Chain of Retrieval: Multi-Aspect Iterative Search Expansion and Post-Order Search Aggregation for Full Paper Retrieval

Anonymous ACL submission

Abstract

Scientific paper retrieval, particularly framed as document-to-document retrieval, aims to identify relevant papers in response to a long-form query paper, rather than a short query string. Previous approaches to this task have focused exclusively on abstracts, embedding them into dense vectors as surrogates for full documents and calculating similarity between them. Yet, abstracts offer only sparse and high-level summaries, and such methods primarily optimize one-to-one similarity, overlooking the dynamic relations that emerge among relevant papers during the retrieval process. To address this, we propose Chain of Retrieval (CoR), a novel iterative framework for full-paper retrieval. Specifically, CoR decomposes each query paper into multiple aspect-specific views, matches them against segmented candidate papers, and iteratively expands the search by promoting top-ranked results as new queries, thereby forming a tree-structured retrieval process. The resulting retrieval tree is then aggregated in a post-order manner: descendants are first combined at the query level, then recursively merged with their parent nodes, to capture hierarchical relations across iterations. To validate this, we present SciFullBench, a large-scale benchmark providing both complete and segmented contexts of full papers for queries and candidates, and results show that CoR significantly outperforms existing retrieval baselines.

1 Introduction

Information Retrieval (IR) is the task of searching for query-relevant documents from a large external corpus, evolving from sparse keyword matching (Sparck Jones, 1972; Robertson et al., 1995) to dense representation-based similarity (Karpukhin et al., 2020; Izacard et al., 2021). Notably, in the era of Large Language Models (LLMs) (Achiam et al., 2023; Team et al., 2023; Dubey et al., 2024; DeepSeek-AI et al., 2025), IR has become increasingly crucial, which allows LLMs to utilize up-to-date external information (Lewis et al., 2020).

In contrast to conventional retrieval tasks, whose queries are short (such as questions or keywords), scientific paper retrieval poses unique challenges. Specifically, queries are structured, long-form documents that encapsulate diverse aspects, ranging from research motivation and proposed methodology to experimental design and empirical findings. Also, relevance in this setting is inherently multi-faceted, as a paper may be considered relevant for various reasons, such as pursuing similar research objectives or employing comparable methods.

However, primarily due to the scarcity of realistic full-paper benchmarks, prior work has focused on abstract-level representations as a proxy for full papers (Cohan et al., 2020; Yasunaga et al., 2022; Ostendorff et al., 2022; Singh et al., 2023; Zhang et al., 2023a), fine-tuning models on abstract pairs connected by citation relationships. While they are effective to some extent, abstracts are inherently sparse and offer only high-level summaries, making them insufficient to capture the nuanced, multi-faceted relationships between scientific works. As a result, abstract-based approaches are limited to shallow similarity and struggle with tasks requiring deeper contextual understanding of full papers beyond surface-level abstracts, such as identifying complementary works, synthesizing literature reviews, or generating novel research directions (Asai et al., 2024; Baek et al., 2024; Chamoun et al., 2024; Jin et al., 2024; D'Arcy et al., 2024).

Nevertheless, it is non-trivial to represent and retrieve full scientific papers. On the one hand, full papers often exceed 100K tokens in length, far surpassing the context limits of most embedding models. Even if encoded into a single vector, representing its diverse aspects (e.g., motivation, methods, and experiments) within a single embedding may lead to oversimplification and blur fine-grained distinctions, particularly when relevance is tied to a

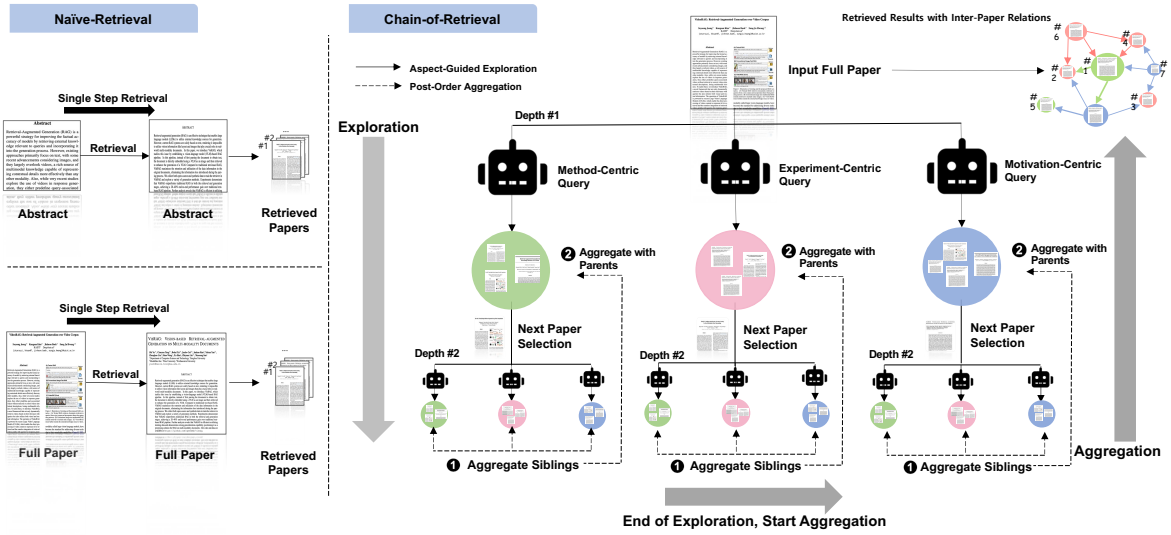

Figure 1: Conceptual illustration of our retrieval method compared to prior retrieval approaches.

specific aspect. Moreover, existing approaches typically optimize semantic similarity between isolated pairs of papers, overlooking the broader collective relations that emerge among relevant works during retrieval. This design stands in contrast to human information-seeking behavior in the information foraging theory (Pirolli and Card, 1995, 1999), which is dynamic and iterative: humans continuously refine their understanding by comparing newly encountered works with prior knowledge.

To address this, we propose Chain of Retrieval (CoR), a novel framework designed to capture dynamic inter-paper relevance while fully leveraging the context of scientific papers, operated through a two-phase process: *Exploration* and *Aggregation*. In the *Exploration* phase, the query paper is decomposed into multiple aspect-specific views, and each aspect is handled by a specialized LLM-powered query optimizer, which is further trained via Direct Preference Optimization (DPO) (Rafailov et al., 2023) on a self-generated preference to mitigate the shortage of realistic supervision for multi-retrieval systems. After retrieving with these aspect-centric queries, the retrieved (non-duplicate) papers most semantically aligned with the original query are promoted for the next iteration, recursively expanding the search, forming a tree-structured retrieval.

In the *Aggregation* phase, results across different depths are merged to capture higher-order relations among retrieved works. Specifically, inspired by the notion of triadic closure in the network theory (Granovetter, 1973), we design an algorithm that first merges top-k results among sibling nodes (papers retrieved from the same parent), then combines these aggregated results with those of their parent node, and continues this process bottom-up until convergence at the root. This hierarchical aggregation reinforces strong multi-hop connections while naturally attenuating weaker, indirect signals, yielding a robust ranking of relevant papers.

To evaluate our CoR framework, we further introduce SciFullBench, a large-scale benchmark suite for paper-to-paper retrieval in the ML and NLP domains, which provides complete papers as both queries and retrieval targets. Notably, this enables a more realistic and challenging evaluation setting, capturing the nuanced, multi-aspect relevance that arises only when full contexts are available and leveraged, unlike existing benchmarks designed for abstract-level retrieval. Experimental results on SciFullBench demonstrate that CoR consistently outperforms retrieval baselines (that either rely on abstracts or naively encode full papers), while remaining compatible with diverse, domain-agnostic embedding models, highlighting both the robustness and generality of our CoR.

2 Related Work

Scientific Paper Retrieval Scientific paper retrieval aims to identify relevant works given a query, which may take the form of short queries (Ajith et al., 2024), research proposals (Garikaparthi et al., 2025), or full papers, and the latter being the focus of this work. Pioneering work in scientific paper-

to-paper retrieval used bibliometric statistics, such as influence scores (Zhou et al., 2012; Mohapatra et al., 2019), co-citations (Small, 1973; Haruna et al., 2018), and bibliographic coupling (Kessler, 1963). Recently, thanks to the capability of neural models, many studies have focused on calculating semantic textual similarities between abstracts of respective documents (Bhagavatula et al., 2018; Ostendorff, 2020), with embedding models specific to this domain. For example, Cohan et al. (2020) and Ostendorff et al. (2022) fine-tune BERT-based models (Devlin et al., 2019) using abstract pairs extracted from the citation graph, and Mysore et al. (2022) further consider the Wasserstein distance between sentence segments within abstract pairs. Moreover, recent studies target multiple tasks (such as paper classification and citation prediction in addition to paper retrieval) in a unified framework by generating their respective representations adaptively (Singh et al., 2023; Zhang et al., 2023a). Complementary to these, hybrid approaches integrate bibliometric signals with vector-based retrieval, reweighting ranks via post-hoc adjustments (Hamedani et al., 2016; Guo et al., 2022). However, such statistical indicators (e.g., citations) develop only after extended periods of scholarly activity, making them less suited for the timely discovery of emerging research. In contrast, we propose modeling inter-paper relations during inference, guided by aspect-driven queries derived from the full context of scientific papers.

Query Optimization with LLMs Effective formulation of queries is central to retrieval performance, and has long been studied from classical relevance feedback approaches (Rocchio Jr, 1971; Salton and Buckley, 1990) to query expansion techniques (Kuzi et al., 2016; Nogueira et al., 2019). More recent methods either leverage LLMs themselves to reformulate queries (Yu et al., 2023a; Wang et al., 2023; Gao et al., 2023), or further augment them with external query-relevant information retrieved in an auxiliary step (Yu et al., 2023b; Shen et al., 2024; Park and Lee, 2024; Lei et al., 2024). Another line of work seeks to reduce query ambiguity by decomposing a complex query into smaller queries (Zheng et al., 2024; Korikov et al., 2024). We note that our approach aligns with this decomposition paradigm but extends it to a more challenging setting: instead of handling short, user-issued queries, we decompose long-form scientific documents into multiple aspect-specific subqueries.

3 Methodology

3.1 Preliminary

Paper-to-Paper Retrieval Given a query paper D (with its abstract as D_{abstract}), paper retrieval is to return a ranked list of relevant candidate papers from the corpus C. In contrast to existing studies that use D_{abstract}, we utilize its complete version D for query formulation and corpus construction.

3.2 Aspect Aware Multi-Vector Retrieval

Aspect-Aware Query Optimization Scientific papers encapsulate multiple facets (e.g., research motivation, methodological design, and experimental validation), which might not be represented by a single query. To capture this diversity, we formulate the query optimization process, whose goal is to transform the full paper D into a set of aspect-specific queries. Formally, we define a family of query optimization functions: $\mathcal{F} = \{f_{\text{M}}, f_{\text{E}}, f_{\text{R}}\}$, where each function f maps D to a query $q = f(D)$ that targets a specific aspect. Notably, each function is instantiated with an LLM guided by an aspect-specific template (see Appendix H). In addition to these fine-grained queries, we also consider the abstract D_{abstract}, since it reflects a general and broad view of the overall content (that can complement the more specialized views), yielding the final query set: $\mathcal{Q} = \{f_i(D) \mid f_i \in \mathcal{F})\} \cup \{D_{\text{abstract}}\}$.

Retrieval with Multi-View Corpora Once the optimized query set $\mathcal{Q}$ is formulated, we perform retrieval individually for each query $q \in \mathcal{Q}$. Specifically, for the abstract-based query D_{abstract}, we use a corpus $\mathcal{C}_{\text{abstract}}$ that contains candidate abstracts (as they are comparable in length and structure). For all other aspect-specific queries derived from the full paper, we use a segmented version of the full corpus: $\mathcal{C}_{\text{chunked}}$, where each paper is split into fixed-length chunks (likely to capture its specific aspect) and indexed using multi-vector representations (Khattab and Zaharia, 2020; Santhanam et al., 2021). Each query then retrieves its top-k relevant segments, which are mapped back to their source papers via the mapping function $h(x)$, as follows: $\mathcal{R}_q = \{h(x) \mid x \in \texttt{Retrieve}(q, C, k)\}$. Finally, the resulting ranked lists of candidate papers from all queries are aggregated to form the multi-view retrieval pool, denoted as follows: $\mathcal{R} = \bigcup_{q \in \mathcal{Q}} \mathcal{R}_q$.

3.3 Iterative Chain-of-Retrieval

Iterative Aspect-Aware Search Expansion The exploration phase aims to iteratively expand the

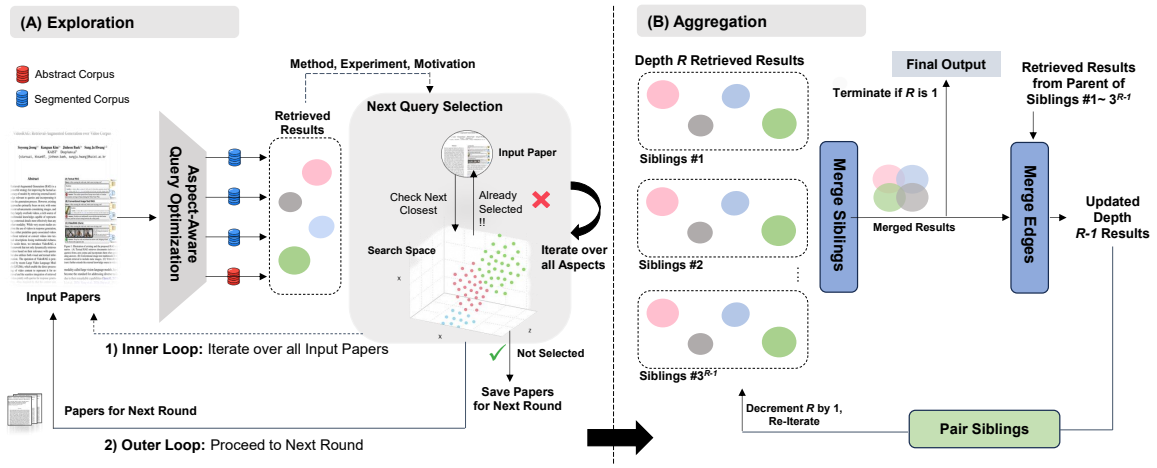

Figure 2: Illustration of our Chain-of-Retrieval (CoR) framework, consisting of (A) Exploration and (B) Aggregation.

search space by promoting promising retrieved results as new queries. Specifically, starting from the root query paper, we branch into aspect-specific queries (namely, motivation, methods, and experiments), using the process described in Section 3.2, while excluding the abstract view. Then, each query retrieves a set of top-k candidate papers from the corpus, which serve as the first layer of expansion. At each subsequent depth r, every aspect-specific branch selects a single representative work from its top-k results and promotes it as the next query paper, ensuring continuity of semantic relevance while avoiding divergence from the original query intent, and this query paper is further decomposed into aspect-specific queries for the next iteration. At the end, this process grows a tree-structured retrieval space: after r iterations, the search expands into $4(3)^{r-1}$ distinct exploration paths, each corresponding to an aspect-driven exploration chain.

To avoid redundancy and ensure efficiency, we store previously selected papers in aspect-specific caches: $\textsc{Cache}[a]$ for $a \in \{\mathrm{M}, \mathrm{E}, \mathrm{R}\}$, where each cache records documents with respect to the initial branching aspects of the root query paper, so that any paper stored in the corresponding cache is excluded from future expansions within that branch (to avoid repeated visits to the same works). Further, every retrieved top-k set across depths is preserved in a memory M, which latter supports aggregation across multiple levels of the search tree.

Recursive Post-Order Aggregation After R rounds of expansion, we aggregate results in a bottom-up, post-order fashion. Starting from the leaf nodes (i.e., the final retrieved sets at depth R), we first merge sibling results that share the same

Algorithm 1 Chain-of-Retrieval (CoR)

Require: Root paper D; Max depth R; Top-k per query K; Corpora $\mathcal{C} = \{\mathcal{C}_{\text{abstract}}, \mathcal{C}_{\text{chunked}}\}$
Ensure: Final top-k candidates for D

$Q \leftarrow [(D, \textsc{Root})]$
$\mathcal{M} \leftarrow [\,]$
$\textsc{Cache}[a] \leftarrow \emptyset \;\; \forall a \in \{\mathrm{M}, \mathrm{E}, \mathrm{R}\}$
for $r = 0$ **to** $R - 1$ **do** ▷ **Exploration**
  $\mathcal{M}[r] \leftarrow [\,], \;\; Q_{\text{next}} \leftarrow [\,]$
  for all $(d, \textsc{parent}) \in Q$ **do**
    $P \leftarrow \textsc{OneHopRetrieval}(d, \textsc{parent}, \mathcal{C}, K)$
    $\mathcal{M}[r].\textsc{Extend}(P)$
    $Q_{\text{next}} \leftarrow \textsc{SelectNextQuery}(P, \textsc{Cache}, \gamma)$
  $Q \leftarrow Q_{\text{next}}$
for $r = R - 1$ **downto** 1 **do** ▷ **Aggregation**
  $S \leftarrow \textsc{MergeSiblings}(\mathcal{M}[r])$
  $\mathcal{M}[r-1] \leftarrow \textsc{MergeEdges}(\mathcal{M}[r-1], S)$
return $\textsc{MergeSiblings}(\mathcal{M}[0])$

parent using Reciprocal Rank Fusion (RRF) (Cormack et al., 2009), defined as follows: $\mathrm{RRF}(P) = \sum_{q \in \mathcal{Q}} \frac{1}{k + \mathrm{rank}_q(P)}$, where $\mathrm{rank}_q(P)$ denotes the position of paper P in the ranked list retrieved by query q, and k is a smoothing constant. This step ($\textsc{MergeSiblings}$ in Algorithm 1) emphasizes documents consistently ranked highly across different aspect queries. The merged sibling sets are then recursively combined upward with their parent results ($\textsc{MergeEdges}$), gradually consolidating evidence across the retrieval tree until the root.

