Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.080190

ARTICLE

DPR-FL: A Dual-Path Retrieval Method with Fusion-Based Filtering and LLM Feedback

Wei Jiang, Weichao Zhang, Haochen Sun*, Yao Meng and Jiayi Wu

School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou, China

*Corresponding Author: Haochen Sun. Email: haochensun@ncwu.edu.cn

Received: 04 February 2026; Accepted: 20 May 2026; Published: 23 July 2026

ABSTRACT: Faced with the surge of massive natural-language content, information retrieval systems must handle increasingly complex queries while filtering noisy information effectively. Although conventional approaches have made notable progress in matching efficiency and general adaptability, they still struggle to precisely model deep semantic associations between query intent and documents in real-world environments. Such limitations can lead to ranking deviations and omission of critical information. Motivated by recent advances in large language models (LLMs) and their capability to capture deep semantics, we propose **DPR-FL**, a **Dual-Path Retrieval** method that integrates fusion-based **Filtering** with structured **LLM Feedback**. It combines direct retrieval with a generation-guided retrieval process to form cooperative information flows. By fusing candidate results from multiple sources, applying a high-dimensional semantic filtering strategy, and leveraging LLM-based semantic feedback, DPR-FL refines and optimizes the selection of relevant documents, improving both coverage and relevance. Furthermore, the framework supports adaptive weighting of candidate sources and semantic signals, enhancing robustness in heterogeneous retrieval scenarios. Together, these components enable finer-grained information selection and semantic enrichment, substantially improving retrieval performance and result reliability in complex contexts. Experimental evaluations on standard web search benchmarks, including TREC DL'19 and DL'20, as well as low-resource BEIR datasets, demonstrate that DPR-FL achieves measurable gains across key metrics such as NDCG@10 and MAP, showing improved generalization, robustness, and adaptability in zero-shot retrieval settings.

KEYWORDS: Information retrieval; dual-path retrieval; fusion-based filtering; large language models; semantic feedback

1 Introduction

In the digital era, information retrieval (IR) serves as a core technology connecting user queries with knowledge resources, supporting efficient access to large-scale heterogeneous data and enabling applications such as question answering, recommender systems, and enterprise knowledge management. Early IR systems were primarily based on statistical models such as Boolean retrieval and TF-IDF. While efficient and easy to implement, these methods rely on surface lexical matching and therefore struggle to capture latent semantic relationships, often leading to retrieval ambiguity and limited recall [1]. To mitigate these issues, techniques such as query expansion, multi-stage retrieval pipelines, and topic modeling approaches including latent semantic analysis (LSA) and probabilistic latent semantic analysis (PLSA) were introduced to provide coarse-grained representations of document topics [2].

The emergence of deep learning shifted IR toward semantic representation learning. Neural retrieval models map queries and documents into a shared semantic space, improving contextual similarity modeling. Transformer-based architectures such as BERT bi-encoders and cross-encoders offer complementary trade-offs between efficiency and ranking accuracy [3,4]. Recent advances in large language models (LLMs) have further expanded retrieval capabilities. Owing to large-scale pre-training and strong semantic reasoning abilities, LLMs enable semantic query expansion and produce informative signals such as summaries or keywords that can be integrated into retrieval pipelines for filtering and reranking across sparse and dense matching spaces [5]. Instruction tuning further improves robustness in zero-shot and low-resource settings, while interactive signals from LLMs allow retrieval strategies to suppress noisy candidates and improve output quality [6]. These developments suggest that combining generative reasoning with principled retrieval mechanisms is a promising direction for complex information environments.

Despite these advances, IR systems still face challenges in scenarios involving complex semantic structures and high information redundancy. Existing methods often fail to capture deep semantic relationships between queries and documents, causing ranking lists to misrepresent the informational value of documents and potentially bury critical evidence among less relevant candidates. Cuconasu et al. [7] emphasize that retrieval quality directly affects downstream generative models, since top-ranked documents lacking the required evidence can significantly degrade generation accuracy.

Therefore, retrieval performance plays a critical role in generation. Sparse retrieval emphasizes lexical coverage, whereas dense retrieval focuses on semantic similarity within embedding spaces. LLM-based generation introduces reasoning-driven semantic priors that are not directly aligned with the two aforementioned retrieval paradigms. These paradigms operate in different representation spaces and optimization regimes, so existing frameworks typically combine them in a loosely coupled manner without explicit coordination. Consequently, current retrieval pipelines lack a principled mechanism for integrating lexical coverage, semantic similarity, and LLM-based generation. Motivated by this observation, this paper proposes DPR-FL, a dual-path retrieval framework that integrates fusion-based filtering with LLM-driven semantic feedback, adopting a pure inference-only paradigm in which all LLM-involved operations are executed without updating model parameters. Building on this paradigm, the proposed framework does not treat sparse, dense, and generative retrieval as independent modules. Instead, it models retrieval as a structured information decomposition problem: a direct retrieval path captures lexical and embedding-level evidence using sparse and dense models, while a generation-guided path leverages LLM-generated semantic cues to infer latent query intent and explore additional contextual evidence. These complementary pathways collectively enable the system to balance retrieval coverage with semantic precision.

Specifically, the framework (shown in Fig. 1) constructs two parallel retrieval paths. The direct path executes the original query against sparse indexes (e.g., BM25) and dense indexes (e.g., Contriever), while the generation-guided path expands the query through LLM-based semantic generation and extracts informative cues for subsequent retrieval. Candidate documents from both paths are merged and deduplicated. A fusion-based semantic filtering stage acts as an interaction mechanism to consolidate heterogeneous cross-path retrieval evidence. Instead of conventional reranking alone, this step evaluates high-dimensional semantic representations generated by the LLM to identify documents consistently supported by multiple retrieval signals. Finally, the filtered document subset is combined with the original query and fed back into the language model to generate refined retrieval prompts for a final retrieval stage. This feedback mechanism forms a closed-loop retrieval process in which intermediate results guide query reformulation, enabling iterative alignment between lexical evidence, semantic similarity, and generative reasoning. The resulting

framework preserves the high-coverage strengths of sparse and dense retrieval while incorporating LLM-based semantic inference and filtering, thereby improving the system's ability to identify key documents in complex and high-redundancy environments.

