International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Adaptive Chunking and Hybrid Retrieval for Enhanced Retrieval-Augmented Generation

Yash Patil
Vidyalankar Institute of Technology Mumbai, India
yn05patil@gmail.com

Punit Jadhav
Vidyalankar Institute of Technology Mumbai, India
punit.jadhav@vit.edu.in

Om Nikam
Vidyalankar Institute of Technology Mumbai, India
om.nikam@vit.edu.in

Deepali Shrikhande
Vidyalankar Institute of Technology Mumbai, India
deepali.shrikhande@vit.edu.in

***Abstract*—RAG uses external knowledge through large language models, but how reliable this technique is depends on two key design aspects that are higher up in the system: chunking (or segmentation) and choosing the right chunks (retrieval).Using a fixed chunk size doesn't take the structure of the document into account and can lose the meaning of the text.Also, dense-vector retrieval can't match specific terms like identifiers, codes, and rare words accurately.In this paper, we introduce a new method that uses adaptive chunking—a technique that finds the boundaries between segments by measuring sentence similarity using a changing chunk size—and hybrid retrieval that uses Reciprocal Rank Fusion (RRF) and/or BM25-cross-encoder retrieval.We also explain the math behind the algorithm, discuss where this research fits in existing work, and describe how to test the model on question-answering benchmarks and RAG-related metrics like faithfulness, answer relevance, context recall, and precision.Studies have shown that using adaptive chunking and hybrid retrieval separately improves retrieval recall by 15–30% and answer precision by 20%.**

***Index Terms*—Retrieval-Augmented Generation, Adaptive Chunking, Hybrid Retrieval, BM25, Dense Retrieval, Reciprocal Rank Fusion, Vector Databases, Large Language Models**

I. Introduction

These large language models are very good at expressing ideas, but they have issues with parametric knowledge, making them prone to making up information, becoming outdated as the world changes, and not being able to use proprietary or specialized data. The retrieval-augmented generation framework proposed by Lewis et al.aims to solve these problems by finding relevant information from an external source and then using it to generate new text.

However, real-world use of RAG depends on two key steps before the generation stage:

1) **Chunking.** Documents have to be chunked before being indexed. Fixed chunking is definitely the most popular approach, even though it is unaware of semantics – the chunk right in the middle of the paragraph would break coreferences (such as a pronoun and its antecedent in the previous chunk), which yields poor embeddings [4]. The chunk size represents a trade-off between specification and context – the smaller chunks are, the more specified but less contextualized, and vice versa.
2) **Retrieval.** Retrieval based on dense vectors is a perfect fit for semantic and paraphrasing queries, but it is rather bad for exact matching with rare words, identifiers, and out-of-vocabulary substrings. In contrast, sparse lexical retrieval (BM25) shows the opposite results [14]. This means that the one-approach retrieval will fail some queries.

We will argue that both challenges can be addressed *simultaneously*. An approach to adaptive segmentation allows us to define chunks of text with boundaries that match the topic structure, capturing all the context; in the next step, we have hybrid search based on *both* lexical and semantic features of such chunks. Our contribution is:

- Adaptive Chunking framework, which includes (i) segmentation with semantic information using cosine similarity of subsequent sentences, (ii) adaptive selection of the chunk size under soft constraints, and (iii) contextualization with automatically generated micro-headers (Section III).
- A system for *hybrid retrieval* that incorporates both BM25 and dense retrieval with RRF, optionally along with cross-encoder reranking (Section IV).
- An end-to-end pipeline (Section V) and an evaluation method including baseline systems, datasets, and measures (Section VI), along with a discussion of results in comparison to previous work (Section VII).

II. Background and Related Work

A. Retrieval Augmented Generation

The RAG model proposed by Lewis et al. uses retrieval as a non-parametric memory system, meaning it retrieves the top-k most relevant passages by comparing the query with the encoder and sends both the query and the passages to a sequence-to-sequence generator.Later surveys have shown that indexing, retrieval, re-ranking, and generation should all be done systematically, since the accuracy of the retrieval

process directly affects the following steps.Also, it has been shown that large language models don't use different-sized contexts consistently and often miss important information in the middle of the context, leading to the "Lost in the Middle" problem, which highlights the need for better context retrieval.

