Scientific Paper Retrieval with LLM-Guided Semantic-Based Ranking

Yunyi Zhang[1], Ruozhen Yang[1], Siqi Jiao[1], SeongKu Kang[2], Jiawei Han[1]
[1]University of Illinois Urbana-Champaign [2]Korea University
`{yzhan238,ruozhen2,sjiao2,hanj}@illinois.edu` `seongkukang@korea.ac.kr`

Abstract

Scientific paper retrieval is essential for supporting literature discovery and research. While dense retrieval methods demonstrate effectiveness in general-purpose tasks, they often fail to capture fine-grained scientific concepts that are essential for accurate understanding of scientific queries. Recent studies also use large language models (LLMs) for query understanding; however, these methods often lack grounding in corpus-specific knowledge and may generate unreliable or unfaithful content. To overcome these limitations, we propose SemRank, an effective and efficient paper retrieval framework that combines LLM-guided query understanding with a concept-based semantic index. Each paper is indexed offline using multi-granular scientific concepts, including general research topics and detailed key phrases. At query time, an LLM identifies core concepts derived from the corpus to explicitly capture the query's information need. These identified concepts enable precise semantic matching, significantly enhancing retrieval accuracy. Experiments show that SemRank consistently improves the performance of various base retrievers, surpasses strong LLM-based baselines, and remains highly efficient.[1]

1 Introduction

Scientific paper retrieval is a crucial task to facilitate literature discovery and accelerate scientific progress (Kang et al., 2024b). Unlike general-purpose information retrieval, scientific paper retrieval is more challenging because queries often involve theme-specific intent and specialized terminology. In addition, acquiring labeled query-passage pairs for supervised fine-tuning is costly and requires domain expertise, making it impractical to continuously annotate more data to adapt the fast-evolving scientific domains.

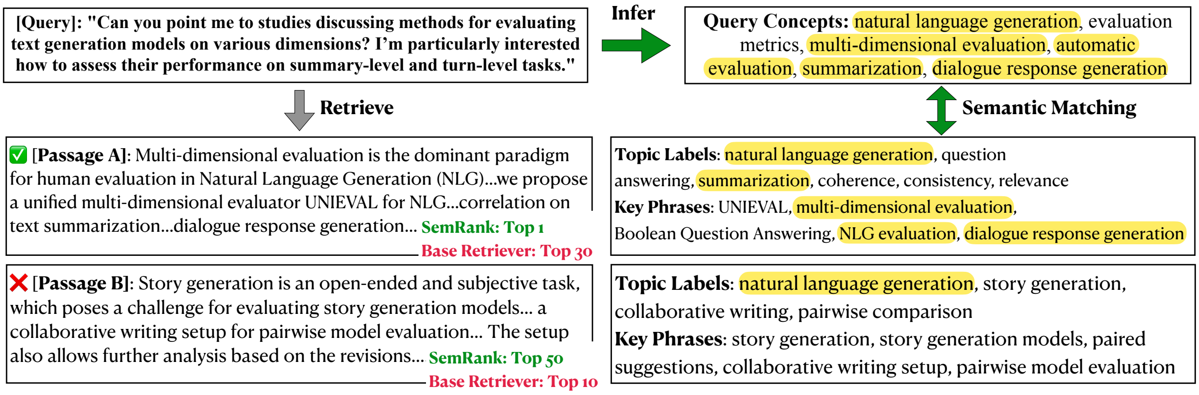
Recently, dense passage retrieval methods have been widely studied in various ad-hoc searches (Karpukhin et al., 2020; Izacard et al., 2021). These methods encode the overall semantics of queries and passages into the same vector space and measure relevance using vector similarity. Although being effective in different general domain applications, they still face challenges in scientific paper retrieval.

Specifically, general-purpose semantic representations learned by dense retrievers often fail to capture fine-grained scientific concepts that are crucial for accurately understanding and satisfying a scientific query (Chen et al., 2022; Shavarani and Sarkar, 2025). For example, the query "*Can you point me to studies discussing methods for evaluating text generation models on various dimensions?*" involves not only general topics like "natural language generation" and "automatic evaluation" that need to be inferred from the text, but also specific details like "multidimensional evaluation". A dense retriever, however, only encodes the text in a holistic view, while it lacks the ability and controllability to focus on the scientific concepts which are the core need of the query.

With the advancements of large language models (LLM) such as GPT (OpenAI, 2023) and Claude (Anthropic, 2024), recent studies also explore how to utilize LLMs in query understanding to help retrieval tasks. For example, HyDE (Gao et al., 2023) uses an LLM to generate a hypothetical passage for encoding, and CSQE (Lei et al., 2024) prompts an LLM to select a set of relevant sentences to expand the original query. However, these methods still rely on a pre-trained dense retriever to encode overall semantics on the document or sentence levels, lacking the ability to explicitly capture what the query is asking for. In addition, LLMs are not inherently retrieval models. They do not have the vast and dynamic knowledge in the scientific literature necessary to understand scientific queries.

[1] Code can be found at: `https://github.com/yzhan238/SemRank`.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2049–2060
November 4-9, 2025 ©2025 Association for Computational Linguistics



[Query]: "Can you point me to studies discussing methods for evaluating text generation models on various dimensions? I'm particularly interested how to assess their performance on summary-level and turn-level tasks."

Query Concepts: natural language generation, evaluation metrics, multi-dimensional evaluation, automatic evaluation, summarization, dialogue response generation

✅ [**Passage A**]: Multi-dimensional evaluation is the dominant paradigm for human evaluation in Natural Language Generation (NLG)...we propose a unified multi-dimensional evaluator UNIEVAL for NLG...correlation on text summarization...dialogue response generation...

