Hybrid Search for RAG-Powered Banking LLM Chatbot

R. Taktashov[1],
[1] Institute of Biomedical Chemistry (IBMC), Moscow, Russia

Abstract

Retrieval-Augmented Generation (RAG) chatbots are transforming banking by providing real-time, accurate assistance through dynamic retrieval from trusted financial sources. However, these systems face significant challenges with hallucinations—plausible but factually incorrect outputs—which pose serious risks in financial contexts, including regulatory violations and erosion of customer trust.
To address these challenges, we present an optimized RAG pipeline that combines hybrid search (lexical retrieval via BM25 and semantic retrieval through Chroma Vector Store) with careful evaluation using both METEOR and RAGAS Factual Correctness metrics. Our best-performing configuration achieves a Factual Correctness score of 0.41 and a METEOR score of 0.39 using hybrid search with recursive text splitting.

1 Introduction

In the fast-paced world of banking, customers demand instant, accurate, and personalized assistance. **R**etrieval-**A**ugmented **G**eneration (**RAG**) chatbots are revolutionizing financial services by combining the power of **L**arge **L**anguage **M**odels (**LLM**) with real-time data retrieval, ensuring responses are both contextually relevant and factually grounded. Unlike traditional chatbots that rely solely on static training data, RAG systems dynamically fetch information from trusted sources—such as bank policies, transaction records, and regulatory guidelines—to provide precise answers tailored to each query.

1.1 Goals

Our primary goal is to address the critical challenge of hallucinations in AI-driven chatbots, where plausible but incorrect information could lead to serious consequences in banking (e.g., misinforming customers about loan terms or providing inaccurate account details). We aim to maximize the **METEOR** score—a robust metric for response quality—while employing hybrid search (lexical + semantic retrieval) with `Chroma` Vector Store and BM25S Index. By complementing **METEOR** with `RAGAS` metric **Factual Correctness**, we hope to ensure that responses are factually consistent and retrieval-aware.

1.2 Metrics

When deploying RAG chatbots in banking, measuring performance is critical—yet complex. No single benchmark serves as a "silver bullet". We decided to limit ourselves with evaluating model responses and synthetic ground-truth responses due to time constraints. Our evaluation combines:

- **METEOR**: based on the harmonic mean of unigram precision and recall
- **Factual Correctness**: based on the harmonic mean of LLM-calculated precision and recall

1.3 Datasets

We were provided with 2 QA datasets:

- **Validation set**: validation examples with synthetic ground-truth responses.
- **Test set**: unlabeled questions for final evaluation

2 Background & Literature Review

2.1 Text Splitting (Chunking)

The choice of text splitting strategy is crucial for retrieval quality and text relevance in RAG systems [1]. While we employed recursive character text splitting, more sophisticated approaches exist. Notably, semantic splitting with different types of breakpoints can create more meaningful chunks based on actual content rather than arbitrary rules. This approach relies on calculating cosine similarities of embedded sentences.

Recent work suggests semantic chunking may improve retrieval accuracy, though its computational cost warrants careful consideration. Some researchers argue that semantic chunking's benefits may not justify the additional effort [2]. The optimal approach depends on both document characteristics and the specific retrieval task. Even though semantic methods show promise, traditional recursive splitting remains widely adopted for its reliability and efficiency in many practical applications.

2.2 Evaluation Metrics

The main metric in this challenge, which we aimed to maximize, is **METEOR** (**M**etric for **E**valuation of **M**achine **T**ranslation with **E**xplicit **OR**dering). It compares the output by comparing it to human-generated reference translations. It addresses limitations of **BLEU** by incorporating explicit word-matching strategies, including stemming, synonymy (via WordNet), and paraphrasing, alongside alignment-based scoring [3] [4]. Neither **BLEU**, nor **METEOR** evaluate whether the answer accurately reflects the retrieved content or addresses the query effectively. Even if the answer is accurate, it might receive low scores due to limited word overlap, paraphrased or reordered explanations of the same concepts.

To address this limiation, we added **Factual Correctness** as a supplementary metric, provided by `RAGAS`. It directly compares factual accuracy of the generated response with the reference [5]. To measure the alignment between the response and the reference, the metric uses the LLM to first break down the response and reference into claims and then uses natural language inference to determine the factual overlap between the response and

the reference.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Factual Correctness} \equiv \text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where:

- TP = Number of claims in the response that are present in reference
- FP = Number of claims in response that are not present in reference
- FN = Number of claims in reference that are not present in response

2.3 Retrieval Methods

Our system employs a hybrid retrieval approach combining two complementary methods. For semantic search, we utilize `ChromaDB` vector store through LangChain's native integration, which provides efficient dense retrieval capabilities using the `Nomic-embed-text-v1.5 embeddings` [6]. This allows the system to capture nuanced contextual relationships between queries and documents.