It is worth noting that, through this recursive process, rank signals from deeper layers are naturally decayed: RRF suppresses the influence of lower-ranked items, papers retrieved only at later depths contribute less to the final ranking. Formally, the effective weight of rank signals from depth R decreases at a rate proportional to $\sigma(2\sigma)^{R-1}$, where σ is the number of query sets per retrieval. As a result,

papers strongly and repeatedly supported across multiple aspects and iterations are reinforced, while weaker or spurious signals are attenuated, yielding a robust top-k ranking for the original query paper.

3.4 Multi-Aspect Preference Optimization

Offline Policy Exploration To improve the quality of aspect-specific queries, we further train each query-generation agent f with the Direct Preference Optimization (DPO) algorithm (Rafailov et al., 2023). A key challenge, however, lies in constructing reliable agent-specific preference datasets: collecting human-labeled preferences for multi-agent systems is both costly and impractical. Moreover, recent studies emphasize that initializing DPO with the Supervised Fine-Tuned (SFT) model is crucial to mitigate distributional shift issues (Feng et al., 2024; Xu et al., 2024; Meng et al., 2024).

To address these challenges, we follow the synthetic preference construction procedure of Meng et al. (2024), treating an off-the-shelf instruction-tuned LLM ($\mathcal{P}_\theta$) as the SFT policy ($\mathcal{P}_{\text{SFT}}$). Specifically, for each input paper D, we generate a set of k rolled-out candidate queries from every functional agent: $Q_d^{\text{Exp}} = \bigcup_{f \in \{f_{\text{M}}, f_{\text{E}}, f_{\text{R}}\}} \{q_{(D,f)}^{(1)}, \ldots, q_{(D,f)}^{(k)}\}$. After that, among these candidates, we identify the best and worst queries according to a reward function: $q_{(D,f)}^{\text{Best}} = \arg\max_j \texttt{Reward}(q_{(D,f)}^{(j)})$ and $q_{(D,f)}^{\text{Worst}} = \arg\min_j \texttt{Reward}(q_{(D,f)}^{(j)})$. In addition, a preference pair is retained only if the reward gap between those two exceeds a margin τ, ensuring that $\mathcal{D}_{\text{pref}}$ contains discriminative signals for training.

Reward Formulation Designing an effective reward function is crucial for preference optimization, as it determines how query-generation agents are guided to improve. Our principle is to provide reward signals that are reliable and reproducible, cost-efficient to compute, and directly aligned with retrieval quality; thus, instead of relying on qualitative judgments from LLMs (LLMs-as-judge), which are costly and potentially inconsistent, we leverage citation links as environmental feedback, providing direct, objective reward signals (which we use Recall@k in practice). Also, for each agent $f \in \{f_{\text{M}}, f_{\text{E}}, f_{\text{R}}\}$, we define an independent reward function $\texttt{Reward}_f$, such that the reward for one agent does not depend on the outputs of others, under the assumption that reinforcing the quality of aspect-specific queries individually is sufficient, since the final retrieval results are aggregated across agents via a linear combination of their outputs.

4 SciFullBench

To evaluate full paper-to-paper retrieval, we construct SciFullBench, a large-scale benchmark for full papers beyond abstract-level proxies. Below, we outline its construction process; additional details are provided in Appendix A.

Query Formulation To support the task of full paper-to-paper retrieval, we collect papers in ML (NeurIPS and ICLR) and NLP (ACL and EMNLP) venues from OpenReview API[1], ACL-Anthology[2], and SEA (Yu et al., 2024), since no single source covers all of them. For each query paper in them, we annotate its ground-truth relevant papers based on their neighboring relationships in the academic graph, such as **References** (outgoing links) and **Citations** (incoming links), from the perspective of query papers. In addition, we parse each document using the AllenAI Science Parse tool[3], and further apply post-processing to remove reference sections, filter citation markers, and eliminate their associated in-context mentions (see Appendix A). Lastly, from the set of papers with at least 10 relevant papers on both the reference and citation criteria, we construct our evaluation suite by randomly sampling 400 query papers per venue.

Corpus Construction To construct a comprehensive corpus for retrieval, we collect papers in the category around CS from 2020 to 2025 in arXiv[4], which offers open access to the full content. Also, to avoid exposing citation or reference information, we apply the same parsing and filtering procedures used in query construction. Lastly, we include labeled documents for all query papers in the corpus, resulting in a collection of about 40K papers with both abstract and full-text available. For segmentation of each document into finer units, we use the tokenization from NLTK (Loper and Bird, 2002).

5 Experiments

5.1 Experimental Setup

We describe here the setup used to evaluate our method. Further implementation details and training procedures are in Appendix B and Appendix C.

Retrieval Models We compare our method with existing retrievers developed for scientific paper re-

[1] https://openreview.net/
[2] https://aclanthology.org/
[3] https://github.com/allenai/science-parse
[4] https://arxiv.org/

Table 1: Main results of various domain-specific and domain-agnostic retrievers on SciFullBench.

	Methods	ICLR-NeurIPS				ACL-EMNLP			
		References		Citations		References		Citations	
		Recall@200	Recall@300	nDCG@200	Recall@300	Recall@100	Recall@200	mAP@30	Recall@100
	Domain-Specific Retrievers (A2A)								
	SciNCL	45.91	51.80	30.87	47.90	28.21	35.58	5.27	25.66
	SPECTER2-Base	44.79	50.41	32.36	49.09	26.88	34.47	5.77	27.10
	SPECTER2-Adapter-MTL CTRL	45.16	51.07	31.07	47.38	27.35	34.44	5.35	25.55
	SciMult-MHAExpert	39.56	45.04	26.36	41.07	24.67	31.70	4.62	22.52
Domain-Agnostic Retrievers	**Jina-Embeddings-v2-BASE-EN**								
	Abstract-to-Abstract (A2A)	44.83	49.92	31.77	47.89	26.92	33.57	5.72	26.07
	Full-to-Full (F2F)	45.22	50.60	33.74	49.63	29.15	36.72	6.68	28.33
	CoR w/ Llama-3.2-3B-Instruct	**49.64**	**55.50**	**36.25**	**56.21**	**31.36**	**39.51**	**7.12**	**32.94**
	CoR w/ QWEN-2.5-3B-Instruct	**49.70**	**55.78**	**37.04**	**56.97**	**31.43**	**39.69**	**7.29**	**34.15**
	BGE-M3								
	Abstract-to-Abstract (A2A)	39.44	44.09	28.57	42.85	24.69	30.69	5.06	24.57
	Full-to-Full (F2F)	41.19	45.84	30.35	44.01	24.98	31.53	5.61	24.45
	CoR w/ Llama-3.2-3B-Instruct	**44.32**	**49.39**	**32.64**	**49.27**	**27.31**	**34.53**	**6.31**	**29.33**
	CoR w/ QWEN-2.5-3B-Instruct	**45.34**	**50.43**	**33.73**	**50.77**	**27.71**	**35.31**	**6.38**	**29.85**
	Inf-Retriever-v1-1.5B								
	Abstract-to-Abstract (A2A)	55.90	61.72	36.46	54.63	36.14	44.82	7.00	31.53
	Full-to-Full (F2F)	38.21	42.27	33.75	48.79	35.50	43.10	7.40	31.21
	CoR w/ Llama-3.2-3B-Instruct	**59.26**	**65.31**	**41.36**	**61.54**	**38.35**	**48.06**	**8.59**	**37.34**
	CoR w/ QWEN-2.5-3B-Instruct	**58.92**	**64.78**	**41.87**	**62.06**	**38.27**	**47.84**	**8.99**	**38.10**

trieval, as follows: **SPECTER2 Base** (Singh et al., 2023); **SciNCL** (Ostendorff et al., 2022); **SciMult-MHAExpert** (Zhang et al., 2023a); **SPECTER2 Adapters + MTL CTRL** (Singh et al., 2023). We also consider off-the-shelf general-purpose embedding models, such as **Jina-Embeddings-V2-Base-EN** (Günther et al., 2023), **BGE-M3** (Chen et al., 2024a), and **Inf-Retriever-v1-1.5B** (Junhan Yang, 2025), leveraged as backbones.

Retrieval Units We consider various retrieval units, where we denote **A** for *Abstract*, **F** for *Full paper*[5], and **C** for *chunked context* (i.e., segmented units of the full paper, each capped at 3K tokens). Using them, we consider four retrieval setups: **A2A** (Abstract-to-Abstract), **F2F** (Full-to-Full), **A2C** (Abstract-to-Chunk), and **F2C** (Full-to-Chunk).

Query Optimizers We instantiate query optimizers using LLMs: the open-weight instruct models Llama-3.2-3B-Instruct (Meta, 2024) and QWEN-2.5-3B-Instruct (Yang et al., 2024; Team, 2024). For reproducibility, we set the temperature to 0 and the maximum generation tokens to 2000.

Training and Inference Configuration For each query, we retrieve the top-300 candidate papers. To prevent repeatedly selecting the same candidate as the next query, we always skip the highest-ranked results (selected previously) and instead promote the next-best. During preference pair construction, we generate 16 candidate queries per aspect and evaluate them based on Recall@30, while excluding pairs whose reward margin is smaller than 3%.

[5]Since full papers often exceed the context length of embedding models, we truncate them up to the maximum length.

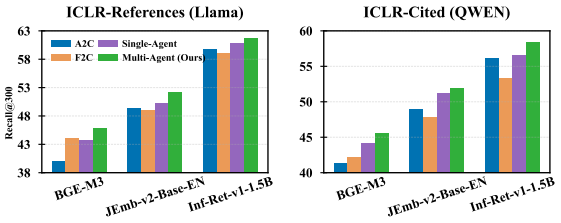

Figure 3: Retrieval performance comparing different query formulations against our multi-agent CoR framework (after a single round of retrieval and with untrained query optimizers).

5.2 Experimental Results and Analysis

Main Results Table 1 presents the main results, where CoR outperforms all baselines across various settings, validating the effectiveness of our proposed framework for full paper-to-paper retrieval. Notably, when using the same domain-agnostic retriever, CoR surpasses abstract-to-abstract (A2A) baselines by an average of 5.16% in Recall, demonstrating that simply relying on abstracts is suboptimal compared to our aspect-driven approach. Also, interestingly, general-purpose embedding models such as Jina-Embeddings-V2-Base-EN and BGE-M3 (whose A2A performance lags behind representative domain-specific retrievers) achieve competitive or superior performance once integrated into CoR, emphasizing that CoR serves as a plug-and-play framework, delivering strong performance regardless of the retrievers. Lastly, when compared against F2F retrieval with long-context embedding models, CoR achieves average gains of 6.8% in Recall, confirming that simply embedding entire

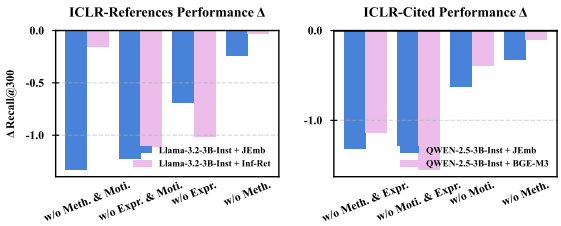

Figure 4: Change (Δ) in retrieval performance (relative to using all aspects) when excluding individual scientific aspects (Moti = Motivation, Meth = Method, Expr = Experiment).

Table 2: Performance comparison with and without the Aspect-Aware Cache in the selection algorithm, reported after retrieval depths of three using DPO-trained query optimizers.

	ICLR-References	ICLR-Citations
{Optimizer} + {Retriever}	Recall@300	Recall@200
Llama-3.2-3B-Inst + JEmb-v2		
w/o Aspect-Aware Cache	53.76	46.81
w/ Aspect-Aware Cache	**54.60**	**47.12**
Llama-3.2-3B-Inst + BGE-M3		
w/o Aspect-Aware Cache	46.83	40.98
w/ Aspect-Aware Cache	**47.24**	**41.44**
Llama-3.2-3B-Inst + Inf-Ret-v1		
w/o Aspect-Aware Cache	62.76	51.77
w/ Aspect-Aware Cache	**63.19**	**52.33**

papers is insufficient, and that structured, iterative search expansion with post-order aggregation is crucial for effective full paper-to-paper retrieval.

Effectiveness of Specialized Query Optimizers Our results in Figure 3 confirm the effectiveness of using multiple specialized LLM agents, each dedicated to a single aspect. More specifically, using multiple specialized agents outperforms a single base agent tasked with generating multiple queries (without aspect separation). Also, when contrasted with setups that use either full papers or abstracts as queries (matched with the same chunked corpus) and that retrieve the same number of total candidates as CoR of a single retrieval round, our multi-aspect queries generated with specialized agents achieve gains of 3.36% and 3.33%, respectively, highlighting the advantage of decomposing noisy, monolithic queries into aspect-aware subqueries.

Importance of Multi-Aspect Coverage To examine the role of aspect diversity in query formulation, we analyze how retrieval performance varies with the number of participating query optimizer agents. As shown in Figure 4, excluding any single aspect leads to measurable performance degradation, which supports the necessity of capturing the multifaceted nature of scientific papers. Not all aspects, however, contribute equally: excluding experimental or research-motivation queries causes larger drops in performance than omitting methodological queries. We attribute this to the inherent characteristics of each aspect. Specifically, experimental and research-motivation views often yield stronger semantic overlap across related papers, grounded in shared entities such as tasks, baselines, and datasets. In contrast, we hypothesize that methodological details are more heterogeneous and less entity-driven, demanding deeper semantic abstraction and reasoning to identify commonalities.

Table 3: Performance comparison of specialized agents trained with DPO versus untrained query optimizers, evaluated after a single round of retrieval (i.e., one depth).

	ICLR-References	NeurIPS-Citations
{Optimizer} + {Retriever}	Recall@100	Recall@300
Llama-3.2-3B-Inst + JEmb-v2		
w/o DPO	36.37	57.46
w/ DPO	**37.05**	**58.26**
Llama-3.2-3B-Inst + BGE-M3		
w/o DPO	31.03	50.30
w/ DPO	**31.53**	**50.98**
Llama-3.2-3B-Inst + Inf-Ret-v1		
w/o DPO	45.16	63.25
w/ DPO	**45.79**	**63.91**
QWEN-2.5-3B-Inst + JEmb-v2		
w/o DPO	36.57	58.66
w/ DPO	**37.21**	**58.77**
QWEN-2.5-3B-Inst + BGE-M3		
w/o DPO	31.13	51.11
w/ DPO	**32.12**	**52.17**

Role of Aspect-Aware Cache Table 2 shows the benefits of incorporating the aspect-aware cache into our *Next Query Selection* strategy, which enables preventing redundant exploration within each aspect branch and thus encouraging more diverse search trajectories. As a result, variants with the aspect-aware cache consistently outperform those without it, demonstrating its effectiveness in steering the retrieval process toward broader coverage and stronger overall performance.

Efficacy of Query Preference Optimization To assess the contribution of reinforcing aspect-aware query optimizers with DPO, we conduct an ablation study isolating the DPO training. As shown in Table 3, integrating DPO consistently improves performance across diverse LLM–retriever configurations in both the reference and citation setups. Notably, although the preference pairs are constructed on the Reference setting with Recall (as the reward signal), the gains generalize beyond this setting.

Analysis on Post-Order Aggregation Figure 5 shows that CoR yields consistent performance gains across successive retrieval rounds, with the gap over baselines widening as depth increases. This highlights the effectiveness of our post-order

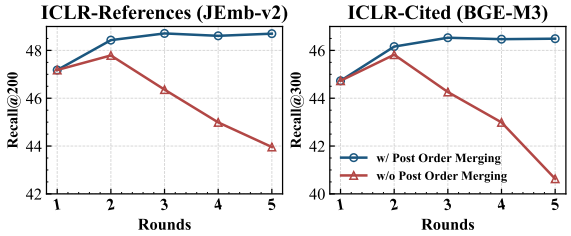

Figure 5: Retrieval results as a function of retrieval depths, with DPO-trained Llama-3.2-3B-Instruct as optimizers.

Table 4: Analysis on the semantic coverage of the Top@300 retrieved candidate documents for the ground truth documents with Convex Hull Volume. The results of CoR are reported after 3 rounds with Llama-3.2-3B-Inst as query optimizers.