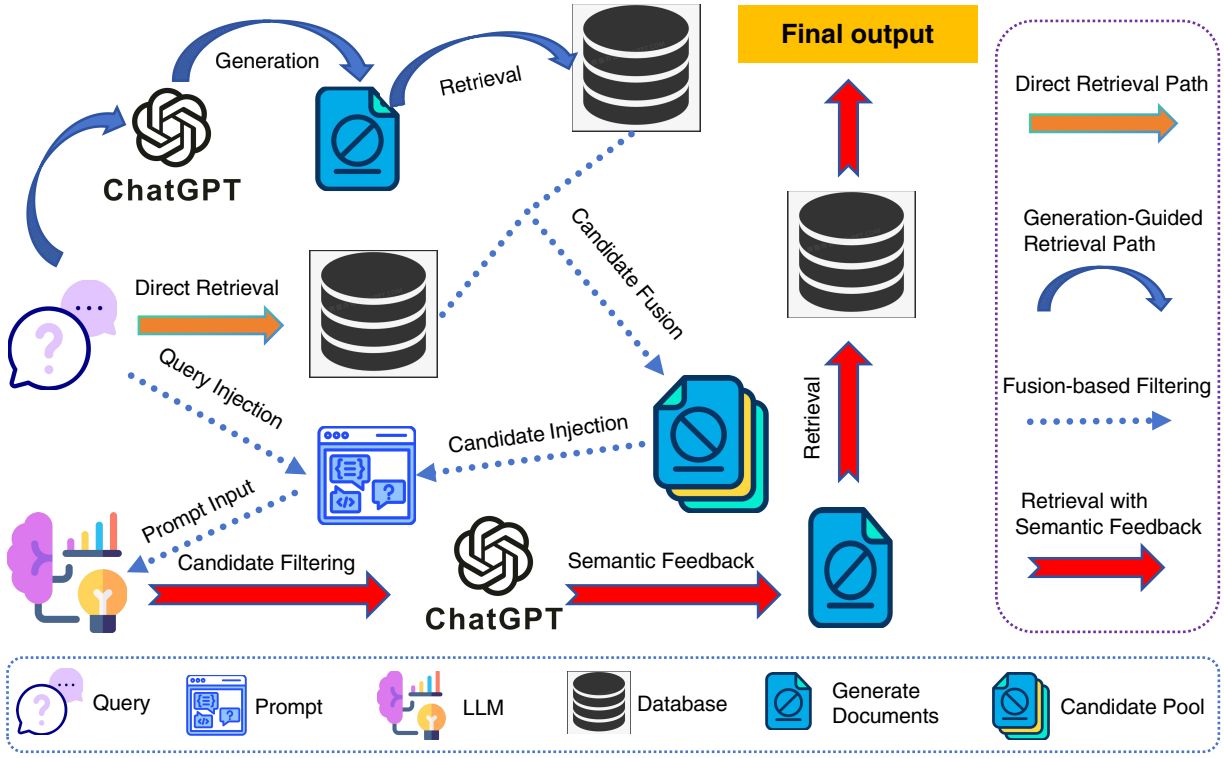

Figure 1: Overall architecture of the proposed DPR-FL framework.

The main contributions of this work are summarized as follows, which, to the best of our knowledge, have not appeared in prior work.

(1) We propose DPR-FL, which is not only a dual-path retrieval framework but also a new retrieval orchestration principle that enables direct retrieval, generation-guided retrieval, and LLM-driven operations to work together to acquire complementary evidence and balances recall with precision.

(2) We introduce a fusion-based semantic filtering mechanism that consolidates cross-path retrieval signals using high-dimensional semantic representations produced by LLMs. The novel mechanism essentially enables interaction among heterogeneous retrieval signals.

(3) We design a semantic feedback mechanism that forms a closed-loop retrieval process, allowing intermediate retrieval results to guide query reformulation and improve retrieval precision. This distinctive mechanism avoids missing relevant documents while refining and distilling dispersed, fragmented information.

2 Related Work

In recent years, information retrieval has shifted from sparse term-matching to dense semantic retrieval, driven by large pretrained language models. This shift has spurred research in dense retrieval, zero-shot transfer, and retrieval-augmented generation. We review key work in each of these areas.

2.1 Dense Retrieval

Dense retrieval typically uses pretrained language models (e.g., BERT [8]) in a dual-encoder framework, encoding queries and documents independently in a shared embedding space. Dense Passage Retrieval (DPR) [9] demonstrated this approach's effectiveness for open-domain question answering. However, the bi-encoder's lack of explicit query–document interaction can limit fine-grained matching. To improve representation learning, subsequent work has proposed advanced training techniques: for example, ANCE [10] incorporates hard negatives via approximate nearest-neighbor mining, and knowledge distillation [11] transfers the fine-grained matching ability of cross-encoders to efficient dual-encoders.

Beyond training strategies, other work rethinks model pretraining and scaling. Condenser [12] proposes a retrieval-oriented pretraining architecture to better align the model with retrieval tasks. Scaling up dual-encoder models likewise enhances cross-domain generalization [13]. From a system perspective, large-scale dense retrieval requires efficient similarity search. GPU-accelerated libraries like FAISS [14] enable billion-scale approximate nearest-neighbor retrieval. Nevertheless, most dense retrieval approaches still treat retrieval as a static mapping from queries to documents, leaving little room to incorporate feedback from intermediate results, so retrieval decisions cannot be revised after encoding, potentially missing relevant documents or yielding suboptimal rankings when queries are ambiguous or contain multiple semantic aspects.

2.2 Zero-Shot Dense Retrieval

Supervised dense retrievers perform well on in-domain data but rely on labeled examples. Unsupervised approaches mitigate this dependency: Contriever [15] learns dense encodings via contrastive learning without relevance labels, achieving competitive zero-shot performance. Generation-based methods further improve retrieval robustness: HyDE [16] uses a large language model to generate a hypothetical document for each query, which serves as a semantic proxy for retrieval. LaPraDoR [17] alternates lexical query expansion with contrastive training over multiple rounds to progressively refine the query representations. These strategies leverage synthetic query contexts and iterative refinement, yielding substantial gains in cross-domain retrieval. However, if the generated proxies misalign with the query's intent, retrieval may still fail, as these approaches remain sensitive to complex or reasoning-intensive queries. When the initial generated representation fails to capture the true intent of the query, the resulting retrieval errors tend to propagate, since most frameworks lack mechanisms for systematically correcting early-stage representation bias.

