Evaluating and Improving LitLLMs with Deep Research

Gaurav Sahu[1,2,3]*, **Shubham Agarwal**[1,2,3]*, **Abhay Puri**[1], **Issam H. Laradji**[1,4]

Krishnamurthy Dj Dvijotham[1], **Jason Stanley**[1]

Laurent Charlin[2,3,5], **Christopher Pal**[1,2,5,6]
ServiceNow Research[1], Mila - Quebec AI Institute[2], HEC Montreal[3]
University of British Columbia[4], Canada CIFAR AI Chair[5], Polytechnique Montreal[6]
Correspondance: gaurav.sahu@mila.quebec

Abstract

Literature reviews are an essential component of scientific research, but they remain time-intensive and challenging to write, especially due to the recent influx of research papers. This paper explores the zero-shot abilities of recent Large Language Models (LLMs) in assisting with the writing of literature reviews based on an abstract. We decompose the task into two components: (1) Retrieving related works given a query abstract and (2) Writing a literature review based on the retrieved results. We analyze how effective LLMs are for both components. For retrieval, we introduce a novel two-step search strategy that first uses an LLM to extract meaningful keywords from the abstract of a paper and then retrieves potentially relevant papers by querying an external knowledge base. Additionally, we study a prompting-based re-ranking mechanism with attribution and show that re-ranking doubles the normalized recall compared to naive search methods while providing insights into the LLM's decision-making process. In the generation phase, we propose a two-step approach that first outlines a plan for the review and then executes steps in the plan to generate the actual review. To evaluate different LLM-based literature review methods, we create test sets from arXiv papers using a protocol designed for rolling use with newly released LLMs to avoid test set contamination in zero-shot evaluations. We release this evaluation protocol to promote additional research and development in this regard. Our empirical results suggest that LLMs show promising potential for writing literature reviews when the task is decomposed into smaller components of retrieval and planning. Particularly, our "Deep Research" retrieval variant improves coverage by over 5x compared to standard keyword search, addressing a key bottleneck in the pipeline. Further, we demonstrate that our planning-based approach achieves higher-quality reviews by minimizing hallucinated references in the generated review by 18-26% compared to existing simpler LLM-based generation methods.

1 Introduction

Writing a literature review—finding, citing, and contextualizing relevant prior work—is a fundamental requirement of scientific writing. It is a complex task that can be broken down into two broad sub-tasks: retrieving relevant papers and generating a related work section. This challenge is amplified in fields like machine learning, where thousands of

[*] Equal Contribution

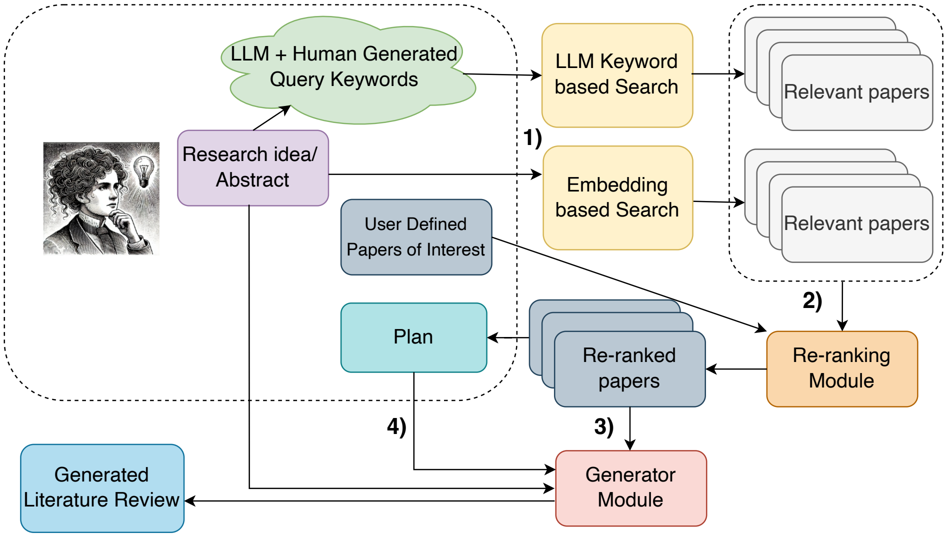


Figure 1: A schematic diagram of our framework, where: 1) Relevant prior work is retrieved using keyword and embedding-based search. 2) LLMs re-rank results to find the most relevant prior work. 3) Based on these papers and the user abstract or idea summary, an LLM generates a literature review, 4) optionally controlled by a sentence plan.

papers appear monthly on arXiv.[1] We explore the utility of large language models (LLMs), combined with retrieval mechanisms, to assist in this process.

Specifically, we investigate using an LLM to generate a paper's related work section based on its abstract (a concise summary of its key contributions). This approach provides valuable early-stage insights for authors, with the capacity to seamlessly incorporate the full manuscript as it develops. Our framework (Figure 1) decomposes the task into four key stages: (1) an LLM generates keywords from the abstract to query search engines (e.g., Google, Semantic Scholar, OpenAlex); (2) the LLM re-ranks retrieved papers for relevance, with an attribution mechanism; (3) the top-k papers are selected; and (4) an LLM generates the final text, optionally guided by a structural "plan" that can be user-provided or model-generated.

