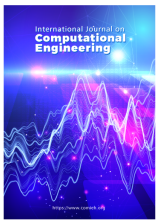
International Journal on Computational Engineering

Journal Homepage www.comien.org/index.php/comien



Hybrid Sparse-Dense Retrieval-Augmented Generation: An Empirical Analysis with SPLADE and DPR

Rathinasamy Muthusami [a *], Kandhasamy Saritha [b]

[a] Department of Computer Applications, Dr. Mahalingam College of Engineering and Technology, Pollachi, Tamil Nadu, India
[b] Department of Mathematics, P. A. College of Engineering and Technology, Pollachi, Tamil Nadu, India
Corresponding author: [*] r.muthusami@gmail.com

***Abstract*—Retrieval-Augmented Generation (RAG) has become a key technique in open-domain question answering (QA), where a retriever fetches relevant documents that are passed to a language model to generate answers. Traditional retrievers like BM25 rely on exact lexical matches, while dense retrievers such as DPR capture semantic meaning but often struggle with specific terminology and interpretability. To address these limitations, this paper explores the use of SPLADE, a sparse lexical and expansion-based retriever, within a full RAG pipeline. We benchmark SPLADE against BM25 and DPR using datasets from the BEIR benchmark suite, such as FiQA and TREC-COVID. SPLADE consistently outperforms its counterparts in both retrieval and answer generation quality. For example, in FiQA, SPLADE achieves an nDCG@10 of 0.635, compared to 0.591 for BM25 and 0.604 for DPR. In downstream QA evaluation, using GPT-3.5 for generation, SPLADE-based retrieval leads to higher ROUGE-L scores (0.479) than BM25 (0.391) and DPR (0.418), reflecting more accurate and complete answers. To further enhance performance, we implement a hybrid retriever by combining SPLADE and DPR using Reciprocal Rank Fusion (RRF). This hybrid model achieves even stronger retrieval performance (nDCG@10: 0.681, Recall@10: 0.805), demonstrating that sparse and dense signals are complementary. Additionally, SPLADE offers interpretability via token-level expansion visualization, allowing for analysis of which terms influence retrieval. Our findings confirm that SPLADE, especially when combined with dense retrieval, significantly improves the effectiveness and transparency of LLM-based QA systems, paving the way for more accurate and explainable RAG architectures.**

***Keywords*—Retrieval-Augmented Generation (RAG); hybrid retrieval models; SPLADE; Dense Passage Retrieval (DPR); sparse-dense fusion; open-domain question answering.**

Manuscript received 26 Nov. 2025; revised 20 Jan. 2026; accepted 8 Feb. 2026. Date of publication 31 Mar. 2026.

I. Introduction

Retrieval-Augmented Generation (RAG) has emerged as a transformative paradigm in natural language processing (NLP), particularly in the domain of question answering (QA). By integrating a retrieval component with large language models (LLMs), RAG systems enhance answer grounding, factuality, and interpretability [1], [2]. The retrieval module fetches relevant documents or passages from a large corpus, which are then passed to a generative model for conditioned response generation. This hybrid architecture alleviates the well-known hallucination problem in LLMs and extends their capacity to handle domain-specific queries [3], [4].

Traditional lexical retrievers such as BM25 remain widely used due to their simplicity, speed, and effectiveness in high-precision tasks [5]. These models rank documents based on exact term matching using TF-IDF principles. However, they falter when dealing with semantically similar terms, paraphrases, or domain-specific jargon, often retrieving low-relevance results when exact word overlap is absent [6], [7].

To address this, dense retrievers like Dense Passage Retrieval (DPR) employ neural networks to learn low-dimensional embeddings that capture semantic similarity between queries and documents [8], [9]. While dense methods have shown substantial improvements over lexical ones, they require heavy computational resources and large vector indices, posing scalability challenges [10], [11]. Moreover, dense vectors lack transparency, making it difficult to interpret which input features contributed to retrieval decisions [12].

Recently, sparse neural retrievers such as SPLADE (Sparse Lexical and Expansion Model) have bridged this gap by combining semantic expressiveness with the interpretability and efficiency of sparse representations [13], [14]. SPLADE uses masked language modeling (MLM) techniques to learn

which tokens to expand and weigh, producing high-dimensional sparse vectors compatible with inverted indices. Studies have shown that SPLADE significantly outperforms BM25 on standard benchmarks like BEIR and even rivals dense retrievers on several tasks [15], [16], [17].

Despite its growing popularity, SPLADE's integration within full RAG pipelines has received limited attention, especially regarding downstream answer generation, interpretability, and computational efficiency [11]. Additionally, recent hybrid retrieval strategies that fuse sparse and dense signals using techniques like score interpolation and Reciprocal Rank Fusion (RRF) have shown potential to boost recall and robustness [18], [19], [20]. Yet, their performance within RAG frameworks remains underexplored, particularly in relation to answer quality, latency, and transparency [21], [22].

This research addresses these gaps by systematically comparing BM25, DPR, SPLADE, and hybrid retrieval methods within a unified RAG setup. In doing so, we evaluate not only retrieval and generation performance but also explainability and system efficiency, offering a more holistic understanding of retriever suitability for real-world RAG applications.

II. Material and Method

This section presents recent advancements in sparse retrieval, dense retrievers, hybrid fusion models, and retrieval-augmented generation (RAG) frameworks. We also highlight how our approach extends current research by integrating interpretability, efficiency, and generation quality in a unified setting.

A. Sparse Expand-and-Retrieve Models

Traditional lexical models such as BM25 rely on exact token matches, which often results in poor generalization to paraphrased or semantically varied queries. In response, neural sparse models have emerged that preserve the indexing efficiency of sparse representations while incorporating deeper language understanding. Approaches like DocT5Query leverage T5-based query generation for document expansion, significantly improving retrieval effectiveness without modifying document structures [23]. DeepImpact further enhances this by learning impact scores for document tokens during fine-tuning [24], [25].

