data, we employ Camelot [2] for table detection and extraction. Tables are serialized into JSON format, capturing metadata such as:

- File Name
- Row and Column Identifiers
- Cell Values
- Split into individual rows, each indexed separately to facilitate row-level retrieval.

When Camelot is insufficient (e.g., for complex formatting), we fallback to pdfplumber for extracting table-like structures. This ensures robust extraction across diverse document types.

3.2 Metadata Enrichment

We apply spaCy’s Named Entity Recognition (NER) [17] to annotate each chunk with entities such as locations, dates, and organizational names. Additional metadata like document type, department, and confidentiality level is simulated for experimentation but can be replaced with real enterprise metadata.

3.3 Hybrid Retrieval Strategy

Our retrieval pipeline combines dense retrieval, sparse retrieval, and reranking, enhancing both semantic understanding and keyword precision.

3.3.1 Dense Retrieval

Chunks are embedded using all-mpnet-base-v2 embeddings [14], a state-of-the-art sentence embedding model. These embeddings are stored in a FAISS HNSW index (with M=32, efConstruction=200, efSearch=50), enabling efficient approximate nearest neighbor searches.

3.3.2 Sparse Retrieval

We employ BM25 [15], a classic keyword-based retrieval algorithm, to complement dense retrieval, particularly beneficial for exact term matching.

3.3.3 Retrieval Fusion

Dense and sparse retrieval scores are combined using a weighted sum:

$$\text{Score}_{\text{combined}} = 0.6 \times \text{Score}_{\text{dense}} + 0.4 \times \text{Score}_{\text{sparse}} \quad (1)$$

This weighting was determined empirically to balance semantic and lexical relevance.

3.4 Contextual Reranking

The top candidate chunks are reranked using a Cross-Encoder reranker based on the ms-marco-MiniLM-L-12-v2 model [10]. This cross-encoder evaluates the query-chunk pairs to reassign relevance scores, ensuring that the most contextually aligned documents are prioritized.

3.5 Query Formulation and Refinement

To enhance initial queries, we incorporate:

- **Query Rewriting:** Rephrasing vague or incomplete queries using LLaMA or Mistral models on ChatGroq.
- **Query Expansion:** Generating alternative query formulations to cover broader aspects of the topic.

If a user flags an answer as unsatisfactory, the query is automatically expanded and retried, leveraging the LLM to guide reformulation.

3.6 Answer Generation with LLMs

Final reranked chunks are passed to LLMs for answer synthesis. We utilize:

- Mistral-7B and LLaMA models on ChatGroq for their balance between accuracy and inference speed.

- A Grounded Prompt Template that instructs the model to:

- Answer strictly based on retrieved sources.
- Use bullet points for clarity.
- Provide citations to source documents.
- Include summaries if the response exceeds three sentences.

3.7 Feedback Loop and Conversational Memory

A ConversationBufferMemory retains up to 10 recent interactions, maintaining session continuity. Additionally, user feedback (thumbs up/down) is logged. Negative feedback triggers automated query reformulation and re-retrieval, enhancing system adaptivity over time.

3.8 Index Optimization

We maintain two FAISS indices:

- **High-Precision Index:** Uses all-mpnet-base-v2 embeddings [14] for general queries.
- **Lightweight Index:** Uses paraphrase-MiniLM-L3-v2 embeddings for resource-constrained environments.

This dual-index system offers a trade-off between computational efficiency and retrieval accuracy.

3.9 Evaluation Dataset and Setup

Our evaluations used a corpus of predominantly HR policies from publicly available sources and corporate reports. The dataset contains a rich mix of text and tables, simulating real enterprise data diversity.

