LitSearch: A Retrieval Benchmark for Scientific Literature Search

Anirudh Ajith[1] **Mengzhou Xia**[1] **Alexis Chevalier**[1,2] **Tanya Goyal**[1]
Danqi Chen[1] **Tianyu Gao**[1]
[1]Princeton Language and Intelligence (PLI), Princeton University [2]BCG X
{anirudh.ajith,mengzhou,achevalier,tanyagoyal,
danqic,tianyug}@princeton.edu

Abstract

Literature search questions, such as "where can I find research on the evaluation of consistency in generated summaries?" pose significant challenges for modern search engines and retrieval systems. These questions often require a deep understanding of research concepts and the ability to reason over entire articles. In this work, we introduce LitSearch, a retrieval benchmark comprising 597 realistic literature search queries about recent ML and NLP papers. LitSearch is constructed using a combination of (1) questions generated by GPT-4 based on paragraphs containing inline citations from research papers and (2) questions about recently published papers, manually written by their authors. All LitSearch questions were manually examined or edited by experts to ensure high quality. We extensively benchmark state-of-the-art retrieval models and also evaluate two LLM-based reranking pipelines. We find a significant performance gap between BM25 and state-of-the-art dense retrievers, with a 24.8% difference in absolute recall@5. The LLM-based reranking strategies further improve the best-performing dense retriever by 4.4%. Additionally, commercial search engines and research tools like Google Search perform poorly on LitSearch, lagging behind the best dense retriever by 32 points. Taken together, these results show that LitSearch is an informative new testbed for retrieval systems while catering to a real-world use case.[1]

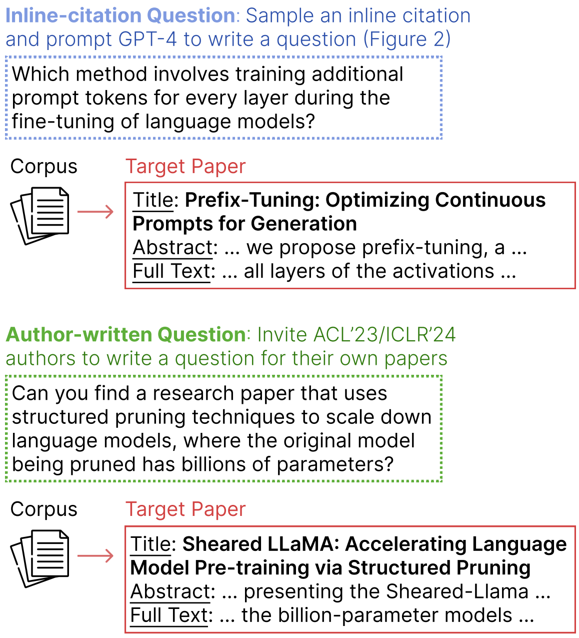

Figure 1: Examples of *inline-citation questions* and *author-written questions* from LitSearch. The questions are often challenging and require deep understanding of target papers to answer correctly.

1 Introduction

Finding literature via a specific search query—for example, to collect related work, to check if a method has been proposed before, or to recall a previously-seen paper—is a critical task for researchers. Developing systems that recommend citations pertinent to such inquiries has the promise to enhance researchers' productivity and expedite scientific discovery (Färber and Jatowt, 2020). However, this task is inherently challenging as it often requires deep domain expertise and reasoning through lengthy papers.

Prior to this study, the task of citation recommendation is often formalized by using inline citation mentions from existing papers as queries, and the cited papers as targets (He et al., 2010; Gu et al., 2022). For instance, given the citation mention "*RoBERTa and T5 are based on recent advances in masked language modeling* `[citation]`," the text surrounding the citation mention is used as a retrieval query, and the cited paper is the target literature. However, directly using inline citations often leads to queries that are noisy, broad (e.g., "*Large Language Models* `[citation]`"), or

[1]Our dataset and code are available at `https://github.com/princeton-nlp/LitSearch`.

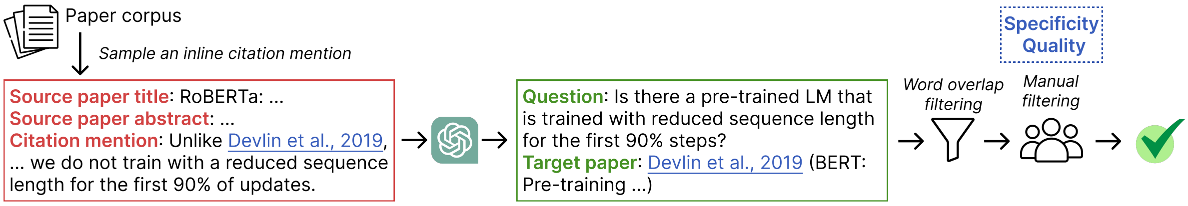

Figure 2: The pipeline to generate inline-citation questions. We first sample a citation mention and prompt GPT-4 to write a question. Then we filter the questions by word overlap (with the target paper title) and manual inspection. The manual inspection annotates the specificity and the quality of the questions (rubrics in Table 1).

context-dependent (e.g., "*We follow the hyperparameters of* `[citation]`").

In this work, we propose a new litearture retrieval benchmark called LitSearch. As illustrated in Figure 1, a literature search question seeks papers that meet specific criteria, closely reflecting actual research workflows. LitSearch consists of two subsets: (1) For **inline-citation questions**, we sample citation mentions from a collection of scientific papers and use GPT-4 (OpenAI, 2023) to rewrite them into literature search questions (Figure 2). We retain questions with a low word overlap with the title of the target papers, and perform manual examination to ensure high quality. (2) For **author-written questions**, we invited authors of ACL 2023 and ICLR 2024 papers to write literature search questions for their own papers. This subset is also manually examined and filtered to remove any inaccurate or easy questions. LitSearch contains 597 questions in total, each paired with one or more scientific papers as the ground truth.

LitSearch has several unique characteristics: (1) To our best knowledge, LitSearch is the *first* dataset featuring realistic literature search questions, providing a new testbed for citation recommendation and retrieval systems. (2) LitSearch is *challenging*, requiring deep understanding and reasoning over entire articles. The average document length (6,041/134 words for full text/titles and abstracts) is significantly longer than most existing retrieval benchmarks (e.g., 56 for Nguyen et al., 2017). (3) LitSearch is of *high quality*, with all questions manually examined by the authors of this work.