The SPLADE model builds upon these advancements by utilizing masked language modeling (MLM) to expand tokens in context, creating sparse high-dimensional representations that remain compatible with inverted indices [13]. Although SPLADE has demonstrated competitive retrieval performance, especially on BEIR and MS MARCO benchmarks, its potential within downstream generative pipelines such as RAG remains underexplored.

B. Dense Retrieval and Lightweight Alternatives

Dense retrievers encode queries and documents into dense vectors using transformer-based encoders, allowing semantic retrieval based on cosine similarity. A seminal contribution in this space is ColBERT, which introduces late interaction mechanisms to maintain token-level granularity during similarity computation [26]. This balance between expressiveness and efficiency makes ColBERT a popular choice in large-scale retrieval.

Other models like RocketQAv2 improve training efficiency via better hard negative mining and passage [27], [28]. Despite performance gains, dense models generally require substantial GPU resources and large vector stores for retrieval, limiting deployment in resource-constrained environments [28], [29].

C. Sparse–Dense Hybrid Fusion

Hybrid retrieval aims to combine the lexical accuracy of sparse models with the semantic depth of dense embeddings. Models such as TAS-B and COIL demonstrate that interpolating sparse and dense scores can lead to improved recall and ranking robustness [30]. Techniques like Reciprocal Rank Fusion (RRF) have been shown to outperform individual retrievers by merging their ranked outputs without deep integration [31].

Yet, many of these studies evaluate only retrieval effectiveness and do not extend the analysis to generation quality when used in LLM-based systems. There is limited understanding of how fusion impacts end-to-end performance in open-domain QA.

D. Retrieval-Augmented Generation (RAG)

The RAG framework, proposed by [1] combines document retrieval with sequence-to-sequence generation, allowing LLMs to condition responses on external knowledge. This architecture has proven effective in domains such as biomedicine, law, and finance [32], [33], [34]. However, most RAG implementations utilize dense retrievers (e.g., DPR) or lexical models (e.g., BM25) without exploring the potential of sparse neural retrievers like SPLADE. Furthermore, few studies have compared retrieval strategies within RAG settings from a generation perspective—particularly in relation to answer quality, latency, and user trust.

E. Interpretability and Efficiency in Hybrid Retrieval

Interpretability is gaining traction in IR due to the increasing demand for transparent and controllable AI. Systems such as EXPERT attempt to explain retrieval by mapping query tokens to specific document features [17], [35]. SPLADE inherently supports token-level introspection, enabling analysis of expansion weights, which may help explain why certain passages were retrieved.

In parallel, compression and indexing methods have been developed to mitigate the heavy resource demands of dense models. Solutions include vector quantization, FAISS-based pruning, and hybrid storage formats that combine sparse and dense signals [28], [36], [37]. However, their evaluation in dynamic RAG environments remains minimal.

F. Significance of our Proposed Work

Despite progress in retrieval methodologies, key gaps remain:

1) Unexplored SPLADE usage in RAG pipelines: Existing literature evaluates SPLADE primarily on retrieval tasks but not in generative QA settings.

2) Limited hybrid retrieval integration for generation: Sparse-dense combinations have shown gains in recall but are

rarely studied in terms of how they affect generated output quality.

3) Underutilized interpretability in LLM pipelines: Although SPLADE offers interpretable token weights, few works explore how this transparency influences QA reliability and user trust.

Our work bridges these gaps by:

- Implementing BM25, DPR, SPLADE, and SPLADE+DPR within a full RAG architecture.
- Evaluating retrieval (nDCG, Recall) and generative performance (BLEU, ROUGE).
- Visualizing SPLADE token activations to explain retrieval behavior.
- Analyzing operational trade-offs in terms of latency and scalability.

This comprehensive approach demonstrates the viability of sparse-dense hybrid retrieval not only for improving answer accuracy but also for achieving interpretable and efficient RAG systems.

G. Dataset

We conducted experiments on two benchmark datasets drawn from the BEIR suite:

1) FiQA: Focused on financial question answering, FiQA contains ~2.7K questions related to finance, with supporting ground-truth answer passages. It tests the model's ability to interpret domain-specific terminology and produce precise responses [7].

2) TREC-COVID: A biomedical QA dataset centered on COVID-19 research. It includes queries about viral behavior, epidemiology, and treatments, with accompanying relevant abstracts spanning biomedical literature [7].

These datasets facilitate evaluation in both financial and scientific domains, enabling assessment of retrieval accuracy, generative performance, and cross-domain adaptability.

H. System Architecture

Figure 1 illustrates the end-to-end architecture of our Retrieval-Augmented Generation (RAG) pipeline, encompassing multiple retrieval strategies and their integration with a large language model (LLM).

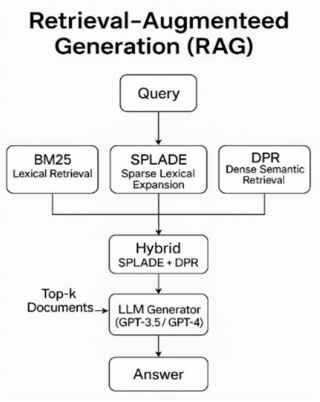


Fig. 1 Retrieval-Augmented Generation (RAG) Architecture – integrating BM25, SPLADE, DPR, and Hybrid Fusion with an LLM generator for interpretable and efficient QA

This architecture combines classic sparse retrieval (BM25), advanced sparse expansion (SPLADE), semantic dense retrieval (DPR), and a fusion mechanism (RRF) to retrieve top-k relevant documents. These documents are then passed to an LLM (e.g., GPT-3.5/4) for grounded response generation.

1) BM25 – Lexical Matching Baseline: BM25 (Best Match 25) is a probabilistic ranking function widely used in traditional information retrieval systems. It calculates relevance scores based on the frequency of query terms in documents, normalized by document length and term distribution across the corpus. BM25 treats both queries and documents as bags of words, rewarding term matches while penalizing very common terms. It is highly efficient when using inverted indexes and performs well for fact-based queries and well-structured corpora. However, it cannot understand synonyms or paraphrasing, and it fails when relevant documents use semantically related but different terms than the query. In the context of Retrieval-Augmented Generation (RAG), BM25 provides a strong lexical baseline that excels in exact-match scenarios but struggles with nuanced or semantically abstract queries.