The main contributions of this work are: **(1)** We present a data collection protocol using a rolling window of recent arXiv papers to avoid test-set contamination when evaluating new LLMs. **(2)** We propose a novel, decomposable pipeline for interactive literature review writing, separating retrieval from generation. This modularity facilitates controlled studies and human-in-the-loop assistance. **(3)** We introduce retrieval innovations including LLM-generated keywords and embedding-based search. To address low initial coverage, we propose a novel **Deep Research** variant: an agentic pipeline that performs multi-stage analysis of candidates to significantly improve retrieval, and an attribution-based debate-ranking method to enhance re-ranking transparency. **(4)** For generation, we propose a plan-based retrieval-augmented approach that gives users greater control and, as our experiments show, substantially improves quality by reducing hallucinations by 18-26% and improving ROUGE scores.

2 Related Work

We decompose the literature review task into retrieval and generation. We now discuss prior work pertinent to both.

[1] E.g. over 4,000 ML papers were submitted to arXiv in October 2024: https://arxiv.org/list/cs.LG/2024-10

Search type	RollingEval-Aug (%)	RollingEval-Dec (%)
arXiv API (Single query)	0.65	1.41
SERP API - Google Search (Single query)	1.23	4.34
Semantic Scholar API (Single query)	**3.93**	**4.76**
arXiv API (Multiple queries)	2.75	1.92
SERP API - Google Search (Multiple queries)	6.80	5.04
Semantic Scholar API (Multiple queries)	6.07	5.09
OpenAlex API (Multiple queries)	5.82	4.87
OpenAlex API (Optimized multiple queries)	18.52	18.68
OpenAlex API (w/ Deep Research)	**36.86**	**37.75**
SPECTER2	8.30	6.80
Semantic Scholar API (Multiple queries) + SPECTER2	9.80	8.20
OpenAlex API (Optimized multiple queries) + SPECTER2	19.73	19.01
OpenAlex API (w/ Deep Research) + SPECTER2	**32.23**	**30.98**

Table 1: Coverage of ground-truth citations for various retrieval strategies. Our Deep Research variant significantly outperforms standard keyword-based methods. Notably, adding SPECTER2 embeddings-based re-ranking on top of Deep Research output hurts the overall coverage.

2.1 Ranking and Retrieval

Traditional information retrieval methods (e.g., TF-IDF, BM25) have been largely superseded by dense vector approaches like Sentence-BERT (Reimers & Gurevych, 2019). More recently, LLMs have been used for re-ranking, where a list of candidate passages is provided as input for the model to re-order based on relevance (Sun et al., 2023; Ma et al., 2023; Pradeep et al., 2023a;b). While effective, these methods often lack interpretability. To address this, some work has focused on attribution, verifying generated statements against cited sources (Yue et al., 2023; Cohen-Wang et al., 2024). These often rely on complex gradient-based or perturbation-based techniques (Sundararajan et al., 2017; Ribeiro et al., 2016), which can be difficult to scale. In contrast, our work proposes a straightforward, prompting-based attribution approach that is scalable and requires only a single pass through the model.

2.2 Literature Review Generation

Our work builds on the foundation of multi-document summarization, particularly the Multi-XScience dataset (Lu et al., 2020). While prior work has explored generating summaries or answers from scientific texts (Pilault et al., 2020; Gao et al., 2023) and using chain-of-thought prompting (Kojima et al., 2022; Zhou et al., 2022), our approach is novel in its use of an explicit, intermediate *plan* to structure the generated text. This parallels traditional NLG pipelines (Reiter & Dale, 1997; Puduppully & Lapata, 2021) but applies it to modern, end-to-end LLMs. Our method introduces controllability, allowing a human user to provide or edit a plan, thereby iteratively refining the output. While models like Galactica (Taylor et al., 2022) have demonstrated strong scientific reasoning, they were not designed for plan-based, citation-grounded generation and suffered from hallucination, a problem our work directly addresses.

3 Retrieval of Related Work

We now detail our methodology for retrieving and ranking related work, evaluated on datasets constructed from recent arXiv papers.

3.1 Dataset Construction and Retrieval Method

To avoid test set contamination from LLM training data, we create two datasets, **RollingEval-Aug** and **RollingEval-Dec**, using papers published on arXiv in August and December 2023, respectively. For each query paper, we use an LLM to generate multiple keyword queries and retrieve a candidate pool of 100 papers published strictly before the query paper from

