LLM-Based Compact Reranking with Document Features for Scientific Retrieval

Anonymous ACL submission

Abstract

Scientific retrieval is essential for advancing academic discovery. Within this process, document reranking plays a critical role by refining first-stage retrieval results. However, standard LLM listwise reranking faces unique challenges in the scientific domain. First-stage retrieval is often suboptimal in the scientific domain, so relevant documents are ranked lower. Moreover, conventional listwise reranking inputs the full text of candidates into the context window, limiting the number of candidates that can be considered. As a result, many relevant documents are excluded before reranking, constraining overall retrieval performance. To address these challenges, we explore compact document representations based on semantic features (e.g., categories, sections and keywords) and propose CoRank, a *training-free*, *model-agnostic* reranking framework for scientific retrieval. It presents a three-stage solution: (*i*) offline extraction of document-level features, (*ii*) coarse reranking using these compact representations, and (*iii*) fine-grained reranking on full texts of the top candidates from (*ii*). This hybrid design provides a high-level abstraction of document semantics, expands candidate coverage, and retains critical details required for precise ranking. Experiments on LitSearch and CSFCube show that CoRank significantly improves reranking performance across different LLM backbones (nDCG@10 from 32.0 to 39.7). Overall, these results highlight the value of information extraction for reranking in scientific retrieval. Code will be publicly available.

1 Introduction

Scientific retrieval (Lawrence et al., 1999; White et al., 2009) is crucial for scientific discovery. While current retrievers are effective at retrieving broadly relevant scientific papers, they often struggle to differentiate between documents covering similar topics (Sciavolino et al., 2021; Liu et al., 2021), making fine-grained relevance estimation essential. Therefore, the reranking stage (Carbonell and Goldstein-Stewart, 1998; Kurland and Lee, 2005) is particularly important (Gao et al., 2021b), as it refines the first-stage retrieval to better distinguish between closely ranked documents.

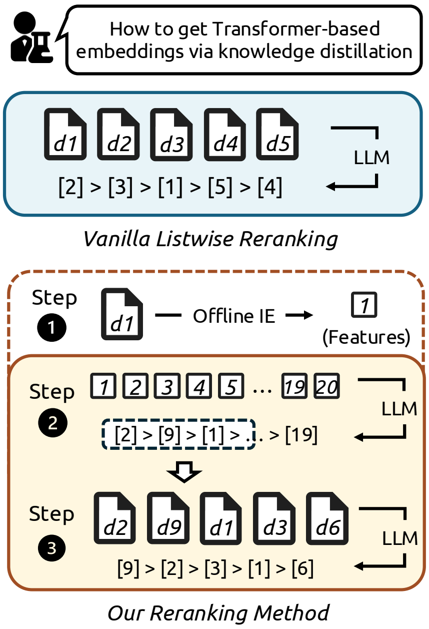


Figure 1: Instead of directly reranking full documents, we extract features, rerank a larger candidate pool with them, and finally refine the shortlist with full documents.

Recently, large language models (LLMs) (Grattafiori et al., 2024; OpenAI et al., 2024) have significantly advanced document reranking, particularly through their application to *listwise reranking* (Sun et al., 2023a; Ma et al., 2023). In this setting, LLMs jointly assess a set of retrieved candidates within their context window and generate a reordered ranking based on their relevance. Previous works (Pradeep et al., 2023a; Gangi Reddy et al., 2024; Liu et al., 2024c; Ren et al., 2024) show that LLM-based listwise rerankers outperform prior embedding-based approaches (Nogueira et al., 2019b, 2020) on benchmarks like BEIR (Thakur et al., 2021).

The standard practice in LLM listwise reranking (Sun et al., 2023a; Pradeep et al., 2023a,b; Gangi Reddy et al., 2024; Liu et al., 2024c) is to input the full text of each candidate document into the context window. However, this approach faces key limitations in scientific retrieval. In the scientific domain, retrievers often show limited performance (Kim et al., 2023; Kang et al., 2024a,b), so truly relevant documents may not rank high enough in the first-stage retrieval. Moreover, due to the substantial token overhead of full text representation and finite context length, rerankers can only operate on a limited number (typically 20 per prompt) (Sun et al., 2023a; Ma et al., 2023) of candidates. As a result, when first-stage retrieval is suboptimal, the reranking performance is inherently constrained (Reddy et al., 2023).

To address these limitations, we explore an alternative document representation that is both compact and informative. Instead of using full text, we investigate an information extraction (IE)–based approach (Niklaus et al., 2018; Zhou et al., 2022; Liu et al., 2022), representing each scientific paper using high-level features such as categories, sections, and keywords. We find that this representation significantly improves per-document token efficiency, enabling a larger pool of candidates to fit within the LLM's context window. This, in turn, makes the reranking process more robust to suboptimal first-stage scientific retrieval results. It also simplifies the input by filtering out irrelevant details, making it easier for LLMs to interpret. However, since IE cannot *guarantee* complete coverage of all relevant content, full-text inputs still serve as a valuable complement when fine-grained relevance comparisons are required.

Motivated by these insights, we propose CoRank, a *training-free*, *model-agnostic* reranking framework for science retrieval. CoRank consists of three stages: (*i*) **Offline Preprocessing**: extract high-level semantic features like categories and keywords from unstructured scientific documents; (*ii*) **Coarse Reranking**: use these compact representations for an initial ranking and select a subset of top candidates; (*iii*) **Fine-grained Reranking**: rerank the top candidates using full scientific documents to recover the potentially missing details during information extraction. Our design enhances the robustness against suboptimal first-stage retrieval with compact feature representation and maximizes the overall effectiveness with final full text reranking.

We evaluate CoRank on two scientific retrieval benchmarks: LitSearch (Ajith et al., 2024) and CSFCube (Mysore et al., 2021). Empirical experiments show that across various LLM backbones, CoRank achieves an absolute improvement of +11.5 nDCG@10 without the sliding window strategy, and retains a +3.9 gain with it. These results demonstrate significant effectiveness of our method for reranking in scientific retrieval. Overall, our contributions are threefold:

# 1 We are the first to reveal the unique limitations of LLM listwise reranking in scientific retrieval.

# 2 We are the first to explore semantic features as compact representations in reranking and propose CoRank, a framework based on this design.

# 3 We show that CoRank significantly improves reranking performance in scientific retrieval.

These findings underscore the value of compact feature extraction as a pre-analysis step for improving the reranking performance in scientific retrieval.

2 Preliminary Analysis

In this section, we first define listwise reranking, then show the limitations of current full text-based methods and explore compact and informative document representation options for reranking in the scientific domain.

2.1 Problem Definition

Reranking Input. Given a query q and a large corpus $\mathcal{P}_n$ of n scientific papers, a first-stage retriever selects a ranked list of m candidate documents $\mathcal{C} = \{p_1, p_2, \dots, p_m\} \subset \mathcal{P}_n$. Typically we have $m \ll n$ due to the scale of the corpus and the finite capacity of rerankers. This candidate list $\mathcal{C}$ is then passed to a reranking model, which aims to reorder the documents such that more relevant ones are ranked higher.

Reranking Objective. The goal of document reranking is to find a permutation $\sigma : \{1, \dots, m\} \to \{1, \dots, m\}$ whose application

$$\mathcal{C}' = [p_{\sigma(1)}, p_{\sigma(2)}, \dots, p_{\sigma(m)}]$$

orders documents in more accurate *descending* relevance to q. The quality is often (Sun et al., 2023a; Gangi Reddy et al., 2024; Liu et al., 2024c) evaluated using top-k metrics such as nDCG@10, with small k values that stress accuracy at the top of the ranked list.