We conduct extensive experiments on both state-of-the-art retrieval models and reranking with large language models (LLMs). On LitSearch, the best dense retrieval model, GritLM (Muennighoff et al., 2024), achieves an average recall@5 of 74.8%, beating BM25 (Robertson et al., 2009) by 24.8%. The recall@5 of GritLM is further improved by 4.4% with GPT-4 reranking. On the other hand, commercial search engines and research tools like Google Search perform poorly on this task, only achieving an average recall@5 of 42.8% at most. Futhermore, compared to existing retrieval datasets from BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2022), LitSearch effectively reflects the performance differences among various embedding models, making it an informative testbed for evaluating state-of-the-art retrieval systems.

2 LitSearch

Our benchmark LitSearch consists of (a) a large corpus of scientific papers $\mathcal{P}$ and (b) pairs of literature search questions and one or more target papers from $\mathcal{P}$. Our desiderata are scientific questions that researchers may use while conducting literature surveys. We use two different strategies to collect such questions: (1) we construct questions using the surrounding context from inline citations in published papers (Section 2.1), and (2) we invited the authors of recent conference publications to manually write questions about their own papers (Section 2.2). For both subsets, we ensure high question quality via manual inspection and filtering conducted by the authors of this work (Section 2.3).

2.1 Inline-citation Questions

We define the following concepts for the ease of description: (a) An *inline citation mention* is a paragraph from the main text of a paper that mentions another paper. For example, this paragraph from the RoBERTa paper (Liu et al., 2019), "... *Unlike Devlin et al., (2019), ... we do not train with a reduced sequence length for the first 90% of updates* ..." mentions the BERT paper (Devlin et al., 2019). (b) The *source paper* is the paper the inline citation mention is sampled from. (c) A *target paper* is a paper that is cited by the inline citation mention.

Figure 2 provides an overview of our data collection methodology for inline-citation questions. We utilize the Semantic Scholar Open Research Corpus

Specificity	
0	*Broad*. There should exist no more than 20 papers that fit the question. Example: What are some parameter-efficient fine-tuning methods?
1	*Specific*. There should exist no more than 5 papers that fit the question. Example: Which method involves training additional prompt tokens for every layer during the fine-tuning of language models?
Quality	
0	*Discarded*. The question is factually wrong, unrealistic, overly broad/specific, or too easy.
1	*Acceptable*. The question can be somewhat out of distribution of what researchers ask, or relatively easy due to high overlap with the title/abstract.
2	*Good*. The question makes a challenging yet meaningful literature search question.

Table 1: Annotation rubrics for the manual filtering (conducted by the authors of LitSearch).

(S2ORC; Lo et al., 2020), a large corpus of academic papers obtained from publishers, archives, and the Internet. We randomly sample inline citation mentions from the S2ORC[2] and prompt GPT-4 to rewrite these citation mentions into literature search questions. These questions are filtered to remove those with a high word overlap with the title of the target papers, and are further manually examined to ensure high quality.

Sampling inline citation mentions. We limit the target papers to be only from the ACL Anthology, for the purpose of aligning with the expertise of the manual annotators, i.e., authors of this work. However, we do not limit where the source papers come from. Depending on the source papers, we call these questions *ACL sourced* or *non-ACL sourced*.

Prompting GPT-4 to generate questions. Given a sampled citation mention, we prompt GPT-4 (OpenAI, 2023) to generate a literature search question. In the prompt, we provide (1) the sampled paragraph (the inline citation mention) from the source paper and (2) the titles of the cited papers, and instruct GPT-4 to generate a literature search question based on the paragraph that would be answered by one or more of the papers cited in the paragraph. We use in-context learning (Brown et al., 2020) and include two demonstrations. The prompt we use can be found in Table 12.

Word-overlap filtering. We notice that inline-citation questions generated in the last step can have very high word overlap with the target paper titles, which makes their retrieval trivial even for BM25 and suggests that the questions may not be of interest to researchers. We calculate the word overlap as the *percentage of words in the generated question that are also included in the target paper titles*. We filter out ACL sourced questions that have an overlap score higher than 0.3 and non-ACL sourced questions that have an overlap score higher than 0.1. In this step, we filter out 5% of the ACL sourced questions and 80% of the non-ACL sourced questions.

2.2 Author-written Questions

Besides generating questions using existing inline citation mentions, we also collect questions directly from human annotators. As writing literature search questions requires deep understanding of the research field and the target paper, we invite researchers to write search queries that are answered by their own published papers. One additional benefit of this setup is that the correctness of the questions is better guaranteed.

We invited authors of ACL 2023 and ICLR 2024 papers to write one literature search question for each of their papers. We chose the two venues as they were among the latest natural language and machine learning conferences at the time of the data collection, hence the papers represent the latest research development and are unlikely to have already been included in the pre-training data of LLMs and retrievers used in our evaluations. We sent out invitations to 623 ACL 2023 authors and 404 ICLR 2024 authors, and received 175 questions from ACL 2023 authors and 117 questions from ICLR 2024 authors.

2.3 Manual Filtering to Ensure High Quality

Finally, the authors of this work manually examine every question from both the inline-citation and author-written subsets and annotate these for *specificity* and *quality* (guidelines in Table 1). Questions that are too general (there are more than 20 papers from the corpus can fit the question) are assigned a quality score of 0 and are excluded. We include only questions with a quality score of 1 or 2 in the final dataset. We also rewrite questions if they

[2]S2ORC provides detailed citation information for most inline citations (including the position of the citation in the paper text and the unique identifier of the target paper), enabling us to easily sample inline citation mentions and match them to S2ORC papers. We used the 2024-03-26 version.

	Broad				Specific				Total #Q
	#Q	Avg. Len	Overlap	Avg. #P	#Q	Avg. Len	Overlap	Avg. #P	
Inline-citation Questions	120	20.6	0.33	1.21	231	22.1	0.34	1.07	351
Author-written Questions	35	15.8	0.43	1.03	211	17.9	0.43	1.00	246

Table 2: Statistics for LitSearch. Please refer to Table 14 for more detailed statistics of each subset. "#Q": number of questions. "Overlap": the fraction of words in the question that are also included in the titles and abstracts of the target papers. "Avg. #P": average number of target papers.

have minor mistakes and can be fixed easily. Each question is assigned to one author for annotation.

As the examples in Table 1 show, questions of both specificity types can be realistic and valuable, but they exhibit distinct traits. We use the specificity scores to distinguish *broad* and *specific* questions in the evaluation.