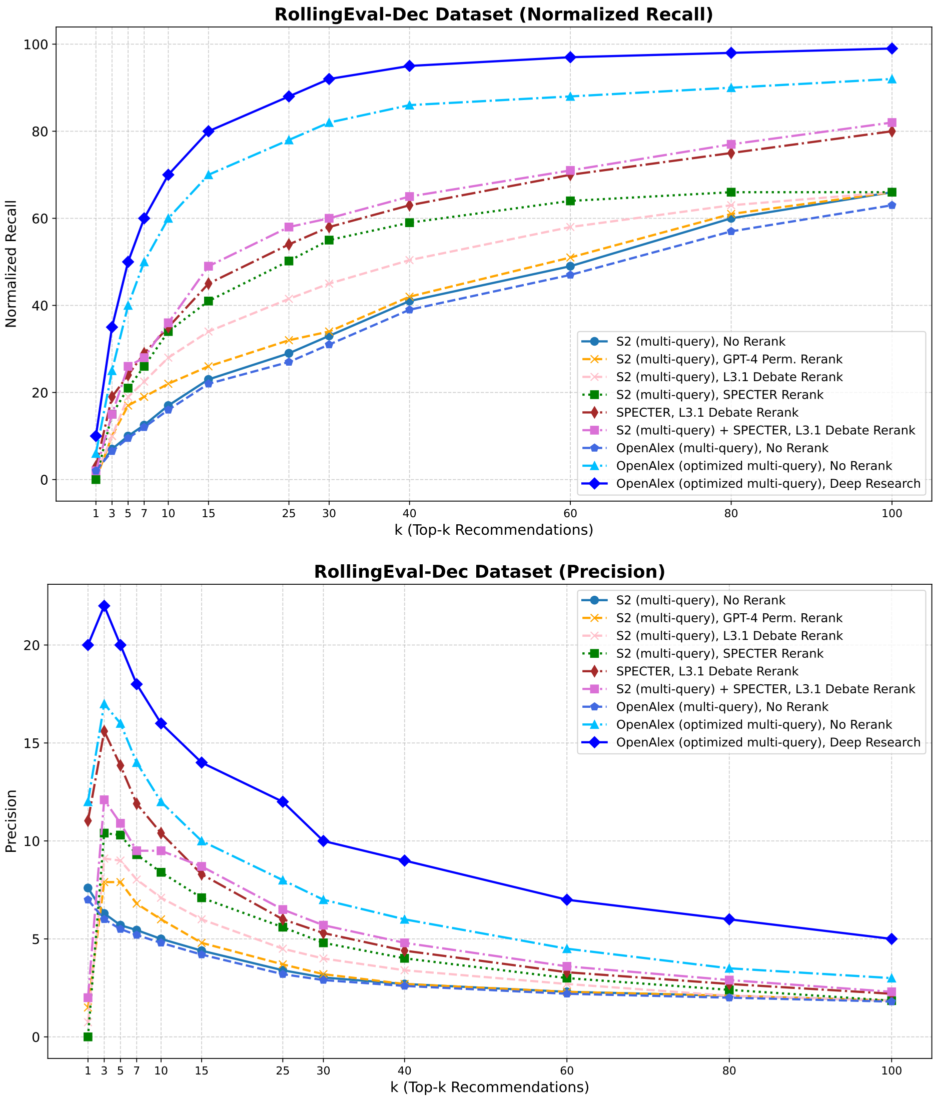


Figure 2: Effect of re-ranking strategies on the RollingEval-Dec dataset ($n = 500, k = 100$). Deep Research generally outperforms other LLM and embedding-based methods.

academic search engines (Semantic Scholar, Google, OpenAlex). We evaluate several query generation strategies, as reflected in Table 1: **(1) Single Query:** A baseline where the LLM generates one single, comprehensive search query from the abstract. **(2) Multiple Queries:** An approach where the LLM generates several (typically three) alternative queries from the same abstract. The final candidate pool is formed by aggregating an equal number of top results from each query's search results to diversify the retrieved set. **(3) Optimized Multiple Queries:** A more advanced strategy where the LLM is prompted to generate a diverse set of queries that target distinct aspects of the abstract, such as the problem domain, the proposed methodology, and the key contribution. We add explicit instructions to the LLM like creating "mutually-exclusive" queries. This aims to increase retrieval breadth by exploring different facets of the research.

Our retrieval pipeline (Algorithm 1) combines two main strategies: (1) LLM-generated keyword search, and (2) embedding-based search using SPECTER2 (Singh et al., 2022). However, as shown in Table 1, these standard methods exhibit low coverage, retrieving less than 7% of ground-truth citations.

Algorithm 1 Retrieval algorithm

```
Require: Input abstract $a$
keywords = LLMKeywords($a$) {Generate keywords from the abstract using an LLM}
candidate_papers = SearchEngine(keywords) {Query a search engine}
reranked_papers = LLMRerank(candidate_papers, $a$) {LLM-based re-ranking}
return reranked_papers
```

3.2 Deep Research: Citation Graph Expansion

To address the low coverage of standard keyword search, we introduce a **Deep Research** variant, powered by Qwen3-14B model (Yang et al., 2025). This method mimics human research behavior by iteratively exploring the citation graph of relevant papers. The algorithm begins with the initial set of seed papers retrieved via keyword search. It then enters a loop: (1) For each paper in the current set, it queries the OpenAlex API to fetch its bibliography (i.e., the papers it cites). If the OpenAlex API does not contain the references for a paper, we use an LLM to extract the references from the PDF itself. (2) This new list of candidate papers is then filtered for relevance against the *original query abstract*. For this, we employ our **Debate Ranking with Attribution** agent (described in Section 3.3), which assesses each candidate's abstract. (3) Candidates that meet a relevance threshold are pruned, and the expansion continues from these newly added nodes. The process terminates when a maximum number of 200 papers is collected or a maximum search depth of 4 is reached. This recursive expansion allows the system to uncover highly relevant papers that are not discoverable through direct keyword matches with the initial abstract, leading to the substantial coverage improvements shown in Table 1. We outline the Deep Research algorithm in Algorithm 2.

Algorithm 2 Deep Research: Citation Graph Expansion

```
Require: Query Abstract $a_q$, Initial Seed Papers $P_{seed}$, Max Depth $D_{max}$, Max Papers $N_{max}$
$Q \leftarrow$ new Queue() {Queue for papers to visit}
$P_{final} \leftarrow P_{seed}$ {The final set of relevant papers}
$V_{ids} \leftarrow \{\text{id}(p) \text{ for } p \in P_{seed}\}$ {Set of visited paper IDs}
for each $p \in P_{seed}$ do
$Q$.enqueue($(p, 0)$)
end for
while $Q$ is not empty and $|P_{final}| < N_{max}$ do
  $(p_{curr}, d_{curr}) \leftarrow Q$.dequeue()
  if $d_{curr} \geq D_{max}$ then
    continue
  end if
  $P_{cand} \leftarrow$ GetReferences($p_{curr}$) {Expand by getting citations using API or an LLM}
  $P_{relevant} \leftarrow$ FilterByRelevance($P_{cand}, a_q$) {Use Debate Ranking to check relevance}
  for each $p_{new} \in P_{relevant}$ do
    if id($p_{new}$) $\notin V_{ids}$ then
      $V_{ids}$.add(id($p_{new}$))
      $P_{final}$.add($p_{new}$)
      $Q$.enqueue($(p_{new}, d_{curr} + 1)$)
      if $|P_{final}| \geq N_{max}$ then
        break
      end if
    end if
  end for
end while
return $P_{final}$
```