Topic Labels: natural language generation, question answering, summarization, coherence, consistency, relevance
Key Phrases: UNIEVAL, multi-dimensional evaluation, Boolean Question Answering, NLG evaluation, dialogue response generation

❌ [**Passage B**]: Story generation is an open-ended and subjective task, which poses a challenge for evaluating story generation models... a collaborative writing setup for pairwise model evaluation... The setup also allows further analysis based on the revisions...

Topic Labels: natural language generation, story generation, collaborative writing, pairwise comparison
Key Phrases: story generation, story generation models, paired suggestions, collaborative writing setup, pairwise model evaluation

Figure 1: An illustrative example from LitSearch with SPECTER-v2 as base retriever. By capturing the scientific concepts for corpus and query, SemRank substantially improves the ranking results.

Therefore, when used in a zero-shot manner, LLMs cannot identify domain-specific terminology and may generate hallucinated content. Thus, how to effectively augment LLMs with corpus-based knowledge while capturing specific query information remains a challenge.

To overcome these limitations, we propose to utilize LLMs' text understanding ability in the scientific retrieval task with the help of a scientific concept-based semantic index. Unlike earlier studies which construct domain-specific semantic index at topic level (Tsatsaronis et al., 2015), we build concept-based semantic index at various granularities: from broad research topics such as "natural language generation" to specific terms such as "multidimensional evaluation metrics". These multi-granular concepts capture the essential content of the paper. Then, we use the semantic index to improve any existing retrievers with the help of LLMs. Specifically, we prompt an LLM to identify a set of core concepts to explicitly represent what the scientific query is asking for. We augment the prompt with candidate concepts derived from the corpus, which helps the LLM to reduce hallucination and ensure the generated content align with the semantic index of corpus. Figure 1 shows an example that, by accurately capturing the multi-granular concepts of the corpus and query, we can improve the scientific paper retrieval results by focusing on the core need of the query.

We propose the SemRank, LLM-Guided **Sem**antic-Based **Rank**ing, a plug-and-play framework for scientific paper retrieval. First, during the indexing time, we build scientific concept-based semantic index for the corpus by identifying a set of research topics and key phrases for each paper. To ensure the topic labels in their canonical forms, we train an auxiliary topic classifier model to identify a set of candidate topics from a large label space[2]. Then, we prompt an LLM to select the core topic labels and extract key phrases to build the semantic index. Then given a query during retrieval, we first construct a set of candidate concepts from the corpus using a base retriever, from which an LLM can be augmented with corpus-based knowledge and identify a set of core concepts for the query. These concepts, serving as an explicit guidance on what the query is asking for, will be used for concept-level semantic matching to improve the retrieval results.

Our method can be easily integrated with any dense retriever and improve their retrieval quality without relying on any annotated query data. Experiments show that our method is both effective and efficient. During retrieval time, SemRank only needs one LLM prompting per query, no additional call to the base retriever, and all computation can be done easily on CPUs. Yet it can significantly improve the ranking performance of a wide range of base retrievers and outperforms various LLM-based baselines.

The contributions of this paper are as follows:

- We propose SemRank, a plug-and-play framework for scientific paper retrieval, which utilizes concept-based semantic index and LLM guidance to explicit capture the information need for scientific queries.
- We develop a light-weighted method which augments an LLM with semantic index of corpus to accurately identify a set of core scientific concepts for the query, which are then used to improve the retrieval performance through concept-level semantic matching.
- Through extensive experiments, we show that

[2] An academic label space is widely available such as ACM CCS and Microsoft Academic Graph (Sinha et al., 2015)

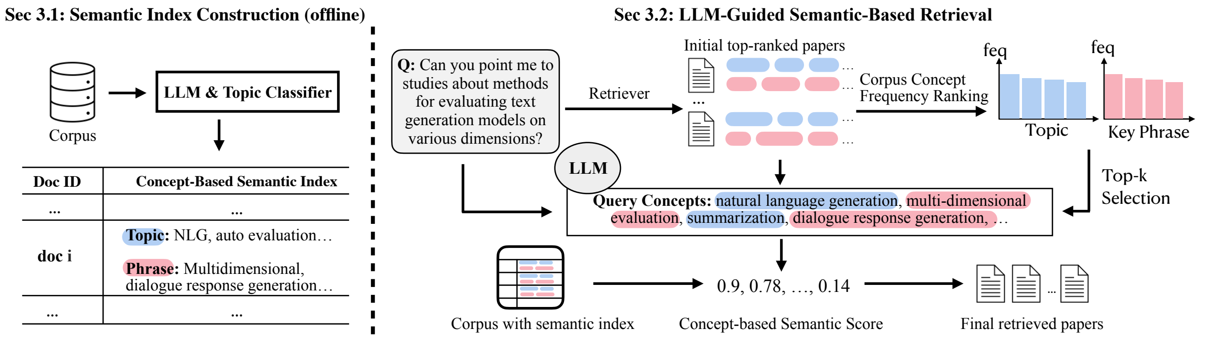

Sec 3.1: Semantic Index Construction (offline)
Corpus
LLM & Topic Classifier
Doc ID
Concept-Based Semantic Index
...
...
doc i
Topic: NLG, auto evaluation…
Phrase: Multidimensional, dialogue response generation…
...
...
Sec 3.2: LLM-Guided Semantic-Based Retrieval
Q: Can you point me to studies about methods for evaluating text generation models on various dimensions?
Retriever
Initial top-ranked papers
Corpus Concept Frequency Ranking
feq
feq
Topic
Key Phrase
Top-k Selection
LLM
Query Concepts: natural language generation, multi-dimensional evaluation, summarization, dialogue response generation, …
Corpus with semantic index
0.9, 0.78, …, 0.14
Concept-based Semantic Score
Final retrieved papers

Figure 2: Overview of the SemRank framework.