2.3 Retrieval-Augmented Generation (RAG)

Large language models often hallucinate on knowledge-intensive tasks. Retrieval-Augmented Generation (RAG) [18] addresses this by combining an external retriever with a sequence-to-sequence generator, grounding outputs in retrieved evidence. Fusion-in-Decoder (FiD) [19] improves this approach by encoding each retrieved passage independently and fusing them during decoding, significantly boosting open-domain QA performance. Retrieval augmentation also benefits dialogue systems, reducing hallucination and improving factuality [20]. However, the success of RAG critically depends on retrieval quality: irrelevant or noisy documents can mislead the generator. To mitigate this, recent work applies reranking and filtering to the retrieved results [21]. These findings underscore that high-quality retrieval is essential for effective generation, yet a recurring challenge in RAG systems remains the quality of retrieved candidates. When retrieval introduces redundant or weakly relevant passages, the generator may incorporate noisy evidence, negatively affecting both factual accuracy and coherence in multi-document generation [21].

2.4 Generation-Augmented Retrieval (GAR)

To improve retrieval quality, recent work leverages large language models for query expansion and iterative feedback. TART [22] uses instruction-aware multitask training to adapt retrieval models across diverse tasks and query intents. Other approaches generate pseudo-documents or context: for example, Generation-Augmented Retrieval (GAR) [23] and Query2Doc [24] enrich queries with LLM-generated documents, while LameR [25] incorporates candidate answers into retrieval prompts. More fine-grained methods like CSQE [26] use sentence-level signals from initial retrievals to guide query expansion. Iterative frameworks such as InteR [27] build multi-round interaction between the retriever and an LLM, with each refining the other's output. These approaches reflect a shift toward iterative retrieval–generation collaboration. Nonetheless, many current methods still rely on loosely structured steps, leaving room for more systematic integration of retrieval feedback, as existing GAR frameworks largely depend on loosely coupled generation–retrieval processes. The lack of explicit coordination across retrieval paths and structured feedback further limits their ability to achieve stable improvements, especially in complex or zero-shot retrieval scenarios.

2.5 Summary and Discussion

Overall, existing retrieval paradigms improve retrieval effectiveness from multiple perspectives. Dense retrieval primarily focuses on representation learning and scalable similarity search, while generation-based approaches expand queries through synthetic semantic contexts. RAG integrates retrieval with generation to ground model outputs in external evidence, and GAR-style methods further introduce iterative interaction between retrieval systems and large language models. Despite these advances, most approaches still rely on static retrieval pipelines or loosely coupled generation–retrieval processes, making it difficult to incorporate structured feedback during candidate refinement. In this context, the proposed DPR-FL can be regarded as a GAR framework, but it is not a purely generation-driven retrieval method. Building on generative retrieval, it integrates direct retrieval and dense vector retrieval, and operates in coordination with a large language model to further filter and re-rank initial candidate documents, thereby achieving a more robust selection of candidates by combining lexical matching and semantic representations.

3 Methodology

This section, referring to Fig. 1, introduces DPR-FL, which comprises four stages: direct retrieval, generation-guided retrieval, fusion-based filtering, and semantic feedback. In this work, the LLM is treated as three operators—LLM_{gen}, $\text{LLM}_{\text{filter}}$, and $\text{LLM}_{\text{summary}}$ (see definitions below).

This chapter introduces DPR-FL referring to Fig. 1 and Algorithm 1, which comprises four stages: direct retrieval, generation-guided retrieval, fusion-based filtering, and semantic feedback corresponding to Sections 3.1–3.4. In this work, the LLM is treated as three operators: LLM_{gen}, LLM_{filter}, and $\text{LLM}_{summary}$.

Algorithm 1: Framework of DPR-FL

Require: $query\_content$
Ensure: $retrieval\_results$
1: $direct\_retrieval\_candidates$ = DirectRetrieval($query\_content$, BM25&Contriever)
2: $generation\_guided\_retrieval\_candidates$ = GenerationGuidedRetrieval($query\_content$)
3: $filtered\_candidates$ = FusionBasedFiltering(
$direct\_retrieval\_candidates$, $generation\_guided\_retrieval\_candidates$)

(Continued)

Algorithm 1 (continued)

4: $\text{LLM}_{summary}$ = SemanticFeedback($filtered\_candidates$)
5: $retrieval\_results$ = DirectRetrieval($\text{LLM}_{summary}$, BM25)
6: **return** $retrieval\_results$

3.1 Direct Retrieval

In this stage, the system executes two retrieval pathways in parallel to achieve comprehensive coverage of explicit lexical matches and latent semantic relevance.

3.1.1 BM25 Retrieval

A traditional inverted-index mechanism based on the BM25 algorithm computes TF-IDF weighted scores between the query and documents, and quickly returns the top k_1 documents that contain the query keywords, ensuring high recall for explicit lexical matches.

3.1.2 Contriever Retrieval

A pretrained language model (Contriever) encodes queries and documents into a shared high-dimensional vector space. Retrieval is performed using an Approximate Nearest Neighbor (ANN) search to obtain the top k_2 documents whose embeddings are closest to the query embedding, thereby capturing deep semantic associations and enabling stronger semantic matching.

3.1.3 Retrieval Result Fusion

Given a query q and a document collection, a retriever R accepts q and returns a set of relevant documents:

$$\{d_1, d_2, \ldots, d_k\} = R(q) \tag{1}$$

R runs the BM25 and Contriever paths in parallel; we denote their outputs as

$$R_{\text{lex}}(q) = \text{BM25}(q) \tag{2}$$

$$R_{\text{dense}}(q) = \text{Contriever}(q) \tag{3}$$

where R_{lex} and R_{dense} are the lexical (BM25) and semantic (Contriever) retrievers, respectively. The top k_1 documents from $R_{\text{lex}}(q)$ and the top k_2 documents from $R_{\text{dense}}(q)$ are combined, and duplicates are removed to form the initial candidate pool:

$$C = \text{Top}_{k_1}\left(R_{\text{lex}}(q)\right) \cup \text{Top}_{k_2}\left(R_{\text{dense}}(q)\right) \tag{4}$$

This sparse and dense candidate strategy balances precise lexical recall with deep semantic coverage: it preserves documents that match important keywords while introducing semantically related documents that might otherwise be missed due to lexical variation.

3.2 Generation-Guided Retrieval

In the generative retrieval stage, a preliminary response generated by a large language model (LLM) is used as an expanded query to further enrich the candidate pool. The process consists of three steps.