3.3 Re-ranking and Attribution

Once a candidate pool is retrieved, we explore three re-ranking strategies: (a) **Instructional permutation generation**, where an LLM directly outputs a ranked permutation of candidates (Sun et al., 2023); (b) **SPECTER2 embeddings**, where candidates are ranked by cosine similarity to the query abstract's embedding; and (c) our proposed **Debate Ranking with Attribution**. In this approach, an LLM generates arguments for and against citing each candidate and provides a final relevance score. Crucially, we require the LLM to attribute its arguments to verbatim quotes from the candidate's abstract, re-prompting if verification fails. This enhances both reliability and interpretability.

3.4 Retrieval and Re-ranking Experiments

We evaluate retrieval quality using precision and normalized recall, where normalized recall measures the fraction of *retrieved* ground-truth papers that appear in the top-k results. This metric assesses ranking quality independently of initial retrieval coverage. Formally:

$$\text{Normalized Recall@k} = \frac{|\text{Retrieved@k} \cap \text{GT}|}{|\text{Retrieved} \cap \text{GT}|}; \quad \text{Precision@k} = \frac{|\text{Retrieved@k} \cap \text{GT}|}{k} \tag{1}$$

Figure 2 shows that for re-ranking, SPECTER2 embeddings generally outperform LLM-based permutation, while our Debate Ranking strategy also shows strong performance. However, LLM-based re-ranking can be brittle; Table 2 shows that GPT-4 produces incomplete or flawed lists over 40% of the time.

In an ablation study (Figure 3), we found that removing the attribution verification step from our Debate Ranking strategy significantly degrades performance (p < .001), confirming that grounding the LLM's reasoning in textual evidence is crucial for ranking accuracy.

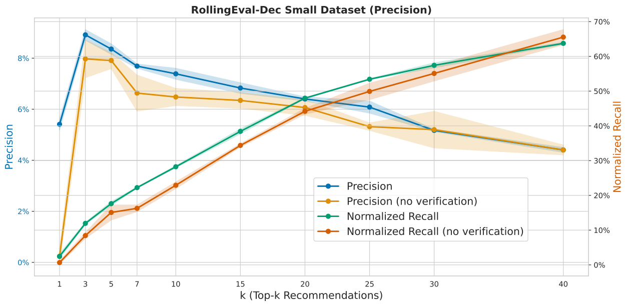


Figure 3: Ablation study showing the positive effect of attribution verification in our debate ranking strategy ($n = 100, k = 40$). Removing verification significantly lowers precision and normalized recall.

Ranker Prediction Type	RollingEval-Aug (%)	RollingEval-Dec (%)
Complete Ranked list	55.1	59.7
Incomplete list	41.5	40.2
Repeated Value	3.3	0.1

Table 2: Error modes of GPT-4 permutation-based re-ranking, showing high rates of incomplete or flawed outputs.

4 Literature Review Generation

After retrieving relevant papers, we focus on the generation task. To mitigate hallucinations common in LLMs (Athaluri et al., 2023), we use a Retrieval Augmented Generation (RAG) approach, conditioning the LLM only on the abstracts of the retrieved papers.

Plan-Based Generation. To improve quality and controllability, we introduce a **writing plan**, an intermediate, human-readable representation that specifies the structure of the output. We explore several strategies: (1) **Plan-based**: The LLM is given a user-provided plan specifying sentence count and citation placement (e.g., ”Generate 5 sentences. Cite paper X at line 2.”). This ”teacher-forcing” approach strongly guides the output. (2) **Prompted Plan**: The LLM first generates its own plan, then follows it. This allows for iterative refinement by a human user. We also experiment with simpler **Per-cite** and **Sentence-by-sentence** generation strategies.

5 Generation Experiments and Results

We evaluate our generation strategies on the Multi-XScience dataset and our new RollingEval-Aug dataset, using ROUGE, BERTScore, and Llama-3-Eval as metrics. We compare against extractive baselines (LexRank), finetuned models (Pointer-Generator), and various zero-shot LLMs. To provide a more tangible illustration of our method's impact, Table 3 presents a qualitative comparison for a single example.

Ground-Truth Snippet	**GPT-4 (0-shot)**	**GPT-4 (Plan-based)**
Multi-document summarization is a challenging task for which there exists little large-scale datasets. (Lu et al., 2020) We present a method to produce abstractive summaries of long documents... We show that this extractive step significantly improves summarization results. (Pilault et al., 2020)	The field of multi-document summarization has been a challenging one due to the lack of large-scale datasets. (Misses cite 1) A significant attempt to address this gap is the Multi-XScience dataset... (Misses cite 2) This study also ensures validity by creating a new test corpus from recent arXiv papers. (Hallucinates detail not in sources.)	The concept of generating the related-work section from abstracts and referenced articles is not entirely new, as evidenced by the Multi-XScience dataset. (Lu et al., 2020) The approach of extending abstractive summarization to lengthy documents via an extractive step aligns with the proposed methodology. (Pilault et al., 2020)

Table 3: Qualitative comparison of generated text. The plan-based output more closely follows the structure and citation pattern of the ground-truth, while the 0-shot version misses citations and hallucinates details not present in the provided source abstracts.