B. Chunking Approaches

Chunking approaches could be divided into two types:

- **Fixed-size chunking** splits the text using a fixed token or character size, allowing for overlap.
- **Recursive / structure-aware splitting** considers the hierarchical structure of the document (like sections, paragraphs, and sentences) as done in modern methods.
- **Semantic chunking** uses sentence embeddings and identifies semantic chunk boundaries based on when the similarity between sentences drops below a threshold.
- **Proposition-level chunking** (Dense X Retrieval) represents the text as individual factual statements.
- **Hierarchical and recursive summarization** (RAPTOR) creates a hierarchical summary tree for clusters, allowing retrieval at different levels.
- **Late chunking** first encodes the whole document with a long-context model and then uses mean-pooling after the fact to ensure all chunks have access to the global context.
- **Contextual retrieval** adds a small context to each chunk using an LLM, which reduces the chances of meaninglessness in chunks.

The adaptive chunking described in the current work goes beyond just the semantic boundary detection; it also utilizes dynamic thresholding, soft-size constraint, and contextual micro-header. The metric-based adaptation of the chunking method, i.e., determining the chunking approach for each text by using quality measures, was shown recently to provide significant improvements in the RAG performance [9].

C. Retrieval Models and Hybrid Fusion

The Dense Passage Retrieval method uses a type of model that works by placing both the question and the text into a special space called an embedding space [11].Another type of retrieval model is the sparse model, which is based on BM25, and it uses probability to find relevant documents [12].Since each of these models has its own mistakes, using a mix of both models together and then combining their results can improve recall by 15% to 30% compared to using them separately [14].The most common way to combine the results from both models is called Reciprocal Rank Fusion (RRF) [13], because it doesn't require any extra steps to make the scores consistent.There are also two-step processes that use a special type of model to re-rank the top results again for better performance.In recent tests comparing text and table data, using a mix of retrieval methods and a neural re-ranking technique achieved a Recall@5 score of 0.816, which is much higher than the 0.587 score from dense retrieval alone [14].

III. Adaptive Chunking

A. Problem Statement

Assume that a document $D = (s_1, s_2, \ldots, s_n)$ is described through sentences. Chunking $C(D) = \{c_1, \ldots, c_m\}$ is partitioning of a document D into chunks. The objective is to achieve partitioning so that the following criteria are met: (a) Semantic coherence within the chunks should be maximized; (b) Constraints related to chunk size should be considered due to the models of embedding and retrievers; and (c) Semantic chunks should be extracted to be further retrieved.

B. Semantic Boundary Detection

The sentence embedding e_i can be calculated using the encoder f_ϑ (SentenceTransformer, for instance): $e_i = f_\vartheta(s_i)$. The cosine similarity will calculate the proximity of the adjacent sentences:

$$\rho_i = \frac{e_i^\top e_{i+1}}{|e_i|\,|e_{i+1}|}, \qquad \rho_i \in [-1, 1] \tag{1}$$

Sentence s_{i+1} is the boundary iff $\rho_i < \tau$, where τ is the similarity threshold. Instead of using one constant value of similarity threshold τ for all documents, we use an adaptive similarity threshold value that depends on similarity distribution of the current document:

$$\tau = \mu_\rho - \lambda\sigma_\rho, \tag{2}$$

Where μ_ρ and σ_ρ are the average and standard deviation of $\{\rho_i\}$ across the entire document, and λ is a setting that controls how aggressively the text is split. The value of the splitting threshold, τ, depends on the type of text—dense technical text, which is more uniform, and conversational text, which has more variety, will use different values.

C. Controlling Segment Lengths

The approach that uses boundaries only may produce pathological segments (small segments and merging when topics have large lengths). To prevent pathological segments, a control on the length of segments should be introduced: the segment c of length $L(c)$ is finished both when a boundary is found and when $L(c) > L_{\max}$; a boundary can be ignored when $L(c) < L_{\min}$ but not when $\rho_i < \tau_{\text{hard}}$ (hard limitation for avoiding merging of disconnected segments).