3.2.1 Semantic Expansion Generation

Given the original query q, we invoke a pretrained large language model Q to generate an initial response set:

$$S = \{s_1, s_2, \ldots, s_g\} = Q(q) \tag{5}$$

where each candidate response s_i is produced through semantic interpretation and reasoning conditioned on q. These responses provide diverse extensions and refinements of the query intent. Compared with a single expansion, the candidate set S captures multiple facets of the query more comprehensively and offers richer semantic signals for subsequent fusion and selection.

3.2.2 Query Augmentation Construction

The original query q is merged with the generated candidate set S to construct an augmented retrieval input:

$$\tilde{q} = (q, S) \tag{6}$$

This fusion preserves the explicit user question while incorporating implicit cues from the model about potential information needs.

3.2.3 Semantic-Guided Retrieval

Dense retrieval is performed using $\tilde{q}$ over the document collection:

$$C' = R(\tilde{q}) = \mathrm{Top}_{k_3}\big(\mathrm{Contriever}(\tilde{q})\big) \tag{7}$$

where $\mathrm{Contriever}(\tilde{q})$ denotes the semantic vector retrieval results produced by the Contriever model for the augmented query, and Top_{k_3} indicates that the top k_3 most relevant documents are selected to form the candidate set. Here, k_3 represents the retrieval cutoff threshold.

In this stage, the augmented queries generated by the model are used in vector-based retrieval, allowing the generative and retrieval components to work together and providing a set of high-quality candidate documents for the next fusion-based filtering step.

3.3 Fusion-Based Filtering

After the generative retrieval stage produces the expanded candidate set C', DPR-FL forms the enhanced candidate pool by taking the union of C' and the initial candidate set C and removing duplicates:

$$D = C \cup C' \tag{8}$$

Subsequently, following the templated prompt p, the original query q and the enhanced candidate set D are assembled into the model input $p(q, D)$ and submitted to a large language model for semantic evaluation.

Prompt (p) for Candidate Filtering.

Given a user query and a numbered set of candidate documents, identify the top-k documents that best answer the query, considering explicit, implicit, and inferred information.

Input:

- Query: {query}

- Candidate documents (1..N): {doc_1}, {doc_2},..., {doc_N}

Instructions:

1. Read the query and all candidate documents.
2. Evaluate explicit statements, indirect implications or paraphrases, and contextual reasoning linking document content to query intent.
3. Select the top-k documents that most completely and directly answer the query.
4. Output only the document numbers in ranked order (e.g., 3, 7, 9, 1, 6, 15, 5, 8, 4, 23), with no explanation or extra text.

Example Output: 3, 7, 9, 1, 6, 15, 5, 8, 4, 23

In this stage, the language model functions as a semantic filter rather than a generator. Instead of producing free-form responses, the model compares the candidate documents with respect to the query intent and returns their indices according to relevance judgments.

To make the LLM role and behavior precise, we distinguish three LLM operators used in DPR-FL with explicit input–output signatures:

$$\mathrm{LLM}_{\mathrm{gen}} : q \mapsto S = \{s_i\}_{i=1}^{g} \qquad \mathrm{LLM}_{\mathrm{filter}} : (q, D, p, k) \mapsto D' \subseteq D \qquad \mathrm{LLM}_{\mathrm{summary}} : (q, D') \mapsto \hat{s}$$

Here $\mathrm{LLM}_{\mathrm{gen}}$ produces generation-guided expansions (used in Section 3.2), $\mathrm{LLM}_{\mathrm{filter}}$ directly selects the top-k relevant candidates in D according to the prompt p (the prompt shown above is an instance of p), and $\mathrm{LLM}_{\mathrm{summary}}$ generates a compact, query-focused synopsis from the filtered set D'.

Formally, the filtering operation used in DPR-FL can therefore be written as

$$D' = \mathrm{LLM}_{\mathrm{filter}}(q, D, p, k) \tag{9}$$

where the output is a top-k subset of documents sorted by relevance, directly returned by the model under the given prompt.

This process exploits the model's ability to model semantic relationships between context and documents, enabling fine-grained relevance assessment and quality filtering. The resulting subset maintains coverage while being more information-dense and better aligned with query intent.

3.4 Semantic Feedback

DPR-FL mitigates limitations of conventional retrieval: keyword search may miss semantically relevant documents, while semantic retrieval can include marginal content. The dual-path design ensures coverage, fusion-based filtering refines candidates, and summary-oriented semantic feedback distills key semantic signals, conceptually analogous to pseudo relevance feedback, producing a focused and information-rich candidate set.

Following direct retrieval, generative retrieval, and semantic selection, the system obtains an initial candidate subset D' that captures multi-level semantic cues and contextual signals. However, D' may still contain dispersed information, redundant passages, or fragments on marginal topics.

To increase information density and retrieval specificity, we feed the entire candidate subset D' into the LLM to produce a compact, query-focused synopsis. The model generates a compressed summary $\hat{s}$ that consolidates key facts and salient content across documents in D' while remaining tightly aligned with the original query q:

$$\hat{s} = \mathrm{LLM}_{\mathrm{summary}}(q, D') \tag{10}$$

The summary $\hat{s}$ is then used as a new retrieval query to perform a final BM25 search over the corpus:

$$F = \mathrm{Top}_{k_4}\left(\mathrm{BM25}(\hat{s})\right) \tag{11}$$

where F denotes the final document set and k_4 is the retrieval cutoff. Centering the final retrieval on a distilled, high-value summary reduces redundancy, mitigates semantic drift, and produces a document set that is more topically coherent and information-rich.

4 Experiments

4.1 Datasets and Evaluation Metrics

To ensure comparability with prior work, we strictly follow the datasets and evaluation metrics employed in existing studies when evaluating DPR-FL. Concretely, for multiple comparative experiments we select two representative web-search benchmarks: TREC Deep Learning 2019 (DL'19) [28] and TREC Deep Learning 2020 (DL'20) [29], which together allow a systematic assessment of the proposed method in large-scale web search scenarios. In addition, we evaluate DPR-FL's generalization on six low-resource datasets from the BEIR benchmark [30] (SciFact, ArguAna, TREC-COVID, FiQA, DBpedia, and TREC-NEWS).

Following prior work, we report mean average precision (MAP) and normalized discounted cumulative gain at rank 10 (NDCG@10) as the primary metrics. For DL'19 and DL'20, we further compute recall at the top 1000 retrieved documents (Recall@1K). For the BEIR datasets, we uniformly report NDCG@10.

4.2 Benchmarks

To comprehensively evaluate both zero-shot and supervised performance of the proposed method, we divide existing retrieval baselines into two categories.