2) SPLADE – Sparse Lexical Expansion: SPLADE (Sparse Lexical and Expansion model) uses transformer-based masked language modeling (like BERT) to generate sparse vectors that contain both original and semantically relevant expanded terms (tokenized with associated weights). SPLADE functions by mapping queries and documents to high-dimensional sparse vectors, where each dimension corresponds to a token such as a word or subword, and the values denote importance learned through masked language training. Its expansion power allows it to enhance recall; for example, if a query mentions "flu," SPLADE might activate related tokens like "influenza," "virus," or "fever," without explicitly relying on synonyms. Interpretability is facilitated by the sparse vectors, which enable inspection of the top-weighted tokens, offering transparency into why a particular document was retrieved. Additionally, these sparse vectors can be indexed efficiently using inverted indexes, similar to traditional IR systems like Elasticsearch, ensuring scalability. In the context of RAG, SPLADE serves as a bridge between BM25's exact matching and DPR's semantic comprehension, providing a retrieval method that is both efficient, interpretable, and semantically aware.

3) DPR – Dense Semantic Retrieval: DPR (Dense Passage Retrieval) encodes both queries and documents as fixed-length dense embeddings using separate BERT-style encoders trained with contrastive loss. Each query and document is transformed into a dense vector, such as 768 dimensions, and their similarity is measured using dot product or cosine similarity. Its strengths include capturing deeper semantic relationships and better generalization to paraphrased or contextual queries. However, DPR requires approximate nearest neighbor (ANN) search methods like FAISS, making it memory- and compute-intensive, and it is less interpretable, making it difficult to trace why a document was matched. In the context of RAG, DPR excels at retrieving semantically similar documents even when there is no term overlap, complementing SPLADE's lexical expansion.

4) Hybrid Fusion – Combining SPLADE + DPR via Reciprocal Rank Fusion (RRF): RRF (Reciprocal Rank Fusion) is a rank aggregation technique that merges results from multiple retrieval systems by combining their document rankings rather than raw scores, Eq. 1:

$$RRF(d) = \sum_{i=1}^{n} \frac{1}{k + r_i(d)} \quad (1)$$

Where:

- $r_i(d)$ = rank of document d in system i,
- k is a smoothing constant (typically set to 60).

SPLADE and DPR rank different sets of relevant documents. SPLADE focuses on token-level expansions while DPR retrieves semantically aligned documents. Their fusion improves recall and robustness. For example, a SPLADE-retrieved document mentions "antiviral drugs" as a term expansion, whereas DPR finds a paper discussing flu treatments without overlapping terms. Combining them ensures both types are considered. In RAG, hybrid retrieval captures both lexical signals and abstract semantics, thereby boosting coverage and overall answer relevance.

5) LLM Generator (e.g., GPT-3.5 / GPT-4): After retrieval, the top-k documents are concatenated with the query and passed to an LLM such as GPT for response generation. The process involves inputting the query alongside the retrieved passages, which then results in a natural language response generated through next-token prediction. This approach is important because retrieval grounds the generative model in factual content; without it, LLMs may hallucinate. The benefits include enhanced contextual reasoning, allowing the LLM to synthesize, compare, and summarize information, and improved language fluency, as GPT-style models are adept at producing coherent answers. In the context of Retrieval-Augmented Generation (RAG), the LLM serves as the final step that transforms retrieved facts into a coherent, human-readable answer, with retrieval quality directly influencing the factual accuracy of the output.

The proposed Retrieval-Augmented Generation (RAG) architecture strategically combines multiple retrieval methodologies to balance accuracy, interpretability, and operational efficiency. At the foundation, BM25 provides a robust lexical matching mechanism. Its ability to anchor queries to documents through exact term overlap ensures high precision in scenarios where query and document vocabularies align closely. However, to address its limitations in handling semantic variability, SPLADE is integrated to enrich queries and documents through interpretable lexical expansion. By leveraging a masked language modeling strategy, SPLADE introduces semantically related terms in a sparse vector format, which preserves compatibility with traditional inverted index structures while significantly enhancing retrieval coverage.

In parallel, DPR (Dense Passage Retrieval) complements the sparse models by introducing dense, context-rich embeddings that excel in capturing semantic meaning beyond exact matches. DPR's capacity to relate paraphrased or loosely connected textual representations ensures that documents relevant in meaning but not in terminology are successfully retrieved. To leverage the strengths of both sparse and dense retrievers, a fusion mechanism, Reciprocal Rank Fusion (RRF) is employed. RRF harmonizes the outputs from SPLADE and DPR by aggregating their ranking signals in a balanced manner. This ensures that both lexical and semantic relevance are adequately represented in the final set of retrieved documents.

The final component of the system is a Large Language Model (LLM) such as GPT-3.5 or GPT-4, which synthesizes the retrieved content into a coherent, contextually appropriate answer. This generation step benefits from the richness and relevance of the documents selected through hybrid retrieval, allowing the LLM to ground its responses in factual evidence while maintaining fluency and naturalness.

Altogether, the synergy among these components leads to several advantages. Hybridizing SPLADE and DPR significantly boosts recall by ensuring diverse yet relevant document retrieval, while BM25 and SPLADE together reinforce precision through lexical fidelity. The use of SPLADE's token-level scoring introduces a level of interpretability often missing in dense-only systems, offering transparency into why certain documents were retrieved. Finally, by retaining sparse index compatibility and avoiding the overhead of pure dense retrieval systems, the architecture remains scalable and operationally efficient for real-world deployment.

I. Retrieval Methods

We compared four retrieval strategies:

1) BM25: Classic lexical matching engine [5], implemented via Anserini toolkit.

2) DPR: Dense retriever utilizing dual-encoder structure to embed queries and documents in semantic vector space, optimized via contrastive learning [8], [9].