SemRank consistently improves the base retrievers' performance and outperforms existing baselines while being more efficient.

2 Problem Formulation

Given a corpus $\mathcal{D}$ of scientific papers, our goal is to retrieve and rank the papers according to their relevance to a given query q. Specifically, we will build concept-based semantic index for the corpus by identifying a set of scientific concepts representing the core information of each paper. We assume a scientific topic label space $\mathcal{T}$, which is widely available and normally contains a large number of research topics. In this work, we use Microsoft Academic Graph (Sinha et al., 2015), which contains 13,613 Computer Science topics. Then, we utilize LLMs and the semantic index to improve the scientific paper retrieval performance.

3 Methodology

In this section, we will present our SemRank framework. We first introduce the offline semantic index construction module (Sect. 3.1), then we introduce the LLM-guided semantic-based ranking for scientific paper retrieval (Sect. 3.2). Figure 2 shows an overview of SemRank.

3.1 Semantic Index Construction

To capture the core research concepts at various granularities for scientific papers, we propose to build a semantic index of the corpus that contains general research topics and specific key phrases. The broad topics aim to cover the overall themes of the papers that are not explicitly mentioned such as "natural language generation" and "automatic evaluation," while the key phrases aim to capture the detailed information specific to the paper such as "multidimensional valuation metrics". Such a well-structured corpus foster flexible query matching at different granularities.

While directly prompting LLMs could be a viable solution to assign each document a set of topics and key phrases, it is hard to ensure the faithfulness of the LLM-generated content. Besides, the research topics are often not explicitly mentioned in the text and LLMs can generate topics at random granularity, which is less controllable and hard to match during retrieval. Therefore, we propose to first fine-tune a multi-label topic classifier for scientific papers with a domain-specific topic label space, from which we can obtain a set of candidate topics for each scientific paper.

Candidate Topic Prediction We first fine-tune a multi-label text classifier to estimate the likelihood of a paper belonging to a research topic. We use a simple log-bilinear text matching network as our model architecture (Zhang et al., 2025). We initialize the paper encoder with a pre-trained scientific domain language model (e.g., SPECTER-v2 (Singh et al., 2022)). We also get the topic embeddings using the same pre-trained model and detach them from the encoder. This ensures only the embeddings are updated without back-propagating to the backbone for saving cost. Then, the classifier predicts the probability of document d_i belonging to topic $t_j \in \mathcal{T}$ with log-bilinear matching:

$$p(t_j|d_i) = \sigma(\exp(\mathbf{t}_j^T \mathbf{W} \mathbf{d}_i)),$$

where σ is the sigmoid function, $\mathbf{W}$ is a learnable interaction matrix, and $\mathbf{t}_j$ and $\mathbf{d}_i$ are the encoded topic and document.

We fine-tune the multi-label topic classifier using the binary cross entropy loss. Given the positive topics of paper $d_i \in \mathcal{D}$ in the label space $\mathcal{T}_i \subset \mathcal{T}$, we train the classifier with:

$$\mathcal{L} = -\sum_{d_i \in \mathcal{D}} \Big(\sum_{t_j \in \mathcal{T}_i} \log p(t_j|d_i) + \alpha \sum_{t_j \notin \mathcal{T}_i} \log\left(1 - p(t_j|d_i)\right) \Big),$$

where the coefficient α is a small constant number to counter the imbalanced numbers of positive and negative labels in a large label space.

Core Concepts Identification With the fine-tuned scientific topic classifier, we first predict a set of candidate topics likely relevant to each paper in the retrieval corpus. Then, we prompt an LLM to perform two tasks: (1) select a set of topics from the candidate list that are not too broad or irrelevant, and (2) extract a set of key phrases from the paper. By providing a candidate topic list, we turn a generation task (of zero-shot prompting) to a selection and extraction task for LLMs, which reduces the chance of hallucination and also ensures the selected topics following a label space. Figure 4 shows our prompt.

By first predicting candidate topic labels with a text classifier, not only can we resolve the issue of LLMs' poor performance on a large structured label space (U et al., 2023; Zhang et al., 2025), but also reduce the cost of prompting LLMs by 98%. A more detailed cost analysis is in Section 4.4. Also note that we only need to train one topic classifier for a domain and use it for all corpora in the same domain (e.g., Computer Science).

In summary, for each scientific paper $d_i \in \mathcal{D}$, we identify a set of research topics belonging to a label space, denoted as $T_i \in \mathcal{T}$, and a set of key phrases extracted from it, denoted as P_i.

3.2 LLM-Guided Semantic-Based Retrieval

With the constructed semantic index capturing the core concepts discussed in the corpus, we now present how SemRank leverages it to enhance retrieval by explicitly modeling a query's information need. Given a base retriever, SemRank first identifies a set of candidate concepts relevant to the query and prompts an LLM to analyze the retrieval context and select the most salient core concepts. These core concepts are then used to refine the initial retrieval results, yielding a concept-aware ranking that better aligns with the query intent.

Candidate Concepts Construction for Query Directly prompting an LLM to assign a set of scientific concepts to a query is not optimal, because the generated content may not align with the semantic index structure of the corpus and thus make it difficult to match between the query and the papers. Therefore, we propose to first construct a set of candidate concepts for a query from the corpus.