(1) **Methods without relevance judgments (zero-shot).** These methods require no query–document relevance annotations and perform retrieval in zero-shot settings using term statistics or pretrained vector representations. Representative approaches include the classical keyword-matching BM25 [31]; Contriever, which leverages unsupervised contrastive learning to map queries and documents into a shared semantic space [15]; and recent LLM-augmented approaches such as HyDE [16] and InteR [27], which improve zero-shot retrieval by synthesizing hypothetical documents or by establishing collaborative interaction between retrievers and large language models to enrich query signals.

(2) **Methods with relevance judgments (supervised).** These methods exploit large-scale labeled datasets (*e.g.*, MS MARCO [32]) for supervised fine-tuning to capture finer-grained relevance signals between queries and documents. Representative work includes DeepCT, which enhances sparse term representations via learned term-weighting [33]; DPR, a bi-encoder dense retrieval framework; ANCE, which employs dynamic negative sampling to improve contrastive training; and Contriever$^{\mathrm{FT}}$, i.e., Contriever fine-tuned under the same supervised framework [15]. These supervised approaches typically yield substantial gains in both precision and recall.

4.3 Implementation Details

For inference, we evaluate three representative large language models that span proprietary and open-source paradigms. First, OpenAI's proprietary conversational model GPT-3.5-Turbo, chosen for its efficient inference and strong semantic understanding. Second, Qwen-Max (Alibaba), a proprietary model noted for its few-shot capabilities and domain adaptation performance. Third, the community-maintained

open-source model DeepSeek-R1, whose fully open training and deployment pipeline facilitates reproducibility and customization.

Preliminary experiments on the InteR benchmark indicated that GPT-3.5-Turbo achieved the best performance among the candidate inference models. To ensure direct comparability with the strongest baseline (InteR) and to maintain consistency across key evaluations, we therefore use GPT-3.5-Turbo as the primary inference engine in the main experiments reported below.

4.4 Experimental Results

In the web-search experiments (Table 1), we evaluate DPR-FL on TREC DL'19 and DL'20 against two baseline groups: methods without relevance judgments (BM25, Contriever, HyDE, InteR) and supervised methods trained with relevance judgments (DeepCT, DPR, ANCE, Contriever^{FT}). We further conduct paired t-tests to statistically validate the performance differences, where ** denotes $p < 0.01$ indicating extremely significant improvements. Within the first group, DPR-FL achieves statistically significant improvements over the state-of-the-art InteR, with approximately +3.0 and +6.0 percentage points in MAP and NDCG@10, respectively. Such significant gains confirm that the superiority of DPR-FL is not caused by random fluctuations, but stems from the rationality of the proposed framework. Despite operating in a zero-shot setting, DPR-FL also outperforms supervised baselines on reported metrics. We note that InteR (gpt-3.5-turbo) attains slightly higher Recall@1K on DL'20, suggesting a trade-off: DPR-FL prioritizes top-rank precision rather than maximizing long-tail coverage. Overall, DPR-FL shows consistent advantages over compared methods in zero-shot retrieval.

Table 1: Retrieval results on TREC DL'19 and DL'20. Best scores are in bold. Significance marks: $^{**}p < 0.01$ (paired t-test against InteR (gpt-3.5-turbo)).

Methods	DL'19			DL'20		
	MAP	**NDCG@10**	**R@1k**	**MAP**	**NDCG@10**	**R@1k**
Methods without relevance judgment						
BM25	30.1	50.6	75.0	28.6	48.0	78.6
Contriever	24.0	44.5	74.6	24.0	42.1	75.4
HyDE	41.8	61.3	88.0	38.2	57.9	84.4
InteR (gpt-3.5-turbo)	50.0	68.3	89.3	46.8	63.5	**88.8**
DPR-FL (gpt-3.5-turbo)	**53.0****	**74.8****	**90.7****	**50.0****	**70.1****	87.0
Methods with relevance judgment						
DeepCT	–	55.1	–	–	55.6	–
DPR	36.5	62.2	76.9	41.8	**65.3**	81.4
ANCE	37.1	**64.5**	75.5	40.8	64.6	77.6
Contriever^{FT}	**41.7**	62.1	**83.6**	**43.6**	63.2	**85.8**

To evaluate our LLM-based relevance matching mechanism from multiple perspectives, we further validate the performance of DPR-FL in diverse low-resource scenarios. Results on the six low-resource subsets of the BEIR benchmark are shown in Table 2, where DPR-FL yields statistically significant improvements over existing baselines. Specifically, on the medical retrieval task TREC-COVID and the scientific literature task SciFact, DPR-FL achieves about 6% and 8% absolute gains in NDCG@10 over InteR, with strong statistical significance. We also observe consistent significant improvements on the financial QA task (FiQA) and the

general knowledge task (DBpedia). However, DPR-FL performs slightly worse than the best baseline on the dialog-based QA dataset ArguAna and the news retrieval dataset TREC-NEWS. This gap may be related to using a general-domain language model, which has limited ability to capture temporal information and specialized terminology in news retrieval. Moreover, longer query contexts in ArguAna make it harder to focus on key information and achieve semantic alignment, leading to a slight decline on these two tasks.

Table 2: Retrieval performance (NDCG@10) on the low-resource BEIR datasets. Best-performing values are highlighted in bold. Significance marks: $^{**}p < 0.01$ (paired t-test against InteR (gpt-3.5-turbo)).

Methods	SciFact	ArguAna	TREC-COVID	FiQA	DBpedia	TREC-NEWS
Methods without relevance judgment						
BM25	67.9	39.7	59.5	23.6	31.8	39.5
Contriever	64.9	37.9	27.3	24.5	29.2	34.8
HyDE	69.1	**46.6**	59.3	27.3	36.8	44.0
InteR (gpt-3.5-turbo)	71.7	40.9	69.7	26.0	42.1	**52.8**
DPR-FL (gpt-3.5-turbo)	**77.2****	38.4	**77.6****	**29.6****	**45.3****	51.4
Methods with relevance judgment						
DPR	31.8	17.5	33.2	29.5	26.3	16.1
ANCE	50.7	41.5	**65.4**	30.0	28.1	38.2
Contriever^{FT}	**67.7**	**44.6**	59.6	**32.6**	**41.3**	**42.8**

Combined with the performance deltas shown in Fig. 2, we further discuss the cross-dataset generalization of DPR-FL at a per-dataset level. Our approach achieves the most notable gains on fact-focused, formally structured domains including SciFact, TREC-COVID, FiQA, and DBpedia, where LLM feedback and dual-path retrieval effectively improve semantic alignment. By contrast, improvements are limited on conversational and time-sensitive domains like ArguAna and TREC-NEWS, due to longer contextual dependencies and domain-specific linguistic characteristics. This dataset-wise breakdown shows that DPR-FL generalizes robustly across standard retrieval scenarios, while its adaptability to highly specialized or conversational domains can be further improved, providing a clearer picture of its cross-domain behavior.