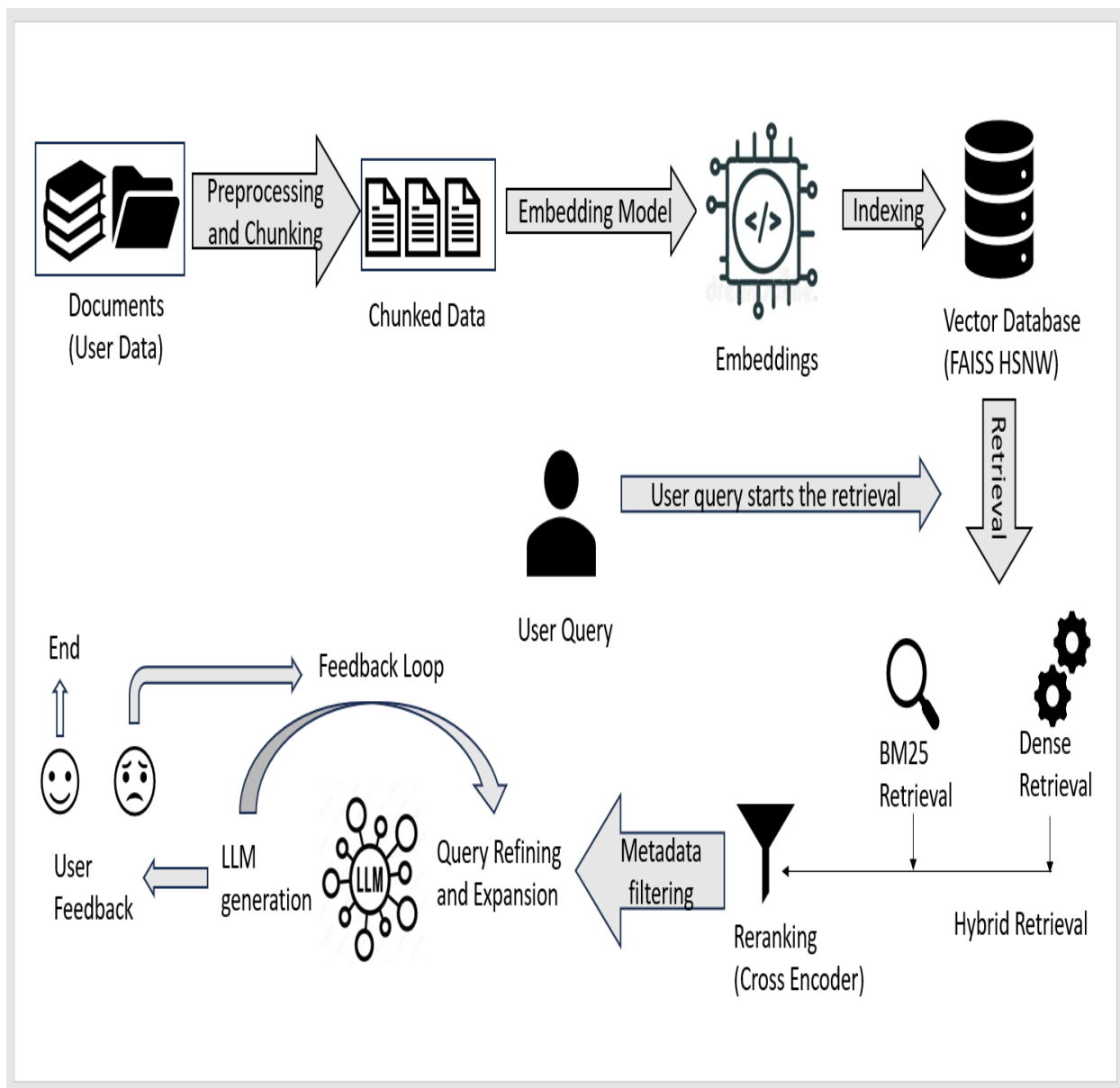


Figure 1: Architecture Diagram of the Proposed RAG Framework

3.10 Metrics

We evaluated the system quantitatively and qualitatively:

- Precision@5, Recall@5, MRR: Standard retrieval metrics.
- Faithfulness, Completeness, Relevance: Assessed on a 5-point Likert scale by human evaluators.