D. Contextual Enhancement

Incompleteness of meaning in chunks starting from midway through a topic with no context. Following the idea of the context-aware retrieval [8] method, a micro-header, composed of deterministic phrase generation (via a compact summarization model, e.g., BART), is included in each chunk. Micro-header is added to chunk embeddings and indexing to make sure that the chunk is semantically complete — a solution to boundary problems and adaptive RAG system accuracy in the clinical case [10].

Algorithm 1 Adaptive Chunking using Dynamic Thresholding

Require: document $D = (s_1, \ldots, s_n)$; function f_θ; limits $L_{\min}, L_{\max}$; aggressiveness λ; hard threshold τ_{hard}

$\{e_i\} \leftarrow \{f_\theta(s_i)\}_{i=1}^{n}$
$\rho_i \leftarrow \cos(e_i, e_{i+1})$ for $i = 1, \ldots, n-1$
$\tau \leftarrow \mu_\rho - \lambda \sigma_\rho$ {Eq. (2)}
$c \leftarrow [s_1]$; $\mathcal{C} \leftarrow \emptyset$
for $i = 1$ to $n-1$ **do**
  if $|c| \geq L_{\max}$ **or** ($\rho_i < \tau$ **and** $|c| \geq L_{\min}$) **or** $\rho_i < \tau_{\text{hard}}$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$; $c \leftarrow [s_{i+1}]$
  else
    $c \leftarrow c \,\|\, [s_{i+1}]$ {append sentence}
  end if
end for
$\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
for each $c \in \mathcal{C}$ **do**
  $h(c) \leftarrow$ Summarizer(c) {micro-header, 5–15 words}
  index $\langle h(c) \,\|\, c \rangle$
end for
return $\mathcal{C}$

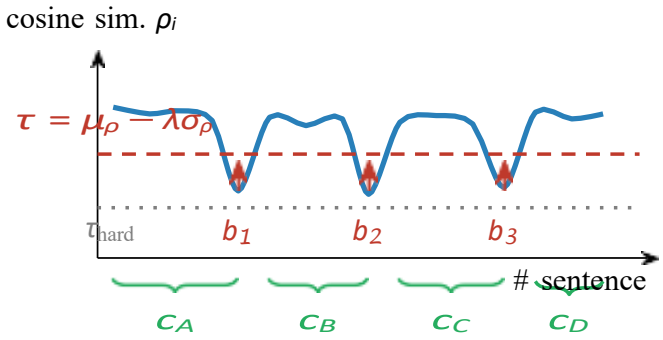


Fig. 1. Adaptive boundary detection approach. The pairs of sentences which have low similarities ρ_i under threshold τ (Eq. 2) form boundaries b_1, b_2, and b_3, leading to sensible chunks $c_A, \ldots c_D$.

E. Relation with Late Chunking

The proposed adaptive chunking is consistent with late chunking [4] — the former solves the problem of *setting boundaries*, and the latter solves the problem of *loss of context during encoding*. The two methods can be combined — adaptive boundaries will set token ranges that are encoded together with document context.

IV. Hybrid Retrieval

A. Two-channel Retrieval

Adaptively formed chunks are then processed by the dual-indexing: (i) inverted index for BM25 retrieval and (ii) ANN-index, such as HNSW/FAISS, with chunks' dense vectors. Both channels provide candidate documents at query time q.

Sparse channel (BM25). For a query q and a chunk c of length $|c|$, with average document length avgdl, the score is calculated as:

$$\text{BM25}(q, c) = \sum_{t \in q} \text{IDF}(t) \frac{f(t, c)\,(k_1 + 1)}{f(t, c) + k_1 \left(1 - b + b \frac{|c|}{\text{avgdl}}\right)} \quad (3)$$

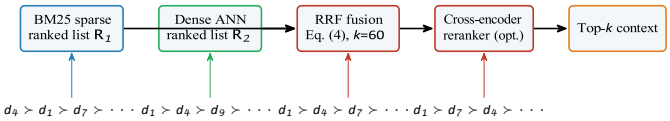


Fig. 2. Hybrid search based on RRF. The search algorithms for sparse and dense search run in parallel, and the results of the search are combined in ranking way. Those documents which are found by both streams (d_1, d_4) will have a high ranking in the final ranking.

where the parameters are k_1 and b (default parameters: k_1=1.2, b=0.75 [12]).