	ACL (Citations)	NeurIPS (Citations)
{Type} w/ {Retriever}	CHV Ratio	CHV Ratio
Relative Coverage over A2A		
F2F w/ JEmb-v2	1.126 (+12.6%)	1.135 (+13.5%)
F2F w/ BGE-M3	1.088 (+8.8%)	1.069 (+6.9%)
F2F w/ Inf-Ret-v1	1.128 (+12.8%)	1.097 (+9.7%)
Relative Coverage over A2A		
CoR w/ JEmb-v2	**1.190 (+19.0%)**	**1.198 (+19.8%)**
CoR w/ BGE-M3	**1.107 (+10.7%)**	**1.097 (+9.7%)**
CoR w/ Inf-Ret-v1	**1.167 (+16.7%)**	**1.140 (+14.0%)**
Relative Coverage over F2F		
CoR w/ JEmb-v2	**1.095 (+9.5%)**	**1.078 (+7.8%)**
CoR w/ BGE-M3	**1.044 (+4.4%)**	**1.042 (+4.2%)**
CoR w/ Inf-Ret-v1	**1.060 (+6.0%)**	**1.072 (+7.2%)**

aggregation, which progressively attenuates weak signals from distant connections and thereby enhances robustness against noise in deeper rounds. By contrast, the baseline strategy (merging results from all branches at once) fails to adaptively penalize such signals, leading to steadily diminishing performance. However, improvements plateau beyond depth 3, which suggests diminishing returns, likely due to added noise with deeper expansions.

Diversity Analysis Recall that CoR expands the search space iteratively across multiple aspects of the query paper, and we further analyze its ability to capture semantic diversity compared to the vanilla retrieval settings. Specifically, to quantify this, we measure the semantic coverage of retrieved ground-truth candidates in the latent embedding space using the Convex Hull Volume Ratio (CHV Ratio), which compares coverage between two experimental configurations over the same test set (see Appendix C.3 for details). As shown in Table 4, CoR consistently improves the diversity of semantic coverage among retrieved relevant documents: relative to the baseline A2A setup, it achieves a 14.98% gain in CHV Ratio, surpassing the 10.72% gain of the F2F setup. Moreover, when directly compared against F2F, CoR still provides an additional 6.52% improvement. These results highlight that beyond improving retrieval accuracy, the proposed CoR enables broader semantic exploration, which allows discovering relevant papers that would likely be missed under the vanilla retrieval strategies.

Table 5: Experimental results on patent-to-patent retrieval using PATENTFULLBENCH (self-constructed; See Appendix 8). Results for CoR are reported after two retrieval rounds.

	References	Citations
Methods	Recall@100	Recall@100
Baseline		
PAT-SPECTER (A2A) (Ghosh et al., 2024)	47.36	52.16
PaECTER (A2A) (Ghosh et al., 2024)	52.77	56.25
BGE-M3 (A2A)	46.42	49.57
BGE-M3 (F2F)	52.30	55.81
Inf-Ret-v1 (A2A)	52.78	57.94
Inf-Ret-v1 (F2F)	56.23	58.59
JEmb-v2-Base-EN (A2A)	49.39	53.27
JEmb-v2-Base-EN (F2F)	56.57	59.61
CoR (Ours): {Optimizer} + {Retriever}		
GPT-4o-Mini-2024-0718 + BGE-M3	**54.71**	**56.78**
GPT-4o-Mini-2024-0718 + Inf-Ret-v1	**58.62**	**63.06**
GPT-4o-Mini-2024-0718 + JEmb-v2	**57.23**	**60.74**

Generalization to Other Domains To examine the generality of CoR beyond scientific paper retrieval, we further evaluate its performance on a different but challenging document-to-document retrieval task: patent-to-patent retrieval, which has substantially long queries (please see Appendix D for details on task design and implementation). As shown in Table 5, CoR delivers robust gains over both domain-agnostic and domain-specific baselines, surpassing F2F retrieval by 2.01% and A2A retrieval by 6.96%. Importantly, these improvements are achieved without any training (i.e., using the off-the-shelf GPT-4o for query optimization), highlighting the versatility and broad applicability of CoR for long document-to-document retrieval.

6 Conclusion

In this work, we introduced Chain of Retrieval (CoR), a novel framework for paper-to-paper retrieval that iteratively expands search space through aspect-aware query optimization (reinforced with DPO training) and recursively merges results via post-order aggregation. For realistic evaluation, we constructed SCIFULLBENCH, a large-scale benchmark designed for full-context scientific retrieval, and the results showcase that CoR consistently outperforms abstract-level and full-context baselines across diverse settings, even with off-the-shelf embedding models and for a different patent retrieval task, highlighting its robustness and generality.

Limitations

While our work introduces a novel retrieval approach that iteratively expands the search space via multi-aspect query optimization over the full context of scientific papers, it still has room for future work. First, the current approach lacks an adaptive branch pruning mechanism during the exploration phase, and thus the search results accumulate primarily as a function of retrieval depth (rather than being guided by the semantic alignment of individual papers with the root query). Although effective in practice, incorporating adaptive pruning strategies that account for semantic relevance at each depth would be an interesting direction for future work, to enhance both efficiency and precision. Second, while our framework effectively retrieves diverse and relevant papers, deeper exploration and repeated LLM inference introduce additional computational overhead and latency. While we partially mitigate this with a lightweight selection mechanism (in place of expensive LLM verifiers), further optimizations on it might be needed.

Ethics Statement

Although our Chain of Retrieval framework improves retrieval performance over prior approaches, it can still retrieve irrelevant or misleading papers, which may propagate incorrect or harmful information to both human users and downstream AI models. To ensure the development of trustworthy automated systems, future work should incorporate robust verification or re-ranking mechanisms that can effectively filter irrelevant or potentially harmful documents from the retrieved set.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. LitSearch: A retrieval benchmark for scientific literature search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15068–15083, Miami, Florida, USA. Association for Computational Linguistics.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, and 1 others. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

Parul Awasthy, Aashka Trivedi, Yulong Li, Mihaela Bornea, David Cox, Abraham Daniels, Martin Franz, Gabe Goodhart, Bhavani Iyer, Vishwajeet Kumar, Luis Lastras, Scott McCarley, Rudra Murthy, Vignesh P, Sara Rosenthal, Salim Roukos, Jaydeep Sen, Sukriti Sharma, Avirup Sil, and 3 others. 2025. Granite embedding models. *Preprint*, arXiv:2502.20204.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2024. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*.

Chandra Bhagavatula, Sergey Feldman, Russell Power, and Waleed Ammar. 2018. Content-based citation recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 238–251, New Orleans, Louisiana. Association for Computational Linguistics.

Peter Brown and Yaoqi Zhou. 2019. Large expert-curated database for benchmarking document similarity detection in biomedical literature search. *Database*, 2019:baz085.

Eric Chamoun, Michael Schlichtkrull, and Andreas Vlachos. 2024. Automated focused feedback generation for scientific writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9742–9763, Bangkok, Thailand. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Jianlyu Chen, Nan Wang, Chaofan Li, Bo Wang, Shitao Xiao, Han Xiao, Hao Liao, Defu Lian, and Zheng Liu. 2024b. Air-bench: Automated heterogeneous information retrieval benchmark. *Preprint*, arXiv:2412.13102.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

Mike D'Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. 2024. Marg: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

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.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Duanyu Feng, Bowen Qin, Chen Huang, Zheng Zhang, and Wenqiang Lei. 2024. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Zhaolin Gao, Kianté Brantley, and Thorsten Joachims. 2024. Reviewer2: Optimizing review generation through prompt generation. *arXiv preprint arXiv:2402.10886*.

Aniketh Garikaparthi, Manasi Patwardhan, Aditya Sanjiv Kanade, Aman Hassan, Lovekesh Vig, and Arman Cohan. 2025. MIR: Methodology inspiration retrieval for scientific research problems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28614–28659, Vienna, Austria. Association for Computational Linguistics.

Mainak Ghosh, Sebastian Erhardt, Michael E. Rose, Erik Buunk, and Dietmar Harhoff. 2024. Paecter: Patent-level representation learning using citation-informed transformers. *Preprint*, arXiv:2402.19411.

Ronald L. Graham. 1972. An efficient algorithm for determining the convex hull of a finite planar set. *Info. Proc. Lett.*, 1:132–133.

Mark S Granovetter. 1973. The strength of weak ties. *American journal of sociology*, 78(6):1360–1380.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, and 1 others. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Hongmei Guo, Zhesi Shen, Jianxun Zeng, and Na Hong. 2022. Hybrid methods of bibliographic coupling and text similarity measurement for biomedical paper recommendation. In *MEDINFO 2021: One World, One Health–Global Partnership for Digital Innovation*, pages 287–291. IOS Press.

Masoud Reyhani Hamedani, Sang-Wook Kim, and Dong-Jin Kim. 2016. Simcc: A novel method to consider both content and citations for computing similarity of scientific papers. *Information Sciences*, 334:273–292.

Khalid Haruna, Maizatul Akmar Ismail, Abdullahi Baffa Bichi, Victor Chang, Sutrisna Wibawa, and Tutut Herawan. 2018. A citation-based recommender system for scholarly paper recommendation. In *Computational Science and Its Applications–ICCSA 2018: 18th International Conference, Melbourne, VIC, Australia, July 2-5, 2018, Proceedings, Part I 18*, pages 514–525. Springer.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024. Agentreview: Exploring peer review dynamics with llm agents. *arXiv preprint arXiv:2406.12708*.

Yichen Yao Wei Chu Yinghui Xu Yuan Qi Junhan Yang, Jiahe Wan. 2025. inf-retriever-v1 (revision 5f469d7).

Anshul Kanakia, Zhihong Shen, Darrin Eide, and Kuansan Wang. 2019. A scalable hybrid research paper recommender system for microsoft academic. In *The world wide web conference*, pages 2893–2899.

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.

Maxwell Mirton Kessler. 1963. Bibliographic coupling between scientific papers. *American documentation*, 14(1):10–25.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Anton Korikov, George Saad, Ethan Baron, Mustafa Khan, Manav Shah, and Scott Sanner. 2024. Multi-aspect reviewed-item retrieval via LLM query decomposition and aspect fusion. In *Proceedings of the Workshop Information Retrieval's Role in RAG Systems (IR-RAG 2024) co-located with the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024), Washington DC, USA, 07 18, 2024*, volume 3784 of *CEUR Workshop Proceedings*, pages 23–33. CEUR-WS.org.

Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1929–1932.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Anne Lauscher, Brandon Ko, Bailey Kuehl, Sophie Johnson, David Jurgens, Arman Cohan, and Kyle Lo. 2021. MultiCite: Modeling realistic citations requires moving beyond the single-sentence single-label setting. *Preprint*, arXiv:2107.00414.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401, St. Julian's, Malta. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Zoran Medić and Jan Snajder. 2022. Large-scale evaluation of transformer-based article encoders on the task of citation recommendation. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 19–31, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models.

Dattatreya Mohapatra, Abhishek Maiti, Sumit Bhatia, and Tanmoy Chakraborty. 2019. Go wide, go deep: Quantifying the impact of scientific papers through influence dispersion trees. In *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 305–314. IEEE.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Sheshera Mysore, Arman Cohan, and Tom Hope. 2022. Multi-vector models with textual guidance for fine-grained scientific document similarity. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4453–4470, Seattle, United States. Association for Computational Linguistics.

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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

Malte Ostendorff. 2020. Contextual document similarity for content-based literature recommender systems. *arXiv preprint arXiv:2008.00202*.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings. *arXiv preprint arXiv:2202.06671*.

Jeonghyun Park and Hwanhee Lee. 2024. Conversational query reformulation with the guidance of retrieved documents. *arXiv preprint arXiv:2407.12363*.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2023. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*.

Peter Pirolli and Stuart Card. 1995. Information foraging in information access environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 51–58.

Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review*, 106(4):643.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, and 1 others. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Gerard Salton and Chris Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American society for information science*, 41(4):288–297.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. 2024. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15933–15946, Bangkok, Thailand. Association for Computational Linguistics.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore. Association for Computational Linguistics.

Henry Small. 1973. Co-citation in the scientific literature: a new measure of the relationship between documents. *J. Am. Soc. Inf. Sci.*, 42:676–684.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Siwei Wu, Yizhi Li, Kang Zhu, Ge Zhang, Yiming Liang, Kaijing Ma, Chenghao Xiao, Haoran Zhang, Bohao Yang, Wenhu Chen, Wenhao Huang, Noura Al Moubayed, Jie Fu, and Chenghua Lin. 2024. SciMMIR: Benchmarking scientific multi-modal information retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12560–12574, Bangkok, Thailand. Association for Computational Linguistics.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Jianxiang Yu, Zichen Ding, Jiaqi Tan, Kangyang Luo, Zhenmin Weng, Chenghua Gong, Long Zeng, RenJing Cui, Chengcheng Han, Qiushi Sun, Zhiyong Wu, Yunshi Lan, and Xiang Li. 2024. Automated peer reviewing in paper SEA: Standardization, evaluation, and analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10164–10184, Miami, Florida, USA. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu,

Michael Zeng, and Meng Jiang. 2023a. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023b. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*.

Dun Zhang, Panxiang Zou, and Yudong Zhou. 2025a. Dewey long context embedding model: A technical report. *Preprint*, arXiv:2503.20376.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yu Zhang, Hao Cheng, Zhihong Shen, Xiaodong Liu, Ye-Yi Wang, and Jianfeng Gao. 2023a. Pre-training multi-task contrastive learning models for scientific literature understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12259–12275, Singapore. Association for Computational Linguistics.

Yu Zhang, Bowen Jin, Xiusi Chen, Yanzhen Shen, Yunyi Zhang, Yu Meng, and Jiawei Han. 2023b. Weakly supervised multi-label classification of full-text scientific papers. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3458–3469.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

Yan-Bo Zhou, Linyuan Lü, and Menghui Li. 2012. Quantifying the influence of scientists and their publications: distinguishing between prestige and popularity. *New Journal of Physics*, 14(3):033033.

A SciFullBench

A.1 Comparison with Prior Benchmarks

Previous benchmarks for scientific literature search where the full context is available for both query and candidates rarely exist. (Kanakia et al., 2019) introduce expert-annotated paper recommendation benchmark using abstract and citations within Microsoft Academic Graph(MAG). SciDocs (Cohan et al., 2020) and SciRepEval (Singh et al., 2023) reveal an evaluation set to search relevant scientific literature within a pool of 30 candidates per query document with 5 gold labels in the computer science domain, while MDCR (Medić and Snajder, 2022) disclose a benchmark with 60 candidates per query from 19 scientific fields sourced from MAG. RELISH (Brown and Zhou, 2019) also provides expert-annotated gold candidates that are relevant to the respective input documents in the biomedical domain. In addition, (Mysore et al., 2021) formulates a CFSCube benchmark to evaluate fine-grained sentence-wise alignment between abstract passages. SciMMIR (Wu et al., 2024) also presents a multimodal document retrieval evaluation set to evaluate the performance of figure-wise document retrieval frameworks. However, such benchmarks do not include candidates or queries in which their entire lexical content is fully disclosed. Although (Zhang et al., 2023b) intends to evaluate their classifier pipeline using the complete scientific context, it contains five potential candidates per query, which is infeasible to evaluate large-scale literature retrieval frameworks. Meanwhile, MultiCite (Lauscher et al., 2021) dataset provides full context with fine-grained citation-intent annotation on the paper's citation context(sentences), to evaluate the model's ability to accurately identify citation intents. However, its scope is inherently limited to a trivial classification task, and does not reveal metadata information(especially full context) on the cited papers nor properly filtered and preprocessed, inappropriate for evaluating real-world paper-to-paper retrieval frameworks. Conversely, our proposed benchmark consists of fully-disclosed document context, along with massive number of target candidates, adhering to the realistic setting for evaluating scientific literature retrieval.

A.2 Construction Procedure

In this section, we provide more details on the step-by-step construction process of SciFullBench.

Step 1 As mentioned in the main content, we initially crawled research papers from top-tier machine learning venues including ICLR 2024, 2025, NeurIPS 2024, 2023, ACL 2024, 2023, and EMNLP 2024, 2023 for our query documents. We scraped existing publications from the main conference for ACL and EMNLP, including both long- and short- papers from the ACL-Anthology website[6], and used the OpenReview[7] API to obtain submitted research papers for ICLR and NeurIPS. Moreover, for ICLR 2024 and NeurIPS 2023, we included a subset of documents uploaded by authors from SEA (Yu et al., 2024). Afterwards, we obtained metadata from the arxiv database uploaded to the Kaggle[8] website, with available data up to January 2025. Moreover, we constructed our initial temporary raw corpus where its id starts with 20 to 25 which refers to its uploaded date on Arxiv database, limiting our target corpus to fairly recent papers. Also, we only filtered papers that are in the machine learning domain, to those belong in cs.AI, cs.LG, cs.CL, cs.CV, cs.NE, cs.IR, cs.DS, cs.CC, cs.DL, cs.HC, cs.RO, cs.MM, cs.CG, cs.SY since our queries are sampled from ML venues.

Step 2 After devising raw data for both queries and candidates, we subsequently formulate gold candidates using citations to represent contextual proximity between scientific documents. We collect the title of papers that cite our potential query documents using the Semantic Scholar API[9] and check the existence of a paper with a matched title (case-insensitive) in our arxiv metadata, and ruled out any potential query documents with fewer than ten gold candidates that met such conditions. As for benchmark split with references, we use Allen AI Science Parse to obtain reference information for respective documents, and filtered documents based on the implemented criteria from above. Using such data, we formulate our raw (query document, gold candidate) pairs where its abstract information is intact and where more than 10 deduplicated candidates exist, organized by splits in respective years belonging to four venues (ICLR, NeurIPS, ACL and EMNLP). For each split, we aggregate all the filtered (query, candidate) set from each venue without regarding its published year. For example, we merge all the candidates in the

[6] https://aclanthology.org/
[7] https://openreview.net/
[8] https://www.kaggle.com/
[9] https://www.semanticscholar.org

Table 6: Query Paper statistics in SCIFULLBENCH benchmark.