LLM Listwise Reranking. Large language models can inspect m candidate documents in their context window and generate a permutation that reflects their relevance ordering. Formally, the model acts as follows:

$$\sigma_{\text{LLM}} = \text{LLM}(q, \mathcal{C}) \in S_m,$$

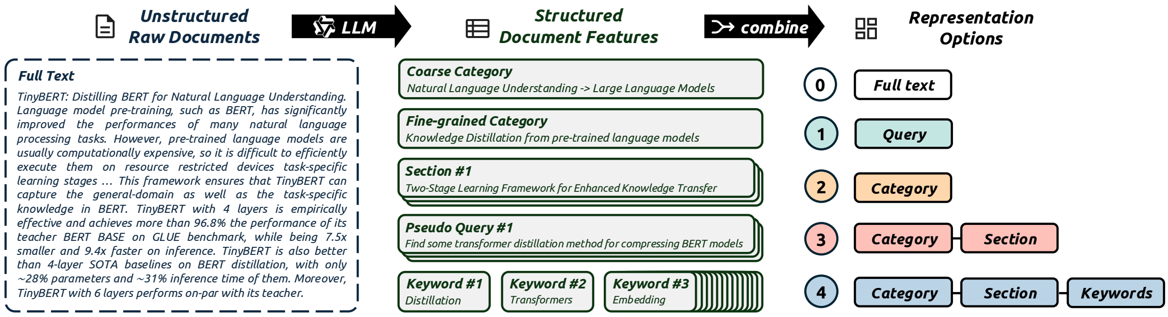
where S_m is the set of all permutations on $\{1, \dots, m\}$. Applying this permutation to the candidate list yields

$$\mathcal{C}' = \big[p_{\sigma_{\text{LLM}}(1)},\, p_{\sigma_{\text{LLM}}(2)},\, \dots,\, p_{\sigma_{\text{LLM}}(m)}\big].$$

2.2 Current Limitations

The standard approach in LLM-based listwise reranking (Sun et al., 2023a; Pradeep et al., 2023a,b; Gangi Reddy et al., 2024; Liu et al., 2024c) feeds the *full text* of each candidate into the model's context window. While this allows LLMs to capture the complete content of each document, it incurs some limitations, especially in scientific retrieval.

Suboptimal First-Stage Retrieval. Unlike in general-domain scenarios, first-stage retrievers—whether sparse (Robertson et al., 2009; Nogueira et al., 2019a) or dense (Izacard et al., 2021; Wang et al., 2022)—struggle to generalize to the scientific domain (Thakur et al., 2021; Bonifacio et al., 2022). This is due to the complexity of long-tail concepts (Kang et al., 2024b, 2025) and the lack of large-scale supervised training data (Bonifacio et al., 2022; Li et al., 2023). Consequently, their retrieval performance is limited, and truly relevant documents are often ranked much lower (Ajith et al., 2024).



TinyBERT: Distilling BERT for Natural Language Understanding. Language model pre-training, such as BERT, has significantly improved the performances of many natural language processing tasks. However, pre-trained language models are usually computationally expensive, so it is difficult to efficiently execute them on resource restricted devices task-specific learning stages … This framework ensures that TinyBERT can capture the general-domain as well as the task-specific knowledge in BERT. TinyBERT with 4 layers is empirically effective and achieves more than 96.8% the performance of its teacher BERT BASE on GLUE benchmark, while being 7.5x smaller and 9.4x faster on inference. TinyBERT is also better than 4-layer SOTA baselines on BERT distillation, with only ~28% parameters and ~31% inference time of them. Moreover, TinyBERT with 6 layers performs on-par with its teacher.

Coarse Category
Natural Language Understanding -> Large Language Models
Fine-grained Category
Knowledge Distillation from pre-trained language models
Section #1
Two-Stage Learning Framework for Enhanced Knowledge Transfer
Pseudo Query #1
Find some transformer distillation method for compressing BERT models
Keyword #1
Distillation
Keyword #2
Transformers
Keyword #3
Embedding
0
Full text
1
Query
2
Category
3
Category
Section
4
Category
Section
Keywords

Figure 2: Overview of our feature extraction pipeline: from unstructured documents, we apply zero-shot LLM information extraction to obtain document features including categories, sections, pseudo queries, and keywords, which are then combined into compact representations.

Token Consumption. Full text representation also introduces significant token overhead, with individual scientific papers often consuming hundreds or even thousands of tokens (Thakur et al., 2021; Ajith et al., 2024; Mysore et al., 2021). At the same time, LLMs have a limited effective context length (Dai et al., 2019; Xiong et al., 2024) and are known to suffer from issues such as positional bias for long inputs (Liu et al., 2024a; Tian et al., 2024). As a result, full-text rerankers are constrained to operate on a small number of candidates, typically 20 documents per input prompt (Sun et al., 2023a; Ma et al., 2023; Pradeep et al., 2023a,b).

When first-stage retrieval is suboptimal and reranking operates over a narrow candidate set, the overall reranking performance is significantly constrained.

2.3 Semantic Features as Alternatives

Given the limitations of full-text representations in scientific-domain reranking, we explore an alternative approach, which is based on document-level semantic features such as categories and keywords. The intuition is that these features are both significantly more concise and capable of preserving the core semantics. Therefore, more candidate documents can be effectively considered during reranking. Specifically, we examine the following four types of document semantic features:

Category. Categories (Sun et al., 2023b; Zhang et al., 2024) offer a high-level topical overview of scientific documents. Each document is assigned a three-level hierarchical category path in the format $\{Category\} \rightarrow \{Subcategory\} \rightarrow \{Subsubcategory\}$, which provides a broad-to-specific classification on its topic.

Section. Sections (Zhou et al., 2023) consist of multiple subtitle-style strings, each summarizing a major part of the scientific document. They capture the internal structure of the document and serve as mid-level semantic signals that enhance category-level summaries.

Keyword. Keywords (Rose et al., 2010; Lee et al., 2023) are terms or entities that represent fine-grained lexical concepts within a document. They provide the most specific information among different granularities.

Pseudo Query. Pseudo queries (Sachan et al., 2022; Kang et al., 2025) are synthetic user questions based on the content. This feature offers a unique query-aligned perspective of the document, simulating how the document might be retrieved in real-world scenarios.

These features cover most common types of document-level IE and vary in granularity and style. All semantic features are extracted using LLM-based *zero-shot information extraction*. The prompt templates used for this extraction are provided in Appendix A.

Among these features, *sections*, *keywords*, and *pseudo queries* have multiple elements for each document. To reduce noise and focus on the most relevant content, we apply an adaptive selection strategy at inference time. We compute dense embedding similarities between the query and each extracted element and retain only the most relevant: 5 keywords (from 30), 1 pseudo query (from 20), and 1 section (from 3). This improves content relevance while minimizing token overhead. Ablation of this strategy can be found in Section 4.2.

However, relying on a single feature often lacks representational power. For example, using only the category tends to be too coarse-grained and offers limited discriminative ability, while keywords alone may lack sufficient context or background information. To address this, we represent each document using *combinations* of features. Specifically, we explore the following four configurations of different overall granularities.

Form 1. *Pseudo Query*

Form 2. *Category*

Form 3. *Category + Section*

Form 4. *Category + Section + Keywords*

Ablation studies on the effect of each specific component can further be found in Section 4.2. The overview of representation construction is presented in Figure 2.