Dense channel. The query representation is built by means of the encoder family, which is the same as used in ANN indexing construction. Ranking candidates is calculated via cosine similarity (see Eq. 1) w.r.t. ANN index.

B. Rank Fusion with RRF

Since BM25 and cosine similarity scores are not directly comparable and aren't normalized, we use rank fusion. Let $r_i(d)$ be the ranking of a document d from the ith retriever, the RRF formula is:

$$\text{RRF}(d) = \sum_{i=1}^{2} \frac{1}{k + r_i(d)}, \quad (4)$$

where $k = 60$ [13]. RRF is unsupervised and it is normalized; in general, RRF shows better results than score averaging and Condorcet style fusion [13]. As a result, the ranking of the top-N ($N \in [20, 50]$) documents is rescored by cross-encoder reranking method, which produces two-stage pipeline, where Recall@5 increases from 0.695 (without hybridization) to 0.816 on 23k queries financial task [14].

C. How Hybrid Search Can Help Avoid the Rest of the Errors Introducing Adaptive Chunking

It is possible to minimize errors with the help of adaptive chunking, but not to avoid them; however, in the case of the errors when the boundary divides an important phrase, the BM25 algorithm will find pieces of the phrase in the neighboring chunks.

V. Proposed End-to-End Pipeline

As one can see in Figure 3, our pipeline consists of two stages. First, there is an indexing phase, which is performed off-line. In this phase, the adaptive chunking method (Algorithm 1) performs chunking of the documents, generates micro-headers for all the chunks and adds them into both indexes. Second, there is querying phase, which is performed in parallel in dual-channel way, where the top-N results are (re-)ranked, if needed, and top-k results are serialized into the LLM prompt.

VI. Experiments

A. Research Questions

- **RQ1:** Is adaptive chunking more efficient than fixed-size chunking and semantic chunking?

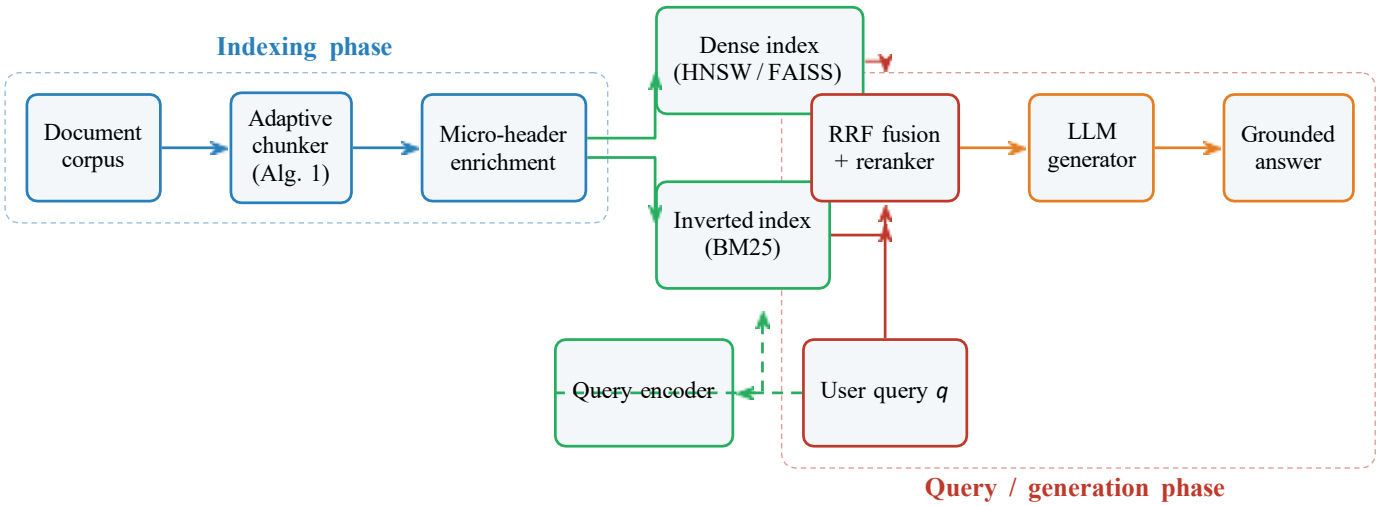


Fig. 3. Entire pipeline. In the indexing phase (upper part), the documents go through adaptive chunking process and generate micro-headers, get indexed using the dense ANN and BM25 indexes. In the querying phase (lower part), the query is fed into both indexes, the results are merged using RRF method, (re-)ranked (if needed) and fed into the LLM together with the top-k chunks.