Our quantitative results (Tables 4 and 5) show that:

- Our teacher-forced **Plan-based framework consistently improves scores** over the zero-shot baseline for all models. GPT-4 with a plan achieves the highest scores across all metrics.
- Open-source models like Llama 2 70B see a significant uplift from plan-based generation, achieving performance comparable to zero-shot GPT-3.5.
- GPT-4 is the most **controllable**, following the exact plan structure over 60% of the time (Table 8 in Appendix).
- Plan-based prompting dramatically improves **citation coverage**, with GPT-4 (Plan) successfully including 98.5% of specified citations.

Human Evaluation. We conducted a study with 6 expert annotators who ranked outputs from zero-shot and plan-based models. As shown in Figure 4, annotators significantly preferred plan-based generations for both Llama 2 and GPT-4. Crucially, plan-based prompting reduced instances of hallucinated content by 29.6% to 11.6% for GPT-4. GPT-4's superior ability to adhere to a generation plan (Table 8) directly correlates with its higher preference ratings in our human evaluation, suggesting that controllability is a key factor in perceived quality. This confirms that providing a structural scaffold (the plan) improves measurable quality by constraining the LLM's output space, forcing it to focus on fulfilling factual and citation requirements rather than narrative invention.

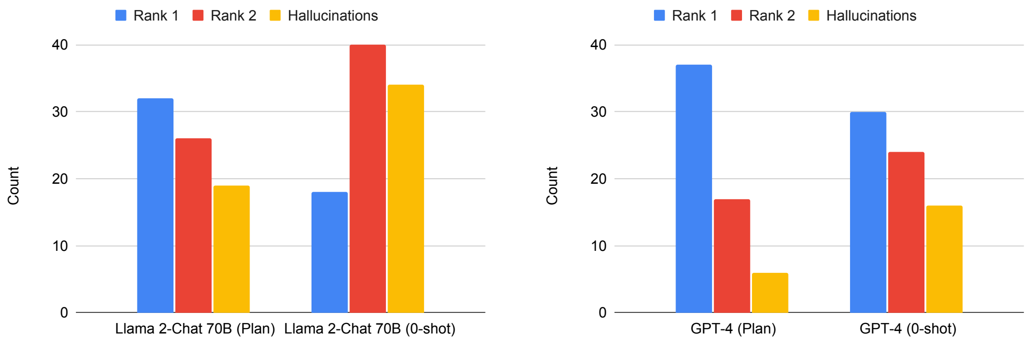


Figure 4: Human evaluation results. Annotators preferred plan-based outputs and identified significantly fewer hallucinations compared to zero-shot generations.

Model Class	**Model**	**R-1 ↑**	**R-2 ↑**	**R-L ↑**
Closed-source 0-shot	GPT-3.5-turbo	29.69	7.32	14.56
	GPT-4	33.21	7.60	15.79
Plan-based	Llama 2-Chat 70B	34.65	8.37	17.08
	GPT-3.5-turbo	35.04	8.42	17.13
	GPT-4	**37.19**	**8.85**	**18.77**

Table 4: Abridged zero-shot and plan-based results on Multi-XScience. Plan-based prompting consistently outperforms zero-shot baselines. Full results in Table 6 in Appendix.

6 Conclusions

This work establishes and evaluates a pipeline for LLM-assisted literature review. To ensure valid evaluation, we introduce a rolling data collection protocol using recent arXiv papers to prevent test set contamination. Our experiments show that decomposing the task into retrieval and generation makes it tractable for modern LLMs. Our Deep Research retrieval variant significantly boosts coverage, a critical first step, and our attribution-based debate ranking improves re-ranking. Most importantly, our plan-based generation approach dramatically reduces hallucinations and improves output quality and controllability.

The landscape of AI-assisted research is rapidly evolving, with new tools demonstrating remarkable improvements. Our work highlights clear remaining challenges: achieving comprehensive coverage of retrieval remains difficult, and hallucinations, though reduced, are not eliminated. However, our modular, plan-driven framework represents a significant step forward.

Limitations and Future Work. The primary limitation is the persistent low coverage of retrieval, even with our best methods. We evaluated components independently due to this, but an end-to-end system requires further improvement in retrieval. Our current methods rely on abstracts; while this mirrors early-stage research, incorporating full-text analysis will be crucial. Our Deep Research variant supports full-text but at the cost of significantly increased runtime. We would also like to point out that LLama-3-70B and Qwen3-14B may have seem some of the papers in our dataset, even if they were not pretrained exactly for the task of judging relevance; however, we confirmed their performance persists on recent papers as well. Future work should explore more advanced embedding and search strategies and continue to refine plan-based generation to further enhance factual accuracy and provide researchers with powerful, reliable tools.

References

Sai Anirudh Athaluri, Sandeep Varma Manthena, V S R Krishna Manoj Kesapragada, Vineel Yarlagadda, Tirth Dave, and Rama Tulasi Siri Duddumpudi. Exploring the boundaries of

Model	**Coverage ↑**	**Avg. words**
Llama 2-Chat 70B (0-shot)	59.31%	284.65
Llama 2-Chat 70B (Plan)	82.62%	191.45
GPT-4 (0-shot)	91.34%	215.15
GPT-4 (Plan)	98.52%	125.10

Table 5: Citation coverage on Multi-XScience. Plan-based prompting significantly improves coverage and produces more concise text.

reality: Investigating the phenomenon of artificial intelligence hallucination in scientific writing through chatgpt references. *Cureus*, 15, 2023. URL https://api.semanticscholar.org/CorpusID:258097853.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents, 2023.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4766–4777, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.428. URL https://aclanthology.org/2020.findings-emnlp.428.