	ICLR		**NeurIPS**		**EMNLP**		**ACL**	
	References	Citations	References	Citations	References	Citations	References	Citations
domain	ML		ML		ML/CL		ML/CL	
years included	2024,2025		2023,2024		2023,2024		2023,2024	
# of papers per year	(141,259)	(324,76)	(170, 230)	(353,47)	(226, 174)	(334, 66)	(184,216)	(256,144)
# of tokens per paper	9402.35	8183.37	11184.39	9035.43	5908.36	6226.45	6190.06	6614.6
average # of candidates per sample	21.74	42.25	24.58	40.37	19.64	39.24	18.05	33.89
minimum # of candidates per sample	10	10	10	10	10	10	10	10

Table 7: Corpus statistics for SCIFULLBENCH.

	SCIFULLBENCH
Total # of papers	40,782
Avg. # of tokens per abstract	200.39
Avg. # of tokens per paper	6987.67
Total # of segmented corpus	115004
Avg. # of tokens per segment	2479.90

citation split for ICLR 2024 and ICLR 2025 into a single set of ICLR-citations split, not ICLR-2024-citations or ICLR-2025-citations. From this filtered pool, we randomly sampled 500 potential query candidates and formulated a corpus containing gold candidates for each query. This process enables our target corpus to be kept within manageable size. Next, we assembled the original pdf files of the papers in our temporarily formulated target corpus. Since Arxiv APIs do not support large-scale requests, we used the Google Cloud API[10] to access the full paper dump directly and collect the latest updated versions for each paper by matching its unique ids within the Arxiv database.

Step 3 Subsequently, using the Allen AI Science Parse tool, we parsed the query and candidate pdf files collected in Step 2 into a json format file containing title, abstract, main content list containing dictionaries with header and contents as keys, and list of reference paper information along with its mentions for both queries and candidates. Since we use citation information as the main signal for measuring document similarities, excluding the citations and the entire reference section was crucial to validate the fairness of our benchmark. Although Allen AI Science Parse generally separates reference from main content, several issues persist, where direct reference information is intermixed within the main content. Thus, we applied an auxiliary filtering algorithm to remove such issues. Since our tool parses a pdf file into a structured json file, we were able to obtain text information segmented into multiple passages. Hence, prior to reformatting it back to complete text, we eliminated sections or headers that contain at least one reference title (case-insensitive) completely while reformatting into a full paper. However, for some cases the reference section was left vacant. Since our filtering algorithm depends on the set of titles within parsed reference section, it was unable to correctly exclude cases intermixed with reference information when such data are inaccessible. Hence, we ruled out any query documents or papers within our target corpus that did not include more than four references in the respective parsed documents. In this way, it resolves problematic cases in which we give a pass on perturbed documents that still contained reference information. Moreover, due to our filtering heuristics, there remains a problem where numerous documents experience severe loss of their original content, since any passage or header that had at least 1 reference title was excluded. To mitigate such concerns, we excluded documents in which such problematic passages comprise more than half of the total main-content list. Furthermore, we ensured to eliminate any residing title(case-insensitive) included in the reference section.

Step 4 Based on the preprocessed full document, we formulate a pseudo-definitive benchmark consisting of (query, gold candidate), and target corpus set by random sampling 400 query documents per split in each venue that still 10 gold candidates residing subsequent to the filtering procedure. Ultimately, we formulate a total of 3200 query documents, along with large-scale target corpora consisting of all the gold candidates of such queries. Finally, we remove citations and interlinked mentions from the entire set of formatted papers and finalize our benchmark. Since this process does not lead to exclusion of query document from our benchmark, we have called the previous stage a

[10]https://cloud.google.com/apis?hl=ko

pseudo-definitive benchmark. Citation mentions are removed using the information provided from our parsing tool, along with 10 different citation patterns using the Python regular expression. Consequently, we devise a definitive, final benchmark in which all documents for both query and candidate corpus consist of title, abstracts, full-paper text, and list of segmented content of corresponding full-paper text.

B Training Procedures

B.1 Query Document and Corpus Construction

Composition Our input query document for offline rollout includes publications and submissions from leading venues in ML likewise SciFullBench, but with different published or submitted years. We incorporate publications and submissions from **ICLR** 2017 to 2023, **NeurIPS** 2016 to 2022 from the data set revealed in REVIEWER2 (Gao et al., 2024), **ACL** 2017 to 2022, **EMNLP** 2017,2018,2020,2021,2022, and **NAACL** 2017,2018,2019,2021,2022. As for the target candidate corpus construction, we follow a similar approach to SciFullBench in which we crawled ML papers in the arxiv database, with dates ranging from 2020 to 2025 as mentioned in Appendix A. Ultimately, we formulate 15K input query documents with at least 10 ground-truth candidates, along with 97k target corpus consisting of all the ground-truth candidates of input query documents.

Formulation Workflow The formulation procedure for building the training set is similar to the process explained in Appendix A, with several key differences. **First**, to prepare query documents that do not overlap with our SCIFULLBENCH benchmark, we collected publications and submissions from different venues(or years) from ones in our evaluation set. Yet, there still exists a slim possibility that duplicate papers may exist in both the benchmark and train set since research papers can be resubmitted to other venues if rejected. To preclude such issues, we excluded query documents from our train set where its abstracts match (using a custom matching function, including lowering case, removing empty sequences, etc.) query documents in SCIFULLBENCH. We particularly used abstracts to check duplicate papers because several queries in our SCIFULLBENCH may not contain title information. **Secondly**, compared to the benchmark construction process that filters query documents with at least 10 ground truth documents for each citation and reference split, we rule out queries with at least 10 ground truth documents including both citation and reference split. Moreover, since the number of our input query documents drastically exceeds those in SCIFULLBENCH(nearly 5 times), our target candidate corpus size inevitably becomes exponentially larger. Thus, we randomly sampled 140K samples from the initial filtered target corpus, and reformulated our queries accordingly with at least 10 candidates as well. Finally, another minor difference in train set construction compared to SCIFULLBENCH was its order in applying the filtering algorithm (**reference removing**, **citation removing**), where we executed both algorithms for query documents immediately after acquiring full content, followed by repeated citation removing algorithm to further enhance the extent of removal.

B.2 Preference Set Construction

Using the query documents and the formulated corpus from Appendix B.1, we exclusively used references as gold labels, as mentioned in the main section. Furthermore, we excluded query documents that exist in the SciFullBench corpus, since our trained query optimizer agents are provided with documents from our benchmark corpus during the multi-round retrieval process. To ensure fairness in evaluating the effectiveness of our algorithm, we excluded potential query documents from our training set that exist in the corpus used for the inference setting, as the documents in the corpus are used as query for later rounds of retrieval. During the offline policy exploration process, we set the maximum input tokens to 60k and the maximum completion tokens to 2000. Moreover, we set the temperature to 0.7 and the nucleus sampling ratio to 0.8 to promote exploration, with the branching factor of 16 per aspect-aware LLM agent, using a single 48GB A6000 GPU with VLLM gpu memory utilization rate of 0.8 for each respective LLM agent + Retriever setup. Subsequent to offline exploration, we measured the recall@30 reward of the generated queries and paired the query response with the highest reward and the lowest reward, sampling only those with a recall@30 difference of at least 3%.

B.3 Train Details and Hyper Parameters

We trained two instruct models, Llama-3.2-3B-Instruct and QWEN-2.5-3B-Instruct, using the Unsloth (Daniel Han and team, 2023) library to re-

duce memory consumption. We trained the respective query optimizer agents with LoRA (Hu et al., 2022) adapters and 4-bit quantization on the default models. In addition, we set the maximum total number of tokens to 4000, allocating 2000 tokens each for input and output. This ensures that the responses from our preference set matches the output token limits in the inference setup, allowing us to exclusively assess the effect of training rather than other factors. Meanwhile, shortened input tokens significantly reduce training time, mitigating overfitting to details within the query document, and encourage learning of favorable patterns from the preferred response queries. Furthermore, we trained each LLM agent with a linear scheduler, a learning rate of 1e-5, a weight decay rate of 0.01, and a maximum gradient norm of 0.6 for 3 epochs, with a global batch size of 32 on a single A5000 GPU with 24GB VRAM, while occasionally using A6000 GPU with 48GB VRAM to speedup training process. Under such setup, it required approximately 14 hours to train a single agent. As for the LoRA configuration, we set both the rank and α scale to 64, and applied the adapters to both the attention layers and the feedforward network, without leveraging dropout or bias factors.

C Experiment Details

C.1 Inference Settings

Prompt Construction We constructed prompts for scientific paper-to-paper retrieval based on the three major aspects that comprise a scientific paper: **Methodology**, **Experiments**, and **Research Questions**. The initial raw prompts were first created using Chat-GPT, and human-in-the loop revision was conducted, to devise the optimal prompt style suitable for our framework.

Retrieval Setting We primarily use the Euclidean distance to measure similarities between our adaptively generated queries and candidates, where we acquired candidates with the minimum L2 distance. Since minimizing the L2 distance was objective for most of our baseline domain-specific embedding models such as SciNCL, SPECTER2-Base, SPECTER2-Adapter-MTL CTRL, we matched such settings when we used jina-embeddings-v2-base-en, BGE-M3, and Inf-Retriever-v1-1.5b. However, for SciMult, we measured MIPs (Maximum Inner Product), since it was trained to maximize MIPs likewise in DPR. In addition, we utilize the FAISS (Douze et al., 2024) library, which enables efficient retrieval from large-scale corpora. We implemented an efficient L2 search using FlatL2 and FlatIP for MIPs.

Query Optimizers The main results are produced by setting the temperature and the maximum generation token hyperparameter to 0 and 2000 respectively, while setting the repetition penalty to 1.2. In addition, we used the VLLM (Kwon et al., 2023) library for faster inference of open source models. For Llama-3.2-3B-Instruct, we used the default context window size of 131072, while using YARN (Peng et al., 2023) to extrapolate the default context window size of QWEN-2.5-3B-Instruct from 32768 to 131072. Moreover, for inference, we used three A5000 GPUs or three RTX3090/4090 GPUs with VRAM of 24GB, along with vllm gpu memory utlization rate of 1.0, to deploy three trained aspect-aware query optimizer LLM agents simultaneously, using fp16 precision for faster inference. As for the inference setup for untrained query optimizer models, we experimented on a single A6000/RTX 4090 GPU with 48GB VRAM using the same context window, along with a fixed gpu memory utilization rate of 0.7. When it comes to experiments using GPT-4o-Mini-2024-0718 and GPT-4o-2024-11-20 as query optimizers, we used temperature 0 and default hyper parameters of the OpenAI client.chat.completions API [11], while completely using the default hyperparameter settings for gpt-4.1-2025-04-14 (temperature is also set to default). Moreover, 60 is used as the hyperparameter k for Reciprocal Rank Fusion. For base agent experiments in Table 3, we equalized the hyper parameters to those used in its respective counterpart comparison groups, and are prompted to generate three different queries in a structured manner, using Python Pydantic BaseModel [12] module.

C.2 Models

Domain Specific Retrievers We compare our method with retrieval using domain-specific embedding models, as mentioned in the main section, to demonstrate robust improvement over approaches that seek to devise optimal representations of paper abstracts through extensive training. For fair assessment, we mainly compare our method with models that demonstrated SoTA performance in previous citation link prediction and paper retrieval benchmarks. SPECTER2 models with adapters

[11] https://platform.openai.com/docs/guides/
[12] https://docs.pydantic.dev/latest/api

and its multitask control codes have been reported to have achieved SoTA performance on the MDCR benchmark (Medić and Snajder, 2022). We used proximity control code adapter, as it was specialized in acquiring and ranking relevant papers, and also base models uploaded to huggingface [13]. In addition, we experiment with SPECTER2-base models, trained with triplet loss by inducing neural retrievers to favor positive abstract pairs over vice versa given abstract of query documents sampled from discrete citation graphs. SciNCL [14] is also trained in a similar manner, where its abstract pairs are derived from the continuous citation embedding space, achieving the strongest performance on the SciDocs benchmark to date. In addition, we compare our method with the SciMult-MHAExpert model implemented with Mixture-of-Experts (Fedus et al., 2022) architecture which compartmentalize internal transformer layers for different tasks in scientific literature tasks, and is reported to outperform previous models on the recommendation benchmark (Kanakia et al., 2019). For SciMult-MHAExpert, we experimented with the model released on the SciMult github repository [15], utilizing an expert model specially trained for link prediction tasks that predicts linked documents given the abstract of the source document.

Domain Agnostic Retrievers As for our domain-agnostic embedding models, we particularly chose off-the shelf retrievers with long context window. This was to ensure that we provide fair experimental setting for our baselines, most notably full document-to-full document retrieval setting. Since general full-documents exceed the context window of most embedding models, we specifically chose long-context window embedding models to mitigate unfair penalization on our full document-to-document retrieval baseline. In particular, for our main experiment, we use three different neural retrievers, Jina-Embeddings-v2-Base-EN, and BGE-M3. both highly cited general-purpose embedding models with a context window of 8192 tokens, showing that our method is compatible with the most widely used retriever models. In addition, we used Inf-Retriever-v1-1.5B, a lightweight version of Inf-Retriever-v1, demonstrating SoTA performance among open source embedding models for long-context retrieval(LongEmbed) in MTEB (Muennighoff et al., 2022)[16] as of September 12th, 2025, with context window size of 32768, along with SoTA performance on AIR-Bench(general QA tasks) (Chen et al., 2024b) benchmark. This further shows that our method can perform robustly with the most recent general purpose embedding models with advanced long-context handling capabilities.

Furthermore, we provide additional experiments using other recent general-purpose embedding models varying in context window size, such as QWEN3-Embedding-0.6B (Zhang et al., 2025b), Dewey-EN-Beta (Zhang et al., 2025a), and Granite-Embeddings-English-R2 (Awasthy et al., 2025). QWEN3-Embedding-0.6B is a light weight version of QWEN3-Embedding model series, recently released model that displays SoTA performance on MTEB-multilingual evaluation split, while Granite-Embeddings-English-R2 is the most recently released model as of August 2025. In addition, Dewey-EN-Beta is another recent model that has context window size of 131072, which is the longest among up-to-date existing models, demonstrating outstanding performance on long-context retrieval tasks, attaining SoTA performance on MTEB LongEmbed benchmark split.(yet to be formally updated on public leader board).

C.3 Metrics

In our experiments, to measure the precision and relevance of our retrieved candidates, we adopt three primary metrics for our main results. **Recall@K** is used as our main metric, which measures the ratio of correct candidates within the Top@k retrieved results. This aligns with our objective, where we seek to acquire a more relevant pool of papers using various aspect-aware queries. Moreover, we report **Normalized Discount Cumulative Gain, nDCG@K**, **Mean Average Precision, mAP@K**, and **MRR@K** to validate the robustness of our method. Unlike the standard nDCG metric, which assigns higher weights to gold candidates with higher significance, we treat all gold candidates for a given query equally, assigning each a uniform weight of 1. Moreover, we report our results using relatively high K value(such as 100, 200, 300), to ensure the fairness of our evaluation suite. This is due to the large average number of annotated ground truth candidates per query in our self-constructed SciFullBench benchmark.

[13] https://huggingface.co/allenai/specter2
[14] https://huggingface.co/malteos/scincl
[15] https://github.com/yuzhimanhua/SciMult
[16] https://huggingface.co/spaces/mteb/leaderboard

(See Table 6). Hence, to ensure that the retrieved candidates is capable of covering all the ground-truth candidates, we set K to a relatively high value.

In addition, we provide **Convex Hull** (Graham, 1972) **Volume** as a metric to quantitatively measure the dispersion of the embedding representations of retrieved candidates, which computes the volume of the smallest possible bounding convex closure of the data points after projecting the embedding vectors to the three-dimensional vector space, presenting the semantic coverage that each retrieval method presents. Notably, we employ convex hull volume in two different ways. As for the analysis provided in Table 4, we measured the relative convex hull volume ratio of ground-truth candidates present among the Top@300 retrieved results using the two equal samples(benchmark data with same query paper, ground truth candidates) used for binary comparison. This metric represents the relative semantic coverage of relevant documents from respective retrieval systems in a more fine grained manner. To ensure the fairness of our analysis, we excluded any sample that were unable to compute convex hull volume using retrieved ground truth candidates (due to its inability to retrieve sufficient number of ground-truth candidates) from either A2A, F2F, and CoR on a given benchmark split, assuring that the comparisons were held using the same samples for respective three configurations. On the other hand, as for the experiments regarding Figure 9, we measured the absolute convex hull volume of total Top@k retrieved samples, to analyze the trend of semantic coverage with respect to retrieval depth progression.

D Patent-to-Patent Retrieval

D.1 PatentFullBench Construction Workflow

Step 1) Collecting Raw Data: We initially obtained a full XML dump from the USPTO Open Data portal website[17], and initially crawled US patents from 2020 to 2025 June 17th.

Step 2) Formatting Raw Data: We parsed XML dump from the USPTO Open Data portal website, obtaining publication number, publication date, application number, application date, title, abstract, inventors, priority claims, claims, descriptions, references(citations from given patent), and main classification for respective patents. As for citations and references, we initially filtered U.S patents exclusively, and ones where its date is over 2020, and obtained its classified country, date, document number(id), citation patterns.(et al,), collecting a group of raw patents where each patent at least had 10 citations from the parsed XML dump.