For lexical retrieval, we implemented a custom BM25S solution rather than using LangChain's standard BM25 implementation [7]. This decision was motivated by two critical factors. First, BM25S demonstrates significantly faster retrieval speeds - often by orders of magnitude - which is essential for maintaining real-time responsiveness in banking applications. Second, unlike the default in-memory BM25 implementation in LangChain, our BM25S solution supports persistent on-disk storage of indices. This persistence capability eliminates the need to rebuild indices between sessions while maintaining efficient memory usage.

The combination of these retrieval methods through reciprocal rank fusion creates a robust hybrid system that leverages both precise term matching (BM25S) and semantic understanding (ChromaDB). This dual approach is particularly valuable in banking contexts where queries may contain both specific financial terminology requiring exact matches and more conceptual questions.

3 Methods

3.1 Document Processing

The knowledge base undergoes preprocessing:

- **Recursive character splitting with**:
  - Chunk size: 1100 characters
  - Overlap: 110 characters (10% of Chunk Size)
  - Primary separator: $\backslash n \backslash n$
- **Semantic Chunking**:
  - Breakpoint threshold type: interquartile
  - Embeddings: `Nomic-embed-text-v1.5`
- **Embedding Generation**: Using `Nomic-embed-text-v1.5 with`:
  - `CUDA` acceleration (batch size=32)
  - L2-normalized embeddings
- **Stores**:
  - Persistent storage at `./chroma_db_ollama` and `./bm25s_index`

3.2 Query Processing

The QA chain implements:

- **Retrieval**: Hybrid search with:
  - Two stores: Chroma and BM25S
  - Top-4 document retrieval using Reciprocal Rank Fusion
- **Generation**: Constrained LLM inference:
  - Model: local Qwen2.5 (4k context in tokens) with 7B parameters
  - Temperature: 0.0 for deterministic outputs

3.3 Implementation Details

Component	Specification
Embedding Model	Nomic-embed-text-v1.5
LLM	Qwen2.5 (Ollama local deployment)
Stores	ChromaDB and BM25S Index with persistent storage
Hardware	NVIDIA GPU RTX 3090 Mobile

In our implementation, we used the same LLM for both answer generation and evaluation of the Factual Correctness metric: Qwen2.5 with 7B parameters, using Ollama's LangChain integration with `Python 3.10`. While specialized fine-tuned LLMs exist specifically for evaluation purposes, there is currently no established gold standard for this task. Our system architecture is shown in **Figure 1**.

4 Results

The evaluation of our hybrid search RAG pipeline for the banking LLM chatbot yielded significant insights into the effectiveness of different retrieval and text-splitting configurations. The results, summarized in **Table 1**, highlight the performance of each approach in terms of **Factual Correctness** and **METEOR** scores. Key findings include:

- **Splitting**: This configuration achieved the highest scores, with a **Factual Correctness** of 0.41 and a **METEOR** score of 0.39. The combination of BM25 (lexical retrieval) and Chroma Vector Store (semantic retrieval) proved most effective in balancing precision and recall, ensuring both factual accuracy and high-quality responses.
- **Semantic Splitting Performance**: While semantic splitting showed promise, it underperformed compared to recursive splitting, with the best hybrid configuration scoring 0.37 in **Factual Correctness** and 0.36 in **METEOR**.
- **Retrieval Methods**
  - **BM25S Alone**: Achieved better results, particularly with recursive splitting (**Factual Correctness**: 0.41, **METEOR**: 0.38).
  - **Vector Store Alone**: Consistently lagged behind hybrid and BM25, indicating the limitations of purely semantic retrieval in banking contexts where precise term matching is critical.

- **Pipeline Efficiency**: The integration of Reciprocal Rank Fusion (RRF) and FlashRank reranking ensured that the top-4 documents from each retriever were optimally combined, further enhancing the relevance and accuracy of the final LLM-generated responses.

The **Factual Correctness** and **METEOR** scores differ only slightly for each configuration, making **Factual Correctness** metric slightly more optimistic than **METEOR**. We didn't use metrics provided by RAGAS which leverage retrieved contexts for each query due to time constraints.

We have not modified the knowledge base itself. However, upon inspecting each text file, we observed inconsistencies in tabulation and document structure. These variations complicate the development of a reliable regular expression for recursive document splitting. As such, picking $\backslash n \backslash n$ as a primary separator was the safest option.

It should be noted that we retained the default prompt provided by the project organizers without modification. Our objective was to optimize evaluation metrics while avoiding undue reliance on prompt engineering, thereby ensuring that performance improvements stemmed primarily from methodological enhancements rather than linguistic adjustments to the input instructions. Below is the prompt template that was used to generate RAG answers:

```
Answer the client's question concisely, clearly, and to the
point—without speculation or digressions. You cannot ask for more
context or ask questions, just answer. Use information from given
context below.
Context: {context}
Question: {question}
Answer:
```

5 Conclusion

This study demonstrates the effectiveness of a hybrid search RAG pipeline for banking chatbots, combining lexical (BM25S) and semantic (Chroma Vector Store) retrieval methods with recursive text splitting. Our best-performing configuration achieved a **Factual Correctness score** of **0.41** and a **METEOR** score of **0.39**. Future work could explore optimizations, such as retrieval parameter tuning combined with switching to SoTA LLMs.