4 Experiments

4.1 Dataset

We evaluated our framework on a dataset comprising publicly available HR policies (e.g., companies and institutions data) and corporate reports. This dataset represents a diverse collection of unstructured text, structured data, and tabular content, simulating real-world enterprise knowledge repositories. Additionally, supplementary experiments utilized datasets from public repositories such as, (<https://www.data.gov.in/>) (<https://archive.ics.uci.edu/ml/datasets/adult>) HR Analytics GitHub Repository.

4.2 Experimental Procedure

We adopted a progressive experimental methodology, beginning with a naive RAG baseline and incrementally integrating advanced retrieval and generation techniques. This stepwise approach allowed us to quantify the impact of each enhancement in the pipeline.

4.3 Naive RAG Baseline

The baseline system was configured as follows:

- Recursive Character-Level Chunking: Documents were segmented into chunks of 500, 700, and 1000 characters to balance context preservation and retrievability. A chunk size of approximately 700 characters yielded optimal performance, although variations across sizes were marginal.

- Dense Retrieval Only: Document embeddings were generated using all-mpnet-base-v2 [14], indexed via FAISS for dense retrieval.
- Direct LLM Generation: Retrieved chunks were passed directly to the LLM without any reranking or filtering mechanisms.
- Tabular Data: Initially, tables were treated as plain text and chunked similarly to unstructured text, which led to suboptimal performance, especially for row-specific queries.

4.4 Table Handling Strategies

Recognizing the inadequacies in handling tabular data, we experimented with several strategies:

1. Storing Entire Tables as Chunks: Effective for small tables (less than 10 rows), but impractical for larger tables due to context window constraints and poor precision in row-specific retrievals.
2. Azure Document Intelligence: Employed to parse tables into structured formats enriched with detailed metadata for rows, columns, and headers.
3. Camelot Integration: Utilized Camelot [2], an open-source tool, to extract tables into JSON format, preserving structural relationships.
4. Row-Level Indexing: Each table row was indexed separately within FAISS, enabling fine-grained retrieval for row-specific queries.

4.5 Full Advanced RAG Pipeline

Upon refining the table handling mechanisms, we implemented the complete advanced RAG pipeline:

- Hybrid Retrieval: Combined dense retrieval (all-mpnet-base-v2 [14]) with sparse retrieval (BM25 [15]) using a weighted fusion: 0.6 (dense) and 0.4 (sparse).

- **Semantic and Table-Aware Chunking:** Enhanced chunking strategies ensured coherence for textual data and structural integrity for tables.
- **Cross-Encoder Reranking:** Applied ms-marco-MiniLM-L-12-v2 [10] as a cross-encoder reranker to score and reorder the top-k retrieved chunks.
- **Query Refinement:** Integrated automatic query rewriting and expansion using LLaMA and Mistral models on ChatGroq, especially in response to negative user feedback.
- **Feedback Loop:** A human-in-the-loop mechanism where user feedback triggered query reformulation and re-retrieval.

4.6 Evaluation Metrics

We assessed the system using both quantitative and qualitative metrics:

- **Precision@5:** The proportion of relevant documents within the top 5 retrieved results, indicating immediate utility.
- **Recall@5:** The proportion of all relevant documents captured within the top 5 results, reflecting coverage.
- **Mean Reciprocal Rank (MRR):** Measures the rank position of the first relevant document, with higher values indicating quicker retrieval.

Additionally, qualitative evaluations were conducted with human annotators who rated responses on:

- **Faithfulness:** The degree to which generated answers accurately reflect retrieved content without hallucination.
- **Completeness:** The extent to which the response comprehensively addresses the query.
- **Relevance:** The pertinence of the response to the user’s query.

And also we use LLMs as an another evaluator. Each qualitative metric was assessed on a 5-point Likert scale.