Step 3) Initial Gold Label Annotation and Target Corpus Construction: After forming a pool of raw extracted patent contents from XML dump data, we obtained patent invention title, abstracts,id, and date, using PatentsView API[18] for respective cited patents from given pool of raw extracted content. Since PatentsView API does not support full-content access, we made sure to exclude patents where its id did not belong in our pool of extracted content from **step 2**. Furthermore, we made sure to only include references where its returned year from PatentsView API was the same as the queried year, and where there exists a patent whose title matches the returned title from PatentsView API.

After obtaining actual reference to full patents per input query document, we attempt to further label each query document with inward citations(incoming links) by labeling with gold documents that cite the query document, compared to the labels obtained from **step 3**, where the query document cites the gold labels. Using the reference information constructed from **step 3**, we further process citation relations through traversing all the patent files, and record the query patent B that cites A patent as B for inward citation of A, by matching patent title and document number. Subsequent to obtaining both inward citations and outward citations(references) for respective query patent documents, we removed duplicated labels that had either the same title or abstract content, and made sure that the given label's patent ids match the ids of the ids of the raw patent information accumulated in **step 2**, since numerous versions with same content with different ids may exist.

For respective splits, we excluded query documents from our benchmark set that had less than 10 labels, and initially randomly sampled 800 query documents for respective splits. Successively, we formulated our target corpus via accumulating the labels that exist within our randomly sampled query document splits without duplicates. Given the initially sampled query documents, we collected gold candidates for each query document, removing duplicates that had either the same patent title or abstract content. Then, we made sure to filter out gold

[17] https://data.uspto.gov/bulkdata/datasets

[18] https://patentsview.org/apis/purpose

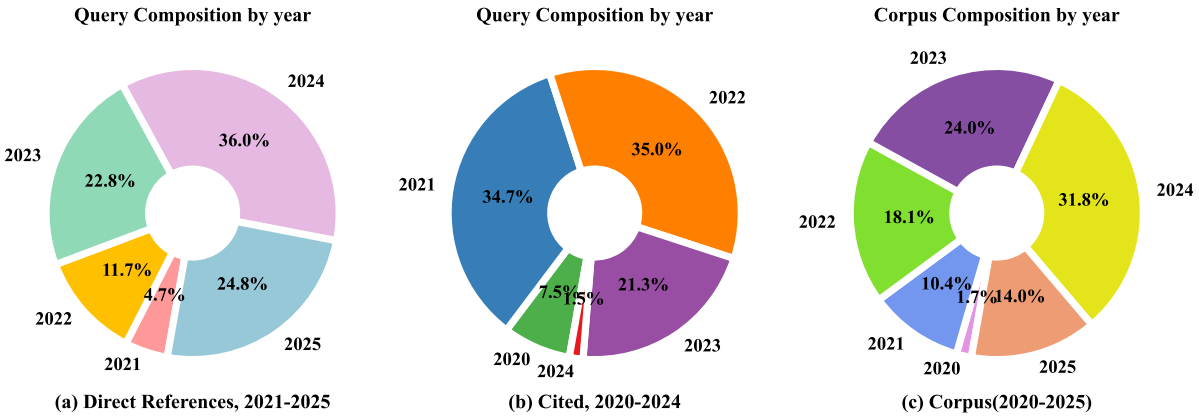

Figure 6: PatentFullBench composition by published year.

Table 8: Query Document statistics in PatentFullBench.

	References	Citations
Nation	United States	
# of documents per split	400	400
# of tokens per abstract	119.50	120.31
# of tokens per document	190335.46	186504.29
average # of candidates per sample	37.9	53.93

Table 9: Corpus statistics for PATENTFULLBENCH.

	PATENTFULLBENCH
Total # of Documents	5150
Avg. # of tokens per abstract	120.97
Avg. # of tokens per document	29141.85
Total # of segmented corpus	52584
Avg. # of tokens per segment	2865.06

candidates of initially sampled query documents that did not exist in the target corpus due to exclusion of candidates that had partial similarities with one another. For example, if one candidate and the other has different title but same abstract content, the latter case is excluded from our target corpus. This process ensures that our potential gold candidates for queries in our benchmark always exist in the target corpus. Finally, we randomly sampled 400 queries that had at least 10 gold candidates after the above filtering step, and reformulated our target corpus making sure that duplicate candidates do not exist in our corpus, obtaining a corpus containing total of 5171 documents.

Step 4) Formatting Full Context of Query and Candidate Documents and Final Filtering Successively, we further processed the full context of documents that are going to be either used as query or candidates for our evaluation. As for query documents, we concatenated title, abstract, claims, and description content, to formulate a full context for respective patents. Likewise the construction process of SciFullBench, we made sure to eliminate potential cues that reveal hindsight information on cited patents. First, using python regular expression, we eliminated information from the section within **"CROSS REFERENCE TO RELATED APPLICATIONS"**, sentences that start with **"US"**, ending with next line sign or phrase **"reference herein"**. Moreover, using the citation information obtained from the XML dump in **step 2**, we eliminated citation patterns, and document numbers within the full context of patents. Finally, using the title and abstract information of constructed gold candidates from **Step 3**, we eliminated title and abstract information that exist in the full content of respective patents. During the processing of candidate documents, we followed the same procedure as query documents, except for the last step where instead of using title and abstract information from the constructed gold labels, we removed every title and abstract information from the query documents that use the candidate document as its gold candidates, making sure that the candidate documents do not reveal direct hints on the relevant patents. After sanitizing the full documents, we segmented full context of candidate documents for every 3000 tokens(based on NLTK, similar to SciFullBench construction). Since each full-context of patents had partial overlap, we excluded the patent candidate from our corpus that had at least single duplicate partial chunks, and ensured to remove such patent candidates from the gold labels(candidates) of our benchmark split. This additional filtering process ensures that the same chunk does not lead to mapping to different patent content, which could potentially hinder the fairness of evaluation. Finally, we constructed chunked corpus containing a total of 5150 different patent documents with both full context and title, abstracts, and 52584 chunks of patent documents

along with 3000 token granularity.

D.2 Experiment Details

Prompt Construction We constructed prompts for patent-to-patent retrieval based on the three major aspects that comprise a patent: **Claim**, **Backgrounds**, and **Method**. **Method** aspect refers to technical details and implementation structure of the novel invention claimed by the given patent, while **Backgrounds** provide background information on the technical problems of prior works and its limited technical objectives. Moreover, **Claim** aspects comprise of the scope of its legal protection, and the main technical contributions of the invention that respective patents intend to claim. The prompt construction process consumed roughly one hour to devise appropriate prompts for query optimizers, with the help of Chat-GPT to generate initial raw prompts and revising the content and structure with human feedback to ensure that the prompts are suitable for patent-to-patent retrieval featuring multiple aspects.

Inference Setting We evaluate our patent to patent retrieval setting with the same domain-agnostic embedding models that we utilized for our evaluation, additionally reporting abstract-to-abstract retrieval results with domain-specific retrievers, PAT-SPECTER and PaECTER (Ghosh et al., 2024), which is the most recent model for patent retrieval trained via minimizing the triplet loss using the binary patent abstract samples, similar to the training methods employed in devising domain specific retrievers for paper retrieval in the academic domain. The train dataset samples were obtained using the European Patent Office(EPO) [19] website, comprising of 30,000 English patents ranging from 1985 to 2022. Moreover, we utilized proprietary LLM, GPT-4o-Mini-2024-0718 for query optimizer and truncated the input document context to 120000 tokens using the tiktoken[20] library since patent context length is significantly longer than scientific papers, occasionally exceeding the context window of GPT-4o-Mini-2024-0718 model. As for the hyper parameters, we set the temperature to 0, same as mentioned in Appendix C.1.

E Supplementary Experiments

Consistent Generalization Across Embedding Models In our main results, we showed three variations of embedding models to demonstrate the effectiveness of our CoR framework. In this section, we further provide additional experiments on other existing long-context embedding models, namely **Granite-Embeddings-R2**, **QWEN3-0.6B-Embedding**, and **Dewey-en-Beta** in Table 10. The experimental results strongly support the robustness and general applicability of our research findings, as even without a trained model, by using proprietary **GPT-4o-2024-11-20** and **GPT-4.1-2025-04-14** with just a single round of retrieval, we were able to observe consistent performance improvements compared to naive F2F and A2A retrieval setups.

Table 10: Additional report that shows robust performance of our Method incorporating other LLM backbone models and neural retrievers, after single round of retrieval

Model + Retriever	ICLR-Citations Recall@300	NeurIPS-Citations Recall@100
Baseline		
Granite-Emb-Eng-R2 - A2A	50.19	41.39
Granite-Emb-Eng-R2 - F2F	54.00	44.37
QWEN3-0.6B-Emb - A2A	51.47	41.76
QWEN3-0.6B-Emb - F2F	54.03	44.81
Dewey-en-Beta - A2A	47.76	40.54
Dewey-en-Beta - F2F	53.15	42.06
Ours		
GPT-4o + Granite-Emb-Eng-R2	**55.64**	**47.22**
GPT-4o + QWEN3-0.6B-Emb	**56.94**	**47.15**
GPT-4.1 + Dewey-en-Beta	**56.73**	**47.99**

The improvements achieved using Granite-Embeddings-R2 indicate that our framework can perform robustly under a context window of 8192 as repeatedly demonstrated in our main results, and is compatible with even the most recently released model. Moreover, the results from the QWEN3-0.6B-Embedding model imply that our model can be applied to embedding models with longer context windows(32768) likewise Inf-Retriever-v1-1.5B in Table 1, as well as to popular state-of-the-art long-context embedding models for general domain-agnostic multilingual retrieval tasks in MTEB benchmark. Furthermore, we provide evaluation results on Dewey-En-Beta, which is an embedding model with currently the longest context window among existing models, while demonstrating State-of-the-Art performance on LongEmbed benchmark. This suggests that our paper-retrieval framework is applicable to models with the longest context windows up-to date, indicating the effectiveness and robustness of our suggested method to handle long context retrieval.

Robustness without Abstract-to-Abstract Retrieval Meanwhile, in Table 11, the results are

[19] https://www.epo.org/en

[20] https://github.com/openai/tiktoken

Table 11: Performance analysis of our pipeline when the original abstract to abstract retrieval is not utilized in our pipeline. The results are reported with untrained query optimizers on a single round of retrieval.

Retrieval Method	ACL-Citations Recall@200	ICLR-Citations Recall@200
Baseline		
JEmb-v2 - A2A	35.33	39.19
JEmb-v2 - F2F	35.88	42.44
BGE-M3 - A2A	33.76	35.16
BGE-M3 - F2F	32.64	36.06
Inf-Ret-v1 - A2A	41.77	45.42
Inf-Ret-v1 - F2F	38.86	43.00
Ours w/o A2A		
QWEN-2.5-3B-Inst + JEmb-v2	**41.13**	**44.41**
Llama-3.2-3B-Inst + BGE-M3	**35.83**	**37.49**
Llama-3.2-3B-Inst + Inf-Ret-v1	**47.05**	**50.65**

Table 12: Ablation study on the effect of our search framework compared to naive chain-of retrieval methods based on A2A and F2F retrieval setting. The results are reported after 3 rounds of retrieval using untrained query optimizers.

Model + Retriever	ICLR-References Recall@100	ICLR-Citations Recall@200
Baseline		
JEmb-v2 - A2A	33.04	39.1
JEmb-v2 - F2F	33.19	41.54
BGE-M3 - A2A	28.71	34.72
BGE-M3 - F2F	31.00	36.12
Inf-Ret-v1 - A2A	42.50	44.67
Inf-Ret-v1 - F2F	36.67	42.38
Ours(w/o DPO)		
Llama-3.2-3B-Inst + JEmb-v2	**38.34**	**46.73**
Llama-3.2-3B-Inst + BGE-M3	**32.62**	**41.84**
Llama-3.2-3B-Inst + Inf-Ret-v1	**46.74**	**52.42**

reported when abstract-to-abstract retrieval(A2A) is not utilized within CoR. Despite A2A setup being omitted from its original multi-aspect queries, consistent improvement over baselines can be observed, although not to the extent when abstract-abstract retrieval is integrated into our pipeline, indicating the positive, balancing effect that A2A retrieval presents, along with robustness of our method. This in turn highlights the effectiveness of our pipeline in generating multiple aspect-aware queries in the absence of complementary abstract to abstract retrieval, strongly supporting our research contributions.

Comparison with Naive Chaining Table 12 shows the benefits of employing multi-round retrieval with CoR compared to that of naive chain-of-retrieval methods using Full Paper-to-Full Paper and Abstract-to-Abstract retrieval setup. The results indicate that our aspect-guided chain-of-retrieval along with post-order recursive aggregation is more effective, and trivial iterative retrieval is not sufficient. Note that the baseline experiments were conducted with under the same hyper parameter setting, with a cache to avoid redundant sampling likewise our framework, with 300 candidates retrieved and aggregated via Reciprocal Rank Fusion. This shows that our approach in exploration and aggregation guided by aspect-centric queries is more effective compared to iterative top-k retrieval using raw content as queries, supporting the validity of our proposed retrieval system.

Starting Index Hyper Parameter Ablation In this section, we provide experiments on the hyper parameter selection starting index γ that we used for **Next Query Selection Algorithm**, to choose the starting point of choosing the representative paper with highest semantic similarity. In our case, value 1 is used to report our results in the main paper(always start from the second highest rank), to select the paper with highest similarity while avoiding the same papers as input query that may exist in our corpus(which inevitably appears at rank 0, since same papers display 100% semantic similarity). To further validate the benefits of selecting paper with highest-semantic similarity for subsequent retrieval, variations of CoR with different selection starting index γ is reported in Figure 7. The results demonstrate steady decrease in performance as selection starting index is increased, implying that chain-of-retrieval using the nearest-possible paper for next query selection is more desirable compared to vice versa, due to the semantic drift that occurs when relatively unrelated papers are selected as query for subsequent round of retrieval.

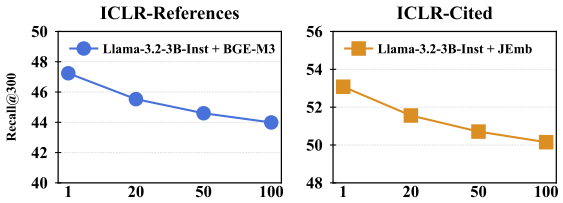

Figure 7: Hyperparameter Ablation Study on the effect of selection starting index γ on the overall performance. Note that we used DPO-trained query optimizers on 3 rounds of retrieval.

Corpus Granularity Ablation Study Furthermore, in order to see how the granularity of candidate representations affects the performance on our multi-vector retrieval system, we conduct further analysis. The results in Figure 8 demonstrate the effectiveness of our multi-vector approach: segmenting each candidate document into 3K-token chunks yields the best performance, which outperforms not only abstract-only and full-document single-vector representations but also coarser chunking strategies (such as 6K-token segments), suggesting that finer-

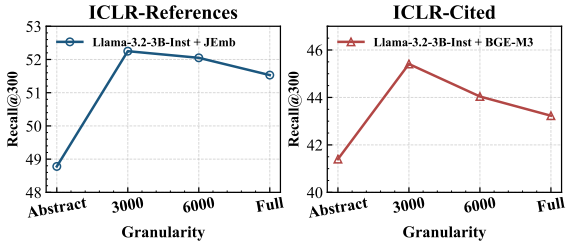

Figure 8: Performance analysis with different candidate paper granularities. Note that we conduct evaluation under single round of retrieval using untrained query optimizers.

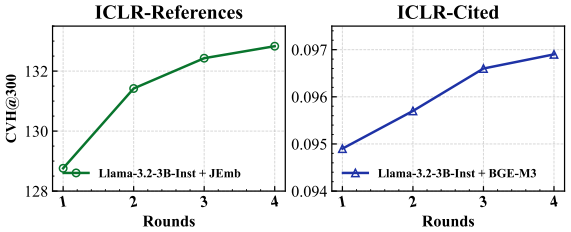

Figure 9: Diversity Analysis with retrieval depth progression. Note that we perform evaluation using DPO-trained Query Optimizers.

grained representations are effective in capturing diverse and localized signals within full-length papers, which enables our query to capture a more multi-faceted dimension of target papers, resulting in an enhanced performance.

Effect of Reciprocal Rank Fusion We conduct an additional ablation study on the effect of RRF (Reciprocal Rank Fusion) in our hybrid retrieval system(single round of CoR), compared to embedding-level merging approach in prior multi-vector retrieval approaches. We primarily compare with two traditional approaches in embedding-level merging strategy, the naive aggregation strategy that forcefully computes similarities (in our case L2 distance) between all the sub-vectors of respective query and candidates and naively sums it up to acquire similarity score of original query and documents. In our experiments, to mitigate the unfair penalization due to document length, we normalize the aggregated results with the total number of subvectors used for similarity calculation.