For the inline-citation subset, we manually examined 382 ACL sourced questions and 450 non-ACL sourced questions. 26% (98 instances) of the ACL questions and 56% (253 instances) of the non-ACL questions are kept. For the author-written subset, since all questions are written by experts, we avoid rewriting them as much as possible. In the end, we kept 89% (155) questions from ACL 2023 authors and 78% (91) questions from ICLR 2024 authors.

2.4 Dataset Statistics

Our final dataset contains 597 questions, with 351 in the inline-citation subset and 246 in the author-written subset. Dataset statistics, including the number of questions, the average question length, and the average word overlap between the question and the target papers (titles and abstracts), are presented in Table 2. We find that author-written questions are shorter and have a higher word overlap rate with the target papers (0.43 vs. 0.33 for inline-citation questions). This is expected: when writing questions for their own papers, authors tend to re-use terminology from their papers and focus on the main findings which are usually included in the abstracts or titles. In contrast, inline-citation questions can be anchored to any span of the reference documents, irrespective of the main findings or the main focuses of the target papers.

2.5 The Retrieval Corpus

The LitSearch retrieval corpus $\mathcal{P}$ consists of ACL Anthology and ICLR papers extracted from S2ORC (see Appendix C for details). We do not use the full S2ORC corpus for efficiency reasons. In total, this yields $64,183$ papers ($59,383$ ACL Anthology papers and $4,807$ ICLR papers)[3]. The average number of words for the documents in $\mathcal{P}$ is 134 / 6,041 (titles and abstracts / full texts).

3 Experiments

3.1 Experimental Setup

We compare the performance of different retrieval systems (enumerated below) on our LitSearch benchmark. Due to the limited context sizes of existing embedding models, we only use the paper titles and abstracts to embed the papers in our retrieval corpus $\mathcal{P}$ by default.

For all systems we compare, we report the recall@K for both the broad and specific subsets of LitSearch. We report results for $K = 5, 20$ for the specific subset and $K = 20$ for the broad subset; these values (5 and 20) correspond to the guidelines followed by the authors while determining the specificity of a given question (see Table 1).

3.2 Baselines

We benchmark both retrieval models and LLM-based rerankers in this work.

Retriever models. We evaluate using the classic BM25 algorithm (Robertson et al., 2009), as well as several state-of-the-art dense retrieval (embedding) models, including GTR (Ni et al., 2022), Instructor (Su et al., 2023), E5 (Wang et al., 2022), and GritLM (Muennighoff et al., 2024).[4] More details are provided in Appendix D.

LLM-based reranking. In addition to vanilla retrieval, we also use strong LLMs (GPT-4[5] in our case) to rerank the top retrieved results from the above retrievers. We use two strategies:

Vanilla reranking. We include the top-n retrieved papers (titles and abstracts) in the context

[3] 7 papers are common to both subsets.

[4] We use the following corresponding checkpoints from `https://huggingface.co/`: GTR-T5-large, Instructor-XL, E5-large-v2, and GritLM-7B.

[5] We use gpt-4-1106-preview in our experiments.

	Inline-citation			Author-written			Avg.	Avg.
	Broad	Specific		Broad	Specific		Broad	Specific
	R@20	R@5	R@20	R@20	R@5	R@20	R@20	R@5
BM25	37.4	38.5	55.8	48.6	62.6	73.5	39.9	50.0
GTR-T5-large	45.7	38.5	51.5	37.1	40.8	55.9	43.8	39.6
Instructor-XL	56.3	48.9	60.0	57.1	55.9	70.1	56.5	52.3
E5-large-v2	55.8	50.4	63.9	54.3	62.6	75.8	55.4	56.2
GritLM-7B	**69.7**	**67.7**	**77.9**	**74.3**	**82.5**	**89.1**	**70.8**	**74.8**
GPT-4 reranking (w/ BM25)	54.9	60.0	67.5	**77.1**	76.8	82.9	59.9	68.0
GPT-4 one-hop (w/ BM25)	62.0	64.1	71.6	74.3	73.5	77.7	64.8	68.6
GPT-4 reranking (w/ GritLM)	**74.7**	**73.2**	**79.9**	**77.1**	**85.8**	**92.4**	**75.3**	**79.2**
GPT-4 one-hop (w/ GritLM)	72.9	70.3	78.4	74.3	84.4	87.2	73.2	77.0

Table 3: Main experiment results of LitSearch. Here we only use the titles and abstracts of papers for retrieval and reranking. We report recall@20 (R@20) for broad questions and recall@5,20 (R@5, R@20) for specific questions. "Broad" and "specific" correspond to the annotations during our manual filtering stage (defined in Table 1).

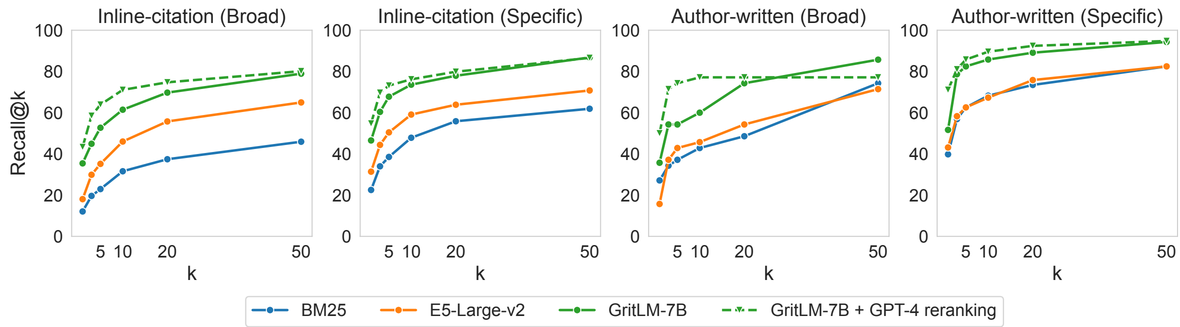

Figure 3: We demonstrate detailed retrieval results using BM25, E5 and GritLM up to $k = 50$. Additionally, we show the effect of applying GPT-4 reranking over GritLM retrieval results.

and prompt GPT-4 to rerank these based on the question (see our prompt in Table 13). This is similar to prior works (Sun et al., 2023b; Ma et al., 2023). We use $n = 100$, resulting in an average context length of 13,844 words.