2.4 Empirical Validation

To evaluate the effectiveness of our feature-based representations, we conduct a series of experiments on the `LitSearch` (Ajith et al., 2024), using `GPT-4.1-mini` (OpenAI, 2024) as the reranking backbone. Semantic features are extracted

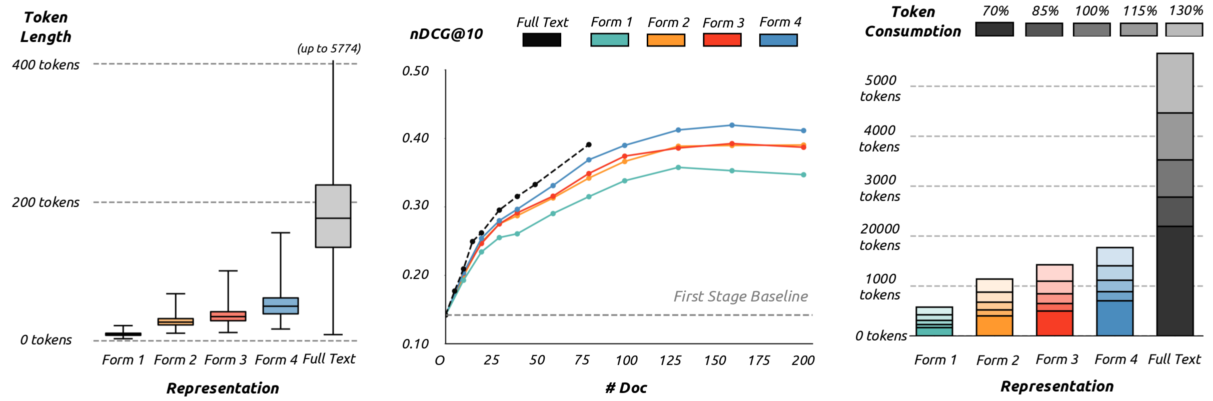


Figure 3: Token efficiency and performance comparison across different document representations. (a) Per-document token lengths distribution. (b) nDCG@10 scores for different number of documents in the context window of single LLM input (c) Context token overhead to reach equal reranking performance.

using `Qwen3-8B-Instruct` (Qwen et al., 2024). Following Gangi Reddy et al. (2024), we use `Contriever` (Izacard et al., 2021) as the first-stage retriever and the similarity encoder for adaptive selection.

Token Length Per Document. We first evaluate the token cost of different representations. As shown in Figure 3 (a), full-text inputs consume around 200 tokens per document on average, with some exceeding 5,000 tokens, limiting the number of documents that can fit into the context window. In contrast, semantic feature-based representations (Forms 1-Form 4) are much more compact, each averaging between 10 and 50 tokens.

Number of Documents in a Single Input. We then evaluate how different representations perform as the number of candidate documents increases (up to 200) within a 32k-token context window. As shown in Figure 3 (b), all representations improve as more documents are included. In the range of 0 to 80 candidates, full-text representations perform better than feature-based ones. This is expected, since full text retains complete document information, though at a much higher token cost. However, beyond 80 candidates, full-text inputs frequently exceed the context length limit. In contrast, feature-based methods, especially Form 4, remain compact and can handle more candidates without issue, while still achieving comparable performance.

Token Cost for Equal Reranking Performance. Finally, we examine how many context tokens are needed to achieve the same reranking performance across different representations. Using results from earlier experiments, we estimate the total token cost by combining the number of documents required with the average token count per document. We take full-text reranking with 20 documents as the 100% performance base, and compare how many tokens each method needs to reach 70%, 85%, 100%, 115%, and 130% of that level. As shown in Figure 3 (c), compact feature-based representations can match full-text performance with significantly fewer tokens. For instance, reaching 100% performance requires around 3,500 tokens for full text, but only 1,000 or even a few hundred tokens for feature-based inputs.

2.5 Discussion

The results above demonstrate that feature-based document representations offer clear advantages over full-text inputs in reranking. They can express the core content of a document using far fewer tokens, allowing significantly more retrieved candidates to be included within the same token overhead. This is especially valuable for recovering relevant documents that were assigned lower scores by the first-stage retriever. Considering both token efficiency and reranking performance, we select Form 4 as our feature-based representation.

However, in the second set of experiments where the number of documents is held constant, we also observe that feature-based representations may underperform full text. After all, offline information extraction may omit subtle but important cues present in the full text.

Therefore, these results motivate a hybrid reranking strategy: using semantic features in the early stage to cover a broader range of candidates, followed by a refinement stage that reranks the top candidates using full-text inputs for more in-depth comparison.

3 Methodology

Based on the preliminary analysis, we propose a hybrid reranking framework in scientific domain. It proceeds in three steps: (*i*) offline extraction of structured semantic features; (*ii*) coarse-grained reranking using the features to select a subset of top candidates; and (*iii*) fine-grained reranking over the subset with full text inputs.

3.1 Compact and Informative Representation

To construct compact, feature-based representations for each scientific document, we first perform document-level information extraction offline. As described in Section 2.3, we use zero-shot information extraction to obtain target semantic features: *category*, *section*, and *keyword*. Formally, for a given document p_i, we obtain:

$$\text{LLM}(p_i) = \text{Category}_i,\ [\text{Section}_i^j],\ [\text{Keyword}_i^j]$$

These features are designed to capture the core semantics of scientific documents in a token-efficient manner. The extraction is performed offline, and the results are cached and reused at inference time to avoid any additional runtime delay. The prompt templates for extracting different features can be found in Appendix A.

3.2 Coarse Reranking w. Compact Features

Given a query q, we obtain a document relevance ranking $\mathcal{C} = [p_1, p_2, \ldots, p_m]$ from a first-stage retriever. We then replace each document with a compact semantic representation and apply LLM-based listwise reranking to identify a high-quality subset.

For sections and keywords, we apply *adaptive selection* to select only those most relevant to the query. Specifically, for each unit (single section or keyword) u, we compute the cosine similarity with text embedding and select the top-5 keywords and the most relevant section. (Analysis on the number of keyword used can be found in Section 4.4) The final feature-based representation r_i is formed as:

$$r_i = \text{Concat}\left[\text{Category}_i, \text{Section}_i^*, \text{Keywords}_i^*\right]$$

where * indicates adaptively selected elements.

With the feature-based representations $\mathcal{R} = [r_1, r_2, \ldots, r_m]$, we then use the LLM to perform listwise reranking, producing a permutation $\sigma_{\text{feat}} = \text{LLM}(q, \mathcal{R}) \in S_m$. Applying this permutation to the original candidate list $\mathcal{C} = [p_1, p_2, \ldots, p_m]$ yields the reranked output for the coarse reranking:

$$\mathcal{C}_{\text{feat}} = [p_{\sigma_{\text{feat}}(1)}, p_{\sigma_{\text{feat}}(2)}, \ldots, p_{\sigma_{\text{feat}}(m)}]$$

We then keep the top-k documents from this list to form the seed set for full text reranking:

$$\mathcal{C}_{\text{seed}} = [p_{\sigma_{\text{feat}}(1)}, \ldots, p_{\sigma_{\text{feat}}(k)}], \quad k < m$$

Coarse reranking with compact document representations greatly expands the number of candidates that can be considered in the same LLM input. This broader coverage helps recover scientific documents that were initially assigned low scores by the first-stage retriever.

3.3 Fine-grained Reranking w. Full Text

In the second reranking stage, we refine the ranking over the seed set of candidates using full documents. Specifically, we take the seed set $\mathcal{C}_{\text{seed}} = [p'_1, p'_2, \ldots, p'_k]$, (we use $'$ to distinguish them from the first-stage input) obtained from the previous stage, and replace each compact representation with its original document text.

Let $\mathcal{T} = [t_1, t_2, \ldots, t_k]$ denote the full text of the selected documents, where t_i corresponds to the full text content of passage p'_i. We then once again use LLM to perform listwise reranking:

$$\sigma_{\text{text}} = \text{LLM}(q, \mathcal{T}) \in S_k$$

With the permutation σ_{text} we get the final ranking:

$$\mathcal{C}_{\text{final}} = [p'_{\sigma_{\text{text}}(1)}, \ldots, p'_{\sigma_{\text{text}}(k)}]$$

Fine-grained reranking with full-text inputs recovers details that may be lost during the information extraction process. Since the candidate set has already been narrowed down, full documents can now be used without exceeding the LLM's context limit.