Table 13: Analysis on the impact of incorporating rank fusion to attain unified results of our hybrid retrieval framework on EMNLP -Citations split

Retrieval Method	Recall@100	MRR@50
Ours w/o RRF & w/o A2A		
Naive Aggregation		
GPT-4o-Mini-2024-0718 + BGE-M3	21.61	32.61
Late Interaction(MaxSim)		
GPT-4o-Mini-2024-0718 + BGE-M3	26.00	31.53
Ours w/o A2A		
GPT-4o-Mini-2024-0718 + BGE-M3	**26.88**	**34.54**

Meanwhile, we also present comparison with late interaction strategy (Khattab and Zaharia, 2020; Santhanam et al., 2021), where only the maximum similarity for subquery and its corresponding sub-documents is aggregated to form original query, document similarity.(in our case, minimum l2 distance) Table 13 illustrates a performance drop when queries optimized in various aspects are used as subvector representations of original documents and are used to compute a single similarity value when matched with the segmented corpus of target candidates. This supports the validity of our design choice, where our ranking-merging system is more robust to noise, while allowing candidates that are strongly aligned in one aspect but unaligned otherwise to be retrieved.

Additional Diversity Analysis In this section, we provide further analysis on the improved diversity of semantics that our retrieval system presents. Figure 9 presents a linearly increasing trend where the Top@300 absolute Convex Hull Volume of 400 samples per benchmark split proportionally increases with respect to the retrieval depth, indicating that our Chain of Retrieval framework is able to diversify the semantics of search results as the rounds progress. Moreover, along with the quantitative metrics to represent the enhanced coverage that CoR presents, we visualize the distribution of the embedding vectors of Top@300 retrieved documents for eight sampled queries on a two-dimensional plane using **t-distributed Stochastic Neighbor Embedding(t-SNE)** (Van der Maaten and Hinton, 2008) and its distribution boundaries via a two-dimensional convex hull area of the projected data points in Figure 10. We simultaneously visualize the distribution of t-SNE data points for both ground truth candidates, and baseline retrieved results(including A2A retrieved results and F2F results using the same embedding model), and retrieved candidates from CoR. The T-SNE Convex Hull visualization results also advocate our claim, where our method typically displays a more dispersed embedding distribution compared to dispersion of retrieved document embeddings from baselines. In addition, the cases in Figure 10 demonstrate that through improved diversity within the retrieved pool of papers, CoR is able to retrieve more relevant papers that exist in ground-truth candidates, as coverage of the documents has increased compared to previous retrieval approaches.

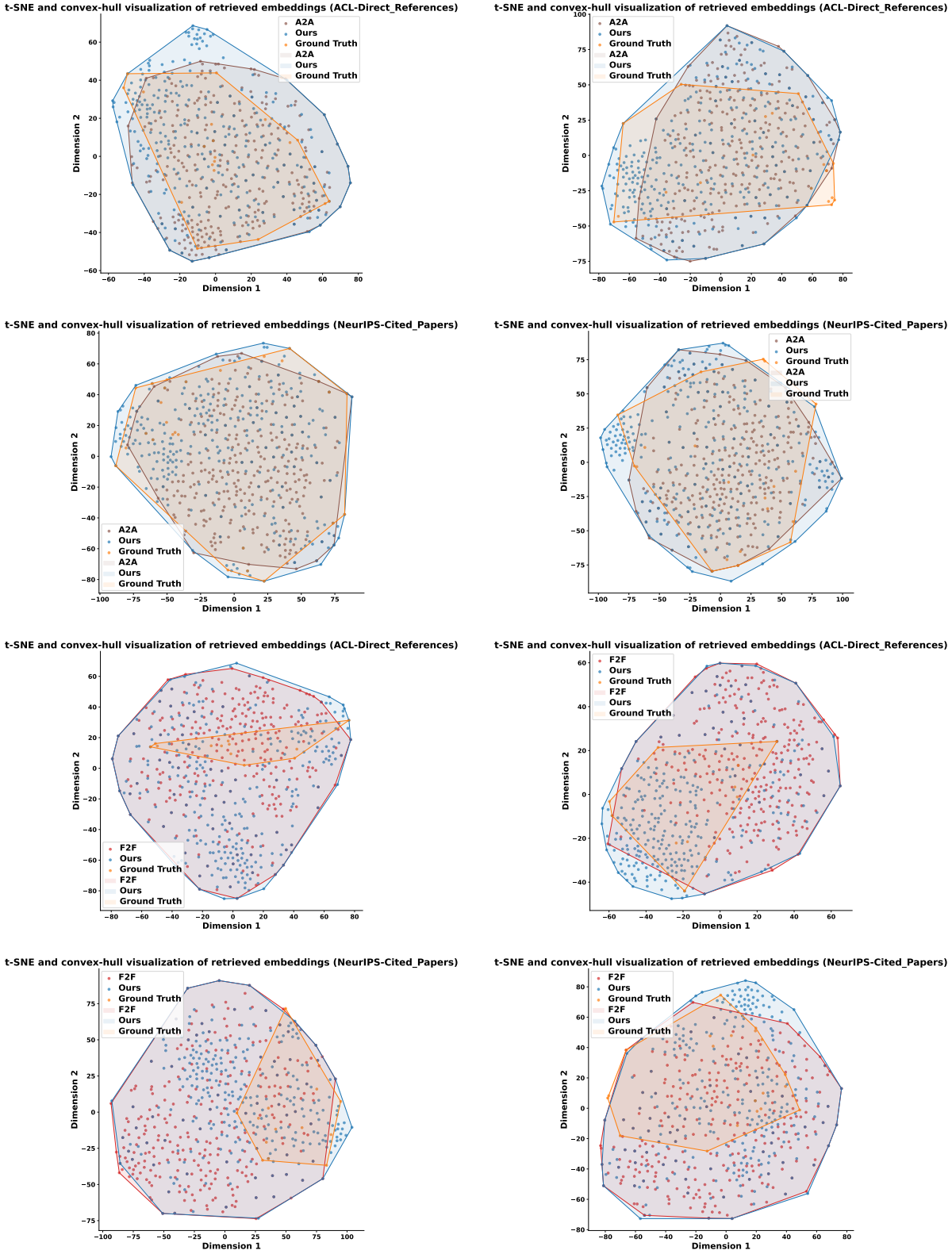

Figure 10: t-SNE and convex hull boundary visualization of Top@300 retrieved document embeddings across eight sampled queries from respective benchmark splits. We analyzed retrieved results when Jina-Embeddings-v2-Base-EN is used as the embedding model. Convex hull boundaries are shown for baseline F2F and A2A retrieval, and ARPA results are illustrated after 3 rounds of retrieval using our DPO-trained Llama-3.2-3B-Instruct query optimizer.

Table 14: Expanded Main Results on SCIFULLBENCH. Note that the results are reported after 3 rounds of retrieval.

		ICLR				NeurIPS			
		References		**Citations**		**References**		**Citations**	
	IR Method	Recall@200	Recall@300	nDCG@200	Recall@300	Recall@200	Recall@300	nDCG@200	Recall@300
	Domain-Specific Retriever								
	SciNCL-**A2A**	44.55	50.51	27.33	44.13	47.37	53.08	34.40	51.67
	SPECTER2-Base-**A2A**	43.43	48.99	28.66	45.23	46.15	51.82	36.05	52.95
	SPECTER2-Adapter-MTL CTRL-**A2A**	43.72	49.98	27.43	43.29	46.60	52.16	34.70	51.46
	SciMult-MHAExpert-**A2A**	37.13	42.56	22.57	36.88	41.98	47.52	30.14	45.25
Domain-Agnostic Retriever	**Jina-Embeddings-v2-BASE-EN**								
	A2A (Abstract-to-Abstract)	43.35	48.38	28.38	44.49	46.30	51.45	35.16	51.28
	F2F (Full-to-Full)	44.29	49.58	31.60	47.85	46.15	51.62	35.88	51.41
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**48.71**	**54.60**	**33.94**	**53.08**	**50.57**	**56.4**	**38.57**	**59.33**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**48.33**	**54.37**	**34.16**	**53.59**	**51.07**	**57.19**	**39.91**	**60.35**
	BGE-M3								
	A2A (Abstract-to-Abstract)	37.23	41.95	25.44	39.61	41.65	46.22	31.69	46.08
	F2F (Full-to-Full)	39.41	44.10	27.12	41.13	42.98	47.57	33.57	46.89
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**42.47**	**47.24**	**29.99**	**46.53**	**46.17**	**51.53**	**35.29**	**52.01**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**43.28**	**48.16**	**30.95**	**47.97**	**47.40**	**52.69**	**36.50**	**53.57**
	Inf-Retriever-v1-1.5B								
	A2A (Abstract-to-Abstract)	53.91	59.66	33.03	51.56	57.89	63.77	39.88	57.70
	F2F (Full-to-Full)	48.37	53.85	32.15	47.89	28.06	30.69	35.35	49.69
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**57.14**	**63.19**	**38.26**	**58.41**	**61.38**	**67.43**	**44.45**	**64.66**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**57.22**	**62.93**	**38.83**	**59.24**	**60.62**	**66.62**	**44.91**	**64.88**

		ACL				EMNLP			
		References		**Citations**		**References**		**Citations**	
	IR Method	Recall@100	Recall@200	mAP@30	Recall@100	Recall@100	Recall@200	mAP@30	Recall@100
	Domain-Specific Retriever								
	SciNCL-**A2A**	29.18	36.62	5.67	26.59	27.23	34.55	4.87	24.73
	SPECTER2-Base-**A2A**	28.28	35.81	6.21	27.96	25.47	33.12	5.33	26.23
	SPECTER2-Adapter-MTL CTRL-**A2A**	28.42	35.73	5.81	26.70	26.28	33.14	4.88	24.40
	SciMult-MHAExpert-**A2A**	25.96	33.07	5.11	24.11	23.37	30.32	4.13	20.92
Domain-Agnostic Retriever	**Jina-Embeddings-v2-BASE-EN**								
	A2A (Abstract-to-Abstract)	27.75	34.49	6.20	27.39	26.08	32.64	5.24	24.75
	F2F (Full-to-Full)	28.78	36.46	7.03	28.46	29.51	36.98	6.32	28.19
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**31.96**	**39.81**	**7.67**	**34.37**	**30.76**	**39.21**	**6.57**	**31.51**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**32.02**	**40.41**	**7.92**	**35.40**	**30.83**	**38.96**	**6.66**	**32.90**
	BGE-M3								
	A2A (Abstract-to-Abstract)	25.88	32.01	5.56	26.29	23.50	29.36	4.57	22.85
	F2F (Full-to-Full)	25.15	32.21	5.98	25.71	24.81	30.84	5.24	23.18
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**28.43**	**35.75**	**6.84**	**30.82**	**26.19**	**33.31**	**5.78**	**27.83**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**28.65**	**36.33**	**6.75**	**31.64**	**26.77**	**34.28**	**6.01**	**28.06**
	Inf-Retriever-v1-1.5B								
	A2A (Abstract-to-Abstract)	37.55	46.24	7.65	32.70	34.73	43.40	6.35	30.36
	F2F (Full-to-Full)	36.35	44.37	7.81	31.55	34.65	41.83	6.98	30.87
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**39.44**	**49.30**	**9.13**	**38.39**	**37.26**	**46.81**	**8.05**	**36.29**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**39.73**	**49.31**	**9.60**	**38.90**	**36.80**	**46.37**	**8.38**	**37.30**

Table 15: Main result with additional metrics. Note that the results are reported after 3 rounds of retrieval.

		ICLR				NeurIPS			
		References		**Citations**		**References**		**Citations**	
	IR Method	nDCG@300	Recall@100	Recall@100	Recall@200	nDCG@300	Recall@100	Recall@100	Recall@200
	Domain-Specific Retriever								
	SciNCL-**A2A**	34.04	34.92	29.49	38.40	38.44	38.10	36.90	46.04
	SPECTER2-Base-**A2A**	32.71	33.47	30.32	39.27	37.77	36.65	37.74	47.90
	SPECTER2-Adapter-MTL CTRL-**A2A**	33.91	34.51	29.13	38.09	38.54	37.30	37.24	46.04
	SciMult-MHAExpert-**A2A**	28.26	28.44	23.51	31.48	33.64	33.01	31.66	40.30
Domain-Agnostic Retriever	**Jina-Embeddings-v2-BASE-EN**								
	A2A (Abstract-to-Abstract)	34.43	34.71	30.09	39.19	39.12	37.91	36.85	45.87
	F2F (Full-to-Full)	35.27	35.17	33.25	42.44	38.82	37.25	37.43	46.15
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**37.19**	**38.36**	**36.50**	**47.12**	**40.56**	**41.16**	**42.55**	**53.21**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**36.82**	**38.52**	**36.52**	**47.43**	**41.00**	**41.36**	**43.81**	**54.53**
	BGE-M3								
	A2A (Abstract-to-Abstract)	29.51	29.69	26.66	35.16	34.65	33.28	33.40	41.42
	F2F (Full-to-Full)	31.78	32.46	28.50	36.06	35.63	34.81	34.31	42.05
	CoR w Llama-3.2-3B-Instruct (w DPO)	31.21	**32.88**	**32.13**	**41.44**	**36.40**	**36.47**	**37.58**	**47.20**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**32.10**	**33.76**	**33.07**	**42.81**	**37.58**	**37.34**	**39.19**	**48.33**
	Inf-Retriever-v1-1.5B								
	A2A (Abstract-to-Abstract)	43.03	44.49	35.03	45.42	48.64	48.20	42.97	52.69
	F2F (Full-to-Full)	39.01	39.21	34.65	43.00	23.17	22.95	37.46	45.09
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**44.65**	**47.15**	**41.44**	**52.33**	**49.96**	**50.87**	**48.48**	**59.01**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**44.65**	**46.81**	**41.99**	**52.79**	**49.77**	**50.78**	**49.11**	**59.37**

		ACL				EMNLP			
		References		**Citations**		**References**		**Citations**	
	IR Method	nDCG@300	Recall@300	Recall@200	Recall@300	nDCG@300	Recall@300	Recall@200	Recall@300
	Domain-Specific Retriever								
	SciNCL-**A2A**	26.72	41.63	34.69	39.79	25.52	39.78	33.05	38.02
	SPECTER2-Base-**A2A**	25.82	40.48	36.38	41.19	24.32	37.55	34.14	39.22
	SPECTER2-Adapter-MTL CTRL-**A2A**	25.76	40.13	34.77	39.61	24.87	37.59	32.36	37.47
	SciMult-MHAExpert-**A2A**	24.19	37.49	30.78	35.31	22.02	34.75	27.35	32.26
Domain-Agnostic Retriever	**Jina-Embeddings-v2-BASE-EN**								
	A2A (Abstract-to-Abstract)	25.50	38.80	35.33	40.52	24.41	37.00	32.43	37.32
	F2F (Full-to-Full)	27.27	41.66	35.88	40.95	27.83	41.73	35.51	40.34
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**28.52**	**44.92**	**42.93**	**48.05**	**28.15**	**43.96**	**40.31**	**46.00**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**28.62**	**45.67**	**43.75**	**49.00**	**27.80**	**43.99**	**41.86**	**47.60**
	BGE-M3								
	A2A (Abstract-to-Abstract)	23.65	35.81	33.76	38.40	22.25	33.29	29.63	34.15
	F2F (Full-to-Full)	24.39	36.67	32.64	37.01	23.88	35.08	30.08	34.73
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**25.12**	**40.19**	**39.48**	**44.46**	**24.12**	**38.24**	**35.63**	**40.87**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**25.77**	**40.71**	**40.22**	**45.29**	**24.66**	**38.97**	**37.01**	**42.14**
	Inf-Retriever-v1-1.5B								
	A2A (Abstract-to-Abstract)	34.17	51.51	41.77	47.33	32.68	48.68	39.00	43.72
	F2F (Full-to-Full)	33.35	49.59	38.86	43.72	32.37	46.40	38.92	43.57
	CoR w/ Llama-3.2-3B-Instruct (w/ DPO)	**35.09**	**55.14**	**48.23**	**54.13**	**33.86**	**52.11**	**46.29**	**51.77**
	CoR w/ QWEN-2.5-3B-Instruct (w/ DPO)	**35.16**	**54.68**	**48.82**	**54.78**	**33.65**	**51.91**	**46.51**	**52.40**

F Algorithm Details

Algorithm 2 Aspect-Aware Multi-Vector Retrieval

Require: Input paper D; Input paper abstract D_{abstract}; Top-k per query K;
Name of the Parent Retrieved Results PARENT; Corpora $\mathcal{C} \leftarrow \{\mathcal{C}_{\text{abstract}}, \mathcal{C}_{\text{chunked}}\}$;

Initialize: Functional aspect-aware LLM query optimizer agents $\mathcal{F} = \{f_{\text{M}}, f_{\text{E}}, f_{\text{R}}\}$

function ONEHOPRETRIEVAL(D, PARENT, $\mathcal{C}$, K)
  $R \leftarrow [\,]$ ▷ Memory to save aspect-aware retrieved results per document
  for all $f_i \in \mathcal{F}$ **do**
    NAME $\leftarrow$ PARENT$\|f_i$ ▷ Update name for aspect-aware retrieval for document D
    $q \leftarrow f_i(D)$
    $T \leftarrow \texttt{Retrieve}(q, C_{\text{chunked}}, K)$ ▷ Top-K Retrieved Results from chunked corpus
    $\mathcal{R}_q \leftarrow [\,h(x) \mid x \in T\,]$ ▷ Retrieved Results mapped back to paper
    R.APPEND(($\mathcal{R}_q$, PARENT, NAME, f_i))
  NAME $\leftarrow$ PARENT$\|$abstract
  $q \leftarrow D_{\text{abstract}}$
  $T \leftarrow \texttt{Retrieve}(q, \mathcal{C}_{\text{abstract}}, K)$ ▷ Top-K Retrieved Results from abstract corpus
  $\mathcal{R}_q \leftarrow [\,h(x) \mid x \in T\,]$ ▷ Retrieved Results mapped to paper
  R.APPEND(($\mathcal{R}_q$, PARENT, NAME, abstract))
  return R

Algorithm 3 Aspect-Aware Next Query Selection

Require: Root paper D; Max depth R; selection starting index γ; Top-k per query K
Aspect Aware Cache CACHE; memory to save query paper for next round retrieval Q_{next}

function INITIALASPECT(NAME) ▷ Obtain Initial Branching Aspect of the root document
  Parse NAME as (ROOT $\|\, a_1 \,\|\, a_2 \,\|\, \cdots$)
  return a_1

function SELECTNEXTQUERY(P, CACHE, γ)
  for all $r \in P$ **do**
    TOPK, PARENT, NAME, PREVIOUS ASPECT $\leftarrow$ r
    if PREVIOUS ASPECT $=$ abstract **then**
      CONTINUE
      ▷ Skip selection if previous retrieved results were from abstract query

    INITIAL ASPECT $\leftarrow$ INITIALASPECT(NAME)
    for $j = \gamma$ **to** K **do**
      DOC $\leftarrow$ TOPK[j]
      if DOC $\notin$ CACHE[INITIAL ASPECT] **then**
        CACHE[INITIAL ASPECT].append(DOC)
        Q_{next}.APPEND((DOC, NAME))
        break

      ▷ Select most similar document not in aspect-aware cache starting from the γ index.