Inspired by the idea of pseudo relevance feedback, given a base retriever s^{base} and its initial ranking results, we collect a set of topics and key phrases that are frequently mentioned in the top-ranked papers. Specifically, for a query q, we collect lists of top-k most frequent topics and key phrases mentioned by the top-ranked papers using their pre-constructed semantic index T_i and P_i, and denote the lists as $T^0(q)$ and $P^0(q)$. These concepts, being frequently mentioned by top-ranked papers, are likely relevant to the initial query.

LLM-Guided Core Concept Identification With the constructed candidate concepts of the query, we can now prompt an LLM to identify a set of core concepts for the query that can most likely identify its relevant papers. Given the topic and key phrases lists $T^0(q)$ and $P^0(q)$ as well as the top-k papers in the current ranking $D^0(q)$, we instruct an LLM to select a set of concepts from the candidate lists in order to improve the current ranking results. Figure 5 shows our prompt.

Because the candidate concepts are collected from the semantic index, they contribute to two advantages: (1) they serve as a high-level summary of the current retrieval results to help the LLM interpret the query and current results; (2) they provide a high-quality candidate list for the LLM, ensuring the faithfulness and reducing hallucination.

After prompting the LLM, we get a set of core concepts selected from the candidate lists, which we denote as $C(q)$. This set contains the most important concepts relevant to the original query and thus explicitly represents the information need of the query. Also, the concepts are in different granularities, proper for matching at the needed level of details. For example, for the query shown in Fig 1, the LLM identifies general topics like “natural language generation” and “automatic evaluation”, and specific terms like “multidimensional evaluation” and “dialogue response generation”.

Core Concept-Based Ranking Finally, with the identified set of core concepts of the query, we can use the semantic index of the corpus to re-evaluate the ranking. Specifically, given the core concepts of the query $C(q)$ and the the concepts of a paper $C_i = T_i \cup P_i$, we calculate their similarity with a multi-vector similarity matching score.

$$s^{sem}(q, d_i) = \frac{1}{|C(q)|} \sum_{c \in C(q)} \max_{c' \in C_i} sim(\mathbf{c}, \mathbf{c}'),$$

Table 1: Datasets overview.

Dataset	corpus size	# test query	doc/query
CSFCube	4,207	34	13.32
DORISMAE	8,482	90	19.49
LitSearch	64,183	597	1.07

where $\mathbf{c}$ denotes the embedding of a concept by a semantic encoder (e.g., SPECTER-v2) and $sim()$ represents the cosine similarity function. This embedding-based soft matching process identifies the most similar concept in a paper for each query's concept, which accounts for the situation where similar concepts are expressed slightly differently (e.g., "hallucination" and "hallucinated content"). Then, we combine this semantic-based score with the base retriever's score by z-score normalization (denoted by $z(\cdot)$), based on which we can re-rank the papers to get a new ranked list.

$$s(q, d_i) = z(s^{base}(q, d_i)) + z(s^{sem}(q, d_i)).$$

Efficiency of SemRank Retrieval Given the ranking results of a base retriever, our LLM-guided semantic-based ranking process is highly efficient. First, it only requires one LLM call, and the output length is minimal because it is highly-structured with a list of scientific concepts. Second, because all the query's concepts are selected from the semantic index and all concept embeddings can be pre-computed offline, we only need the cosine similarity computation (i.e., dot product with normalized vectors) during retrieval, which is highly efficient and can be done on CPUs. Besides, the pairwise similarities between concepts can also be pre-computed for the maximal inference efficiency, but it requires substantial amount of storage, so we opt to compute the similarity during retrieval. We conduct a detailed efficiency analysis in Sect. 4.4.

4 Experiments

4.1 Experiment Setup

Datasets We use three public datasets on scientific paper retrieval: **CSFCube** (Mysore et al., 2021), **DORISMAE** (Wang et al., 2023a), and **LitSearch** (Ajith et al., 2024), including both human-annotated and LLM-generated relevance labels. We use the processed version of CSFCube and DORISMAE released by Kang et al. (2024b)[3] and LitSearch from its official github[4]. Table 1 summarizes the overall statistics of the datasets.

Base Retrievers We use a wide range of base retrievers in our experiments. We include a sparse retriever **BM25**, an unsupervised dense retriever **SPECTER-v2** (Singh et al., 2022), and the **Hybrid** of these two. We also include two instruction-tuned dense retrievers, **E5-large-v2** (Wang et al., 2022) and **GritLM-7B** (Muennighoff et al., 2025)[5].

Baselines We compare SemRank with a collection of retrieval methods using corpus knowledge and/or LLMs to enhance base retrievers.

- **Boudin et al.** (Boudin et al., 2020) uses a seq2seq keyphrase generation model to enrich the corpus indexing.
- **BERT-QE** (Zheng et al., 2020) expands the query with relevant text chunks selected from top-ranked papers returned by the base retriever.
- **ToTER** (Kang et al., 2024a) improves the retrieval performance with a topical taxonomy by comparing the topic distributions of the query and documents predicted by a text classifier.
- **HyDE** (Gao et al., 2023) prompts an LLM to generate hypothetical document that answers the query and encode it as the query vector.
- **GRF** (Mackie et al., 2023) generates relevant context by an LLM. We choose to generate scientific concepts for fair comparison.
- **CSQE** (Lei et al., 2024) uses an LLM to select relevant sentences from top-ranked documents of the base retriever, which are then used to expand the query together with a hypothetical document.

Evaluation Metrics We use Recall@K (R@K) as our evaluation metric. Following previous studies, we use $K = 50, 100$ for CSFCube and DORISMAE, and $K = 5, 20, 100$ for LitSearch.