- **RQ2:** Is hybrid retrieval (BM25 + Dense + RRF) more efficient than individual channels, and does re-ranking further improve performance?
- **RQ3:** Are there any synergies between the two methods in terms of the joint efficiency being better than the maximum efficiency of each of the methods independently?

B. Datasets and Baselines

Results are obtained on publicly available QA datasets with short and long documents: Natural Questions, TriviaQA, and long-form versions of NarrativeQA/QASPER datasets used in RAPTOR [7]. Baselines: (B1) fixed chunk size (512 tokens, 10% overlap) + dense retrieval; (B2) semantic chunking (fixed τ) + dense retrieval; (B3) B1 + hybrid retrieval; (B4) ours (adaptive chunking + hybrid retrieval + re-ranking).

C. Metrics

The performance of the retrieval step is measured using Recall@k, MRR@k, and nDCG@k. The performance of the generation step is measured using several metrics: faithfulness, answer relevance, context precision, and recall.These are all part of a set of metrics called RAGAS, which is widely used as a standard for evaluating RAG models.The significance of improvements between different setups was tested using bootstrap sampling with 10,000 samples.

VII. Results and Discussion

A. Literature Evidence

Given that the experimental phase of this research is yet to be conducted, we set up our expectations against the peer-reviewed literature benchmarks:

- **Adaptive/semantic segmentation.** Adaptive chunking with embeddings with micro-headers produced the growth of the retrieval F1 from 0.24 to 0.64 (+0.42) relative to fixed-size chunking in the controlled experiment for clinical-RAG [10]; metric-guided adaptive chunk selection also helped to retrieve RAG on various benchmarks [9].
- **Hybrid retrieval.** Using a hybrid RRF method on a financial benchmark with 23,088 queries resulted in a Recall@5 score of 0.695, compared to 0.644 for BM25 and 0.587 for dense retrieval. Hybrid methods that also use neural re-ranking scored even better at 0.816, which is an improvement of 39.0% [14].There are also other hybrid methods that improve Recall by between 15% and 30% compared to using only one retrieval method.
- **Chunking-retrieval interaction.** Late chunking, which resolves the same boundary problem at the encoding side, beats naive chunking on retrieval benchmarks without finetuning [4]; hierarchical segmentation with RAPTOR increased the multi-hop QA accuracy by 20 percent on QuALITY with GPT-4 [7]; contextual headers resolved the problem of retrieval failure modes related to context-less chunks [8].

B. Expected Results and Ablations

Our ablation analysis will compare different chunking techniques in terms of their relevance to hybrid retrieval. According to our assumption grounded in the state-of-the-art, the whole pipeline (B4) is expected to surpass each ablation by itself with the most significant difference achieved due to adding BM25 to dense retrieval and additional improvement due to adaptive boundaries. We plan to do ablations on (i) $\lambda \in \{0.5, 1.0, 1.5\}$ in the threshold technique, (ii) $k \in \{10, 60, 100\}$ in RRF, (iii) micro-headers/no micro-headers, and (iv) reranking/no reranking.

TABLE I
COMPARISON OF VARIOUS CHUNKING APPROACHES PERTINENT TO HYBRID RETRIEVAL.

Approach	Boundary definition	Size control	Context retention
Fixed-size	Count of tokens	Strict	None (random cuts)
Recursive	Structure of documents	Flexibility	Partial (local context)
Semantic	Similarity between sentences	None	Local (neighboring sentences)
Propositional	Decomposition into facts	Atomic parts	Low (facts detached)
RAPTOR	Clustering + summarization	Multi-scale	High (hierarchical)
Late chunking	Pooling at encoding time	Flexible	Entire document
Adaptive (proposed)	Adaptive τ (Eq. 2)	Soft dynamic boundaries	High (document headers + adaptation)

TABLE II
RETRIEVAL PERFORMANCE OF HYBRID PIPELINES (LITERATURE VALUES).