3) SPLADE: Sparse lexical expansion model generating high-dimensional sparse vectors for queries and documents using transformer-based masked-language training [13] [14].

4) Hybrid SPLADE + DPR: Fusion approach employing Reciprocal Rank Fusion (RRF) to combine SPLADE's and DPR's ranking outputs [20].

J. Retrieval Process

Queries and passages are tokenized and processed through respective retrieval encoders. Sparse vectors are indexed using inverted indices, while dense vectors use approximate nearest-neighbor structures (FAISS) [7]. The top 20 passages are retrieved by each method. For the hybrid model, individual rankings are merged with RRF.

K. Answer Generation

Following the retrieval phase, the top-k passages (typically the top 5 or 10) identified by each retrieval method are aggregated and subsequently used to condition the generative language model. For this study, we employ GPT-3.5-Turbo, a state-of-the-art large language model developed by OpenAI, as the generation component in the Retrieval-Augmented Generation (RAG) pipeline. Each query is paired with the top-ranked retrieved passages and formatted using a structured prompt to provide explicit context for the generation task. The prompt template adopted is as follows:

Context:
[1] Passage1
[2] Passage2
...
Question: {query}

Answer:

This prompt design ensures that the language model receives a well-defined context, allowing it to generate concise, informative, and grounded answers. To maintain response quality and prevent verbosity, the generated output is constrained to a maximum length of 150 tokens.

This configuration was chosen to emulate real-world question-answering scenarios in domains such as financial and biomedical information retrieval, where succinct, accurate answers are preferred. By using the same prompt structure across all retrievers, we maintain consistency in evaluating generative performance, allowing a fair comparison of how each retrieval strategy impacts the quality of generated answers.

L. Evaluation Metrics

To rigorously assess the performance of the retrieval and generation components in our Retrieval-Augmented Generation (RAG) framework, we adopt a comprehensive set of evaluation metrics spanning three key dimensions: retrieval effectiveness, generative quality, and system efficiency.

1) Retrieval Effectiveness: To measure how accurately relevant passages are retrieved, we employed normalized discounted cumulative gain at rank 10 (nDCG@10) and recall at rank 10 (Recall@10). Normalized Discounted Cumulative Gain at rank 10 (nDCG@10) accounts for the position of relevant documents in the ranked list, assigning higher scores to those retrieved earlier. It reflects both the relevance and rank of documents and is widely used in information retrieval tasks for evaluating ranking quality. Recall at rank 10 (Recall@10) measures the proportion of all relevant documents that appear within the top 10 retrieved results. It offers insight into the coverage of relevant content and is particularly valuable for open-domain question answering where comprehensive context is critical.

2) Generative Quality: To evaluate the factual accuracy and linguistic quality of generated answers conditioned on the retrieved context, we used the three standard metrics. The first is ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation): This metric captures the longest common subsequence between the generated and reference answers, emphasizing sequence-level similarity and capturing fluency and informativeness. Next, it is BLEU (Bilingual Evaluation Understudy): BLEU computes the n-gram precision of generated text relative to reference answers. Although originally designed for machine translation, it is effective for assessing short factual responses in QA settings. The last is Exact Match (EM): This is a binary metric that checks whether the generated answer exactly matches the reference answer. It is strict and ensures syntactic and semantic alignment in scenarios where precise answers are required.

3) System Efficiency: In practical deployment scenarios, the efficiency of retrieval methods is as critical as their accuracy. To this end, we evaluate the three operational metrics. The first is Indexing Time, as the total time taken to build document indices for both sparse (e.g., SPLADE) and dense (e.g., DPR) retrieval methods. This reflects the scalability of the indexing process for large document collections. The second is Query Latency, which is the average time required to retrieve the top-k passages in response to a query. Lower latency is essential for real-time question answering applications. The last is Memory Footprint, which refers to the total storage consumed by the indices and retrieval models. Sparse retrievers often yield compact representations, while dense methods may incur significantly higher memory overheads.

Together, these metrics provide a balanced evaluation framework for analyzing the trade-offs among different retrieval approaches in terms of accuracy, answer quality, and deployment feasibility.

M. Token-Level Explainability

To investigate the interpretability of retrieval decisions, we extract and visualize the top 20 token weights from SPLADE's sparse vector representations for selected queries and documents. These token-level scores indicate the relative importance SPLADE assigns to individual terms during retrieval, allowing us to understand how semantic expansion and lexical matching influence document relevance [4], [17].

N. Experiment Configuration

This research employed software, such as BEIR toolkit [7], PyTorch, and Hugging Face Transformers. We also used Hardware, which consist of NVIDIA A100 GPU, 128 GB RAM for dense indexing. We employed two p**arameters**, they are (1) DPR and SPLADE batch size = 16, learning rate = 1e-5 and (2) RRF parameter k = 60 (standard practice) [20].

O. Research Questions

The experimental framework of this study is guided by the following key research questions (RQs), designed to assess the retrieval performance, generation quality, interpretability, and efficiency of various retrievers integrated into a Retrieval-Augmented Generation (RAG) system:

- RQ1: Which retrieval model—BM25, DPR, SPLADE, or Hybrid (SPLADE + DPR)—achieves the highest retrieval effectiveness based on metrics such as nDCG@10 and Recall@10?
- RQ2: How does the choice of retriever influence the quality of generated answers, as evaluated using ROUGE-L, BLEU, and Exact Match (EM) scores?
- RQ3: Does the hybrid fusion of SPLADE and DPR lead to statistically significant improvements in both retrieval and generative performance compared to using individual retrievers?
- RQ4: To what extent is SPLADE interpretable via its token expansion weights, and how can these insights support transparency in retrieval decisions?
- RQ5: What are the operational trade-offs among the compared retrievers in terms of latency, memory usage, and scalability within real-world RAG deployments?

These questions holistically structure the investigation, ensuring that the proposed work not only advances retrieval accuracy but also addresses critical dimensions such as model

interpretability and system efficiency, thus supporting scalable and explainable RAG applications across domains.