One-hop reranking. Inspired by Tang et al. (2023), we leverage the fact that for some questions, there may exist lexically similar inline citation mentions in the retrieval corpus. Due to our data collection pipeline, this is particularly true for the ACL sourced inline citation questions. We posit that the retrieval models will be able to retrieve these source papers (that cite the target papers) based on the questions.

We extract the top m retrieved papers and construct a new candidate list by adding papers cited by each of these seed retrieved papers. We concatenate papers in the following order, skipping duplicates: [rank-1 paper p_1, papers cited by p_1, rank-2 paper p_2, papers cited by p_2, ..., p_m, papers cited by p_m]. To avoid very long contexts, we truncate this list after the first n papers and use the same prompt for GPT-4 based reranking as above. In our experiments, we use $m = 50$ and $n = 200$ resulting in average length of 27,544 words.

3.3 Results

We outline the performance of the above systems on LitSearch in Table 3. First, we observe that all instruction-finetuned embedding models, e.g. Instructor, E5, and GritLM, substantially outperform BM25 on our benchmark. In fact, they also perform better than the GTR model. Overall, we found that GritLM-7B achieves the best performance (70.8 recall@20 on broad questions and 74.8 recall@5 on specific questions), leaving a large gap compared to other baselines.

Impact of reranking. We also report the performance improvement brought by the reranking methods on the weakest (BM25) and strongest (GritLM) retrievers in Table 3. We observe that both vanilla and one-hop reranking improve over the base retrieval performance. For example, on the specific subset of inline questions, the vanilla GPT-4 reranking improves the recall@5 of BM25 and GritLM by 21.5% and 5.5% respectively. Interestingly, the

	Inline (specific)		Author (specific)	
	Qual=1 R@5	Qual=2 R@5	Qual=1 R@5	Qual=2 R@5
BM25	36.4	30.6	62.2	55.0
GTR-T5-large	42.0	31.4	40.7	36.9
Instructor-XL	55.1	39.5	58.5	48.6
E5-large-v2	48.4	42.6	61.5	57.7
GritLM-7B	67.3	58.7	80.0	76.6

Table 4: Comparison of retrieval performance on different quality (qual) questions. Generally, retrievers report lower performance on the Qual=2 questions, i.e. those deemed more challenging in our manual annotation.

improvements from one-hop reranking are generally lower than vanilla reranking across all subsets, when using GritLM as the base retriever; this shows that our benchmark cannot be easily "gamed" by mimicking the data collection pipeline or exploiting similar citation mentions to the question from other papers.

Impact of question specificity. Table 3 and Figure 3 show that retrieval systems generally report higher recall performance on the specific subset. This is expected: there exists a smaller number of "competing" papers, i.e. those that also satisfy the search question, for the specific subset in the retrieval corpus. Note that our human annotation tagged questions with approximately 5 relevant papers as specific and 20 relevant papers as broad (see Table 1 for details). We keep both subsets in our dataset as this stratified reporting presents a more nuanced view of retriever capabilities.

Inline-citation vs. author-written questions. We observe very different performance trends for the two subsets (Table 3 and Figure 3). In particular, inline-citation questions are harder than author-written questions for all retriever systems, on both broad and specific questions.

We attribute this difference to the higher semantic or lexical overlap of author-written questions with the paper titles and abstracts (see Table 2 for statistics). This reflects expected tendency of paper authors to formulate the questions around the main contributions from the abstracts and re-use terminologies. Such annotator biases have been widely discussed in prior data collection efforts as well, particularly when humans write content from scratch (Gururangan et al., 2018).

Impact of question quality. Table 4 compares how retrieval models perform on different quality subsets. Recall that we manually annotated

	Inline		Author	
	Broad R@20	Spec R@5	Broad R@20	Spec R@5
BM25	**37.4**	**38.5**	48.6	62.6
w/ full	18.6	23.8	**65.7**	**71.6**
GTR-T5-large	**45.7**	38.5	37.1	**40.8**
w/ full	43.9	**39.4**	**45.7**	39.8
Instructor-XL	**56.3**	48.9	**57.1**	55.9
w/ full	53.0	**50.9**	**57.1**	**56.9**
E5-large-v2	55.8	**50.4**	54.3	**62.6**
w/ full	**56.9**	48.7	**60.0**	62.1
GritLM-7B	69.7	**67.7**	**74.3**	**82.5**
w/ full	**70.8**	63.4	65.7	73.0

Table 5: Retrieval results of using only titles and abstracts vs. using titles, abstracts, and full text (w/ full). We do not observe consistent improvements from including the full text for existing retrieval models.

the quality of all questions (Table 1). We observe that questions with a quality score of 2, i.e. determined to be more realistic and difficult by manual annotators, are consistently more challenging for retrieval models. This demonstrates the high annotation quality of our manual inspection step. The presence of these different quality questions in our dataset leads to higher diversity and better coverage over the varied information seeking needs of users.

4 Analysis

4.1 Does Including More Paper Content Improve Retrieval Performance?

In the previous section, we only used the titles and abstracts (on average 134 words) to encode the papers in the retrieval corpus. Here, we evaluate whether encoding more paper content can improve retrieval performance. For all retriever models compared, we create embeddings using the full paper text (on average 6,041 words) up to their allowed context lengths.[6] We compare this setting against our default setting (only titles and abstracts).

Our results are outlined in Table 5. Surprisingly, we find that the addition of more paper text does not improve performance on LitSearch consistently. In fact, we only observe substantial improvement on the author-written broad questions for BM25 and some embedding models. In other cases, more text more often hinders instead of improving performance. Note that the maximum con-

[6]The maximum context lengths for GTR-T5-large, Instructor-XL, E5-large-v2 and GritLM-7B are 512, 512, 512 and 2048 tokens respectively.

	ACL		Non-ACL	
	Broad R@20	Specific R@5	Broad R@20	Specific R@5
BM25	38.8	39.4	36.9	38.2
GTR-T5-large	37.2	39.4	48.9	38.2
Instructor-XL	48.6	43.9	59.1	50.9
E5-large-v2	46.6	46.2	59.1	52.1
GritLM-7B	72.4	65.9	68.8	68.5
With BM25				
Reranking	51.1	59.8	56.2	60.0
One-hop	**65.2**	**71.2**	**60.8**	**61.2**
With GritLM				
Reranking	80.3	**72.7**	**72.7**	**73.3**
One-hop	**81.3**	67.4	69.9	71.5

Table 6: Comparison of retrieval performance on the ACL vs. non-ACL sourced inline-citation questions. Results show that the performance improvement from one-hop reranking over BM25 is subtantially higher for ACL sourced questions.

text length of the tested models is 2,048 (GritLM) and the average length of their training data is even shorter—for example, the commonly used MS-MARCO (Nguyen et al., 2017) and NaturalQuestions (Lee et al., 2019) have an average document length of 56 and 79. This is significantly shorter than the full text of papers from our retrieval corpus averaging 6,041 words in length, potentially leading to the unsatisfying performance when using full texts with embedding models.