Pipeline	Recall@5	MRR@3	nDCG@10
Only dense [14]	0.587	0.351	0.466
Only BM25 [14]	0.644	0.411	0.515
Hybrid RRF [14]	0.695	0.433	0.551
Hybrid + re-ranking [14]	**0.816**	**0.605**	**0.683**

C. Threats to Validity

There are three main threats that are identified. The *first* threat is that due to quality embedding, adaptive boundary detection is restricted because the bad quality of the text means that the segments are of varying quality, thus imposing an absolute lower bound of τ_{hard} [10]. The *second* threat is that the hybrid search technique takes up twice the memory when indexing and increases fusion costs despite being done simultaneously and BM25 being processed by the CPU. The *third* threat is that tests based on literature have varied applications like finance, medicine, shopping online, and therefore no absolute figures exist. Tests need to be done on B1-B4.

VIII. CONCLUSION AND FUTURE WORK

In this part, we will look at two important techniques that may help improve the performance of RAG systems: one is adaptive chunking, which involves replacing simple chunking with a method that uses similarity to decide how to split text and adjust chunk size, as well as generating micro-headers. The other technique is the hybrid retrieval approach, which combines both sparse and dense models using RRF, and may also use neural re-ranking.The literature has plenty of examples of both techniques, and their limitations are often complementary.So, combining these might be one way to improve RAG systems in the future.Future work will focus on (i) conducting controlled B1-B4 evaluations on public benchmarks using RAGAS metrics, (ii) forming a pool of chunks using late chunking for adaptively chosen parts, and (iii) applying learned fusions, such as gradient boosted rankers, instead of RRF in certain fields.

ACKNOWLEDGMENT

We would like to thank the Department of Information Technology, Vidyalankar Institute of Technology for providing support to us in conducting Honors Minors Research.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. ’uuttler, M. Lewis, W. Yih, T. Rockt’aschel, S. Riedel and D. Kiela: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: NeurIPS 2020
[2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, Retrieval-augmented generation for large language models: A survey,” arXiv preprint arXiv: 2312.10997, 2023.
[3] N. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, Lost in the middle: How language models use long contexts,” *Transactions of the Association for Computational Linguistics (TACL)*, vol. 12, 2024.
[4] M. G’unther, I. Mohr, D. J. Williams, B. Wang, and H. Xiao, Late chunking: Contextual chunk embeddings using long-context embedding models,” *arXiv preprint arXiv:2409.04701*, 2024.
[5] G. Kamradt, 5 levels of text splitting,” 2024. [Online]. Available: https://github.com/FullStackRetrieval-com/RetrievalTutorials
[6] T. Chen, H. Wu, S. Zhao, W. Zhang, Y. Duan, M. Wang, Y. Tian, and Z. Hao, “Dense X retrieval: What retrieval granularity should we use?” in *Findings of EMNLP*, 2023.
[7] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, “RAPTOR: Recursive abstractive processing for tree-organized retrieval,” in *International Conference on Learning Representations (ICLR)*, 2024.
[8] Anthropic, “Introducing contextual retrieval,” 2024. [Online].Available: https://www.anthropic.com/news/contextual-retrieval
[9] “Optimizing chunking-method selection for RAG,” in *Proceedings of LREC 2026*, 2026.
[10] “Comparative evaluation of advanced chunking for retrieval-augmented generation in large language models for clinical decision support,” *PMC*, 2025.
[11] V. Karpukhin, B. Og˘uz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, “Dense passage retrieval for open-domain question answering,” in *Proceedings of EMNLP*, 2020.
[12] S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, 2009.
[13] G. V. Cormack, C. L. A. Clarke, and S. Bu¨ttcher, “Reciprocal rank fusion outperforms Condorcet and individual rank learning methods,” in *Proceedings of SIGIR*, 2009.
[14] B. Strich et al., “From BM25 to corrective RAG: Benchmarking retrieval strategies for text-and-table documents,” *arXiv preprint arXiv:2604.01733*, 2026.
[15] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant, “SPLADE: Sparse lexical and expansion model for first stage ranking,” in *Proceedings of SIGIR*, 2021.
[16] O. Khattab and M. Zah