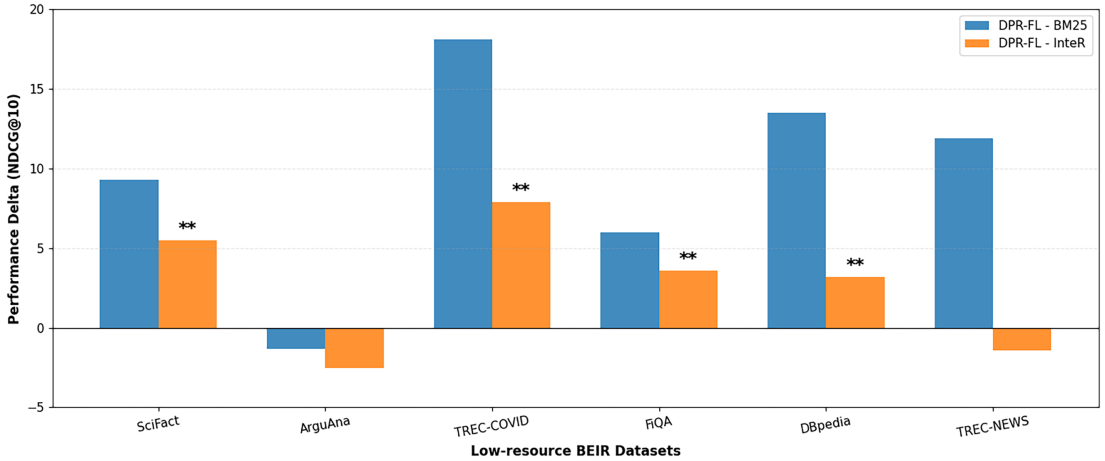

Figure 2: Performance delta of DPR-FL against BM25 and InteR on low-resource BEIR datasets. $^{**}p < 0.01$ (paired t-test against InteR).

To further validate the contribution of each component in DPR-FL, we conduct an ablation study. As shown in Table 3, removing any single module (including the dual-path structure, LLM feedback, or fusion filtering) leads to a noticeable performance decline. Specifically, the dual-path design, which integrates both lexical and semantic retrieval signals, plays a pivotal role in improving ranking accuracy—its removal results in a 4.2–4.5 percentage point drop in MAP on both DL'19 and DL'20 datasets. By contrast, the impact of LLM feedback and fusion filtering varies slightly across datasets: on DL'19 (with more standardized query expressions), the LLM feedback module contributes a 2.0 percentage point MAP improvement, while on DL'20 (with more complex document distributions), the fusion filtering module is more effective in reducing noise and improving relevance matching. This variation is closely related to the dataset characteristics: DL'19 focuses more on precise semantic matching, while DL'20 requires stronger noise reduction due to its more diverse document types. Such a detailed cross-dataset analysis not only confirms the necessity of each component but also provides clear guidance for the practical application of the proposed framework across different retrieval scenarios.

Table 3: Ablation results of DPR-FL on TREC DL'19 and DL'20, with best values in bold.

Methods	DL'19			DL'20		
	MAP	NDCG@10	R@1k	MAP	NDCG@10	R@1k
Full DPR-FL	**53.0**	**74.8**	90.7	**50.0**	**70.1**	87.0
w/o Dual-path (Sparse-only)	48.8	68.2	**90.8**	45.8	64.2	**87.2**
w/o Dual-path (Dense-only)	49.5	69.0	90.0	46.5	65.0	86.5
w/o LLM Feedback	51.0	71.5	90.2	48.0	67.0	87.0
w/o Fusion Filtering	50.0	70.5	89.8	47.5	66.2	86.8

4.5 Discussion

In our experimental design, we set the retrieval-related parameters uniformly as $k_1 = k_2 = k_3 = k$ and fixed $k_4 = 1000$. This configuration aims to establish a consistent basis for comparison across the three evaluation metrics: MAP, NDCG@10, and Recall@1K. Specifically, the first two metrics are computed on the same top-k results, while Recall is measured at a fixed cutoff depth of 1000. Such a design minimizes interference caused by inconsistent parameter settings, thereby ensuring the comparability of results. On this foundation, we further investigate the impact of the number of generated candidates g, the retrieval size k, output stability, and the choice of large language models, ensuring that observed performance differences primarily stem from variations in the key variables rather than from experimental inconsistencies.

In the generative retrieval stage of the DPR-FL method, multiple candidate documents generated by the language model are introduced, and their number g is controlled to investigate its impact on overall retrieval performance. As shown in Table 4, on both TREC DL'19 and DL'20 datasets, as the number of generated candidates g increases, the model steadily improves in MAP and NDCG@10, reaching the optimal performance when $g = 10$. Further increasing g results in minimal performance changes, indicating that the retrieval performance becomes stable after a certain point. Meanwhile, the Recall@1K metric fluctuates minimally across different settings, remaining stable overall. This suggests that the number of generated candidate documents has a limited effect on recall ability but provides a more direct improvement to ranking quality.

Table 4: Impact of the generated candidate size g on DFR-S performance (GPT-3.5-Turbo).

g	DL'19			DL'20		
	MAP	NDCG@10	R@1K	MAP	NDCG@10	R@1K
4	51.1	72.7	90.2	48.4	68.2	88.0
6	51.0	72.9	90.1	47.7	67.4	88.5
8	51.8	73.8	89.7	49.5	68.1	88.2
10	53.0	74.8	90.7	48.9	67.8	88.3
12	51.0	73.3	89.7	49.8	68.3	89.0

In light of the minor fluctuations observed in some metrics in Table 4, we attribute this variability to both the number of generated candidates g and the inherent randomness of large language model outputs. Specifically, the generated content for the same query may differ slightly across runs. To further assess the output stability of DPR-FL under such stochastic generation conditions, we conduct five repeated experiments with a fixed number of generated candidate documents ($g = 10$), which corresponds to the optimal setting identified in Table 4. As shown in Table 5, although minor variations are observed across different runs, the overall performance fluctuations remain small. This indicates that DPR-FL exhibits good stability and robustness, and that the observed performance gains are consistent rather than resulting from random variation.