Overall, this hybrid strategy effectively addresses the challenges of LLM-based listwise reranking in the scientific domain. The coarse reranking stage expands the reranking pool, improving robustness to suboptimal first-stage retrieval in the scientific domain. The fine-grained reranking stage, in turn, preserves sufficient detail for precise relevance comparisons.

4 Experiments

In this section, we first outline the experimental setup, then present the reranking results, followed by ablation studies and further analysis. Detailed qualitative studies can be found in Appendix B.

4.1 Experimental Setup

Datasets & Metric. We evaluate different reranking methods on two high-quality scientific retrieval benchmarks: `LitSearch` (Ajith et al., 2024) and `CSFCube` (Mysore et al., 2021). The former is a benchmark of expert-annotated complex literature queries targeting recent ML and NLP papers. The latter is a human-annotated testbed for faceted query-by-example retrieval. Since reranking focuses on the top results, we follow standard practice (Sun et al., 2023a; Ma et al., 2023; Gangi Reddy et al., 2024; Liu et al., 2024c), reporting metrics for @5 and @10. We include nDCG, MAP, and Recall as our evaluation metrics.

Models & Hyperparameters. For the reranking backbones, we use `Qwen3-32B-Instruct` (Qwen et al., 2024) as the representative open-source model, `Gemini 2.0 Flash` (Google DeepMind, 2025) and `GPT-4.1-mini` (OpenAI, 2024) as proprietary examples. For generative parameters, we use a temperature of 1.0 and a fixed random seed of 42 for reproducibility. The API costs are detailed in Appendix C.

For semantic feature extraction, we find that small open-source models are sufficiently effective; we therefore adopt `Qwen3-8B-Instruct` (Qwen et al., 2024) for all extraction tasks (Templates in Appendix A). For both first-stage retrieval and semantic similarity scoring for filtering features, we use `Contriever` (Izacard et al., 2021) following Gangi Reddy et al. (2024).

Baselines. For supervised, model-specific baselines, we compare against `RankVicuna` (Pradeep et al., 2023a), `RankZephyr` (Pradeep et al., 2023b), and `RankMistral` (Liu et al., 2024c). These models are trained on large-scale general-domain datasets such as MS MARCO (Bajaj et al., 2016) and RankGPT (Sun et al., 2023a) to enhance reranking performance.

For model-specific baselines, we use vanilla listwise reranking (Sun et al., 2023a) with zero-shot instruction-following. We also consider the sliding window strat-

Model	Strategy	LitSearch						CSFCube					
		N@5	N@10	M@5	M@10	R@5	R@10	N@5	N@10	M@5	M@10	R@5	R@10
Initial Retriever													
Contriever	Dense Retrieval	12.6	14.2	11.0	11.7	16.4	21.2	22.0	23.6	5.8	8.6	9.5	16.4
Supervised, Model-Specific Methods													
RankMistral	Vanilla $_{Full}$	19.5	20.1	18.0	18.2	23.6	24.3	23.1	21.9	5.8	7.8	8.4	13.3
RankVicuna	Vanilla $_{Sliding}$	25.9	26.5	23.9	24.1	30.8	32.4	30.0	28.7	7.5	10.6	10.7	17.5
RankZephyr	Vanilla $_{Sliding}$	33.7	34.1	31.9	32.1	37.7	39.0	35.7	34.1	9.4	13.6	14.6	22.7
Zero-Shot, Model-Agnostic Reranking Strategies													
Qwen3-32B	Vanilla $_{Full}$	25.4	25.7	24.5	24.6	26.6	27.3	28.5	28.4	6.9	10.4	13.0	20.2
	CoRank $_{Full}$	**39.4**	**39.6**	**38.3**	**38.4**	**41.5**	**41.9**	**32.2**	**31.8**	**8.8**	**12.8**	**15.9**	**23.4**
Qwen3-32B	Vanilla $_{Sliding}$	39.6	39.8	37.9	37.9	43.1	43.5	31.6	31.2	7.4	11.3	14.3	22.9
	CoRank $_{Sliding}$	**41.9**	**42.2**	**40.3**	**40.4**	**45.0**	**46.2**	**36.2**	**35.5**	**9.8**	**14.6**	**17.7**	**26.6**
Gemini 2.0 Flash	Vanilla $_{Full}$	26.0	26.1	25.1	25.2	26.9	27.3	31.0	28.5	7.8	10.6	13.2	18.8
	CoRank $_{Full}$	**40.5**	**40.7**	**38.7**	**38.8**	**43.6**	**44.2**	**36.6**	**35.1**	**11.5**	**15.4**	**20.7**	**27.8**
Gemini 2.0 Flash	Vanilla $_{Sliding}$	40.7	40.8	38.9	39.0	44.4	44.6	32.6	32.2	8.3	12.5	15.3	23.2
	CoRank $_{Sliding}$	**43.1**	**43.3**	**40.7**	**40.9**	**48.1**	**48.9**	**36.3**	**38.1**	**10.2**	**16.5**	**17.8**	**31.0**
GPT-4.1-mini	Vanilla $_{Full}$	25.9	26.1	25.1	25.2	26.7	27.2	32.2	28.7	7.8	10.7	14.3	19.7
	CoRank $_{Full}$	**46.0**	**46.3**	**44.5**	**44.7**	**48.3**	**49.4**	**39.2**	**39.0**	**10.6**	**16.5**	**18.6**	**30.5**
GPT-4.1-mini	Vanilla $_{Sliding}$	41.6	41.9	40.3	40.4	43.8	44.4	34.1	34.9	8.6	13.5	15.2	25.3
	CoRank $_{Sliding}$	**45.5**	**45.8**	**43.9**	**44.0**	**48.3**	**49.1**	**40.1**	**39.4**	**11.0**	**16.9**	**19.2**	**30.4**

Table 1: Reranking performance on the `LitSearch` and `CSFCube` reflected by nDCG (N@k), MAP (M@k) and Recall (R@k). We compare CoRank with supervised reranking models and zero-shot, model-agnostic strategy.

egy (Sun et al., 2023a; Ma et al., 2023), a test-time scaling (Xia et al., 2025) technique designed to enhance reranking performance. We report results both with and without the use of sliding windows. Although CoRank is orthogonal to this technique, we include a comparison in Section 4.3 to further show our effectiveness.

Reranking Parameters. We follow the standardized setup (Sun et al., 2023a; Ma et al., 2023; Pradeep et al., 2023a,b; Gangi Reddy et al., 2024) from prior work to unify reranking parameters. For vanilla reranking without the sliding window strategy, we include 20 full-text documents within the context. An exception is `RankMistral` (Liu et al., 2024c), which benefits from long-context training and is able to process 100 full-text documents in a single input. When using the sliding window strategy, we rank a total of 100 full-text documents by applying a window size of 20 with a step size of 10. For CoRank, we include 200 compact representations in the coarse reranking stage, followed by 20 full-text documents in the fine-grained reranking stage.

4.2 Main Results

Reranking Performance. We begin by evaluating the performance of different reranking methods on the target academic retrieval benchmarks, as shown in Table 1.

First, we observe that all reranking methods outperform the first-stage dense retriever baseline, confirming the critical role of reranking in scientific document retrieval. However, while model-specific rerankers such as `RankVicuna` (Pradeep et al., 2023a) and zero-shot listwise baselines do yield noticeable improvements, their gains are relatively modest compared to results in general-domain settings (Sun et al., 2023a), particularly when the sliding window strategy is not used.