Hong Chen, Hiroya Takamura, and Hideki Nakayama. SciXGen: A scientific paper dataset for context-aware text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1483–1492, Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.128. URL https://aclanthology.org/2021.findings-emnlp.128.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. Capturing relations between scientific papers: An abstractive model for related work section generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6068–6077, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.473. URL https://aclanthology.org/2021.acl-long.473.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. Contextcite: Attributing model generation to context, 2024. URL https://arxiv.org/abs/2409.00729.

Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. Citebench: A benchmark for scientific citation text generation. *arXiv preprint arXiv:2212.09577*, 2022.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*, 2023.

M Waleed Kadous. Llama 2 is about as factually accurate as gpt-4 for summaries and is 30x cheaper, Aug 2023. URL https://www.anyscale.com/blog/llama-2-is-about-as-factually-accurate-as-gpt-4-for-summaries-and-is-30x-cheaper.

Rodney Kinney, Chloe Anastasiades, Russell Authur, Iz Beltagy, Jonathan Bragg, Alexandra Buraczynski, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Arman Cohan, et al. The semantic scholar open data platform. *arXiv preprint arXiv:2301.10140*, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Xiangci Li, Biswadip Mandal, and Jessica Ouyang. CORWA: A citation-oriented related work annotation dataset. In *Proceedings of the 2022 Conference of the North American Chapter* *of the Association for Computational Linguistics: Human Language Technologies*, pp. 5426–5440, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.397. URL https://aclanthology.org/2022.naacl-main.397.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen. Generating a structured summary of numerous academic papers: Dataset and method. *arXiv preprint arXiv:2302.04580*, 2023.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4969–4983, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.447. URL https://aclanthology.org/2020.acl-main.447.

Patrice Lopez. GROBID, February 2023. URL https://github.com/kermitt2/grobid. original-date: 2012-09-13T15:48:54Z.

Yao Lu, Yue Dong, and Laurent Charlin. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8068–8074. Association for Computational Linguistics, November 2020. doi: 10.18653/v1/2020.emnlp-main.648. URL https://aclanthology.org/2020.emnlp-main.648.

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

Laura Nguyen, Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. LoRaLay: A multilingual and multimodal dataset for long range and layout-aware summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 636–651, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.46. URL https://aclanthology.org/2023.eacl-main.46.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS-W*, 2017. URL https://openreview.net/forum?id=BJJsrmfCZ.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9308–9319, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.748. URL https://aclanthology.org/2020.emnlp-main.748.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*, 2023a.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*, 2023b.

Jason Priem, Heather Piwowar, and Richard Orr. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *arXiv preprint arXiv:2205.01833*, 2022.

Ratish Puduppully and Mirella Lapata. Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, 9:510–527, 2021. URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00381/101876/Data-to-text-Generation-with-Macro-Planning.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144. ACM, 2016.

Tarek Saier and Michael Färber. unarXive: A Large Scholarly Data Set with Publications' Full-Text, Annotated In-Text Citations, and Links to Metadata. *Scientometrics*, 125(3): 3085–3108, December 2020. ISSN 1588-2861. doi: 10.1007/s11192-020-03382-z.

Tarek Saier, Johan Krause, and Michael Färber. unarXive 2022: All arXiv Publications Pre-Processed for NLP, Including Structured Full-Text and Citation Network. In *Proceedings of the 23rd ACM/IEEE Joint Conference on Digital Libraries*, JCDL '23, 2023.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. SciRepEval: A multi-format benchmark for scientific document representations. *arXiv preprint arXiv:2211.13308*, 2022.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*, pp. 243–246, 2015.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Pengjie Ren, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*, 2023.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. Automatic evaluation of attribution by large language models, 2023. URL https://arxiv.org/abs/2305.06311.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

Kun Zhu, Xiaocheng Feng, Xiachong Feng, Yingsheng Wu, and Bing Qin. Hierarchical catalogue generation for literature review: A benchmark. *arXiv preprint arXiv:2304.03512*, 2023.

Appendix

A Ethics Statement

The rapid advancements in LLMs and NLP technologies for scientific writing have led to the emergence of increasingly powerful systems such as OpenAI DeepResearch, AI Co-Scientist, and ScholarQA. These tools extend beyond earlier systems like Explainpaper and Writefull[2], which assist in paper comprehension and abstract generation, and Scite[3], which helps with citation discovery. As AI-powered tools become more deeply integrated into the scientific workflow, ethical considerations around their use continue to evolve. Many conferences, such as ICLR, have begun collecting statistics on authors' usage of LLMs for literature review generation and paraphrasing, and have issued guidelines on responsible usage.[4] While writing assistant technology could have great promise as an aide to scientists, we think their use should be disclosed to the reader. As such assistants become more powerful, they might be abused in certain contexts, for example, where students are supposed to create a literature review as a part of their learning process. The use of such tools might also be problematic as authors of scientific work should read the articles that they cite, and heavy reliance on such tools could lead to short-term gains at the cost of a deeper understanding of a subject over the longer term. Any commercially deployed systems authors use should also contain appropriate mechanisms to detect if words have been copied exactly from the source material and provide that content in a quoted style. Additionally, as newer tools like DeepResearch, AI Co-Scientist, and ScholarQA continue to improve, it is crucial to assess their long-term impact on scientific research. The use of these tools should complement, rather than replace, human expertise in literature analysis. Finally, the rolling evaluations we present here do not involve training LLMs on arXiv papers. This mitigates concerns regarding the copyright status of arXiv papers and their use for LLM training.