4.7 Results

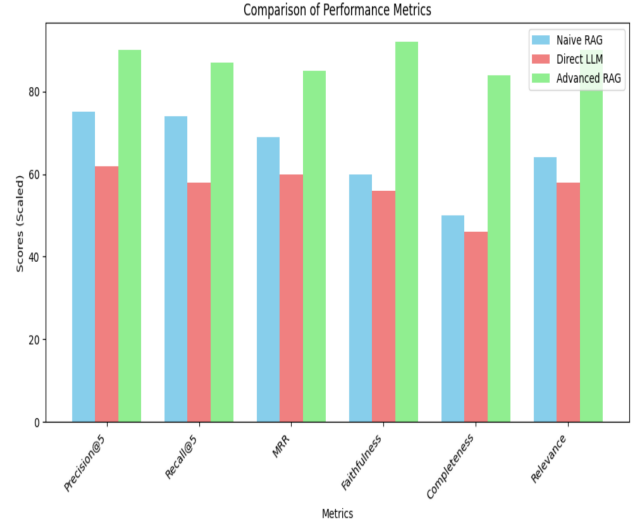


Figure 2: Comparison of the performance metrics

Metric	Direct LLM	Naive RAG	Advanced RAG
Precision@5	62%	75%	90%
Recall@5	58%	74%	87%
MRR	0.60	0.69	0.85
Faithfulness	2.8	3.0	4.6
Completeness	2.3	2.5	4.2
Relevance	2.9	3.2	4.5

Table 1: Comparison of performance metrics across Naive RAG, Direct LLM, and Advanced RAG.

4.8 Key Observations

- **Chunking:** A 700-character chunk size offered a balanced trade-off between context preservation and retrievability, though the impact of chunk size was not significant beyond certain thresholds.
- **Table Handling:** Implementing row-level indexing drastically improved retrieval precision for tabular queries compared to naive chunking.

- **Component Contributions:** Each enhancement—hybrid retrieval, cross-encoder reranking, and query refinement—provided incremental improvements, cumulatively resulting in substantial gains in both quantitative and qualitative metrics.

4.9 Summary

Our experimental results demonstrate that the proposed advanced RAG framework significantly outperforms both naive RAG and direct LLM prompting approaches. The combination of hybrid retrieval, semantic and structure-aware chunking, cross-encoder reranking, and dynamic query refinement enables effective handling of heterogeneous enterprise data, including complex tabular formats. Consistent improvements across Precision@5, Recall@5, and MRR, alongside higher human ratings for faithfulness, completeness, and relevance, affirm the robustness of our pipeline for real-world enterprise knowledge augmentation tasks. These findings validate the necessity of tailored retrieval strategies and structured data handling within enterprise RAG systems.

5 Advantages of the Proposed Framework

Our advanced RAG framework introduces several key improvements that make it well-suited for handling enterprise data retrieval and generation tasks. First, the framework is capable of effectively working with a wide variety of data formats commonly found in enterprises, including unstructured text, structured documents, and tabular data. This versatility makes it practical for real-world scenarios where information is dispersed across different formats. Second, the hybrid retrieval approach—combining dense embeddings with sparse keyword-based methods—strikes a balance between semantic understanding and exact keyword matching. This ensures that the system retrieves information that is both contextually relevant and factually precise. The additional layer of cross-encoder

reranking further refines the results, prioritizing the most relevant content. Another strength of the framework is its approach to tabular data. By implementing table-aware chunking and indexing each row individually, the system achieves a level of granularity that allows it to answer row-specific queries more effectively than standard text chunking would allow. Additionally, the system includes dynamic query optimization through LLM-based rewriting and expansion, enabling it to refine ambiguous or incomplete queries. This makes the retrieval process more robust, especially when dealing with varied user inputs. For generation, the use of a grounded prompting strategy ensures that LLM responses remain anchored in the retrieved evidence, with citations and summaries provided where necessary. This not only enhances the credibility of the answers but also mitigates the risk of hallucinations. Lastly, the framework is designed with scalability in mind through a dual-index approach—offering a high-precision index for accuracy-demanding tasks and a lightweight alternative for resource-constrained environments. The system’s modular design also makes it adaptable to domains beyond HR and corporate data, such as healthcare, legal, and financial applications.