Implementation Details For building semantic index, we train our text classifier using MAPLE (Zhang et al., 2023b) which contains topic labels from Microsoft Academic Graph (Sinha et al., 2015). We initialize it with `SPECTER-v2-base` and fine-tune it with learning rate at 5e-5 for 10 epochs. The balancing factor α is set to 1e-2. We also use `SPECTER-v2-base` for concept encoding. During retrieval, the number of candidate topics and key phrases is $k = 50$. We use `GPT-4.1-mini` as the LLM for SemRank and all LLM-based baselines for fair comparison. The

[3] https://aclanthology.org/attachments/2024.emnlp-main.407.data.zip

[4] https://github.com/princeton-nlp/LitSearch

[5] We also experiment with other retrievers such as NV-Embed-v2 (Lee et al., 2025). GritLM achieves the strongest performance on scientific paper retrieval, potentially because it specifically includes scientific corpora in its training data.

Table 2: Performance of baselines in Recall@K, with the best score **boldfaced** and the second best underlined.

Base	Methods	CSFCube		DORISMAE		LitSearch		
		R@50	R@100	R@50	R@100	R@5	R@20	R@100
SPECTER-v2	Retriever	0.5331	0.6860	0.5305	0.7208	0.3931	0.5551	0.7205
	Boudin et al.	0.5088	0.6739	0.4695	0.6040	0.4345	0.5671	0.7250
	BERT-QE	0.5243	0.6689	0.5284	0.6942	0.3959	0.5551	0.7364
	ToTER	0.5443	0.7131	0.5319	0.7234	0.3948	0.5568	0.7239
	HyDE	0.5879	0.7473	0.5110	0.6789	0.4241	0.5923	0.7682
	GRF	0.5599	0.6758	0.5442	0.7283	0.4319	0.5830	0.7604
	CSQE	0.5586	0.7149	0.5022	0.6491	0.4366	0.5747	0.7223
	SemRank	**0.6222**	**0.7601**	**0.5894**	**0.7451**	**0.5028**	**0.6316**	**0.7746**
E5-large-v2	Retriever	0.6111	0.7362	0.5548	0.7162	0.5137	0.6573	0.7765
	BERT-QE	0.6399	0.7589	0.5943	0.7451	0.4906	0.6361	0.7881
	ToTER	0.6134	0.7553	0.5596	0.7120	0.4981	0.6627	0.7951
	HyDE	0.6203	0.7255	0.5559	0.7236	0.4854	0.6592	0.8149
	GRF	0.6183	0.7797	0.6025	0.7501	0.5347	0.6984	**0.8319**
	CSQE	0.6416	0.7549	0.4787	0.5977	0.5503	0.6389	0.7719
	SemRank	**0.6661**	**0.8177**	**0.6286**	**0.7754**	**0.5807**	**0.7042**	0.8312

Table 3: Performance of SemRank on three datasets with different base retrievers, with the best score **boldfaced**.

Methods	CSFCube		DORISMAE		LitSearch		
	R@50	R@100	R@50	R@100	R@5	R@20	R@100
SPECTER-v2	0.5331	0.6860	0.5305	0.7208	0.3931	0.5551	0.7205
+ SemRank	**0.6222**	**0.7601**	**0.5894**	**0.7451**	**0.5028**	**0.6316**	**0.7746**
BM25	0.4651	0.5966	0.5721	0.7441	0.4381	0.5794	0.7362
+ SemRank	**0.5840**	**0.7076**	**0.6507**	**0.8104**	**0.5126**	**0.6612**	**0.7920**
Hybrid	0.5855	0.7131	0.6743	0.8297	0.5397	0.6877	0.7881
+ SemRank	**0.6434**	**0.7673**	**0.6779**	**0.8358**	**0.5835**	**0.7161**	**0.7922**
E5-large-v2	0.6111	0.7362	0.5548	0.7162	0.5137	0.6573	0.7765
+ SemRank	**0.6661**	**0.8177**	**0.6286**	**0.7754**	**0.5807**	**0.7042**	**0.8312**
GritLM-7B	0.6732	0.7742	0.6415	0.8037	0.6908	0.8001	0.9149
+ SemRank	**0.7290**	**0.8466**	**0.6743**	**0.8187**	**0.6955**	**0.8171**	**0.9221**

experiments are run on one NVIDIA RTX A6000 when a GPU is needed.

4.2 Retrieval Performance Comparison

Table 2 shows the results of compared baselines with two retrievers, SPECTER-v2 and E5-large. We clearly see that SemRank overall outperforms the compared baselines on all datasets. Specifically, we observe that methods using LLMs for query understanding achieves better performance, showing the strength of text understanding ability of LLMs in the retrieval task. Comparing with previous methods, SemRank uses the LLM for concept-based query understanding and matching, which captures the query's need more explicitly and thus achieves stronger performance.

Table 3 additionally shows the results of SemRank on different base retrievers. We can see that SemRank consistently improves the performance of all kinds of retrievers. Even for GritLM-7B, the SOTA model reported in Ajith et al. (2024) and trained with scientific data, SemRank still improves its performance on all datasets. Besides, SemRank also improves the performance of Hybrid model, showing that concept-level semantic matching is not a combination of typical sparse and dense features, but in another intermediate level that really captures scientific knowledge.

4.3 Ablation Studies

We conduct ablation studies to show the effectiveness of each component of SemRank. We include the following ablated versions:

- **No Topic**: excludes topics in semantic index.

Table 4: Ablation studies of SemRank on LitSearch.