Model Class	**Model**	**ROUGE1 ↑**	**ROUGE2 ↑**	**ROUGEL ↑**
Extractive	One line baseline	26.869	4.469	14.386
	LexRank	30.916	5.966	15.916
	TextRank	31.439	5.817	16.398
Abstractive Finetuned	Hiersum	29.861	5.029	16.429
	Pointer-Generator	33.947	6.754	18.203
	PRIMER	26.926	5.024	14.131
Abstractive 0-shot	Long T5	19.515	3.361	12.201
	Flan T5	21.959	3.992	12.778
	Galactica-1.3B	18.461	4.562	9.894
	Falcon-180B	22.876	2.818	12.087
Open-source 0-shot	Llama 2-Chat 7B (No plan)	24.636	5.189	13.133
	Llama 2-Chat 13B (No plan)	26.719	5.958	13.635
	Llama 2-Chat 70B (No plan)	28.866	6.919	14.407
	LLama-3.1-70B (No Plan)	33.289	8.050	15.898
Closed-source 2-stage	GPT-3.5-turbo (Per cite) 1st stage	26.483	6.311	13.718
	GPT-3.5-turbo (Per cite) 2nd stage	24.359	5.594	12.859
	GPT-3.5-turbo (Sentence by sentence)	31.654	6.442	15.577
Closed-source 0-shot	GPT-3.5-turbo (No plan)	29.696	7.325	14.562
	GPT-4 (No plan)	33.213	7.609	15.798
Plan	Llama 2-Chat 70B (Prompted plan)	30.389	7.221	14.911
	GPT-3.5-turbo (Prompted plan)	32.187	7.788	15.398
	GPT-4 (Prompted plan)	34.819	7.892	16.634
	Llama 2-Chat 70B (Plan)	34.654	8.371	17.089
	GPT-3.5-turbo (Plan)	35.042	8.423	17.136
	GPT-4 (Plan)	37.198	8.859	18.772
	Llama-3.1-70B (Plan)	35.575	9.406	18.772

Table 6: Complete zero-shot results for different models on the Multi-XScience dataset.

[2] https://www.explainpaper.com/, https://x.writefull.com/

[3] https://scite.ai/

[4] ICLR'24 Large Language Models guidelines https://iclr.cc/Conferences/2024/CallForPapers

Model	ROUGE1 ↑	ROUGE2 ↑	ROUGEL↑	BERTScore↑	Llama-3-Eval↑
CodeLlama 34B-Instruct	22.608	5.032	12.553	82.418	66.898
CodeLlama 34B-Instruct (Plan)	27.369	5.829	14.705	83.386	67.362
Llama 2-Chat 7B	23.276	5.104	12.583	82.841	68.689
Llama 2-Chat 13B	23.998	5.472	12.923	82.855	69.237
Llama 2-Chat 70B	23.769	5.619	12.745	82.943	70.980
GPT-3.5-turbo (0-shot)	25.112	6.118	13.171	83.352	72.434
GPT-4 (0-shot)	29.289	6.479	15.048	84.208	72.951
Llama 2-Chat 70B (Plan)	30.919	7.079	15.991	84.392	71.354
GPT-3.5-turbo (Plan)	30.192	7.028	15.551	84.203	72.729
GPT-4 (Plan)	33.044	7.352	17.624	85.151	75.240

Table 7: Complete zero-shot results on the proposed RollingEval-Aug dataset.

Model	Multi-XScience			RollingEval-Aug		
	% ↑	Mean ↓	Max ↓	% ↑	Mean ↓	Max ↓
GPT-3.5-turbo (Plan)	4.73	3.65	17	3	4.7	16
Llama 2-Chat 70B (Plan)	19.04	2.66	22	17.4	2.72	18
GPT-4 (Plan)	60.7	−0.01	8	70.6	0.16	5

Table 8: Analysis of how closely different models follow the provided generation plan. We show the percentage of responses with the same number of lines as the plan, and the mean/max difference in lines. GPT-4 follows the plan most faithfully.

B Datasets

While there are datasets available for different tasks in academic literature (see Table 9), we use the Multi-XScience dataset (Lu et al., 2020) for our experiments. Recent work (Chen et al., 2021b; Funkquist et al., 2022) also focuses on related work generation and provides a similar dataset. As part of this work, we release two corpora: 1. We extend the Multi-XScience corpus to include the full text of research papers, and 2. We create a new test corpus, RollingEval-Aug, consisting of recent (August 2023) arXiv papers (with full content).

Dataset	Task
BigSurvey-MDS (Liu et al., 2023)	Survey Introduction
HiCaD (Zhu et al., 2023)	Survey Catalogue
SciXGen (Chen et al., 2021a)	Context-aware text generation
CORWA (Li et al., 2022)	Citation Span Generation
TLDR (Cachola et al., 2020)	TLDR generation
Multi-XScience Lu et al. (2020)	Related Work Generation

Table 9: Different tasks for academic literature

Multi-XScience full text We create these datasets based on the latest release (2023-09-12) of the S2ORC corpus[5] (Lo et al., 2020) available at the Semantic Scholar Open Data Platform (Kinney et al., 2023). The S2 Platform provides access to multiple datasets, including paper metadata, authors, S2AG (Semantic Scholar Academic Graph), paper embeddings, etc. While the 'Papers' dataset consists of 200M+ metadata records, S2ORC consists of 11+M full-text publicly available records with annotations chunked into 30 files (~215G compressed json) where research documents are linked with arXiv and Microsoft Academic Graph (MAG) (Sinha et al., 2015) IDs, when available. This corpus provides full text of the research papers (parsed using a complex pipeline consisting of multiple LaTeX and PDF parsers such as GROBID (Lopez, 2023) and in-house parsers.[6]). The full text is also aligned with annotation spans (character level on the full text), which identify sections, paragraphs, and other useful information. It also includes spans for citation mentions and the matching semantic corpus-based ID for bibliographical entries, making it easier to align