In contrast, our proposed method, CoRank, is fully zero-shot, model-agnostic, and training-free, yet achieves strong and consistent gains across different LLM backbones. Notably, without using sliding windows, CoRank improves average nDCG@10 from 27.3 to 38.8, a relative improvement of over 40%. Even when combined with sliding window inputs, it still yields an average gain of +3.9 nDCG@10.

We also find that both CoRank and the sliding window strategy independently lead to substantial performance gains (ours being larger and more efficient; see Section 4.3). This is likely because both methods expand the reranking scope, which is particularly valuable in scientific domains considering the suboptimal first-stage retrieval. These findings reinforce our earlier analysis of current limitations presented in Section 2.2.

Ablation Studies. To evaluate the contribution of each component in our design, we conduct ablation studies on the `LitSearch` (Ajith et al., 2024) dataset using `GPT-4.1-mini` (OpenAI, 2024) as the backbone.

Specifically, we remove individual semantic features—*category*, *section*, and *keywords*—from the compact representation to assess their relative importance. In addition, we ablate two key components of our pipeline: *adaptive selection* and *fine-grained reranking*, to understand their impact on overall performance.

As shown in Table 2, removing any single semantic component (category, section, or keywords) consistently degrades reranking performance, confirming that each feature contributes complementary information. Category captures coarse-grained background context that tends to align more easily with the query but offers less specific detail. In contrast, section and keywords reflect finer-grained semantics that are harder to match but,

Model	Strategy	N@10	M@10	R@10
Vanilla	Full Text	26.1	25.2	27.3
CoRank	– Category	43.4	41.9	45.9
CoRank	– Section	45.4	43.8	48.2
CoRank	– Keywords	44.6	42.8	48.0
CoRank	– Selection	45.5	44.1	48.2
CoRank	– Fine. Rank.	41.1	38.7	46.7
CoRank	Full	**46.3**	**44.7**	**49.4**

Table 2: Ablation results showing the impact of removing individual features and two key design components.

when relevant, provide highly targeted signals. This balance underscores the value of combining semantic features across different levels of abstraction.

Additionally, we find that both adaptive selection and fine-grained reranking contribute to the final performance. Adaptive selection ensures that the most query-relevant features are retained, while the fine-grained reranking stage substantially compensates for potential information loss in compact representations by recovering more nuanced evidence from the full text.

4.3 Comparison with Sliding Window

Sliding window strategy (Sun et al., 2023a) is a widely used (Ma et al., 2023; Pradeep et al., 2023a,b; Gangi Reddy et al., 2024) method for overcoming the context length limitations of LLMs in listwise reranking. Instead of ranking all candidates in a single prompt, it partitions the candidate list into overlapping windows and reranks each window independently from the bottom up. This method has been shown to notably improve reranking performance (Sun et al., 2023a).

As noted in Section 4.1, we evaluate both the standalone and combined use of sliding windows with our method. While CoRank and sliding window are orthogonal and compatible, isolating them in comparison allows us to clearly demonstrate the superior efficiency and effectiveness of our approach when used alone.

Method	N@10	M@10	R@10	Token Usage	Cost
Vanilla	26.0	25.0	27.3	1.01M	$0.40
Sliding	40.8	39.1	44.2	9.06M	$3.62
CoRank	42.2	40.6	45.2	3.60M	$1.44
Both	43.8	41.8	48.1	11.65M	$4.66

Table 3: Comparison of performance, token usage, and API cost between CoRank and Sliding Window.

We evaluate CoRank and the sliding window strategy on `LitSearch` and `CSFCube` on three tested models, reporting nDCG@10, total token consumption, and API cost for GPT-4.1-mini. As shown in Table 3, CoRank not only achieves better average reranking quality (nDCG@10 improved by 1.4), but also requires only 40% of the token budget compared to sliding windows. These results demonstrate that CoRank provides an efficient yet effective alternative for expanding reranking range. Moreover, combining these two methods can further boost performance, offering an advanced solution for complex retrieval scenarios.

4.4 Hyperparameter Study

We also study the effect of key hyperparameters in CoRank, using the `GPT-4.1-mini` (OpenAI, 2024) reranker on the `LitSearch` (Ajith et al., 2024) dataset. We report results in terms of nDCG@10, MAP@10, Recall@10, token usage, and estimated API cost.

Number of Keywords. With its default configuration, CoRank extracts 30 keywords from each document and selects the top 5 most relevant ones based on cosine similarity from text embedding. These selected keywords are then concatenated into the document representation. Here, we evaluate the impact of the keyword selection on coarse reranking performance by varying the number of concatenated keywords: 0, 1, 3, 5, 10, 15, and 20. The results are in Table 4.

# Keyword	N@10	M@10	R@10	Token Usage	Cost
0 keyword	38.9	36.5	44.2	1.80M	$0.72
1 keyword	40.4	38.0	46.3	1.95M	$0.78
3 keywords	40.1	37.9	44.9	2.26M	$0.90
5 keywords	41.1	38.7	46.7	2.57M	$1.03
10 keywords	42.4	40.4	46.5	3.33M	$1.33
15 keywords	42.6	40.1	48.6	4.10M	$1.64
20 keywords	42.4	40.2	47.4	4.87M	$1.95

Table 4: Effect of the number of selected keywords on coarse reranking: performance, token usage and cost.

Our experiments reveal a trade-off between performance and cost when varying the number of keywords. In general, increasing the number of selected keywords leads to a linear increase in token cost, while also improving coarse reranking performance. However, we observe diminishing returns beyond 10 keywords. This is expected, as keywords are included in order of embedding similarity to the query—those added later tend to be less relevant and contribute weaker relevance signals.

Fine-Grained Pool Size. In the main experiments, for fine-grained reranking, we select the top 20 candidates from the coarse reranking stage to align with other reranking baselines. Here, we explore the impact of varying this candidate selection size. Specifically, we evaluate performance when selecting 5, 10, 20, 40, 60, 80, and 100 candidates for fine-grained reranking. The results are shown in Table 5.

Experimental results show that increasing the fine-grained reranking candidate pool from 0 to 20 documents yields a favorable trade-off between token cost and performance. However, performance saturates beyond 20 candidates and remains stable up to 100 documents, despite the growth of token consumption.

In theory, if an LLM's long-context capability were fully effective, performance should continue to improve as more candidates are included. In practice, however,

Pool Size	N@10	M@10	R@10	Token Usage	Cost
5 docs	43.0	41.3	46.7	0.25M	$0.10
10 docs	44.3	42.8	46.7	0.50M	$0.20
20 docs	46.3	44.7	49.4	1.01M	$0.40
40 docs	45.4	43.9	48.0	2.01M	$0.81
60 docs	45.0	43.4	48.3	3.02M	$1.21
80 docs	45.3	43.8	48.0	4.03M	$1.61
100 docs	46.0	44.4	49.1	5.03M	$2.01

Table 5: Effect of the fine-grained candidate pool size: performance, token usage and cost.

the observed plateau suggests that the effective context length is often much shorter than the theoretical context limit, which is also mentioned in previous studies (Hsieh et al., 2024; Kuratov et al., 2024).

5 Related Work

Classical Document Reranking. Document Reranking (Karpukhin et al., 2020; Ren et al., 2021, 2023) is a critical component in information retrieval (IR) (Baeza-Yates et al., 1999; Singhal et al., 2001), especially when high retrieval precision is required and the first-stage retriever alone is insufficient. Early reranking methods were typically lexical and probabilistic (Salton et al., 1975; Robertson and Jones, 1976), relying on term overlap between the query and document to adjust relevance scores. However, these approaches were limited by reliance on exact matching and failed to capture in-depth semantics (Karpukhin et al., 2020; Gao et al., 2021a).