	R@5	R@20	R@100
SPECTER-v2	0.3931	0.5551	0.7205
Indexing			
No Topic	0.4331	0.6040	0.7687
No Phrase	0.4160	0.5682	0.7447
Retrieval			
No Corpus	0.4060	0.5557	0.7260
No LLM (class)	0.4079	0.5459	0.7320
No LLM (freq)	0.3897	0.5497	0.7267
SemRank	**0.5028**	**0.6316**	**0.7746**

- **No Phrase**: excludes phrases in semantic index.
- **No Corpus**: excludes the candidate scientific concepts from the corpus and thus prompts the LLM to directly generate scientific concepts based on its own knowledge.
- **No LLM (class)**: excludes the LLM and selects scientific concepts for the query using the fine-tuned topic classifier.
- **No LLM (freq)**: excludes the LLM and uses the top-20 most frequent concepts mentioned by top-ranked papers returned by the base retriever.

Table 4 shows the results of ablation studies on the LitSearch dataset with SPECTER-v2 as the base retriever. First, we observe that either ablating topics or phrases from the semantic index will degrade the performance, with phrases affecting more because of its capturing more detailed information. Second, removing the augmented corpus knowledge from LLM prompting will greatly affect the final performance, because LLMs tend to generate terms not matched with the corpus. Finally, removing LLM but using topic classifier or statistic-based metric for query concept identification also drastically decreases the performance, showing the power of LLM on query understanding when augmented with corpus knowledge.

4.4 Efficiency Analysis

We show the efficiency of SemRank by comparing it with other LLM-based baselines. Specifically, we report the following factors of each method: the number of base retriever calls per query (**# RET**), the number of LLM calls per query (**# LLM**), the average number of tokens generated by the LLM per query (**LLM Output Len**), and the average running time per query (**Running Time**). As stated in Sect. 4.1, we use the same LLM checkpoint for all baselines for a fair comparison. Table 5 shows the detailed comparison results on LitSearch. We can clearly see that SemRank takes the least inference time among the compared methods, with 1.5× faster than the second fast method. Not only does SemRank call the retriever and LLM only once, it also expects minimal number of tokens responded by the LLM because of the concept-based structured input and output format.

Table 5: Efficiency analysis of LLM-based methods.

	# RET	# LLM	LLM Output Len	Running Time
HyDE	**1**	**1**	169.39 tok	4.02 sec
GRF	2	**1**	79.72 tok	2.78 sec
CSQE	2	2	462.21 tok	11.27 sec
SemRank	**1**	**1**	**18.92 tok**	**1.82 sec**

Additionally, the offline indexing part of SemRank is efficient as well. For semantic index construction on CSFCube, to get the candidate topics, the text classifier inference time is 52 seconds, and to prompt LLM for topic and keyphrase selection, it takes 84 seconds and $1.74 dollars. In comparison, directly prompting LLMs to build such semantic index on CSFCube will take approximately 42 minutes and $85 dollars.

4.5 Combination with LLM-based Reranking

Recent studies also use LLM for reranking retrieval results by prompting it to provide a new ranked list of top documents. To show that SemRank can naturally integrate with such a method, we compare the performance of LLM-based reranking with and without SemRank. Following (Ajith et al., 2024), we provide top-100 papers to the LLM and use the prompt from Sun et al. (2023). Table 6 shows the results on LitSearch with 3 base retrievers. We can see SemRank does not conflict with LLM-based reranking by consistently improving its performance. While typical reranking only improves recall within provided number of papers (R@100 unchanged), SemRank can brings more relevant papers and also improves R@100.

4.6 Parameter Studies

We study the influence of setting different number of candidates provided to LLM for query concept identification, i.e., k in Sect. 3.2 for prompt in Figure 5. We set the value of $k = 5, 10, 25, 50, 75, 100$ and report their performance on LitSearch with base retriever SPECTER-v2 and GritLM. Results in Figure 3 show that SemRank overall is not very sensitive to the value of k for $k \geq 25$. We notice slightly increased performance for smaller k in the R@5 measures, which shows that an incomplete query concept set may not be sufficient to affect top-ranked documents that are hard to distinguish.

Table 6: Further analysis of SemRank by combining with LLM-based reranking.

	R@5	R@20	R@100
SPECTER-v2	0.3931	0.5551	0.7205
+ Reranking	0.6636	0.7038	0.7205
+ SemRank	**0.6705**	**0.7435**	**0.7746**
E5-large-v2	0.5137	0.6573	0.7765
+ Reranking	0.6989	0.7480	0.7765
+ SemRank	**0.7108**	**0.7963**	**0.8312**
GritLM-7B	0.6908	0.8001	0.9149
+ Reranking	0.7575	0.8470	0.9149
+ SemRank	**0.7774**	**0.8520**	**0.9221**

5 Related Works

Dense Retrieval Dense retrieval has become a core paradigm in modern information retrieval. Early models like Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) and ME-BERT (Luan et al., 2021) leveraged in-batch and BM25-based hard negatives to improve training efficiency. Subsequent methods refined negative sampling: ANCE (Yu et al., 2021) used asynchronously updated indices; RocketQA (Qu et al., 2021) introduced cross-batch and denoised negatives; and ADORE (Zhan et al., 2021) adopted dynamic sampling for greater stability. PAIR (Ren et al., 2021) incorporated passage-level similarity signals, while FiD-KD (Izacard and Grave, 2021) distilled knowledge from reader models.

In academic domains, specialized models like SciBERT (Beltagy et al., 2019) pre-trained on scientific texts laid the groundwork. Parisot and Zavrel (2022) proposes a multi-objective approach that uses general-domain document relevance and scientific domain citation network and self-supervised data. Mandikal and Mooney (2024) presents a hybrid approach that combines sparse and dense retrievers with a weighting parameter. MixGR (Cai et al., 2024) matches queries and documents additionally at subquery and proposition levels and merge them with rank fusion. Recent innovations also focus on knowledge distillation from cross-encoder rankers (Huang and Chen, 2024; Tao et al., 2024; Zhang et al., 2023a). Despite the progress and varied strategies for improving dense retrieval, these methods inherently rely on representing entire documents or queries with single dense embeddings, restraining them to capture fine-grained details crucial for accurately interpreting complex scientific queries. In contrast, our approach explicitly models the multi-granular scientific concepts within both queries and documents.