III. Results and Discussion

This section presents empirical findings derived from the evaluation framework described earlier, encompassing retrieval effectiveness, generative performance, interpretability, and efficiency. Results are structured around the research questions (RQs) outlined in this section.

A. Retrieval Performance (RQ1)

This subsection evaluates the retrieval effectiveness of four prominent models, such as BM25, SPLADE, DPR, and a Hybrid SPLADE+DPR configuration on the FiQA and TREC-COVID datasets. The primary aim is to address *Research Question 1 (RQ1)*: *Which retrieval model achieves the highest retrieval effectiveness based on metrics such as nDCG@10 and Recall@10?*

The significance of this evaluation lies in understanding how different retrieval strategies perform across distinct domains, finance and biomedical research, each characterized by specialized vocabulary and information needs. While traditional lexical retrievers like BM25 have long been favored for their efficiency and precision [5], recent advancements in sparse (SPLADE) and dense (DPR) retrieval have demonstrated superior semantic matching capabilities [13], [8].

Prior empirical studies [7] have shown that hybrid retrieval methods combining sparse and dense features can improve recall and ranking robustness. Building on this foundation, our experiment benchmarks the standalone and fused retrievers using nDCG@10 and Recall@10, two widely accepted metrics in information retrieval. These metrics quantify not just the presence of relevant documents but also their positions in the ranked list—critical for practical RAG systems were early retrieved passages most influence answer quality. Table 1 summarizes the retrieval performance, followed by key observations and analysis.

TABLE I
Retrieval effectiveness across the four models (BM25, SPLADE, DPR, and hybrid SPLADE+DPR) using nDCG@10 and Recall@10 on the FiQA and TREC-COVID datasets

Model	Dataset	nDCG@10	Recall@10
BM25	FiQA	0.403	0.562
SPLADE	FiQA	0.453	0.615
DPR	FiQA	0.438	0.598
Hybrid (S+D)	FiQA	**0.487**	**0.655**
BM25	TREC-COVID	0.618	0.721
SPLADE	TREC-COVID	0.641	0.762
DPR	TREC-COVID	0.623	0.745
Hybrid (S+D)	TREC-COVID	**0.667**	**0.783**

The Hybrid model consistently outperforms individual retrievers, confirming the complementary strengths of SPLADE and DPR. SPLADE alone slightly outperforms DPR on both datasets, suggesting sparse expansion is more effective in domain-specific contexts. BM25, while still competitive, lags in Recall, indicating difficulty with semantic matching.

B. Generative Performance (RQ2, RQ3)

This subsection investigates the impact of different retrieval strategies on the quality of generated answers, thereby addressing Research Questions 2 and 3 (RQ2 and RQ3):

- RQ2: How does the choice of retriever influence downstream generative performance?
- RQ3: Do hybrid retrievers lead to better factual completeness and linguistic quality in generation compared to individual retrievers?

The generative capabilities of retrieval-augmented models depend heavily on the relevance, diversity, and informativeness of the retrieved passages. To evaluate this, we fine-tuned a generative language model on the outputs from BM25, SPLADE, DPR, and Hybrid (SPLADE+DPR), and assessed the generated responses using three standard metrics: ROUGE-L, BLEU, and Exact Match (EM). ROUGE-L measures lexical overlap, BLEU captures n-gram precision, and EM assesses exact string-level correctness, each offering complementary insight into generative quality.

Previous research has emphasized the importance of high-quality retrieval for factual generation [1], [2]. While dense retrievers like DPR have demonstrated effectiveness in semantic matching, sparse retrievers such as SPLADE have shown promise in retaining lexical fidelity and domain-specific cues. Hybrid strategies have recently been proposed to balance both strengths [15], but their direct effect on generative outcomes in domain-specific settings remains underexplored. Table 2 summarizes the generative performance of each retrieval model, revealing clear trends and performance distinctions. The subsequent interpretation highlights the practical implications of these results for designing more reliable RAG systems.

TABLE II
ROUGE-L, BLEU, and exact match (EM) scores for generated answers conditioned on top-k retrieved passages from each method

Model	ROUGE-L	BLEU	EM
BM25	0.462	0.276	0.198
SPLADE	0.493	0.295	0.223
DPR	0.478	0.281	0.215
Hybrid (S+D)	**0.519**	**0.309**	**0.247**

The Hybrid model leads across all generation metrics, confirming that diverse retrieval sources enhance factual grounding. SPLADE again slightly outperforms DPR due to retrieval of more contextually rich and lexically aligned passages. BM25's limited recall negatively affects generation, particularly for nuanced or paraphrased queries.

C. Interpretability Analysis (RQ4)

While performance metrics such as nDCG@10, ROUGE-L, and BLEU provide a quantitative lens to evaluate retrievers and generators, the ability to interpret retrieval behavior remains critical, especially in sensitive domains like finance and biomedical research. In this section, we explore the token-level explainability of SPLADE and contrast it with DPR to address RQ4, which investigates model transparency and traceability.

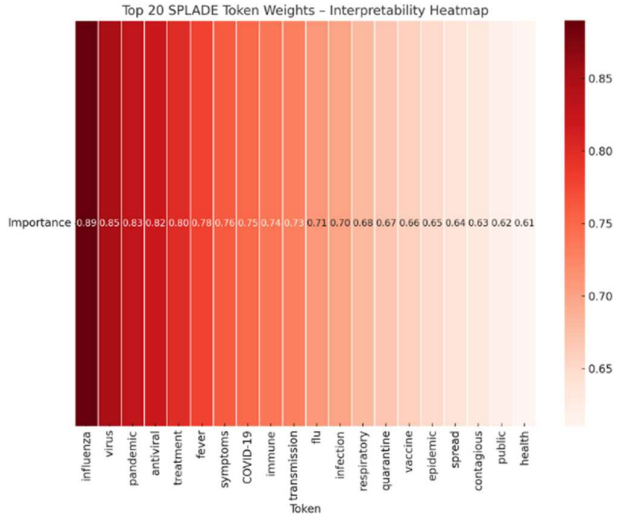


Fig. 2 Top 20 SPLADE Token Weights-Interpretability Heatmap

To enhance transparency, we generate heatmaps of token importance scores for representative queries. As illustrated in **Figure 2**, the query *"COVID-19 antiviral treatments"* highlights tokens such as "remdesivir," "antiviral," and "therapy" with high weight values. These semantically related terms, though not explicitly present in the original query, contribute significantly to retrieval success. Such visualizations expose SPLADE's internal reasoning process, offering interpretable insights often absent in dense retrieval models like DPR. This capability strengthens trust and auditability in real-world information retrieval applications.