4.2 Does the Source of Inline Citation Questions Matter?

Next, we study how the different sources of inline-citation questions affect the model performance. Table 6 outlines the performance of retrieval models on ACL sourced vs. non-ACL sourced inline-citation questions. Our results show that the two different sets report similar trends and model rankings for different retrieval models, particularly on the specific subset of questions. Interestingly, we find that the performance improvement from one-hop reranking is very different for the ACL and non-ACL questions.

For BM25, we observe that one-hop reranking is significantly better than the vanilla reranking on ACL sourced questions (+11.4% recall@5 on specific); but the gap is much smaller on non-ACL sourced questions (+1.2% recall@5 on specific). We posit that this is because BM25 can better exploit the data annotation pipeline on the ACL sourced questions. It can likely first identify the source ACL paper where the citation mention comes from, and then find the target paper via one-hop reranking. Including the non-ACL questions to LitSearch prevents systems from exploiting such "shortcuts" as non-ACL source papers are not part of the retrieval corpus.

For GritLM, we do not observe similarly large performance gains when using one-hop reranking. We hypothesize that this is because when using GritLM, the initial top retrieval results already include the target papers and the one-hop strategy does not bring further improvement.

	Inline (specific) R@5	Author (specific) R@5
BM25	38.5	62.6
GritLM-7B	67.7	82.5
Google Search	23.1	62.5
Google Scholar	20.5	17.5
Elicit	23.1	17.5

Table 7: Recall@5 for commercial search engines on a random subset of 80 specific questions. Search engines generally report poor performance. Note that the comparison is not apples-to-apples as search engines use a much larger retrieval corpus.

4.3 Performance of Search Engines

In practice, researchers use search engines like Google Search, Google Scholar, or Elicit[7] to search for relevant papers for their scientific queries. We conduct a human study to understand how these search engines perform on LitSearch: We randomly sample 80 questions (all specific; 40 inline-citation and 40 author-written) from our dataset. We manually input these questions into the above search engines and report recall@5.[8] We note that this is not an apples-to-apples comparison against the retrieval models in earlier sections due to the discrepancy in the retrieval corpus.

Table 7 outlines the results of our human study. It shows that all three search engines deliver similarly low recalls on inline-citation questions. On the author-written questions, Google Search performs much better than the other two. Although not directly comparable, this performance is generally worse than the embedding models, demonstrating the potential of these strong dense retrieval models for citation recommendation applications.

[7] https://elicit.com/

[8] For Google Search, we consider the top-5 academic papers in the search results and ignore other webpage results. We only consider the academic papers on the first page of search results, even if this number is lower than 5.

	MSMARCO	**SCIDOCS**	**NQ**	**ArXiv**	**LitSearch** (broad)	**LitSearch** (specific)
GTR-T5-large	42.7	15.5	55.1	17.5	23.3	30.4
Instructor-XL	41.6	17.4	57.2	19.8	32.8	41.2
E5-large-v2	43.5	20.5	63.4	27.0	27.1	45.3
GritLM-7B	42.0	24.4	70.3	34.3	44.1	60.3

Table 8: Comparison between LitSearch and existing retrieval benchmarks. All reported numbers are nDCG@10 for a direct comparison.

4.4 Comparing Other Retrieval Benchmarks

We compare model performance on LitSearch to several popular retrieval benchmarks included in BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2022)—namely MSMARCO (Nguyen et al., 2017), SCIDOCS (Cohan et al., 2020), and NQ (Lee et al., 2019).[9] We also compare to ArXiv (Gu et al., 2022), a previous citation recommendation benchmark directly using inline citations as queries. Table 8 shows that LitSearch generally agrees with existing retrieval benchmarks. However, LitSearch can differentiate retriever models better: for example, the gap between GritLM and E5 on LitSearch (specific) is 15 points (nDCG@10), while they perform almost the same on MSMARCO. LitSearch provides an informative testbed that can effectively reflect the recent (and future) advancement in embedding models.

5 Related Work

Citation recommendation. The community has propose a number of citation recommendation datasets (Färber and Jatowt, 2020), including global citation recommendation datasets (directly using a paper as the query and papers it cites as target papers; Cohan et al., 2020; Bhagavatula et al., 2018), and local citation recommendation datasets (using inline citation mentions as queries; He et al., 2010; Medić and Snajder, 2020; Jeong et al., 2020; Gu et al., 2022). There are also language models and retrieval models specifically trained for scientific document understanding and retrieval tasks, such as SciBERT (Beltagy et al., 2019) and SPECTER (Cohan et al., 2020). Compared to existing citation recommendation datasets, LitSearch is comprised of manually annotated, natural language literature search questions, providing a more realistic and challenging evaluation for citation recommendation systems.

[9]We use the results from the MTEB benchmark website: `https://huggingface.co/spaces/mteb/leaderboard`.

Retrieval benchmarks. There have been numerous datasets evaluating retrieval systems from Wikipedia (Kwiatkowski et al., 2019; Lee et al., 2019), web queries (Nguyen et al., 2017), biomedical questions (Voorhees and Tice, 2000), and more. Recently, there have been several benchmarks combining multiple datasets and evaluating retrieval or embedding models across different domains and different use cases, such as KILT (Petroni et al., 2021), BEIR (Thakur et al., 2021), and MTEB (Muennighoff et al., 2022). LitSearch offers a unique perspective by exploring the novel literature search question type, effectively complementing the existing benchmarks.

Retrieval systems. Traditional retrieval systems rely on bag-of-word algorithms such as TF-IDF and BM25. Dense retrieval (embedding) models have gained more popularity due to their abilities to do semantic search without relying on exact keyword matches (Pennington et al., 2014; Reimers and Gurevych, 2019). State-of-the-art dense models are mostly adopted by fine-tuning pre-trained language models (Devlin et al., 2019; Touvron et al., 2023) with a contrastive learning objective on either supervised or unsupervised data (Karpukhin et al., 2020; Gao et al., 2021; Izacard et al., 2022; Ni et al., 2022; Khattab and Zaharia, 2020). Recent development introduces "instructions" when encoding queries and documents, which significantly improves the versatility of embeddings across tasks (Su et al., 2023; Wang et al., 2022; Wu et al., 2022; Lee et al., 2024; BehnamGhader et al., 2024).