Algorithm 4 Chain-of-Retrieval

Require: Root paper D; Max depth R; selection starting index γ; Top-k per query K; Corpora $\mathcal{C} \leftarrow \{\mathcal{C}_{\text{abstract}}, \mathcal{C}_{\text{chunked}}\}$

Initialize:
query queue $Q \leftarrow [(D, \textsc{Root})]$
memory to save retrieved results $\mathcal{M} \leftarrow [\,]$
$\textsc{Cache}[a] \leftarrow \emptyset \quad \forall a \in \{\text{M}, \text{E}, \text{R}\}$ ▷ Initialize Aspect-Aware Cache for Method, Experiment, and Research Question Aspect

for $r = 0$ **to** $R - 1$ **do**
  $\mathcal{M}[r] \leftarrow [\,]$, $Q_{\text{next}} \leftarrow [\,]$
  for all $(d, \textsc{parent})$ in Q **do**
    $P \leftarrow \textsc{OneHopRetrieval}(d, \textsc{parent}, \mathcal{C}, \text{K})$
    $\mathcal{M}[r].\textsc{Extend}(P)$
    $NQ \leftarrow \textsc{SelectNextQuery}(NQ, P, \textsc{Cache}, \gamma)$
  $Q \leftarrow NQ$

Algorithm 5 Recursive Post-Order Merging

Require: Root paper D; Max depth R; All Retrieved Results after depth R of retrieval $\mathcal{M}$;

Ensure: Final top-k candidates for D

function $\textsc{Group}(L)$ ▷ Group list of per round retrieved results with same parent nodes
  return $\{\, p \mapsto [\, x \in L : \text{Parent}(x) = p \,] \,\}$

function $\textsc{MergeSiblings}(L)$ ▷ Merge retrieved results with same parent nodes
  $S \leftarrow \{\}$
  for $(\textsc{Parent}, G) \in \textsc{Group}(L)$ **do**
    $S[\textsc{Parent}] \leftarrow \text{RRF}(G)$
  return S

function $\textsc{MergeEdges}(P, S)$ ▷ Merge retrieved results from parent nodes and merged child nodes
  $O \leftarrow [\,]$
  for $(\text{TopK}, \textsc{Parent}, \textsc{Name}) \in P$ **do**
    if $\textsc{Parent} \in Keys(S)$ **then**
      O.append(RRF(TopK, $S[\textsc{Parent}]$))
  return O

$r \leftarrow R - 1$

while $r > 0$ **do** ▷ Post-Order Merge, starting from the bottom level nodes
  $S \leftarrow \textsc{MergeSiblings}(\mathcal{M}[r])$
  $\mathcal{M}[r-1] \leftarrow \textsc{MergeEdges}(\mathcal{M}[r-1], S)$
  $r \leftarrow r - 1$
return $\text{RRF}(\mathcal{M}[0])$

G Case Study

Table 16: Example of retrieved documents given input paper document from EMNLP-Citations using 3round retrieval with ARPA, using DPO-trained Llama-3.2-3B-Instruct + Jina-Embeddings-V2-Base-EN setup. Note that due to extensive length of input documents, we only provide the title and abstract information of input paper, and titles for retrieved documents. Retrieved documents that belong to ground truth candidates are highlighted in blue.

Input Document Meta Data	**[Title]** Stance Detection on Social Media with Background Knowledge **[Abstract]** Identifying users2019 stances regarding specific targets/topics is a significant route to learning public opinion from social media platforms. Most existing studies of stance detection strive to learn stance information about specific targets from the context, in order to determine the user2019s stance on the target. However, in real-world scenarios, we usually have a certain understanding of a target when we express our stance on it. In this paper, we investigate stance detection from a novel perspective, where the background knowledge of the targets is taken into account for better stance detection. To be specific, we categorize background knowledge into two categories: episodic knowledge and discourse knowledge, and propose a novel Knowledge-Augmented Stance Detection (KASD) framework. For episodic knowledge, we devise a heuristic retrieval algorithm based on the topic to retrieve the Wikipedia documents relevant to the sample. Further, we construct a prompt for ChatGPT to filter the Wikipedia documents to derive episodic knowledge. For discourse knowledge, we construct a prompt for ChatGPT to paraphrase the hashtags, references, etc., in the sample, thereby injecting discourse knowledge into the sample. Experimental results on four benchmark datasets demonstrate that our KASD achieves state-of-the-art performance in in-target and zero-shot stance detection.
Ground-Truth Document Meta Data	**[Title]** A More Advanced Group Polarization Measurement Approach Based on LLM-Based Agents and Graphs **[Title]** A Survey of Stance Detection on Social Media: New Directions and Perspectives **[Title]** Chain of Stance: Stance Detection with Large Language Models **[Title]** A Challenge Dataset and Effective Models for Conversational Stance Detection **[Title]** Mitigating Biases of Large Language Models in Stance Detection with Counterfactual Augmented Calibration **[Title]** Multi-modal Stance Detection: New Datasets and Model **[Title]** A Logically Consistent Chain-of-Thought Approach for Stance Detection **[Title]** Stance Detection with Collaborative Role-Infused LLM-Based Agents **[Title]** Prompting and Fine-Tuning Open-Sourced Large Language Models for Stance Classification **[Title]** Ladder-of-Thought: Using Knowledge as Steps to Elevate Stance Detection **[Title]** Investigating Chain-of-thought with ChatGPT for Stance Detection on Social Media
Top@30 Retrieved Document Meta Data	**[Title]** A Survey of Stance Detection on Social Media: New Directions and Perspectives **[Title]** Stance Detection with Collaborative Role-Infused LLM-Based Agents **[Title]** A Survey on Stance Detection for Mis- and Disinformation Identification **[Title]** Prompting and Fine-Tuning Open-Sourced Large Language Models for Stance Classification **[Title]** Chain of Stance: Stance Detection with Large Language Models **[Title]** A Challenge Dataset and Effective Models for Conversational Stance Detection **[Title]** Enabling Contextual Soft Moderation on Social Media through Contrastive Textual Deviation **[Title]** Stance Detection on Social Media with Fine-Tuned Large Language Models **[Title]** Stance Detection in Web and Social Media: A Comparative Study **[Title]** Multi-modal Stance Detection: New Datasets and Model **[Title]** TATA: Stance Detection via Topic-Agnostic and Topic-Aware Embeddings **[Title]** A Benchmark for Cross-Domain Argumentative Stance Classification on Social Media **[Title]** Mitigating Biases of Large Language Models in Stance Detection with Counterfactual Augmented Calibration **[Title]** Advancing Annotation of Stance in Social Media Posts: A Comparative Analysis of Large Language Models and Crowd Sourcing **[Title]** DEEM: Dynamic Experienced Expert Modeling for Stance Detection **[Title]** FarExStance: Explainable Stance Detection for Farsi **[Title]** Investigating Chain-of-thought with ChatGPT for Stance Detection on Social Media **[Title]** Relative Counterfactual Contrastive Learning for Mitigating Pretrained Stance Bias in Stance Detection **[Title]** A Logically Consistent Chain-of-Thought Approach for Stance Detection **[Title]** Embeddings-Based Clustering for Target Specific Stances: The Case of a Polarized Turkey **[Title]** Reinforcement Tuning for Detecting Stances and Debunking Rumors Jointly with Large Language Models **[Title]** Examining the Influence of Political Bias on Large Language Model Performance in Stance Classification **[Title]** KCD: Knowledge Walks and Textual Cues Enhanced Political Perspective Detection in News Media **[Title]** Deciphering Political Entity Sentiment in News with Large Language Models: Zero-Shot and Few-Shot Strategies **[Title]** Argumentative Stance Prediction: An Exploratory Study on Multimodality and Few-Shot Learning **[Title]** A More Advanced Group Polarization Measurement Approach Based on LLM-Based Agents and Graphs **[Title]** Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research **[Title]** WIBA: What Is Being Argued? A Comprehensive Approach to Argument Mining **[Title]** KGAP: Knowledge Graph Augmented Political Perspective Detection in News Media **[Title]** "We Demand Justice!": Towards Social Context Grounding of Political Texts

Table 17: Example of retrieved documents given input paper document from Citations split in PatentFullBench, using single round retrieval results with ARPA, using untrained GPT-4o-Mini-2024-0718 as Query Optimizers . Note that due to extensive length of input documents, we only provide the title and abstract information of input document, and titles for retrieved documents. Retrieved documents that belong to ground truth candidates are highlighted in blue.

Input Document Meta Data	**[Abstract]** The present disclosure describes methods and systems directed towards providing scaled engagement and views of an e-sports event. Instead of providing the same distribution of live e-sport event data to all remote viewers of a live e-sports event, features associated with e-sports gaming network could be used to customize the distribution of live e-sport event data to promote immersive viewer experience. The enhanced immersion can also be carried out in a virtual reality or augmented reality setting. The features would be capable of providing additional information, different views, and a variety of different commentators for the e-sports event so that the viewer can be more engaged when viewing the particular e-sports event. With the increased engagement from remote viewers, the distribution of live e-sports event data can also be further modified for monetization by incorporating advertisements as well.
Ground-Truth Document Meta Data	**[Title]** Statistical driven tournaments **[Title]** Scaled VR engagement and views in an e-sports event **[Title]** Player to spectator handoff and other spectator controls **[Title]** Creation of winner tournaments with fandom influence **[Title]** User-driven spectator channel for live game play in multi-player games **[Title]** Methods and systems to increase interest in and viewership of content before, during and after a live event **[Title]** Discovery and detection of events in interactive content **[Title]** Integrating commentary content and gameplay content over a multi-user platform **[Title]** Statistically defined game channels **[Title]** Online tournament integration **[Title]** De-interleaving gameplay data
Top@30 Retrieved Document Meta Data	**[Title]** Methods and systems to increase interest in and viewership of content before, during and after a live event **[Title]** Scaled VR engagement and views in an e-sports event **[Title]** Creation of winner tournaments with fandom influence **[Title]** Player to spectator handoff and other spectator controls **[Title]** Discovery and detection of events in interactive content **[Title]** Real-time modifications in augmented reality experiences **[Title]** User-driven spectator channel for live game play in multi-player games **[Title]** Integrating commentary content and gameplay content over a multi-user platform **[Title]** Online tournament integration **[Title]** Integrating augmented reality experiences with other components **[Title]** AR-based connected portal shopping **[Title]** External screen streaming for an eyewear device **[Title]** Systems and methods for generating and facilitating access to a personalized augmented rendering of a user **[Title]** Systems, methods and apparatuses of digital assistants in an augmented reality environment and local determination of virtual object placement and apparatuses of single or multi-directional lens as portals between a physical world and a digital world component of the augmented reality environment **[Title]** Statistically defined game channels **[Title]** AR/VR enabled contact lens **[Title]** Shared augmented reality unboxing experience **[Title]** De-interleaving gameplay data **[Title]** Statistical driven tournaments **[Title]** Automated augmented reality experience creation based on sample source and target images **[Title]** Augmented reality unboxing experience **[Title]** Real-time video dimensional transformations of video for presentation in mixed reality-based virtual spaces **[Title]** Systems and methods for pinning content items to locations in an augmented reality display based on user preferences **[Title]** Avatar customization system **[Title]** Controlling interactive fashion based on body gestures **[Title]** Systems, methods, and apparatus for enhanced headsets **[Title]** User interfaces for wide angle video conference **[Title]** User interface for multi-user communication session **[Title]** Real-time upper-body garment exchange **[Title]** Applying animated 3D avatar in AR experiences

Query Case Analysis for Scientific Paper Retrieval

Document Meta-Data

Abstract: We introduce GAIA, a benchmark for General AI Assistants that, if solved, would represent a milestone in AI research. GAIA proposes real-world questions that require a set of fundamental abilities such as reasoning, multi-modality handling, web browsing, and generally tool-use proficiency. GAIA questions are conceptually simple for humans yet challenging for most advanced AIs: we show that human respondents obtain 92% vs. 15% for GPT-4 equipped with plugins. This notable performance disparity contrasts with the recent trend of LLMs outperforming humans on tasks requiring professional skills in e.g. law or chemistry. GAIA's philosophy departs from the current trend in AI benchmarks suggesting to target tasks that are ever more difficult for humans. We posit that the advent of Artificial General Intelligence (AGI) hinges on a system's capability to exhibit similar robustness as the average human does on such questions. Using GAIA's methodology, we devise 466 questions and their answer. We release our questions while retaining answers to 300 of them to power a leader-board hereby accessible.

Method-Focused Query:

GAIA, a benchmark for General AI Assistants, presents a novel approach to evaluating the capabilities of large language models (LLMs) in performing real-world tasks that require a range of fundamental abilities, including reasoning, multi-modality handling, web browsing, and tool-use proficiency. The methodology employed in developing GAIA involves creating a set of 466 carefully crafted questions and their answers, covering various assistant use cases such as daily personal tasks, science, and general knowledge. Each question is designed to admit a short, single correct answer, enabling simple and robust automatic evaluation.

To address the limitations of traditional LLM benchmarks, GAIA focuses on targeting tasks that are conceptually simple for humans yet challenging for most advanced AIs. The questions are intended to be easy to create, yet vary, root in the real world, and pose a unique combination of factual and inferential demands. Furthermore, GAIA's design principles prioritize interpretability, robustness against memorization, and easiness of use, aiming to provide a more nuanced evaluation framework for assessing the capabilities of LLMs.

The development of GAIA involved several stages, starting with the creation of a seed set of questions by the authors themselves, followed by extensive annotation and validation phases involving human curators and evaluators. Two independent annotators assessed each question for ambivalence, and discrepancies were resolved through iterative refinements. This rigorous process ensured that the questions remained unambiguous and suitable for evaluation.

Upon releasing the benchmark, GAIA evaluates LLMs' performance across various capabilities, including web browsing, multi-modality processing, coding, and diverse filetype reading. The evaluation procedure leverages a zero-shot prompt strategy, utilizing a prefix prompt to guide the model toward producing a factual answer. Results indicate significant disparities between human and LLM performances, highlighting the need for more sophisticated evaluation frameworks that account for the complexities inherent in real-world scenarios.

Notably, GAIA's methodology diverges from previous benchmarks by focusing on the overall performance of AI assistants rather than individual tasks or narrow skill sets. This shift underscores the importance of considering the broader cognitive landscape encompassed by general intelligence, particularly in light of the growing emphasis on achieving true Artificial General Intelligence (AGI). Future directions for GAIA entail exploring extensions to accommodate emerging topics, incorporating human-in-the-loop feedback, and investigating novel methodologies for tackling issues such as linguistic and cultural diversity.

Overall, GAIA represents a pioneering endeavor in establishing a standardized benchmark for

evaluating the capabilities of General AI Assistants, offering valuable insights into the strengths and weaknesses of contemporary LLM architectures. By fostering a deeper understanding of the intricacies underlying effective AI assistance, GAIA paves the way for continued innovation in the pursuit of AGI.

Experiment-Focused Query:

Here's a detailed summary of the experiment setup, datasets, evaluation metrics, baselines, and key findings of the paper "GAIA: A Benchmark for General AI Assistants":

Experiment Setup:

The authors propose a novel benchmark called GAIA (General AI Assistant) consisting of 466 carefully crafted questions and their answers. Each question requires a series of steps to arrive at a factual answer, mimicking real-world scenarios where AI assistants interact with humans. The questions cover various domains, including science, literature, and everyday life, and are designed to be conceptually simple yet challenging for current AI systems.

Datasets:

The authors develop a custom dataset of 466 questions, divided into three levels of difficulty: Level 1 (simplest), Level 2 (moderate), and Level 3 (most challenging). The questions are paired with their answers, reasoning traces, and metadata describing the required capabilities and tools.

Evaluation Metrics:

The authors employ a combination of human evaluation and automated scoring to assess the performance of AI systems on the GAIA benchmark. Specifically:

* **Human Score:**
  The authors annotate a subset of questions with their expected answers and measure the similarity between the predicted answer and the true answer.