6 Code availability

The code for this project is available at github

References

[1] Aadit Kshirsagar. "Enhancing RAG Performance Through Chunking and Text Splitting Techniques". In: *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 10 (Sept. 2024), pp. 151–158. DOI: 10.32628/CSEIT2410593.

[2] Renyi Qu, Ruixuan Tu, and Forrest Bao. *Is Semantic Chunking Worth the Computational Cost?* 2024. arXiv: 2410.13070 [cs.CL]. URL: https://arxiv.org/abs/2410.13070.

[3] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ed. by Jade Goldstein et al. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909/.

[4] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040/.

[5] Shahul Es et al. "RAGAs: Automated Evaluation of Retrieval Augmented Generation". In: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Ed. by Nikolaos Aletras and Orphee De Clercq. St. Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 150–158. DOI: 10.18653/v1/2024.eacl-demo.16. URL: https://aclanthology.org/2024.eacl-demo.16/.

[6] Zach Nussbaum et al. *Nomic Embed: Training a Reproducible Long Context Text Embedder*. 2025. arXiv: 2402.01613 [cs.CL]. URL: https://arxiv.org/abs/2402.01613.

[7] Stephen Robertson and Hugo Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond". In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389. ISSN: 1554-0669. DOI: 10.1561/1500000019. URL: https://doi.org/10.1561/1500000019.

Table 1
Comprehensive RAG Evaluation Metrics

Configuration	Factual Correctness	METEOR
Semantic Splitting		
+ BM25S Only	0.38	0.37
+ Chroma Only	0.33	0.33
+ Hybrid	0.37	0.36
Recursive Splitting		
+ BM25S Only	0.41	0.38
+ Chroma Only	0.38	0.37
+ Hybrid	0.41	0.39

Note: All metrics measured on 0-1 scale (higher is better). Best-performing configuration is highlighted in green.

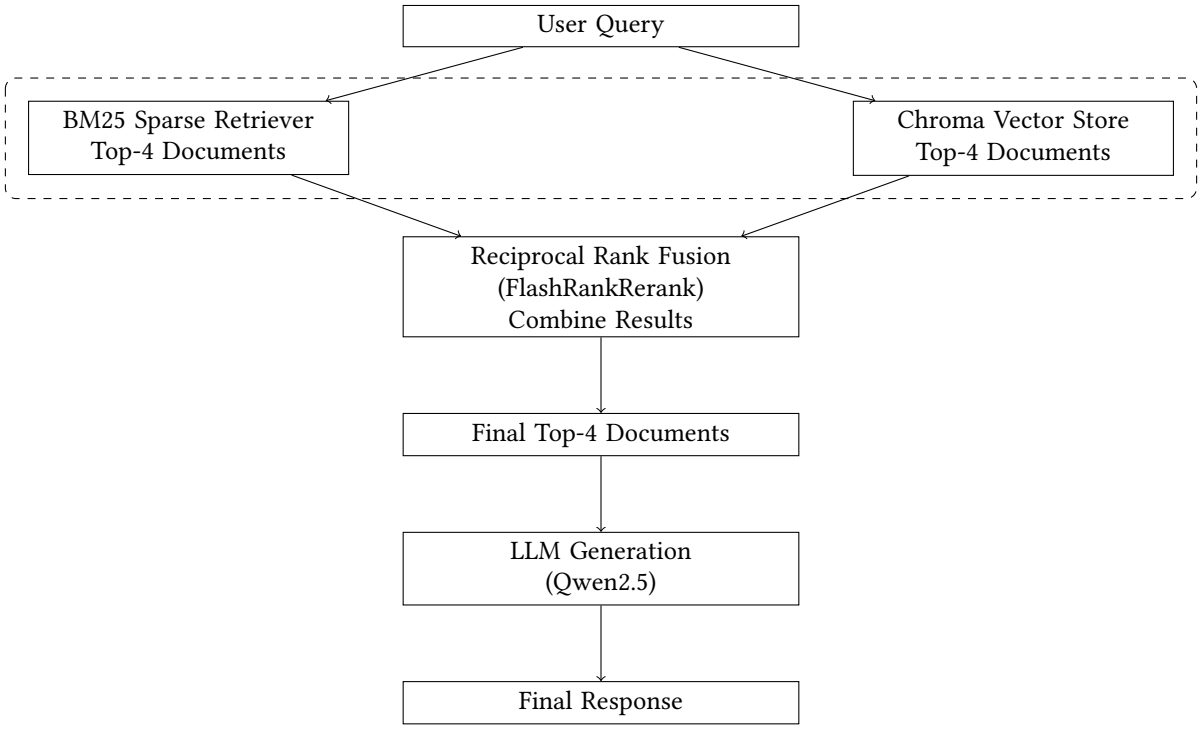

Figure 1. Hybrid search pipeline combining BM25 (sparse) and Chroma (dense) retrievers with Reciprocal Rank Fusion and FlashRank reranking. The system takes top-4 results from each retriever, combines them using RRF, then reranks to produce final top-4 documents for LLM generation.