Table 5: Output stability and variance statistics of DPR-FL across repeated runs ($g = 10$, GPT-3.5-Turbo).

Run	DL'19			DL'20		
	MAP	NDCG@10	R@1K	MAP	NDCG@10	R@1K
1	53.0	74.8	90.7	48.9	67.8	88.3
2	52.5	73.3	89.7	48.1	67.7	88.1
3	52.3	73.9	90.0	48.4	68.0	89.4
4	52.7	74.7	90.4	48.8	67.1	88.7
5	52.2	73.0	89.8	49.2	69.3	89.5
Avg	52.54	73.94	90.12	48.68	67.98	88.78
Std	0.32	0.81	0.42	0.43	0.81	0.63

For completeness, we report the average (Avg) and standard deviation (Std) across these runs in Table 5. The small standard deviations across metrics confirm that the performance improvements of DPR-FL are stable and not caused by stochastic variations in LLM generation.

In the initial phase of the DPR-FL method, both the direct retrieval and generative retrieval paths rely on retrieving candidate documents, making the choice of the retrieval document quantity k crucial to overall performance. To systematically evaluate its impact, Table 6 reports the performance of DPR-FL under different values of k. The results show that as k increases from 10 to 15, retrieval performance consistently improves and reaches its peak at $k = 15$. However, further increasing k leads to a performance decline, indicating that an excessive number of redundant candidates introduces noise, which interferes with subsequent merging, selection, and generation processes, thereby negatively affecting overall retrieval quality.

Table 6: Effect of retrieval document size k on DPR-FL performance (GPT-3.5-Turbo).

k	DL'19			DL'20		
	MAP	NDCG@10	R@1K	MAP	NDCG@10	R@1K
10	50.6	72.7	89.7	48.5	68.9	89.3
13	50.5	73.2	88.8	48.2	67.3	87.9
15	53.0	74.8	90.7	48.9	67.8	88.3
17	51.9	73.1	90.1	50.0	70.1	87.0
20	50.6	71.6	89.8	48.4	67.5	87.5

In the candidate merging phase of DPR-FL, we employ three representative large language models, covering both proprietary and open-source paradigms, as the discriminator module for semantic filtering and ranking of multi-path candidate documents. Table 7 summarizes the overall retrieval performance of different models at this stage. The results show that Qwen-Max achieves the best performance on several metrics, reflecting stronger semantic understanding and discrimination capability. DeepSeek-R1 follows closely and maintains stable performance across both datasets. In contrast, GPT-3.5-Turbo exhibits weaker performance in semantic consistency judgment and document selection precision, which may limit its effectiveness in the filtering stage.

Table 7: Robustness of DPR-FL across different large language models.

LLM	DL'19			DL'20		
	MAP	NDCG@10	R@1K	MAP	NDCG@10	R@1K
Qwen-Max	53.0	74.8	90.7	48.9	67.8	88.3
GPT-3.5-Turbo	49.7	71.3	88.4	47.2	67.5	89.1
DeepSeek-R1	51.3	72.4	89.1	49.5	68.0	87.4

4.6 Computational Cost Analysis

In our method, the large language model is used in three stages: candidate generation (LLM_{gen}), semantic filtering during fusion (LLM_{filter}), and final response generation ($LLM_{summary}$). On the DL19 dataset (43 queries), the mean token usage per query is 120 tokens for LLM_{gen}, 700 tokens for LLM_{filter} (including query, candidate excerpts, and prompt overhead), and 514 tokens for $LLM_{summary}$, totaling 1334 tokens per query. The method involves three LLM inference steps per query, with an average latency of about 3–5 s, including approximately 1 s for candidate generation, 1–3 s for fusion filtering, and about 1 s for final generation. Most of the cost arises from the filtering stage due to its longer inputs, while in practice the overhead can be controlled by truncating candidate excerpts and limiting the number of candidates (top-k), keeping overall cost and latency moderate compared to retrieval-only baselines.

5 Conclusions

This paper presents DPR-FL, a dual-path retrieval method with fusion-based filtering and summary-oriented semantic feedback from large language models (LLMs). DPR-FL integrates dual-path retrieval, fusion-based filtering, and summary-guided semantic feedback, providing a principled mechanism to combine lexical and semantic signals and offering theoretical insights for effective candidate selection beyond experimental gains. Experiments on web search benchmarks and low-resource BEIR datasets show that

DPR-FL consistently outperforms unsupervised and supervised baselines, achieving notable improvements in MAP and NDCG and demonstrating stability and potential for generalization.

Future work will focus on automated prompt optimization, dynamic candidate generation and fusion strategies, and the evaluation of the framework's robustness and applicability in larger-scale web search and low-resource scenarios. Specifically, adjusting the number of generated candidates, fusion filtering strategies, and semantic feedback modules could enable efficient retrieval for complex queries, while assessing different LLMs across diverse tasks. Further exploration of cross-domain datasets and long-text retrieval will enhance the framework's scalability and generalization in practical IR systems.

Additionally, as an emerging and evolving research direction in information retrieval, the use of LLMs as judges for relevance assessment and ranking still requires further systematic investigation [34,35]. While our experimental results have verified the effectiveness of the LLM feedback mechanism in this work, exploring the reliability, consistency, and interpretability of LLM-based judgments will be an important part of our follow-up research to further improve the trustworthiness of the framework.

Acknowledgement: None.

Funding Statement: This research was funded by the National Natural Science Foundation of China (42371466); the Key Research and Development Program of Henan Province (251111211700); the Key Research Projects of Henan Higher Education Institutions (23A520031, 24A520020, 25B520012); the Henan Provincial Archives Bureau Scientific and Technological Project (2025-Z-002); and the Science and Technology Plan Project of Housing and Urban-Rural Development in Henan Province (HNJS-2024-K35).