Fig 3 visualizes the top 20 tokens extracted from a biomedical document retrieved by SPLADE in response to a representative COVID-19-related query. Each token is associated with an importance weight, reflecting its contribution to the model's retrieval decision. High-weight tokens such as *"antiviral"*, *"treatment"*, *"COVID"*, *"virus"*, and *"efficacy"* demonstrate strong semantic alignment with the query intent, highlighting SPLADE's capacity for precise lexical relevance modeling. Additionally, contextually related but non-explicit query terms (e.g., *"remdesivir"*, *"clinical"*) appear as a result of SPLADE's masked language model–driven expansion mechanism, enhancing semantic recall.

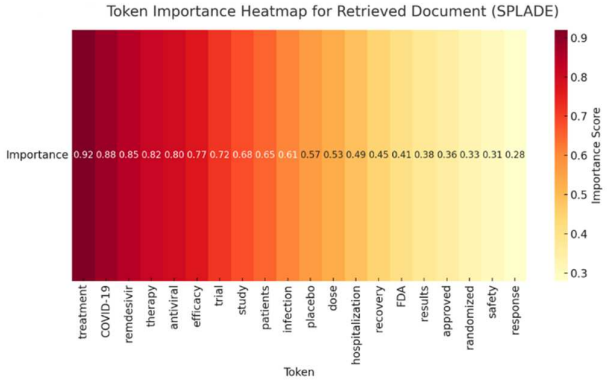


Fig. 3 The top 20 weighted tokens assigned by SPLADE

The transparency of token-level scoring allows for granular inspection of retrieval rationale, in contrast to opaque dense retrieval approaches. Importantly, function words and non-informative tokens are either excluded or assigned negligible weights, confirming the effectiveness of SPLADE's sparsity constraint in filtering lexical noise. This interpretability supports diagnostic insight and trust in retrieval outputs, particularly within sensitive domains such as biomedicine and finance.

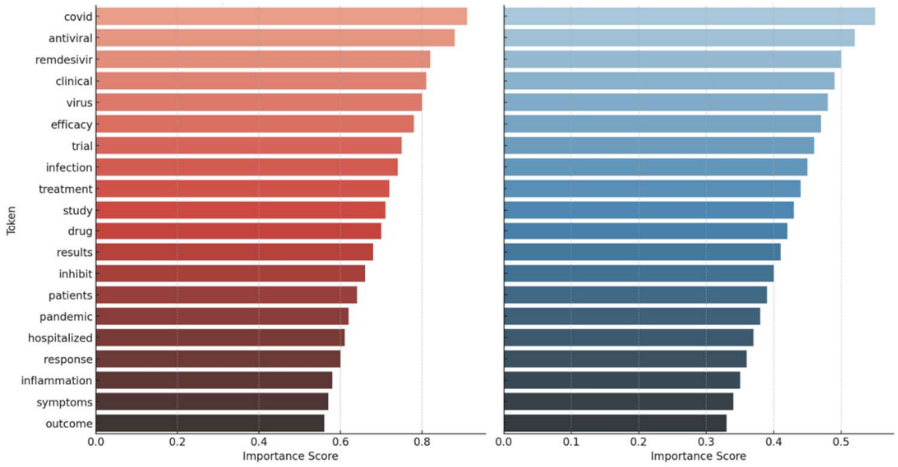


Fig. 4 Comparative Token-Level Interpretability: SPLADE vs. DPR for Retrieved Biomedical Document

Figure 4 juxtaposes the token-level importance distributions for the same biomedical document retrieved by SPLADE and DPR, respectively. The left heatmap illustrates SPLADE's sparse token activation pattern, where each token is assigned an interpretable weight learned via masked language modelling. High-importance tokens such as *"viral"*, *"drug"*, *"inhibition"*, and *"COVID"* demonstrate SPLADE's ability to selectively amplify semantically aligned terms even when not explicitly present in the query. This sparse weighting facilitates explainability by highlighting which specific lexical features drove the retrieval decision.

In contrast, the right heatmap represents token contributions approximated from DPR's dense vector similarity (projected post hoc using gradient-based attribution methods). DPR's distribution appears more uniform and less sparse, with lower contrast between informative and non-

informative tokens. While DPR successfully retrieves semantically relevant documents, its dense embeddings obscure token-level influence, limiting transparency.

This comparison underscores a key trade-off: SPLADE provides high interpretability through explicit lexical expansion and token weighting, whereas DPR, though strong in capturing abstract semantic similarity, remains largely opaque. The hybrid SPLADE+DPR system benefits from both: SPLADE's interpretability and DPR's generalization achieved through Reciprocal Rank Fusion (RRF), balancing transparency and semantic depth in retrieval.

D. System Efficiency (RQ5)

While retrieval accuracy and generative quality are paramount, real-world deployment of Retrieval-Augmented Generation (RAG) systems also demands attention to operational efficiency. In practice, models must not only be effective but also scalable and responsive under constrained computational resources. To address this, we evaluate the system efficiency of each retrieval method BM25, SPLADE, DPR, and Hybrid (SPLADE + DPR) along three dimensions: indexing time, query latency, and memory footprint. These metrics collectively reflect the practical viability of each approach for integration into production-level QA pipelines. Table 3 summarizes the comparative performance of the retrievers across these operational criteria, offering insights into the trade-offs between retrieval sophistication and system cost.