6 Limitations and Future Work

While the framework offers several advantages, there are still some limitations that we aim to address in future work. One of the primary limitations is the reliance on static indexing. As it stands, any updates to the document corpus require full reindexing, which can be time-consuming and impractical in environments where data changes frequently. A more dynamic, incremental indexing mechanism is needed to make the system responsive to real-time data updates. Another area for improvement is in handling highly complex or nested tables. Although our current approach performs well on simple to moderately complex tables, it can struggle with preserving relationships in deeply structured tables with hierarchical headers or merged cells.

This sometimes leads to partial loss of context during retrieval. The feedback mechanism, while useful, currently depends on explicit user input like thumbs up or down. In real-world settings, users may not always provide such feedback, limiting the system’s capacity to learn and adapt automatically. Incorporating passive signals, such as how users interact with the retrieved information, could help the system improve over time without requiring direct input. We also see limitations in the query reformulation process. Presently, it relies on heuristic-based expansions using LLMs, which can occasionally misinterpret user intent or broaden the query too much, leading to less focused results. More intent-aware or interactive clarification methods could address this issue. Finally, while our retrieval fusion strategy is effective on moderately sized datasets, scaling it to very large corpora poses computational challenges, particularly in terms of retrieval latency and memory usage. More efficient fusion and reranking techniques will be essential for handling enterprise-scale deployments.

6.1 Future Work

Looking ahead, we plan to enhance the framework in several ways. One direction is to implement dynamic indexing capabilities, allowing the system to update its indices incrementally as new data arrives. We also intend to integrate advanced table understanding models like TAPAS [6] or TURL [20], which are specifically trained to preserve relational structures in complex tables. In terms of user feedback, we aim to explore methods for leveraging implicit feedback signals—such as click-through rates and time spent on documents—to guide system improvements without relying on explicit ratings. To improve query handling, we are considering the use of agent-based approaches, such as ReAct agents [19], that can reason about query intent and dynamically adapt retrieval strategies. Finally, we plan to extend the system to support multimodal data, including scanned documents, images, and charts, broadening its applicability across various enterprise contexts.

References

- [1] R. Bommasani et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [2] Camelot Project. Camelot: PDF Table Extraction for Humans. 2018. <https://camelot-py.readthedocs.io>.
- [3] A. Chowdhery et al. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [4] Y. Gao et al. Precise Zero-Shot Dense Retrieval without Relevance Labels. *arXiv preprint arXiv:2212.10496*, 2022.
- [5] K. Guu et al. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv preprint arXiv:2002.08909*, 2020.
- [6] J. Herzig et al. TAPAS: Weakly Supervised Table Parsing via Pre-training. *arXiv preprint arXiv:2004.02349*, 2020.
- [7] G. Izacard and E. Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. *arXiv preprint arXiv:2007.01282*, 2020.
- [8] V. Karpukhin et al. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2004.04906*, 2020.
- [9] P. Lewis et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401*, 2020.
- [10] R. Nogueira and K. Cho. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.
- [11] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.
- [12] J. Phang et al. Clustering and Chunking Strategies for Efficient Retrieval. *arXiv preprint arXiv:2104.07511*, 2021.
- [13] O. Press et al. Train Short, Test Long: Attention with Linear Biases. *arXiv preprint arXiv:2108.12409*, 2021.

- [14] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084*, 2019.
- [15] S. Robertson and H. Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 2009. <https://www.nowpublishers.com/article/Details/INR-018>.
- [16] N. Shinn et al. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint arXiv:2303.11366*, 2023.
- [17] spaCy Team. spaCy: Industrial-Strength Natural Language Processing. 2020. <https://spacy.io>.
- [18] H. Touvron et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*, 2023.
- [19] S. Yao et al. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629*, 2022.
- [20] R. Zhang et al. TURL: Table Understanding through Representation Learning. *arXiv preprint arXiv:2006.14806*, 2020.
- [21] G. Mialon et al. Augmented Language Models: A Survey. *arXiv preprint arXiv:2302.07842*, 2023.
- [22] O. Khattab and M. Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *arXiv preprint arXiv:2004.12832*, 2020.