6 Conclusion

In this paper, we propose LitSearch, a new retrieval benchmark comprised of 597 manually-curated literature search questions. LitSearch includes a inline-citation question set and an author-written question set, both undergoing manual inspections from the authors of LitSearch. We conduct extensive experiments with BM25, state-of-the-art embedding models, and LLM reranking. Our experiments demonstrate the superior performance of state-of-the-art instruction-finetuned embedding models, with additional improvement via GPT-4-based reranking. We also verify that commercial search engines like Google struggle on LitSearch questions. The comparison with existing retrieval benchmarks shows that LitSearch better highlights the different performance of retrieval systems.

Limitations

Even though we manually examined the dataset, there still exist questions that are either slightly out of distribution compared to what researchers would ask, or too easy due to high overlap with the target papers. The author-written questions are easier than we expected, as writing challenging literature search questions is non-trivial even for experienced researchers. Even though we experimented with several state-of-the-art systems, it was not an exhausted evaluation and we left out more sophisticated retrieval or reranking systems. This research primarily focuses on only English questions and research papers.

Ethics Statement

The research artifact of this paper, LitSearch, is manually inspected and is ensured to have no unsafe or inappropriate content. However, the process to generate the dataset may introduce certain biases: for example, the inline-citation questions contain more target papers that have high citations due to the sampling; the author-written questions only cover ACL 2023 and ICLR 2024 papers.

Acknowledgements

We want to acknowledge Dan Friedman, Howard Yen, Jiayi Geng, Lucy He, and other members of the Princeton NLP group for their useful feedback and discussion. We also acknowledge all the ACL 2023 and ICLR 2024 authors that contributed questions to LitSearch (listed in Appendix A). Tianyu Gao is supported by an IBM PhD Fellowship. This work is gratefully supported by an NSF CAREER award (IIS-2239290), and Microsoft Azure credits through the "Accelerate Foundation Models Academic Research" Initiative.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *Preprint*, arXiv:2404.05961.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

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.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Michael Färber and Adam Jatowt. 2020. Citation recommendation: approaches and datasets. *Int. J. Digit. Libr.*, 21(4):375–405.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910.

Nianlong Gu, Yingqiang Gao, and Richard H. R. Hahnloser. 2022. Local citation recommendation with hierarchical-attention text encoder and scibert-based reranking. In *Advances in Information Retrieval: 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10–14, 2022, Proceedings, Part I*, page 274–288, Berlin, Heidelberg. Springer-Verlag.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. 2010. Context-aware citation recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 421–430, New York, NY, USA. Association for Computing Machinery.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, 124(3):1907–1922.

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.

O. Khattab and Matei A. Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Association for Computational Linguistics (ACL)*, pages 6086–6096.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Association for Computational Linguistics (ACL)*, pages 4969–4983.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Zoran Medić and Jan Snajder. 2020. Improved local citation recommendation based on context enhanced with global information. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 97–103, Online. Association for Computational Linguistics.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. MS MARCO: A human-generated MAchine reading COmprehension dataset.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 9844–9855.

OpenAI. 2023. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

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

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023a. Is chatgpt good at search? investigating large language models as re-ranking agent. *ArXiv*, abs/2304.09542.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023b. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 14918–14937.

Michael Tang, Shunyu Yao, John Yang, and Karthik Narasimhan. 2023. Referral augmentation for zero-shot information retrieval. *Preprint*, arXiv:2305.15098.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

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

Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. 2022. Grit: A generative region-to-text transformer for object understanding. *Preprint*, arXiv:2212.00280.