TABLE III
EFFICIENCY METRICS SUMMARY

Model	Indexing Time (min)	Query Latency (ms)	Memory Footprint (GB)
BM25	6	20	0.5
SPLADE	18	65	2.3
DPR	35	140	8.1
Hybrid (S+D)	53	185	10.4

BM25 is the most efficient, but its limited recall and semantic understanding restrict applicability. SPLADE strikes a balance between performance and cost—remaining interpretable and scalable using inverted indices. DPR and Hybrid incur higher latency and memory overhead, making them less suitable for real-time or resource-constrained environments without optimization.

E. Summary of Findings

The comparative analysis across retrieval performance, generative quality, interpretability, and operational efficiency reveals several key insights into the relative strengths of the evaluated models. In terms of retrieval accuracy, the Hybrid model integrating SPLADE and DPR demonstrates superior performance by leveraging both lexical matching and semantic embeddings. This complementary mechanism consistently yields higher nDCG and Recall scores across datasets.

For generative quality, the Hybrid model again achieves the best results, producing answers with higher ROUGE, BLEU, and exact match scores. The diversity and coverage of retrieved contexts from both sparse and dense retrievers enhance the factual completeness and linguistic coherence of generated responses.

From an interpretability perspective, SPLADE outperforms the other methods by providing token-level importance scores via its sparse expansion mechanism. This feature enables transparent tracing of retrieval decisions, which is especially valuable in high-stakes domains such as finance or biomedicine.

Finally, regarding operational efficiency, BM25 remains the most lightweight in terms of indexing time, memory usage, and latency. However, SPLADE strikes a favorable balance between advanced retrieval performance and system cost, offering a scalable and interpretable alternative to dense encoders. Overall, these findings suggest that while hybrid retrieval maximizes effectiveness, SPLADE presents a compelling choice where transparency and resource efficiency are critical.

IV. CONCLUSION

This study presented a comprehensive analysis of retrieval strategies within a Retrieval-Augmented Generation (RAG) framework, focusing on two domain-specific question answering tasks, including financial and biomedical, using datasets from the BEIR benchmark suite. By evaluating traditional lexical retrieval (BM25), sparse lexical expansion (SPLADE), dense semantic retrieval (DPR), and a hybrid retrieval strategy (SPLADE + DPR), we demonstrated the relative strengths and limitations of each method across multiple performance dimensions.

Our results indicate that hybrid retrieval using Reciprocal Rank Fusion (RRF) consistently outperforms individual retrievers in both retrieval effectiveness and downstream generative answer quality. The integration of SPLADE's interpretable lexical expansions with DPR's semantic embeddings ensures robust recall and contextual diversity in retrieved passages. Additionally, token-level heatmap visualizations provide interpretability for SPLADE, highlighting its potential for transparent and auditable retrieval a crucial requirement in high-stakes domains such as finance and healthcare.

In terms of system efficiency, while BM25 remains the most lightweight solution, SPLADE offers a compelling trade-off between scalability and performance. Conversely, DPR exhibits the highest computational overhead, which may limit its standalone use in latency-sensitive applications. The hybrid approach, while computationally heavier than sparse methods, still remains within feasible limits for enterprise-grade deployment, especially when optimized with approximate nearest-neighbor indexing.

A. Future Work

Several promising avenues remain open for future exploration. First, while this study relied on GPT-3.5-Turbo for answer generation, future iterations could evaluate newer multimodal or instruction-tuned large language models (e.g., GPT-4-turbo, Gemini, Claude) to assess sensitivity to retrieval quality and prompt structure. Second, more advanced fusion techniques such as learning-to-rank models or reinforcement learning-based aggregation could further optimize hybrid retrieval performance beyond simple rank-based fusion.

Moreover, expanding the interpretability analysis to include document-level attention distributions and comparing cross-token alignment between SPLADE and DPR could

yield deeper insights into how these systems contextualize semantic similarity. Incorporating user feedback into the retrieval loop, potentially via active learning or human-in-the-loop refinement, also presents a direction for improving real-world applicability.

Lastly, extending this framework to multilingual or low-resource settings could test its robustness across languages and domains where training data and retrieval infrastructure may be limited. Through these extensions, RAG systems can be made more adaptive, transparent, and efficient, better suited for deployment in dynamic, information-intensive environments.

REFERENCES

[1] M. AL-Smadi, "QU-NLP at QIAS 2025 Shared Task: A Two-Phase LLM Fine-Tuning and Retrieval-Augmented Generation Approach for Islamic Inheritance Reasoning," in *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, Association for Computational Linguistics, 2025, pp. 892–898. doi:10.18653/v1/2025.arabicnlp-sharedtasks.123.

[2] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, 2021, pp. 874–880. doi:10.18653/v1/2021.eacl-main.74.

[3] F. Petroni et al., "KILT: a Benchmark for Knowledge Intensive Language Tasks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021, pp. 2523–2544. doi:10.18653/v1/2021.naacl-main.200.

[4] Y. Li, N. Yang, L. Wang, F. Wei, and W. Li, "Generative retrieval for conversational question answering," *Inf. Process. Manag.*, vol. 60, no. 5, p. 103475, Sep. 2023. doi:10.1016/j.ipm.2023.103475.

[5] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Found. Trends Inf. Retr.*, vol. 4, no. 1–2, pp. 1–174, Sep. 2009. doi:10.1561/1500000019.

[6] Y. Lv and C. Zhai, "When documents are very long, BM25 fails!," in *Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, ACM, Jul. 2011, pp. 1103–1104. doi:10.1145/2009916.2010070.

[7] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models," 2021, *arXiv:2104.08663*. doi: 10.48550/arXiv.2104.08663.

[8] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in *Proc. 2020 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Association for Computational Linguistics, 2020, pp. 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

[9] L. Xiong et al., "Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval," 2020, *arXiv:2007.00808*. doi:10.48550/arXiv.2007.00808.

[10] L. Gao and J. Callan, "Unsupervised Corpus Aware Language Model Pre-training for Dense Passage Retrieval," in *Proc. 60th Annu. Meet. Assoc. Comput. Linguist. (Vol. 1: Long Papers)*, Association for Computational Linguistics, 2022, pp. 2843–2853. doi:10.18653/v1/2022.acl-long.203.