With advances in deep learning, neural reranking methods (Guo et al., 2019; Trabelsi et al., 2021), particularly those based on learning-to-rank (Cao et al., 2007; Liu, 2010) frameworks became prevalent. These models overcome previous limitations, capturing semantic similarity beyond surface-level term matches.

The emergence of pre-trained language models (PLMs) (Vaswani et al., 2017; Devlin et al., 2019; Reimers and Gurevych, 2019) has further transformed the field of document reranking (Nogueira et al., 2019a; Yates et al., 2021). Cross-encoder-based rerankers (Litschko et al., 2022), in particular, have demonstrated strong performance by modeling query-document relevance scores and optimizing pointwise, pairwise, or listwise loss functions (Zhuang et al., 2023).

LLM-Based Listwise Document Reranking. Recently, large language models (LLMs) (Touvron et al., 2023; OpenAI et al., 2023) have revolutionized the field of NLP. With strong general knowledge and instruction-following (Ouyang et al., 2022) capabilities, LLMs offer a new paradigm for listwise document reranking. A classic proposal (Sun et al., 2023a) involves placing the query and a list of candidate documents into the context window and prompting the LLM to generate a reordered ranking. This zero-shot strategy has been shown to outperform traditional reranking baselines (Sun et al., 2023a; Ma et al., 2023).

After that, LLM listwise reranking has been extended in multiple directions. Works like RankVicuna (Pradeep et al., 2023a) and RankZephyr (Pradeep et al., 2023b) represent early explorations of open-source reranking LLMs. For inference time efficiency, FIRST (Gangi Reddy et al., 2024) modify listwise reranking by using only the first token decoding. RankMistral (Liu et al., 2024c) adopts long-context training to help LLMs adapt to larger document lists in a single input. Test-time scaling (Xia et al., 2025) also emerged as a common enhancement. For example, the sliding window strategy (Sun et al., 2023a; Ma et al., 2023) partitions the candidate list into overlapping chunks, enabling broader coverage within limited context. Building on this, Tang et al. (2024) enhance reranking quality through permutation consistency. ScaLR (Ren et al., 2024) further designs self-calibration to improve consistency across windows.

Notably, two prior studies (Liu et al., 2024b; Li et al., 2024) also identified limitations of full-document representations, but their analyses are on general domain. Moreover, the alternative representations they propose are fundamentally different from ours—one (Liu et al., 2024b) adopts non-natural language embeddings, and the other (Li et al., 2024) relies on document chunking.

To the best of our knowledge, our work is the first to highlight the unique limitations of full-document representations in scientific retrieval. We are also the first to leverage IE-based features as document representations for reranking. This makes our contribution novel and significant within the reranking literature.

6 Conclusion

In this work, we first identify fundamental challenges of standard listwise reranking in scientific retrieval. On the one hand, the first-stage retriever often fails to rank truly relevant documents high due to the lack of domain-specific training data. On the other hand, existing LLM-based listwise reranking methods typically operate over full-document inputs, which are substantial in terms of token usage and therefore limited to a small number of candidates. As a result, when the first-stage retrieval is suboptimal for scientific retrieval, the performance of standard reranking method is massively restricted.

To address this, we propose CoRank, a training-free and model-agnostic reranking framework for scientific retrieval. CoRank employs compact semantic features for coarse reranking. This allows the LLM to consider a broader range of candidates within its context window. A subsequent fine-grained reranking stage then refines the top results using full-text inputs. Experiments on two scientific benchmarks show that CoRank significantly improves reranking performance.

Overall, we are the first to explore semantic features as compact document representations for LLM reranking in scientific retrieval. Our results highlight the value of semantic feature extraction as a pre-analysis step in LLM listwise reranking in the scientific domain.

Limitations

Feature Extraction Quality. In our offline preprocessing stage, we apply zero-shot information extraction using LLMs to obtain semantic features such as categories for each document. This is a relatively simple approach. While it has proven effective based on both quantitative results and case studies, there is still potential room for improvement. For instance, one could explore using more specialized models or introducing multi-turn feedback mechanisms to enhance the quality of extracted features. That said, since the primary focus of our work is to address the reranking bottleneck in scientific retrieval, we did not dedicate significant effort to modifiying the IE component, especially observing that the basic extraction already yielded strong performance.

Applicability Across Domains. This work focuses exclusively on scientific retrieval. The motivation and assumptions behind our method are specifically tailored to the characteristics of this domain, such as long-tail terminology and the limitations of first-stage retrievers. We do not evaluate our method in general-domain settings. The potential effectiveness and limitations of our method outside the scientific domain remain untested and are left for future work.

Ethics Statement

We conduct our experiments on widely recognized and publicly available scientific retrieval datasets. Our proposed method and findings do not pose any foreseeable harm to individuals or groups. Overall, we do not anticipate any significant ethical concerns with this work.

References

Qingyao Ai, Ting Bai, Zhao Cao, Yi Chang, Jiawei Chen, Zhumin Chen, Zhiyong Cheng, Shoubin Dong, Zhicheng Dou, Fuli Feng, et al. 2023. Information retrieval meets large language models: A strategic report from chinese ir community. *AI Open*, 4:80–90.

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

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*, volume 463. ACM press New York.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset.

Luiz Henrique Bonifacio, Hugo Queiroz Abonizio, Marzieh Fadaee, and Rodrigo Frassetto Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 2387–2392. ACM.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136. ACM.

J. Carbonell and Jade Goldstein-Stewart. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *ACM SIGIR Forum*, 51:209–210.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. FIRST: Faster improved listwise reranking with single token decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8642–8652, Miami, Florida, USA. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021a. COIL: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, Online. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021b. Rethink training of bert rerankers in multi-stage retrieval pipeline. *ArXiv preprint*, abs/2101.08751.

Google DeepMind. 2025. Gemini 2.5: Our most intelligent ai model. `https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/`. Google Blog.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

J. Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and

Xueqi Cheng. 2019. A deep look into neural ranking models for information retrieval. *Inf. Process. Manag.*, 57:102067.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *ArXiv preprint*, abs/2404.06654.

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

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 *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, pages 1497–1508. ACM.

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.

SeongKu Kang, Yunyi Zhang, Pengcheng Jiang, Dongha Lee, Jiawei Han, and Hwanjo Yu. 2024b. Taxonomy-guided semantic indexing for academic paper search. *ArXiv preprint*, abs/2410.19218.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Jihyuk Kim, Minsoo Kim, Joonsuk Park, and Seung-won Hwang. 2023. Relevance-assisted generation for robust zero-shot retrieval. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 723–731, Singapore. Association for Computational Linguistics.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Y. Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Oren Kurland and Lillian Lee. 2005. Pagerank without hyperlinks: structural re-ranking using links induced by language models. *ArXiv*, abs/cs/0601045.

Steve Lawrence, Kurt Bollacker, and C Lee Giles. 1999. Indexing and retrieval of scientific literature. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 139–146.

Wanhae Lee, Minki Chun, Hyeonhak Jeong, and Hyunggu Jung. 2023. Toward keyword generation through large language models. *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: structure-aware pre-trained language model for legal case retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1035–1044. ACM.

Minghan Li, É. Gaussier, Juntao Li, and Guodong Zhou. 2024. Keyb2: Selecting key blocks is also important for long document ranking with large language models. *ArXiv preprint*, abs/2411.06254.

Robert Litschko, Ivan Vulić, and Goran Glavaš. 2022. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1071–1082, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pai Liu, Wenyang Gao, Wen Dong, Lin Ai, Songfang Huang, and Yue Zhang. 2022. A survey on open information extraction from rule-based model to large language model. In *Conference on Empirical Methods in Natural Language Processing*.