A Annotator Acknowledgments

We would like to thank Marah I Abdin, Jaewoo Ahn, Kabir Ahuja, Xi Ai, Satoshi Akasaki, Anastasios N Angelopoulos, Jinheon Baek, Eslam Mohamed Bakr, Pablo Barceló, Claudio Battiloro, Jonas Belouadi, Abhik Bhattacharjee, Valeriia Bolotova, Pengshan Cai, Nitay Calderon, Qingqing Cao, Defu Cao, Souradip Chakraborty, Jun Shern Chan, Sachin Chanchani, Yulong Chen, Yiming Chen, Xinyuan Chen, Nuo Chen, Hanjie Chen, Xiudi Chen, Zeming Chen, An-Chieh Cheng, Xize Cheng, Cheng-Han Chiang, Josef Dai, David Dale, Yue Deng, Yifan Deng, Shizhe Diao, Bosheng Ding, Xuan Long Do, Yilun Du, Yupei Du, Salijona Dyrmishi, Dante Everaert, Zhenghan Fang, Bahare Fatemi, Jiazhan Feng, Shangbin Feng, Patrick Fernandes, Javier Ferrando, Christopher Fifty, Sarah E Finch, Matthew Finlayson, Lea Frermann, Mikhail Galkin, Songyang Gao, Ziteng Gao, Silin Gao, Sara Ghazanfari, Nathan Godey, Navita Goyal, Xinran Gu, Yuxian Gu, Yu Gu, Anchun Gui, Jiacheng Guo, Ashim Gupta, Paul Hagemann, Tianxing He, Zhengfu He, Juncai He, Leonhard Hennig, Konstantin Hess, Jennifer Hu, Xiaoyang Hu, Zhilei Hu, Weidong Huang, Yichong Huang, Ayyoob Imani, Qi Jia, Yifan Jiang, Hanwen Jiang, Yiding Jiang, Yang Jin, Youngjin Jin, Zhijing Jin, Emmeran Johnson, Josef Jon, David Jurgens, Ehsan Kamalloo, Junmo Kang, Jian Kang, Mikhail Khodak, Hyunjae Kim, Soroush Abbasi Koohpayegani, Suhas Kotha, Jeongyeol Kwon, Sunjae Kwon, Philippe Laban, Zhibin Lan, Nayoung Lee, Deokjae Lee, Celine Lee, Heejun Lee, Jie Lei, Wenhao Li, Yafu Li, Yufei Li, Yanzeng Li, Yanzhou Li, Ziqiang Li, Zhaoyi Li, Ziheng Li, Xiaonan Li, Yinghao Li, Yu Li, Chengrui Li, Yingjie Li, Yunlong Liang, Baohao Liao, Kezhou Lin, Licong Lin, Enrico Liscio, Xiangyan Liu, Chenzhengyi Liu, Yixin Liu, Xingbin Liu, Haolin Liu, Xiao Liu, Yajiao Liu, Meng Liu, Tianyang Liu, Wei Liu, Qingyu Lu, Pan Lu, Junyu Lu, Zhengyi Luo, Yang Luo, Ang Lv, Junhyung Lyle, Jiajun Ma, Kaixin Ma, Ziqiao Ma, Mounica Maddela, Chaitanya Malaviya, Zhiyu Mei, Ethan Mendes, Fatemehsadat Mireshghallah, Niloofar Mireshghallah, Mircea Mironenco, Takeru Miyato, Fengran Mo, Xinyi Mou, Niklas Muennighoff, Cheolwon Na, Piotr Nawrot, Mang Ning, Longshen Ou, Siqi Ouyang, Lorenzo Pacchiardi, Ziqi Pang, Sara Papi, Letitia Parcalabescu, Tanmay Parekh, Aleksandar Petrov, Lucía Pitarch, Moritz Plenz, Manish Prajapat, Joan Puigcerver, Valentina Pyatkin, Shuofei Qiao, Yujia Qin, Chengwei Qin, Sigal Raab, Hossein A Rahmani, Siyu Ren, Yubing Ren, Ruiyang Ren, Yangjun Ruan, Michael J Ryan, Shoumik Saha, Vageesh Saxena, Michael Saxon, Alexander Scarlatos, Agam Shah, Erfan Shayegani, Behzad Shayegh, Xiangqing Shen, Sheng Shen, Ruizhe Shi, Zhengliang Shi, Kensen Shi, Ziyi Shou, Prasann Singhal, Jasivan Alex Sivakumar, Junru Song, Chunjin Song, Nikita Srivatsan, Michal Štefánik, Hao Sun, Mingjie Sun, Weiwei Sun, Zhiqing Sun, Xiaohang Tang, Liyan Tang, Eshaan Tanwar, Jiayan Teng, Davide Testa, Changyao Tian, Yufei Tian, Eric Todd, Benjamin Towle, Austin Tripp, Yi Tu, Rheeya Uppaal, Lazar Valkov, Neeraj Varshney, Artem Vazhentsev, Yiming Wang, Qifan Wang, Zhaoyang Wang, Lirui Wang, Zhicheng Wang, Weiqi Wang, Jiaan Wang, Boshi Wang, Haiming Wang, Huimin Wang, Yun-Cheng Wang, Runzhe Wang, Yu Wang, Yidong Wang, Licheng Wen, Te-Lin Wu, Yu-Yu Wu, Qianhui Wu, Dongming Wu, Tong Wu, Zijun Wu, Mengzhou Xia, Jian Xie, Yiming Xie, Weiwen Xu, Yi Xu, Xilie Xu, Derek Xu, Shohei Yamasaki, Hao Yan, Chenghao Yang, Xianjun Yang, Sen Yang, Bingsheng Yao, Qinyuan Ye, Fan Yin, Haneul Yoo, Kiyoon Yoo, Xinyan Velocity Yu, Jianfei Yu, Qiying Yu, Mo Yu, Zichun Yu, Yue Yu, Youliang Yuan, Zihao Yue, Xiang Yue, Yuanwen Yue, Daoguang Zan, Zhiyuan Zeng, Guangtao Zeng, Yuheng Zha, Runzhe Zhan, Jiaxu Zhang, Zhexin Zhang, Chen Zhang, Xinlu Zhang, Yabo Zhang, Renrui Zhang, Kechi Zhang, Ruoyu Zhang, Feng Zhang, Siyan Zhao, Junhao Zheng, Wenjie Zheng, Ming Zhong, Yan Zhou, Pei Zhou, Yangqiaoyu Zhou, Aojun Zhou, Xuekai Zhu, Luyao Zhu, Yanqiao Zhu, Dele Zhu, Andrew Zhu, Wenjie Zhuo and Caleb Ziems for contributing author-written questions about their ACL 2023 and/or ICLR 2024 papers.

B Annotation Details

We provide instructions regarding manually inspecting questions in Table 1. We sent out emails and Google Forms to recruit ACL 2023 and ICLR 2024 authors for author-written questions, and the templates can be found in Table 10 and Table 11 respectively.

C Retrieval Corpus

The LitSearch retrieval corpus $\mathcal{P}$ consists of ACL Anthology and ICLR papers extracted from S2ORC. Here we describe how we identify those papers in S2ORC: We isolate ACL anthology papers from S2ORC by identifying entries whose metadata includes an ACL Anthology ID. We identify ICLR papers utilizing a combination of the venue-based queries to Semantic Scholar's Academic Graph API and by title-matching using titles of accepted papers scraped from the official ICLR website.

D Retriever details

We list the full HuggingFace checkpoint paths corresponding to the dense retrievers we use in Table 9. We use the following instructions for the instruction-finetuned embedding models: "`Represent the research question for retrieving relevant research paper abstracts:`" for encoding queries; "`Represent the title and abstract of the research paper for retrieval:`" for encoding papers when using Instructor-XL for retrieval using paper titles and abstracts; when performing retrieval using paper titles and abstacts with GritLM-7B, we use the instruction "`Given a research query, retrieve the title and abstract of the relevant research paper`".

Retriever	HuggingFace Checkpoint
GTR-T5-large	`sentence-transformers/gtr-t5-large`
Instructor-XL	`hkunlp/instructor-xl`
E5-large-v2	`intfloat/e5-large-v2`
GritLM-7B	`GritLM/GritLM-7B`

Table 9: HuggingFace checkpoints we use for each dense retriever.