LLM-Enhanced Retrieval Recent works show that LLMs can boost retrieval quality even when little or no human supervision is available. In zero-shot settings, HyDE (Gao et al., 2023) prompts an instruction-tuned LLM (e.g., InstructGPT) to imagine a hypothetical answer document, then encodes this synthetic text with an unsupervised dual-encoder; the dense representation guides nearest-neighbor search and already outperforms Contriever (Izacard et al., 2021) without any task data. At inference time, LLMs refine the query itself. Rewrite-Retrieve-Read pipeline (Ma et al., 2023) trains a lightweight rewriter with RL from the downstream reader LLM, consistently topping standard retrieve-then-read QA. For expansion, HyDE's hypothetical-document trick underpins GenRead (Yu et al., 2023), which sometimes matches or beats retrieval-based pipelines by generating context first. CSQE (Lei et al., 2024) tempers hallucinations by mixing LLM expansions with sentences extracted from top-ranked corpus hits, outperforming fine-tuned neural expanders on tough TREC queries. CCQGen (Kang et al., 2025) leverages an LLM to select topical concepts derived from a predefined top-down taxonomy. In our work, SemRank leverages an LLM to select core concepts directly from candidate sets of both broad research topics and specific key phrases derived from the corpus, which augment LLMs with multi-granular knowledge to help query understanding and reduce hallucination.

Retrieval with Corpus Knowledge Recent work has explored leveraging corpus-based knowledge to enhance retrieval accuracy through query expansion and refinement techniques. Methods such as BERT-QE (Zheng et al., 2020) select corpus-contextualized text chunks to alleviate vocabulary mismatches, while GAR (Mao et al., 2021) generates pseudo-passages for semantic enrichment of queries, demonstrating substantial gains. Query2doc (Wang et al., 2023b) further utilizes LLMs to create entire pseudo-documents, capturing external knowledge from web-scale training corpora. Similarly, classical pseudo-relevance feedback (PRF) (Wang et al., 2021; Lei et al., 2024) has been adapted to dense retrieval, reducing hallucinations and improving effectiveness. Graph-based methods such as (MacAvaney et al., 2022; Kulkarni et al., 2023) utilize document similarity graphs to dynamically expand search results. ToTER (Kang et al., 2024a) designs taxonomy-based retrieval to

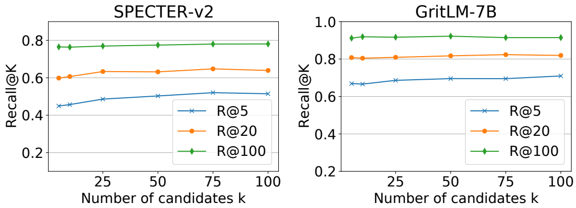


Figure 3: Parameter analysis on LitSearch by varying k, the number of candidate query concepts.

identify the central topic classes and exploit their topical relatedness to supplement PLM-based retrievers. TaxoIndex (Kang et al., 2024b) constructs a semantic index guided by an academic taxonomy, extracting and organizing concepts from documents, and then trains an indexing module to match these concepts with queries. In contrast, our proposed SemRank framework avoids supervision, utilizing corpus-derived concepts in an unsupervised manner to build semantic index and perform pseudo-relevance feedback, thereby enhancing retrieval without additional training.

6 Conclusion

We present SemRank, a novel scientific paper retrieval method that integrates LLM-guided query understanding with a concept-based semantic index. To overcome the limitations of existing methods, SemRank identifies multi-granular scientific concepts to explicitly understand scientific queries at the concept level. By augmenting LLMs with corpus knowledge, SemRank also facilitates LLM's understanding of query and context while reducing hallucination. Experiments demonstrate that SemRank consistently improves retrieval performance across various base retrievers and outperforms various baseline methods while remaining highly efficient.

Limitations

While SemRank shows strong performance and efficiency in scientific paper retrieval, it also has several limitations. First, our current studies limit to scientific paper retrieval dataset with only title and abstract, while retrieving full scientific papers could be more challenging due to the difficulties of effectively understanding long structured text. Second, SemRank only considers scientific concepts as a set, while not considering their internal relationship which could bring more insights to paper and query understanding. Third, although our use of LLMs is efficient, the reliance on prompting still introduces sensitivity to prompt design and model behavior, which may require tuning for different domains. Fourth, our experiments are done mainly on Computer Science domain corpora, because there is limited high-quality retrieval dataset available from other disciplines. We would like to argue that there is still a big gap in constructing paper retrieval benchmarks for different disciplines, which could be a research opportunity for future studies. Finally, we only focus on English scientific paper retrieval in the work, while it remains a challenge on multilingual or multi-modal (e.g., figures, tables) paper retrieval.

Acknowledgments

Research was supported in part by National Science Foundation IIS-19-56151, NSF IIS 25-37827, the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under BRIES Program No. HR0011-24-3-0325. The research has used the Delta/DeltaAI advanced computing and data resource, supported in part by the University of Illinois Urbana-Champaign and through allocation #250851 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants OAC 2320345, #2138259, #2138286, #2138307, #2137603, and #2138296. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

References

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. LitSearch: A retrieval benchmark for scientific literature search. In *EMNLP*.