Qi Liu, Bo Wang, Nan Wang, and Jiaxin Mao. 2024b. Leveraging passage embeddings for efficient listwise reranking with large language models. *Proceedings of the ACM on Web Conference 2025*.

Tie-Yan Liu. 2010. Learning to rank for information retrieval. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, page 904. ACM.

Wenhan Liu, Xinyu Ma, Yutao Zhu, Ziliang Zhao, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2024c. Sliding windows are not the end: Exploring full ranking with long-context large language models.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip Yu. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model.

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.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019a. From doc2query to doctttttquery. *Online preprint*, 6(2).

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019b. Multi-stage document ranking with bert.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, et al. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024. Introducing gpt-4.1 in the api. `https://openai.com/index/gpt-4-1/`. Accessed: 2025-05-19.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. Rankvicuna: Zero-shot listwise document reranking with open-source large language models.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze!

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, et al. 2024. Qwen2.5 technical report.

Revanth Gangi Reddy, Pradeep Dasigi, Md Arafat Sultan, Arman Cohan, Avirup Sil, Heng Ji, and Hannaneh Hajishirzi. 2023. Refit: Relevance feedback from a reranker during inference.

Nils Reimers and Iryna Gurevych. 2019. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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 the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183, Online. Association for Computational Linguistics.

Ruiyang Ren, Yuhao Wang, Kun Zhou, Wayne Xin Zhao, Wenjie Wang, Jing Liu, Ji-Rong Wen, and Tat-Seng Chua. 2024. Self-calibrated listwise reranking with large language models.

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. TOME: A two-stage approach for model-based retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6102–6114, Toronto, Canada. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Stephen E. Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*, 27:129–146.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pages 1–20.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. Gollie: Annotation guidelines improve zero-shot information-extraction. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023a. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023b. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2024. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2327–2340, Mexico City, Mexico. Association for Computational Linguistics.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, A. Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *ArXiv preprint*, abs/2211.09085.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models.

Runchu Tian, Yanghao Li, Yuepeng Fu, Siyang Deng, Qinyu Luo, Cheng Qian, Shuo Wang, Xin Cong, Zhong Zhang, Yesai Wu, et al. 2024. Distance between relevant information pieces causes bias in long-context llms. *ArXiv preprint*, abs/2410.14641.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Mohamed Ali Trabelsi, Zhiyu Chen, Brian D. Davison, and Jeff Heflin. 2021. Neural ranking models for document retrieval. *Information Retrieval Journal*, 24:400–444.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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*, abs/2212.03533.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *ArXiv preprint*, abs/2302.10205.

Howard D White, H Cooper, LV Hedges, et al. 2009. Scientific communication and literature retrieval. *The handbook of research synthesis and meta-analysis*, 2:51–71.

Shijie Xia, Yiwei Qin, Xuefeng Li, Yan Ma, Run-Ze Fan, Steffi Chern, Haoyang Zou, Fan Zhou, Xiangkun Hu, Jiahe Jin, et al. 2025. Generative ai act ii: Test time scaling drives cognition engineering.

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, et al. 2023. Darwin series: Domain specific large language models for natural science. *ArXiv preprint*, abs/2308.13565.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao Ma. 2024. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4643–4663, Mexico City, Mexico. Association for Computational Linguistics.

Andrew Yates, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2666–2668. ACM.

Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Rui Li, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. 2024. Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. *ArXiv preprint*, abs/2403.00165.

Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. 2022. A survey on neural open information extraction: Current status and future directions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5694–5701. ijcai.org.

Xuanhe Zhou, Guoliang Li, and Zhiyuan Liu. 2023. Llm as dba. In *unknown*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *ArXiv preprint*, abs/2308.07107.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning T5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2308–2313. ACM.

A Prompt Templates

In the preprocessing stage, our method uses `Qwen3-8B-Instruct` (Qwen et al., 2024) to perform zero-shot information extraction (Wei et al., 2023; Sainz et al., 2024), instructing the model to extract document-level semantic features such as categories (Sun et al., 2023b; Zhang et al., 2024), sections (Zhou et al., 2023), and keywords (Rose et al., 2010; Lee et al., 2023). During inference, we provide a zero-shot, model-agnostic reranking framework, and evaluate its performance on `Qwen3-32B-Instruct` (Qwen et al., 2024), `Gemini 2.0 Flash` (Google DeepMind, 2025), and `GPT-4.1-mini` (OpenAI, 2024).

Here, we include the prompts used for different types of zero-shot document-level information extraction and model-agnostic reranking.

A.1 Extracting Categories

Prompt Template

Please analyze this document for its topic and categories:
{document}
Provide a comprehensive analysis that includes:
1. The broad category (coarse-grained) this document belongs to
2. The specific category (fine-grained) within that broad category
3. A concise, title-like description of the document's topic
Deliver the analysis in one concise paragraph.

A.2 Extracting Sections

Prompt Template

Identify 3-8 logical sections that would effectively organize this document's content:
{document}
Generate appropriate subtitle-style headings for each section that would help structure the document. Sections should be comprehensive and cover the full scope of the content.

A.3 Extracting Keywords

Prompt Template

Extract a comprehensive list of at least 30 diverse keywords and concepts from this document:
{document}
Generate as many diverse, relevant keywords and concepts as possible. Include both specific terms and broader conceptual themes.

A.4 Generating Pseudo Queries

Prompt Template

Generate 20 diverse search queries that users might enter to find this document:
{document}
Create different types of queries that cover various aspects of the document content. Queries should be diverse in wording, length, and specificity.

A.5 Reranking

Prompt Template

You are an LLM reranker, an intelligent assistant that can rank passages based on their relevancy to the query.
I will provide you with {num} passages (either represented by full text, previous user query, keywords or structured analysis), each indicated by a numerical identifier [].
Rank the passages based on their relevance to the search query: {query}.
[1]{passage}
...
[n]{passage}
Search Query: {query}.
Rank the {num} passages above based on their relevance to the search query. All the passages should be in descending order of relevance.
The output format should be [passage_id] > [passage_id] > ..., (If the full list is very long, generate at least 10) e.g., [4] > [2] > ... Only respond with the ranking results, do not say any word or explain.

B Case Studies

B.1 Distribution of Positives and Negatives

While metrics like nDCG@10 quantify the overall reranking quality, they do not indicate which specific reranking judgement contribute to the improvements. To better understand where our performance gains and limitations come from, we perform a qualitative analysis of how the position of ground-truth documents changes in the ranked list with or without CoRank.

We compare CoRank with the vanilla zero-shot listwise reranker on `LitSearch` (Ajith et al., 2024) with `GPT-4.1-mini` (OpenAI, 2024). Based on the rank position of the ground-truth documents and whether they fall within the top-10 (i.e., Recall@10), we obtain the distribution of positive cases and negative cases.

Analysis on Shared Positives. As shown in figure 4, CoRank successfully covers the majority of positive cases (69 out of 71) of the vanilla reranker. This is expected, as the fine-grained reranking stage in CoRank

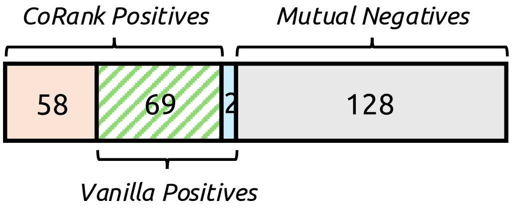


Figure 4: Distribution of Recall@10 positive and negative in Vanilla listwise reranking and CoRank.

inherits the full-text completeness in document representation of the vanilla listwise reranking.