* **Automated Scoring:**
  The authors implement a scoring mechanism that evaluates the correctness of the predicted answer based on the presence of certain keywords and phrases in the answer.

Baselines:

The authors compare the performance of several popular large language models (LLMs) on the GAIA benchmark, including GPT-4, AutoGPT, and human evaluators. The baseline performances serve as a reference point for assessing the effectiveness of the proposed GAIA benchmark.

Key Findings:

* Despite achieving impressive results on various NLP tasks, current LLMs struggle significantly on the GAIA benchmark, particularly on Level 3 questions.
* The authors observe that equipping LLMs with external tools and APIs can significantly enhance their performance on the GAIA benchmark, highlighting the importance of integrating external resources into AI systems.
* The GAIA benchmark demonstrates the limitations of traditional evaluation metrics, such as perplexity and BLEU score, in capturing the nuances of human-AI interaction.

The authors conclude that solving the GAIA benchmark represents a significant milestone toward developing truly general-purpose AI assistants that can effectively assist humans in a wide range of contexts.

Overall, the GAIA benchmark offers a comprehensive framework for evaluating the capabilities of AI systems in simulating real-world human-AI interactions, emphasizing the importance of

integrating external resources, improving reasoning and decision-making, and enhancing overall human-AI alignment.

Research-Question Focused Query:

Here's a detailed explanation of the scientific paper, focusing on its motivation, research questions, and contributions:

Motivation:

The paper introduces a novel benchmark called GAIA (General AI Assistant) aimed at assessing the capabilities of General AI Assistants. The author argues that traditional benchmarks, such as GLUE and MMLU, are insufficient for evaluating AI systems' ability to generalize across various tasks and domains. Current benchmarks tend to focus on narrow tasks, whereas real-world applications require AI systems to tackle complex, multifaceted problems. GAIA addresses this limitation by presenting a comprehensive framework for evaluating AI assistents' reasoning, multi-modality handling, web browsing, and tool-use proficiency.

Research Questions:

The primary research question driving this paper is: "Can we develop a benchmark that comprehensively evaluates the capabilities of General AI Assistants?" Specifically, the author seeks to investigate:

* Can AI systems generalize across various tasks and domains?
* How effective are current benchmarks in capturing the complexities of real-world applications?

To address these questions, the author designs and implements GAIA, a benchmark comprising 466 carefully crafted questions covering various aspects of AI assistance, including reasoning, multi-modality handling, web browsing, and tool-use proficiency.

Contributions: The paper contributes several significant ideas to the field of Natural Language Processing (NLP):

1. **Comprehensive Benchmark for General AI Assistants**: GAIA offers a standardized framework for evaluating AI assistents' capabilities, filling the gap left by existing benchmarks.
2. **Realistic Task Set**: The benchmark features questions that mimic real-world scenarios, encouraging AI systems to apply their knowledge and skills in context-dependent settings.
3. **Focus on Reasoning and Multi-Modality Handling**: GAIA emphasizes the importance of reasoning and multi-modality handling in AI assistents, highlighting areas where current models fall short.
4. **Evaluation Methodology**: The paper presents a systematic approach to evaluating AI assistents, incorporating human evaluation, model-based evaluation, and automatic scoring techniques.

Overall, the paper demonstrates the feasibility of developing a comprehensive benchmark for General AI Assistants, paving the way for improved evaluation methodologies and more sophisticated AI systems. . . .

Query Case Analysis for Patent-to-Patent Retrieval

Document Meta-Data

Abstract: The present disclosure describes methods and systems directed towards providing scaled engagement and views of an e-sports event. Instead of providing the same distribution of live e-sport event data to all remote viewers of a live e-sports event, features associated with e-sports gaming network could be used to customize the distribution of live e-sport event data to promote immersive viewer experience. The enhanced immersion can also be carried out in a virtual reality or augmented reality setting. The features would be capable of providing additional information, different views, and a variety of different commentators for the e-sports event so that the viewer can be more engaged when viewing the particular e-sports event. With the increased engagement from remote viewers, the distribution of live e-sports event data can also be further modified for monetization by incorporating advertisements as well.

Method-Focused Query: The disclosed method for providing scaled engagement and views of a live electronic sports (e-sport) event involves a systematic approach to enhance viewer immersion through customized data distribution. Initially, the method collects comprehensive information from the live e-sport event, including in-game elements, player statistics, and audience reactions. This data is processed to generate detailed statistics about the players, which are then used to create a dynamic display of the event from an in-game camera perspective. The generated display is further modified based on individual viewer preferences, which may include adjustments for augmented reality (AR) environments, different camera angles, and the integration of advertisements. The modifications are tailored to enhance the viewer's experience by providing relevant statistics, commentary, and interactive features, such as the ability to follow specific players or engage with other viewers. The final modified display is distributed to the viewer's device, ensuring a personalized and immersive viewing experience that can also facilitate monetization through targeted advertisements. The system architecture includes an event server for data collection, a gaming server for processing and display generation, and a public network for distribution, all of which work together to deliver a rich, engaging e-sport viewing experience.

Claim-Focused Query: The claims of this patent outline a method and system for enhancing viewer engagement and experience during live electronic sports (e-sports) events through customized data distribution. The independent claim details a method that includes collecting event data, generating player statistics, and creating a display that presents the event from an in-game camera perspective. This display is then modified based on viewer preferences, which may include augmented reality integration and personalized camera angles, before being distributed to remote viewers. The dependent claims specify refinements such as the inclusion of game-related information from player devices, audience reactions, advertisement incorporation, and user profile-based modifications that tailor the viewing experience to individual preferences. Additional claims address the provision of multiple viewing perspectives, interactive functions for viewers, and audio from other viewers, enhancing the immersive experience. The system claims describe an architecture comprising an event server and a gaming server that collectively manage data collection, processing, and distribution, while also allowing for the integration of advertisements and user interactions. Overall, the claims provide broad protection for methods and systems that facilitate a customizable and engaging viewing experience for remote participants in e-sports events.

Background Focused Query: The present invention pertains to the field of electronic sports (e-sports), specifically focusing on enhancing viewer engagement and experience during live e-sports events through customized data distribution. Existing methods of broadcasting e-sports events typically provide a uniform viewing experience to all remote viewers, lacking personalization and immersive features that could significantly enhance engagement. Current streaming platforms do not adequately leverage the wealth of data generated during live competitions, such as player statistics, audience reactions, and in-game dynamics, which limits the depth of viewer interaction and

understanding of the event. Furthermore, the integration of advanced technologies like virtual reality (VR) and augmented reality (AR) remains underutilized, preventing viewers from experiencing events in a more immersive manner akin to being physically present. There is a pressing need for systems that can dynamically modify the presentation of e-sports data based on individual viewer preferences, including customizable perspectives, additional commentary, and interactive features, while also addressing challenges related to monetization through targeted advertisements. Operating constraints include the need for real-time data processing, compatibility with various user devices, and the ability to handle high viewer throughput without compromising latency or quality. The application context spans competitive gaming leagues, online streaming services, and event organizers, all of whom seek to enhance viewer satisfaction and engagement while navigating the complexities of data management and user interaction in a rapidly evolving digital landscape.

H Prompts

Method Agent Prompt for Scientific Paper-to-Paper Retrieval

Instruction:
You are a specialized research assistant tasked with generating a structured, detailed explanation of a scientific paper based on its **Methodology**. Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the **methodology** of given paper.

IMPORTANT
- Your explanation is going to be used as a query to retrieve similar papers **METHOD** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.
- There are also other agents who are tasked with generating explanation on given paper. Unlike you, they are focused on experiments, and research questions of given paper. You must try to avoid overlap with possible explanations that the other two agents might generate.

Input:
You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the **METHODOLOGY** section, to extract its main approaches.

Key Considerations:
Highlight specific method/approach details, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

Output Format:
Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's approach, and key concepts in a concise yet informative manner. Focus on high-level insights rather than excessive detail.
You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Experiment Agent Prompt for Scientific Paper-to-Paper Retrieval

You are a specialized research assistant tasked with generating a structured, detailed explanation of a scientific paper's experimental setup.
Your goal is to clearly outline the **datasets, evaluation metrics, baselines, and key experimental findings**, making it easy to understand how the paper validates its approach.

IMPORTANT
- Your explanation is going to be used as a query to retrieve similar papers **EXPERIMENT** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.
- There are also other agents who are tasked with generating explanation on given paper. Unlike you, they are focused on methods, and research questions of given paper. You must try to avoid overlap with possible explanations that the other two agents might generate.

Input:
You will be provided with the full text of a scientific paper. Carefully analyze its content, paying particular attention to the **Experiments, Results, and Evaluation** sections to extract the key experimental details.

Key Considerations:
Datasets & Benchmarks: Clearly specify the datasets and benchmarks used for evaluation.
Baselines & Comparisons: Identify what methods or models the paper compares against.
Key Results & Insights: Summarize the main experimental findings without excessive detail.

Output Format:
Generate a clear, well structured and detailed paragraph that highlights the experimental methodology, datasets, evaluation metrics, baselines, and key results. Focus on high-level insights rather than excessive detail.

You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Research Question Agent Prompt for Scientific Paper-to-Paper Retrieval

Instruction:
You are a specialized research assistant tasked with generating a structured, detailed explanation of a scientific paper based on its **Motivation, Research Questions, and Contributions**. Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the **core problem, key contributions, and research objectives.**

IMPORTANT
- Your explanation is going to be used as a query to retrieve similar papers **RESEARCH QUESTION** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.
- There are also other agents who are tasked with generating explanation on given paper. Unlike you, they are focused on experiments, and methods of given paper. You must try to avoid overlap with possible explanations that the other two agents might generate.

Input:
You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the Introduction and Conclusion, to extract its main contributions, research questions, and motivations.

Key Considerations:
Highlight specific motivations, research questions, and contributions, avoiding vague or overly general descriptions.
Use precise language to ensure clarity while maintaining depth.

Output Format: Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's motivation, problem statement, research questions, and key contributions in a concise yet informative manner. Focus on high-level insights rather than excessive detail

You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Base Agent Prompt for Paper-to-Paper Retrieval

Instruction: You are a specialized research assistant tasked with generating a structured,detailed explanation of a scientific paper based three different aspects.

1. **Method-Specific Queries:**
   Generate a structured, detailed explanation of a scientific paper based on its Methodology Your goal is to provide a clear yet comprehensive summary that makes it easy to identify relevant papers by emphasizing the methodology of given paper.

   IMPORTANT
   - Your explanation is going to be used as a query to retrieve similar papers METHOD wise.
   - Make sure that your explanation can retrieve highly relevant papers easily.

   Input:
   You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the **METHODOLOGY** section, to extract its main approaches.

   Key Considerations:
   Highlight specific method/approach details, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

   Output Format:
   Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's approach, and key concepts in a concise yet informative manner. Focus on high-level insights rather than excessive detail You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

2. **Experiment-Specific Queries:**
   Generate a structured, detailed explanation of a scientific paper's experimental setup Your goal is to clearly outline the datasets, evaluation metrics, baselines, and key experimental findings, making it easy to understand how the paper validates its approach.

   IMPORTANT
   - Your explanation is going to be used as a query to retrieve similar papers **EXPERIMENT** wise.
   - Make sure that your explanation can retrieve highly relevant papers easily.

   You will be provided with the full text of a scientific paper. Carefully analyze its content, paying particular attention to the Experiments, Results, and Evaluation sections to extract the key experimental details.

   Key Considerations:
   Datasets & Benchmarks: Clearly specify the datasets and benchmarks used for evaluation.
   Baselines & Comparisons: Identify what methods or models the paper compares against.
   Key Results & Insights: Summarize the main experimental findings without excessive detail.

   Output Format:
   Generate a clear, well structured and detailed paragraph that highlights the experimental methodology, datasets, evaluation metrics, baselines, and key results. Focus on high-level insights rather than excessive detail. You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

3. **Research Question-Specific Queries:**
   Generate a structured, detailed explanation of a scientific paper based on its Motivation, Research Questions, and Contributions. Your goal is to provide a clear yet comprehensive summary that

makes it easy to identify relevant papers by emphasizing the core problem, key contributions, and research objectives.

IMPORTANT

- Your explanation is going to be used as a query to retrieve similar papers **RESEARCH QUESTION** wise.
- Make sure that your explanation can retrieve highly relevant papers easily.

Input:

You will be given the full text of a scientific paper. Carefully analyze its content, with a particular focus on the Introduction and Conclusion, to extract its main contributions, research questions, and motivations.

Key Considerations:

Highlight specific motivations, research questions, and contributions, avoiding vague or overly general descriptions. Use precise language to ensure clarity while maintaining depth.

Output Format:

Generate a well-structured, detailed and yet clear paragraph that effectively captures the paper's motivation, problem statement, research questions, and key contributions in a concise yet informative manner. Focus on high-level insights rather than excessive detail You must not include title and abstract of given paper in your answer, and try to put it into your own words with high level reasoning after reading the paper.

Output:

Return a structured json file for respective Method-Specific Queries, Experiment-Specific Queries, and Research Question-Specific queries, with the respective keys as "method_query", "experiment_query", and "research_question_query". Each key should contain the generated explanation as a string.

Method-Focused Agent Prompt for Patent Retrieval

You are a specialized research assistant tasked with generating a structured, detailed explanation of a patent's **METHOD**.

Your goal is to summarize how the invention works in practice: its core technical principles, implementation procedures, system architecture, and functional mechanisms. This explanation will be used as a query to retrieve patents with similar methods or implementation techniques.

IMPORTANT

- Focus **ONLY** on the Detailed Description and Embodiments sections.
- Capture the technical processes, structures, system flows, or algorithms.
- Paraphrase at a high-level while keeping enough technical detail for retrieval.
- Other agents will generate explanations about the invention claims and the background/problems of the given patent. You must avoid overlap with those aspects.

Input

You will be provided with the full text of a patent. Carefully analyze its Detailed Description, Examples, and Figures to extract the methodological details.

Key Considerations

Here are some key aspects that you may focus on.

- **Core technical principle:** what mechanism enables the invention to function?
- **Implementation structure:** key components, modules, or subsystems.
- **Operational flow:** how the method proceeds step by step.
- **Variants or embodiments:** different configurations or modes of execution.
- **Integration context:** how it interacts with external systems or environments.

Output Format

Produce a single, clear, and well-structured paragraph that captures the **METHOD** of the invention. Write in concise, retrieval-friendly technical language that highlights implementation strategies and system functionality.

Claim-Focused Agent Prompt for Patent-to-Patent Retrieval

You are a specialized research assistant tasked with generating a structured, detailed explanation of a patent's **CLAIMS**.

Your goal is to clearly outline the legal protection scope: what is being claimed, how broad the claims are, and what technical features or components are covered.
This explanation will be used as a query to retrieve patents with similar **CLAIM** structures and protection scopes.

IMPORTANT

- Focus **ONLY** on the **CLAIMS** section.
- Emphasize the scope of protection, claimed components, processes, and relationships among them.
- Avoid background information, prior art, or detailed embodiments (those are handled by other agents).
- Do not directly copy claim sentences; paraphrase into concise, high-level, retrieval-friendly language.
- Include both the broad independent claims (core invention scope) and notable dependent claims (specific refinements).
- Other agents will generate explanations about the invention **details/method** and the **background/problems** of the given patent. You must avoid overlap with those aspects.

Input

You will be provided with the full text of a patent. Carefully analyze the **CLAIMS** section.

Output Format

Produce a clear, well-structured paragraph that captures the scope and focus of the **CLAIMS**. Use neutral, technical language suitable for retrieval so that patents with overlapping protection scopes can be surfaced.

Background-Focused Agent Prompt for Patent-to-Patent Retrieval

You are a specialized research assistant tasked with generating a structured, detailed explanation of a patent's **Problem / Background** (i.e., the technical field and the shortcomings of prior art).

Your goal is to clearly state the problem space—technical domain, prior-art limitations, unresolved challenges, operating constraints, and desired (non-solution) performance objectives—so that similar patents can be retrieved by shared problem patterns rather than specific solutions.

IMPORTANT

- Your explanation is going to be used as a query to retrieve patents that are similar in the **PROBLEM SPACE.**
- Include concrete domain terms (components, materials, data types), relevant standards/regulations, operating conditions (e.g., throughput, latency, power, temperature), failure modes, and application contexts that characterize the problem.
- Other agents will generate explanations about the invention details/method and the claims made from the input patent. You must avoid overlap with those aspects.

Input

You will be provided with the full text of a patent.

Key Considerations

Here are some key aspects that you may focus on.

- **Technical Field:** the domain and subdomain (e.g., "wireless edge inference for medical imaging").
- **Prior Art & Limitations:** concrete bottlenecks, inefficiencies, failure cases, safety/privacy concerns, interoperability issues.
- **Unmet Technical Objectives:** what must be achieved (targets or constraints).
- **Operating Constraints & Edge Cases:** data distributions, environmental/thermal limits, network conditions, power/memory budgets, size/weight/cost constraints, lifecycle/maintenance issues.
- **Application Contexts & Stakeholders:** deployment scenarios, industries, users, and integration boundaries.

Output Format

Generate a single, clear, well-structured paragraph in neutral technical language that captures **ONLY** the **Problem / Background**. Paraphrase in your own words at a high level; do not copy text or include any solution, embodiment, or claim content. The paragraph should be information-dense and retrieval-friendly so that it can surface patents that confront the same technical challenges.