Author Contributions: The authors confirm contribution as follows: Conceptualization, Wei Jiang and Weichao Zhang; methodology, Weichao Zhang and Haochen Sun; software, Weichao Zhang and Haochen Sun; validation, Yao Meng and Jiayi Wu; formal analysis, Weichao Zhang; investigation, Haochen Sun and Jiayi Wu; resources, Wei Jiang; data curation, Haochen Sun; writing—original draft preparation, Weichao Zhang; writing—review and editing, Wei Jiang, Jiayi Wu and Yao Meng; visualization, Weichao Zhang and Yao Meng; supervision, Wei Jiang; project administration, Wei Jiang; funding acquisition, Wei Jiang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data available on request from the authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Salton G, McGill MJ. Introduction to modern information retrieval. New York, NY, USA: McGraw-Hill; 1983.
2. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. J Am Soc Inf Sci. 1990;41(6):391–407.
3. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019 Nov 3–7; Hong Kong, China. p. 3982–92.
4. Nogueira R, Cho K. Passage re-ranking with BERT. arXiv:1901.04085. 2019.
5. Baek I, Lee K, Lee S, Yu H. Crafting the path: robust query rewriting for information retrieval. arXiv:2407.12529. 2024.
6. Zeng Q, Yuan S, Lee J, Lee J. Unsupervised text representation learning via instruction-tuning for zero-shot dense retrieval. arXiv:2409.16497. 2024.

7. Cuconasu F, Trappolini G, Siciliano F, Filice S, Campagnano C, Maarek Y, et al. The power of noise: redefining retrieval for RAG systems. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2024 Jul 14–18; Washington, DC, USA. New York, NY, USA: ACM; 2024. p. 719–29.
8. Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers); 2019 Jun 2–7; Minneapolis, MN, USA. p. 4171–86. doi:10.18653/v1/N19-1423.
9. Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, et al. Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov 16–20; Online. p. 6769–81. doi:10.18653/v1/2020.emnlp-main.550.
10. Xiong L, Xiong C, Li Y, Tang KF, Liu JL, Bennett PN, et al. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: Proceedings of the 2021 International Conference on Learning Representations (ICLR); 2021 May 4; Vienna, Austria.
11. Hofstätter S, Lin SC, Yang JH, Lin J, Hanbury A. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; Online. p. 113–22. doi:10.1145/3404835.3462891.
12. Gao L, Callan J. Condenser: a pre-training architecture for dense retrieval. arXiv:2106.00261. 2021.
13. Ni J, Gao L, Callan J, Lin J. Large dual encoders are generalizable retrievers. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing; 2022 Nov 16–20; Abu Dhabi, United Arab Emirates. p. 9844–55. doi:10.18653/v1/2022.emnlp-main.669.
14. Johnson J, Douze M, Jégou H. Billion-scale similarity search with GPUs. arXiv:1702.08734. 2017.
15. Izacard G, Caron M, Hosseini L, Riedel S, Bojanowski P, Joulin A, et al. Unsupervised dense information retrieval with contrastive learning. arXiv:2112.09118. 2022. doi:10.48550/arXiv.2112.09118.
16. Gao L, Ma X, Lin J, Callan J. Precise zero-shot dense retrieval without relevance labels. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2023 Jul 3–8; Toronto, ON, Canada. p. 1762–77. doi:10.18653/v1/2023.acl-long.99.
17. Xu L, Liu B, Zhang Y, Xu C, Tang J. LaPraDoR: a unified paradigm for pre-training dense retrievers via lexical query expansion and contrastive learning. arXiv:2203.06169. 2022. doi:10.48550/arXiv.2203.06169.
18. Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems; 2020 Dec 6–12; Vancouver, BC, Canada. p. 9459–74.
19. Izacard G, Grave E. 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; 2021 Apr 19–23; Online. p. 870–80.
20. Shuster K, Xu D, Ju D, Roller S, Madotto A, Welleck S, et al. Retrieval augmentation reduces hallucination and improves safety in open-domain dialogue. In: Proceedings of the 2021 Findings of the Association for Computational Linguistics; 2021 Nov 7–11; Punta Cana, Dominican Republic. p. 3784–803.
21. Chuang Y-S, Fang W, Li S-W, Yih W-T, Glass J. Expand, rerank, and retrieve: query reranking for open-domain question answering. In: Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023; 2023 Jul 9–14; Toronto, ON, Canada. p. 12131–47. doi:10.18653/v1/2023.findings-acl.768.
22. Asai A, Schick T, Lewis P, Chen X, Izacard G, Riedel S, et al. Task-aware retrieval with instructions. arXiv:2211.09260. 2022.
23. Mao Y, He P, Liu X, Shen Y, Gao J, Han J, et al. Generation-augmented retrieval for open-domain question answering. arXiv:2009.08553. 2020.
24. Wang L, Yang N, Wei F. Query2Doc: query expansion with large language models. arXiv:2303.07678. 2023.
25. Shen T, Long G, Geng X, Tao C, Zhou T, Jiang D. Large language models are strong zero-shot retriever. arXiv:2304.14233. 2023.

26. Lei Y, Cao Y, Zhou T, Shen T, Yates A. Corpus-steered query expansion with large language models. arXiv:2402.18031. 2024.
27. Feng J, Tao C, Geng X, Shen T, Xu C, Long G, et al. Synergistic interplay between search and large language models for information retrieval. arXiv:2305.07402. 2023.
28. Craswell N, Mitra B, Yilmaz E, Campos D, Voorhees EM. Overview of the TREC 2019 deep learning track. arXiv:2003.07820. 2020.
29. Craswell N, Mitra B, Yilmaz E, Campos D. Overview of the TREC 2020 deep learning track. arXiv:2102.07662. 2021.
30. Thakur N, Reimers N, Rücklé A, Srivastava A, Gurevych I. BEIR: a heterogeneous benchmark for zero-shot evaluation of information retrieval models. arXiv:2104.08663. 2021.
31. Robertson SE, Zaragoza H. The probabilistic relevance framework: BM25 and beyond. Found Trends Inf Retr. 2009;3:333–89. doi:10.1561/1500000019.
32. Bajaj P, Campos D, Craswell N, Deng L, Gao J, Liu X, et al. MS MARCO: a human generated machine reading comprehension dataset. arXiv:1611.09268. 2016.
33. Dai Z, Callan J. Context-aware sentence/passage term importance estimation for first stage retrieval. arXiv:1910.10687. 2019.
34. Rahmani HA, Yilmaz E, Craswell N, Mitra B. LLMJudge: LLMs for relevance judgments. arXiv:2408.08896. 2024.
35. Yan L, Qin Z, Zhuang H, Jagerman R, Wang X, Bendersky M, et al. Consolidating ranking and relevance predictions of large language models through post-processing. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; 2024 Nov 12–16; Miami, FL, USA. p. 410–23. doi:10.18653/v1/2024.emnlp-main.25.