Anthropic. 2024. Introducing the next generation of claude.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *EMNLP-IJCNLP*.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase generation for scientific document retrieval. In *ACL*.

Fengyu Cai, Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, Iryna Gurevych, and Heinz Koeppl. 2024. MixGR: Enhancing retriever generalization for scientific domain through complementary granularity. In *EMNLP*.

Xilun Chen, Kushal Lakhotia, Barlas Oguz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2022. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? In *Findings of EMNLP*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *ACL*.

Chao-Wei Huang and Yun-Nung Chen. 2024. PairDistill: Pairwise relevance distillation for dense retrieval. In *EMNLP*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. In *ICLR*.

SeongKu Kang, Shivam Agarwal, Bowen Jin, Dongha Lee, Hwanjo Yu, and Jiawei Han. 2024a. Improving retrieval in theme-specific applications using a corpus topical taxonomy. In *WWW*.

SeongKu Kang, Bowen Jin, Wonbin Kweon, Yu Zhang, Dongha Lee, Jiawei Han, and Hwanjo Yu. 2025. Improving scientific document retrieval with concept coverage-based query set generation. In *WSDM*.

SeongKu Kang, Yunyi Zhang, Pengcheng Jiang, Dongha Lee, Jiawei Han, and Hwanjo Yu. 2024b. Taxonomy-guided semantic indexing for academic paper search. In *EMNLP*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Hrishikesh Kulkarni, Sean MacAvaney, Nazli Goharian, and Ophir Frieder. 2023. Lexically-accelerated dense retrieval. In *SIGIR*.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *ICLR*.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *EACL*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *TACL*, 9:329–345.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *EMNLP*.

Sean MacAvaney, Nicola Tonellotto, and Craig Macdonald. 2022. Adaptive re-ranking with a corpus graph. In *CIKM*.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *SIGIR*.

Priyanka Mandikal and Raymond Mooney. 2024. Sparse meets dense: A hybrid approach to enhance scientific document retrieval. In *SDU@AAAI*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval for open-domain question answering. In *ACL-IJCNLP*.

Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative representational instruction tuning. In *ICLR*.

Sheshera Mysore, Tim O'Gorman, Andrew McCallum, and Hamed Zamani. 2021. CSFCube - a test collection of computer science research articles for faceted query by example. In *NeurIPS Datasets and Benchmarks Track*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mathias Parisot and Jakub Zavrel. 2022. Multi-objective representation learning for scientific document retrieval. In *SDP@COLING*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *NAACL*.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of ACL*.

Hassan Shavarani and Anoop Sarkar. 2025. Entity retrieval for answering entity-centric questions. In *KnowledgeNLP@NAACL*.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2022. Scirepeval: A multi-format benchmark for scientific document representations. In *EMNLP*.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *WWW*.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *EMNLP*.

Chongyang Tao, Chang Liu, Tao Shen, Can Xu, Xiubo Geng, Binxing Jiao, and Daxin Jiang. 2024. ADAM: Dense retrieval distillation with adaptive dark examples. In *Findings of ACL*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artiéres, Axel-Cyrille Ngonga Ngomo, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, and 3 others. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16.

Simon Chi Lok U, Jie He, Víctor Gutiérrez-Basulto, and Jeff Z. Pan. 2023. Instances and labels: Hierarchy-aware joint supervised contrastive learning for hierarchical multi-label text classification. In *Findings of EMNLP*.

Jianyou Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. 2023a. Scientific document retrieval using multi-level aspect-based queries. In *NeurIPS Datasets and Benchmarks Track*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, and Furu Wei. 2023b. Query2doc: Query expansion with large language models. In *EMNLP*.

Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2021. Pseudo-relevance feedback for multiple representation dense retrieval. In *SIGIR*.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. In *CIKM*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *ICLR*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR*.

Hang Zhang, Yeyun Gong, Xingwei He, Dayiheng Liu, Daya Guo, Jiancheng Lv, and Jian Guo. 2023a. Noisy pair corrector for dense retrieval. In *Findings of EMNLP*.

Yu Zhang, Bowen Jin, Qi Zhu, Yu Meng, and Jiawei Han. 2023b. The effect of metadata on scientific literature tagging: A cross-field cross-model study. In *WWW*.

Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Rui Li, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. 2025. TELEClass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In *WWW*.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: Contextualized Query Expansion for Document Re-ranking. In *EMNLP*.

Query Core Concept Identification Prompt

You will receive a paper abstract along with a set of candidate topics for the paper.
Your first task is to select the topics that best align with the core theme of the paper. Exclude topics that are too broad or less relevant.
Only use the topic names in the candidate set.
Your second task is to generate a complete list of key phrases extracted from the paper.
Do some rationalization before outputting the list of relevant topics and key phrases.

Output format: ‘<top> topic 1, topic 2, ... </top> <kp>key phrase 1, key phrase 2, ... </kp>’.

Paper: $\{d\}$

Figure 4: Prompts given to the LLM for building semantic index.

Query Core Concept Identification Prompt

You will receive a query for research papers and a ranked list of papers returned by a retriever.
You will also be provided a list of research topics and key terms with their frequencies that are frequently mentioned by the top-ranked papers returned by the retriever.
Your task is to improve the provided retrieval results by selecting a list of topics and terms that can accurately identify the relevant papers of the query.
Make sure your selection is strictly based on the original query and does not contain repeated concepts.

Output format: ‘<ans>selection 1, selection 2, ...</ans>’.

Retriever result: $\{D^0(q)\}$

Candidate topics: $\{T^0(q)\}$

Candidate key terms: $\{P^0(q)\}$

Original Query: $\{q\}$

Figure 5: Prompts given to the LLM for query core concept identification.