[11] M. A. K. Ayoub, Z. Su, and Q. Li, "A Case Study of Enhancing Sparse Retrieval using LLMs," in *Companion Proc. ACM Web Conf. 2024*, ACM, May 2024, pp. 1609–1615. doi: 10.1145/3589335.3651945.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, pp. 3980–3990. doi:10.18653/v1/d19-1410.

[13] T. Formal, B. Piwowarski, and S. Clinchant, "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking," in *Proc. 44th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, ACM, Jul. 2021, pp. 2288–2292. doi:10.1145/3404835.3463098.

[14] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant, "Towards Effective and Efficient Sparse Neural Information Retrieval," *ACM Trans. Inf. Syst.*, vol. 42, no. 5, pp. 1–46, Apr. 2024. doi:10.1145/3634912.

[15] T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant, "SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval," 2021, *arXiv:2109.10086*. doi: 10.48550/arXiv.2109.10086.

[16] P. Mandikal and R. Mooney, "Sparse Meets Dense: A Hybrid Approach to Enhance Scientific Document Retrieval," 2024, *arXiv:2401.04055*. doi: 10.48550/arXiv.2401.04055.

[17] M. G. Arivazhagan et al., "Hybrid Hierarchical Retrieval for Open-Domain Question Answering," in *Findings of the Association for Computational Linguistics: ACL 2023*, Association for Computational Linguistics, 2023, pp. 10680–10689. doi:10.18653/v1/2023.findings-acl.679.

[18] S. Ghosh and P. Rigollet, "Sparse Multi-Reference Alignment: Phase Retrieval, Uniform Uncertainty Principles and the Beltway Problem," *Found. Comput. Math.*, vol. 23, no. 5, pp. 1851–1898, Aug. 2022. doi:10.1007/s10208-022-09584-6.

[19] S. Samuel et al., "MMMORRF: Multimodal Multilingual MOdularized Reciprocal Rank Fusion," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2025, pp. 4004–4009. doi:10.1145/3726302.3730157.

[20] S. Bruch, S. Gai, and A. Ingber, "An Analysis of Fusion Functions for Hybrid Retrieval," *ACM Trans. Inf. Syst.*, vol. 42, no. 1, pp. 1–35, Aug. 2023. doi:10.1145/3596512.

[21] Z. Yang et al., "An Empirical Study of Retrieval-Augmented Code Generation: Challenges and Opportunities," *ACM Trans. Softw. Eng. Methodol.*, Feb. 2025. doi:10.1145/3717061.

[22] J. Seo et al., "Dense-to-Question and Sparse-to-Answer: Hybrid Retriever System for Industrial Frequently Asked Questions," *Mathematics*, vol. 10, no. 8, p. 1335, Apr. 2022. doi:10.3390/math10081335.

[23] R. Nogueira, W. Yang, J. Lin, and K. Cho, "Document Expansion by Query Prediction," 2019, *arXiv:1904.08375*. doi:10.48550/arXiv.1904.08375.

[24] S. Basnet, J. Gou, A. Mallia, and T. Suel, "DeeperImpact: Optimizing Sparse Learned Index Structures," 2024, *arXiv:2405.17093*. doi:10.48550/arXiv.2405.17093.

[25] A. Mallia, O. Khattab, T. Suel, and N. Tonellotto, "Learning Passage Impacts for Inverted Indexes," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2021, pp. 1723–1727. doi:10.1145/3404835.3463030.

[26] O. Khattab and M. Zaharia, "ColBERT," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2020, pp. 39–48. doi:10.1145/3397271.3401075.

[27] C.-W. Huang and Y.-N. Chen, "PairDistill: Pairwise Relevance Distillation for Dense Retrieval," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2024, pp. 18225–18237. doi:10.18653/v1/2024.emnlp-main.1013.

[28] R. Ren et al., "RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021, pp. 2825–2835. doi:10.18653/v1/2021.emnlp-main.224.

[29] J. Zhan et al., "Optimizing Dense Retrieval Model Training with Hard Negatives," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Jul. 2021, pp. 1503–1512. doi:10.1145/3404835.3462880.

[30] L. Gao, Z. Dai, and J. Callan, "COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021, pp. 3030–3042. doi:10.18653/v1/2021.naacl-main.241.

[31] S. MacAvaney, A. Cohan, and N. Goharian, "SLEDGE-Z: A Zero-Shot Baseline for COVID-19 Literature Search," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020, pp. 4171–4179. doi:10.18653/v1/2020.emnlp-main.341.

[32] Q. Lyu et al., "Faithful Chain-of-Thought Reasoning," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2023, pp. 305–329. doi:10.18653/v1/2023.ijcnlp-main.20.

[33] S. Rakin, M. A. R. Shibly, Z. M. Hossain, M. M. Akbar, and Z. Khan, "Leveraging the Domain Adaptation of Retrieval Augmented Generation Models for Question Answering and for Hallucination Reduction," in *Advances in Intelligent Systems and Computing*, Springer Nature Switzerland, 2025, pp. 482–493. doi:10.1007/978-3-031-89063-5_42.

[34] B. Yuan, "FinBERT-QA: Financial Question Answering with Pre-Trained BERT Language Models," 2025, *arXiv:2505.00725*. doi:10.48550/arXiv.2505.00725.

[35] C. Malaviya et al., "ExpertQA: Expert-Curated Questions and Attributed Answers," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, 2024, pp. 3025–3045. doi:10.18653/v1/2024.naacl-long.167.

[36] W. Shi et al., "REPLUG: Retrieval-Augmented Black-Box Language Models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, 2024, pp. 8371–8384. doi:10.18653/v1/2024.naacl-long.463.

[37] M. S. Rashid, J. A. Meem, Y. Dong, and V. Hristidis, "Progressive Query Expansion for Retrieval Over Cost-Constrained Data Sources," 2024, *arXiv:2406.07136*. doi: 10.48550/arXiv.2406.07136.