Analysis on Exclusive Positives. At the same time, CoRank exclusively attains 58 positive cases. Analyzing these cases, we find that most (57 out of 58) were missed by the vanilla reranker because their first-stage retrieval ranks were too low (i.e., outside the top 20). This supports our analysis in Section 2.2 on the key challenge in scientific-domain reranking: truly relevant documents may be excluded from reranking due to suboptimal first-stage retrieval results. CoRank mitigates this issue with the compactness of feature-based representations in the coarse reranking stage, allowing more candidates to be considered and increasing the likelihood of recovering relevant documents.

Next, we present full instances of both positive and negative examples, along with their ranks in the first-stage retrieval, coarse reranking stage, and fine-grained reranking stage. This allows us to analyze the underlying causes of success or failure in each case.

B.2 Positive Example

This example clearly demonstrates the advantages of our method. The ground-truth document in this case involves long-tailed scientific concepts such as CGA and VAE, which are not well captured by the first-stage retriever. As a result, it is ranked only 92nd in the initial retrieval, making it inaccessible to vanilla rerankers that operate on the top 20 candidates. Even with the sliding window strategy (Sun et al., 2023a; Ma et al., 2023), the document would need to consistently win across 9 windows to enter the top-10, highlighting the limitations of existing reranking methods in scientific domains, as discussed earlier. In contrast, our method successfully ranks this document as the top candidate during the coarse reranking stage. This illustrates two key points: (1) our feature-based representation is highly compact, allowing us to expand the reranking pool to 200 candidates and include low-ranked but relevant documents such as this one; and (2) it is also sufficiently informative, enabling the reranker to recognize and promote the document directly to the top of the list once included.

Query

Can you recommend a foundational paper that provides a scalable framework for generating English sentences with controllable semantic and syntactic attributes for the purpose of augmenting datasets in NLP tasks?

Ground Truth Document

Control, Generate, Augment: A Scalable Framework for Multi-Attribute Text Generation: We introduce CGA, a conditional VAE architecture, to control, generate, and augment text. CGA is able to generate natural English sentences controlling multiple semantic and syntactic attributes by combining adversarial learning with a context-aware loss and a cyclical word dropout routine. We demonstrate the value of the individual model components in an ablation study. The scalability of our approach is ensured through a single discriminator, independently of the number of attributes. We show high quality, diversity and attribute control in the generated sentences through a series of automatic and human assessments. As the main application of our work, we test the potential of this new NLG model in a data augmentation scenario. In a downstream NLP task, the sentences generated by our CGA model show significant improvements over a strong baseline, and a classification performance often comparable to adding same amount of additional real data.

First-Stage Retrieval Ranking

92 of 14256

Compact Representation

Natural Language Processing (NLP) -> Text Generation and Neural Machine Translation -> Conditional VAE-Based Framework for Controllable and Scalable Multi-Attribute Text Generation with Applications in Data Augmentation: CGA for Data Augmentation in NLP Tasks (Text Generation, Multi-Attribute Control, Data Augmentation, Semantic Attributes)

Coarse Reranking

1 of 200

Fine-grained Reranking

1 of 20

B.3 Negative Example

The example above illustrates a case where our method fails to rank the ground-truth document highly—not due to representation quality, but because the document was ranked extremely low (319th) by the first-stage retriever. At such a low position, it is not visible to vanilla reranking methods, nor to those with extended context windows such as the sliding window strategy (Sun et al., 2023a; Ma et al., 2023) or our Method.

As a result, even though our compact representation accurately captures the document's semantic content, it never enters the reranking process and thus has no opportunity to be promoted to a higher position.

Query

Are there any studies that explore post-hoc techniques for hallucination detection at both the token- and sentence-level in neural sequence generation tasks?

Ground Truth Document

Detecting Hallucinated Content in Conditional Neural Sequence Generation: Neural sequence models can generate highly fluent sentences, but recent studies have also shown that they are also prone to hallucinate additional content not supported by the input. These variety of fluent but wrong outputs are particularly problematic, as it will not be possible for users to tell they are being presented incorrect content. To detect these errors, we propose a task to predict whether each token in the output sequence is hallucinated (not contained in the input) and collect new manually annotated evaluation sets for this task. We also introduce a method for learning to detect hallucinations using pretrained language models fine tuned on synthetic data that includes automatically inserted hallucinations Experiments on machine translation (MT) and abstractive summarization demonstrate that our proposed approach consistently outperforms strong baselines on all benchmark datasets. We further demonstrate how to use the token-level hallucination labels to define a fine-grained loss over the target sequence in low-resource MT and achieve significant improvements over strong baseline methods.We also apply our method to word-level quality estimation for MT and show its effectiveness in both supervised and unsupervised settings 1.

Compact Representation

Natural Language Processing -> Hallucination Detection in Neural Sequence Generation -> Token-Level Hallucination Detection in Conditional Text Generation: detecting hallucination using token-level classification and pretrained language models (Hallucination Detection, Neural Sequence Generation, Token-Level Classification, Machine Translation (MT))

First-Stage Retrieval Ranking

319 of 14256

Coarse Reranking

N/A

Fine-grained Reranking

N/A

C Proprietary Model API Cost

As part of our evaluation of different reranking methods, we tested two commercial LLMs: `Gemini 2.0 Flash` (Google DeepMind, 2025) and `GPT-4.1-mini` (OpenAI, 2024). Here, we report the API costs incurred in the main experiments when querying these models. The costs associated with experiments in the Further Analysis section have already been presented as part of the token efficiency comparison.

Model	**Price**	**Setting**	**# Token**	**Cost ($)**
`Gemini 2.0 Flash`	$0.1/M	Vanilla $_{Full}$	1.22M	$0.12
		CoRank $_{Full}$	3.32M	$0.33
		Vanilla $_{Sliding}$	10.98M	$1.10
		CoRank $_{Sliding}$	14.08M	$1.41
		Total Cost	**29.61M**	**$2.96**
`GPT-4.1-mini`	$0.4/M	Vanilla $_{Full}$	1.22M	$0.12
		CoRank $_{Full}$	3.32M	$1.33
		Vanilla $_{Sliding}$	10.98M	$4.39
		CoRank $_{Sliding}$	14.08M	$5.63
		Total Cost	**29.61M**	**$11.84**

Table 6: Reranking token usage and cost across different settings for `Gemini 2.0 Flash` and `GPT-4.1-mini` (OpenAI, 2024). Prices are per 1 million tokens.

As shown in Table 6, across all different settings in the main experiments, we spent $2.96 on `Gemini-2.0-Flash` (Google DeepMind, 2025) and $11.84 on `GPT-4.1-mini` (OpenAI, 2024), resulting in a total API cost of $14.80.

D Future Work

In the previous section, we identified two main limitations of our current approach. These suggest promising directions for future research that could further improve the generality and effectiveness of our method.

D.1 Improving Feature Extraction

Our current pipeline uses zero-shot information extraction (Wei et al., 2023) with general-purpose LLMs (Qwen et al., 2024) to generate semantic features such as categories (Sun et al., 2023b; Zhang et al., 2024) and keywords (Rose et al., 2010; Lee et al., 2023). While this approach is simple and has proven effective, future work could explore more advanced techniques to enhance extraction quality. For example, using domain-specific LLMs (Taylor et al., 2022; Xie et al., 2023) or introducing multi-turn feedback (Shinn et al., 2023) may yield richer and more accurate representations, further improving downstream reranking performance.

D.2 Extending to General Domains

Our method is designed specifically for scientific retrieval (Lawrence et al., 1999; White et al., 2009), where challenges like long-tail terminology (Kang et al., 2024a) and limited first-stage recall (Kim et al., 2023) are particularly severe. An important direction for future work is to investigate the applicability of our framework in general-domain retrieval settings (Zhu et al., 2023; Ai et al., 2023). This includes evaluating how well the feature-based representation and the hybrid reranking strategy transfer beyond the scientific context, and identifying any necessary adaptations.