E Prompts and Additional Statistics

Table 12 shows the prompt we use for generating inline-citation questions via GPT-4. Table 13 shows the reranking prompt for GPT-4. Table 14 shows a more detailed statistics about LitSearch.

```
Hi {{annotator name}},

We hope this email finds you well!

First, congrats on your paper's acceptance to {{conference name}}! We are [REDACTED] from
[REDACTED] who are working on constructing a new challenging retrieval benchmark where the task is
to retrieve relevant research papers given a research query. Would you be willing to dedicate 2
minutes to write a literature-search question about your {{conference name}} paper? Here's the link
to the google form: {{link}}.

Your contribution will help us build better, more challenging evaluations for large language models.
We will make sure to list you as a contributor to our benchmark (unless you prefer otherwise). Thank
you!

Best,
{{author 1}}
{{author 2}}
```

Table 10: Email template sent out to ICLR 2024 and ACL 2023 authors for collecting author-written questions.

```
Dear {{author name}},

Thanks for contributing to our literature-search question collection effort!

We are trying to create a new challenging retrieval benchmark where the task is to retrieve relevant
research papers given a research query. For example, the research query "Which paper first found
that large language models can do in-context learning?" should retrieve the GPT-3 paper. We could
use your help for creating such a query that should be answered by your own paper!

Could you provide one high-quality, literature-search-type query about your paper that you think
would be challenging even for state-of-the-art retrieval systems? Please make sure to follow these
guidelines when writing your query. Your query must be

(1) Realistic: It should be plausible that a researcher working in a related field may ask this
exact question. Do not ask a question that puts too many unrealistic constraints (such as "What work
that uses the NaturalQuestions dataset trains for 15 epochs and uses a learning rate of 3e-5?").

(2) Specific: Your query should be answered by/correspond to one particular (representative) paper
or a small number of papers. Do not ask an overly broad question like "What are some multimodal
models?" Instead, you can ask "Which multimodal model was the first to use interleaved image and
text data?"

(3) Challenging: Our goal is to collect a set of queries that are extremely challenging even for
SOTA retrievers today. Make sure not to submit a query that can easily be answered via keyword
matching or a Google search like "Which paper proposed masked language modeling?" (BERT). Instead,
you can use the following formats to make the question more challenging:
(a) Asking a detailed question that the abstract does not cover: "Which text pre-training paper
first used the data mixture of wikipedia and bookscorpus?" (BERT)
(b) Rephrasing (reducing word overlap to the paper): "Is there such a machine learning dataset,
where for some questions, there is no correct answer and model should abstain?" (SQuAD v2)
(c) Asking about the main technique innovation: "Is there any paper that combines distillation and
structured pruning for language models?" (CoFi pruning)

Thank you for your contribution! We will make sure that:
- Your name will be credited in our paper unless you choose otherwise.
- We will make the dataset available for open-source development.
```

Table 11: Instructions provided in the Google Forms sent to ICLR 2024 and ACL 2023 authors for collecting author-written questions.

```
I will provide you with a excerpt from an scientific article that cites various papers in the
positions {cite_001}, {cite_002}, etc. I would like you to write *general* questions about the
academic literature using this paragraph, which will be answered with the corresponding citations.
Here are the constraints you must follow:
(1) The questions should make sense and be interesting without reading the paragraph. The questions
should be general, and you should only use the paragraph to find the ground-truth answer.
(2) Try to ask questions which cover as many citations as possible and are as detailed as possible.
Pack as much information into the question as you can.
(3) If the context is not clear, skip the citation. Make sure that every question is of the highest
quality, as these questions will be used for important work concerning information extraction from
the scientific literature.
(4) The answer should ONLY contain the citation key.


TEXT: We argue there are two underlying motivations for the ad text generation task, especially for
product descriptions. Application-wise, the utility is to improve the seller experience for
e-commerce services when registering a new product. The generated descriptions can reduce the need
for manual data entry, and potentially improve sales due to better descriptions (in terms of
attractiveness, structure, and persuasiveness). Research-wise, ad text generation is an
under-studied task, and arguably a good proxy for persuasive text generation
{cite_016}{cite_017}{cite_018}{cite_019}.

PAPER TITLES:
- cite_016: Is this post persuasive? ranking argumentative comments in online forum
- cite_017: On the role of discourse relations in persuasive texts
- cite_018: Measuring online debaters’ persuasive skill from text over time
- cite_019: Analyzing the Persuasive Effect of Style in News Editorial Argumentation
- cite_020: Persuaide! an adaptive persuasive text generation system for fashion domain
- cite_021: A statistical framework for product description generation
- cite_022: SILVER: Generating persuasive Chinese product pitch

QUESTION: I want to read some papers that try to study and quantify how persuasive text can be. I’m
coming at this from an applications-perspective, as I’m interested in using these insights for
product development. Could you give me a list of papers to read?
ANSWER: cite_016, cite_017, cite_018, cite_019

QUESTION: Has anyone looked at automating advertisement text generation specifically for fashion
items?
ANSWER: cite_020

QUESTION: I’m thinking about going into the computer-retail business and I’m wondering if it’s
possible to generate persuasive text to sell more computers?
ANSWER: cite_021

QUESTION: Could you recommend some readings for articles that generate persuasive text in Chinese
aimed at advertising?
ANSWER: cite_022

(Additional Demonstration Omitted)

TEXT: {{PARAGRAPH}}

PAPER TITLES:
{{CITATIONS}}
```

Table 12: The prompt used for generating questions from inline citations using GPT-4.

```
(system) You are RankGPT, an intelligent assistant that can rank papers based on their relevancy
to a research query.

(user) I will provide you with the abstracts of 100 papers, each indicated by number identifier
[]. \nRank the papers based on their relevance to research question: {{query}}.

(assistant) Okay, please provide the papers.

(user) [1] Title: {{Title1}}\n Abstract: {{Abstract1}}

(assistant) Received passage [1].

(user) [2] Title: {{Title2}}\n Abstract: {{Abstract2}}

(assistant) Received passage [2].

...

(user) [100] Title: {{Title100}}\n Abstract: {{Abstract100}}

(assistant) Received passage [100].

(user) Search Query: {{query}}. \nRank the 100 papers above based on their relevance to the
research query. The papers should be listed in descending order using identifiers. The most
relevant papers should be listed first. The output format should be [] > [], e.g., [1] > [2].
Only respond with the ranking results, do not say any words or explain.
```

Table 13: The prompt used for reranking retrieved documents using GPT4 (adapted from Sun et al., 2023a).

	Broad			**Specific**			**Total #Q**
	#Q	Avg. L	Overlap	#Q	Avg. L	Overlap	
Inline-Citation Questions							
ACL-sourced	32	24.8	0.33	66	26.0	0.35	98
Non-ACL-sourced	88	19.1	0.33	165	20.5	0.34	253
Author-written Questions							
ACL 2023	25	14.5	0.41	130	18.1	0.42	155
ICLR 2024	10	19.0	0.49	81	17.6	0.45	91

Table 14: Detailed statistics for LitSearch.