[5] Dataset available at `http://api.semanticscholar.org/datasets/v1/`

[6] `https://github.com/allenai/papermage`

with references compared to other academic datasets such as LoRaLay (Nguyen et al., 2023), UnarXive (Saier & Färber, 2020; Saier et al., 2023), etc. or relying on citation graphs like OpenAlex (Priem et al., 2022), next-generation PDF parsers (Blecher et al., 2023) or other HTML webpages.[7] For the Multi-XScience, we obtain the full text of papers for 85% of records from the S2ORC data using the span annotations from the corpus aligned with citation information.

RollingEval datasets Llama 2 was publicly released on 18th July 2023 and GPT-4 on 14 March 2023. Both provide limited information about their training corpus, and academic texts in the Multi-XScience may or may not have been part of their training data. To avoid overlap with the training data of these LLMs, we process a new dataset using papers posted after their release date. To do so, we first filter the papers published in August 2023 from S2ORC that contain an arXiv ID, resulting in ~15k papers. S2ORC does not provide the publication date of the papers directly, so we use regex '2308' on the arXiv ID to extract papers posted in 08'23. We then use section annotations to get the section names and match using synonyms ('Related Work, Literature Review, Background') to extract section spans. We take the rest of the text as conditioning context except the related work section which results in ~4.7k documents. Using the citation annotations, we extract the full text of cited papers from the S2ORC corpus again using corpus ID. Similar to Multi-XScience, we use paragraph annotations to create a dataset for the latest papers (~6.2k rows). We create a subset of 1,000 examples (RollingEval-Aug) where we have the content of all the cited papers. The average length of a related work summary is 95 words, while the average length of abstracts is 195. On average, we have 2 citations per example, which makes the dataset comparable to the original Multi-XScience dataset.

C Other Generation Experiments

Llama 2 fine-tuning In parallel, we also fine-tune Llama 2 models on the train set with the original shorter context, but they are very sensitive to hyperparameter configuration. When we instruct-finetune Llama 2 7B, it initially produces code. We find a slight improvement when fine-tuning the Llama 2 7B model for 30k steps with an LR of 5e-6 over 0-shot model (see Table 10), but it quickly overfits as we increase the LR or the number of steps. We leave hyperparameter optimization, fine-tuning larger models with RoPE scaling and plan-based generation for future work.

Model	**ROUGE1 ↑**	**ROUGE2 ↑**	**ROUGEL ↑**
Llama 2-Chat 7B - 0-shot	26.719	5.958	13.635
Llama 2-Chat 7B - 10k steps (LR 5e-6)	24.789	5.986	12.708
Llama 2-Chat 7B - 30k steps (LR 5e-6)	27.795	**6.601**	14.409
Llama 2-Chat 7B - 60k steps (LR 1e-5)	22.555	5.511	11.749

Table 10: Results after fine-tuning Llama 2-Chat 7B on Multi-XScience dataset

Longer context While Llama 2 can ingest 4096 tokens, recent studies have found that it uses 19% more tokens (Kadous, 2023) than GPT-3.5 or GPT-4 (2048 and 4096 tokens respectively), implying that the effective number of words in Llama 2 is considerably lower than GPT-4 and only a bit higher than GPT-3.5. We experiment with the popular RoPE scaling (Su et al., 2021) in 0-shot Llama models to increase the context length (4k–6k). This permits using the full text of the papers instead of just their abstracts. Results in Table 11 show that directly using RoPE scaling on 0-shot models produces gibberish results. Instead, one needs to fine-tune the model with the longer context. In fact, a plan-based-longer-context CodeLlama (initialized from Llama 2 and trained with a 16k token context through RoPE scaling) improves on ROUGE1/L, but achieves comparable results as a shorter-context

[7] https://ar5iv.labs.arxiv.org/ and https://www.arxiv-vanity.com/

plan-based CodeLlama on ROUGE2. For reporting results with longer context Llama 2 using RoPE scaling (Su et al., 2021), we use HuggingFace Text Generation Inference.[8]

Model	ROUGE1 ↑	ROUGE2 ↑	ROUGEL ↑
Llama 2-Chat 7B (4000 words)	17.844	1.835	10.149
Llama 2-Chat 7B (5000 words)	17.254	1.736	9.986
Llama 2-Chat 7B (6000 words)	17.179	1.647	9.897
Llama 2-Chat 13B (4000 words)	20.071	3.516	10.916
Llama 2-Chat 13B (5000 words)	20.722	3.714	11.13
Llama 2-Chat 13B (6000 words)	17.179	1.647	9.897
Llama 2-Chat 70 (4000 words)	19.916	2.741	10.456
Llama 2-Chat 70B (5000 words)	19.675	2.605	10.48
Llama 2-Chat 70B (6000 words)	20.437	2.976	10.756
CodeLlama 34B-Instruct (4000 words)	27.425	5.815	14.744

Table 11: Zero-shot results using RoPE scaling for larger context on RollingEval-Aug dataset. Here we report the max number of words used for truncation instead of the tokens.

Code LLMs We evaluate the performance of code-generating LLMs to write related-work sections requiring more formal and structured language. Since Code LLMs are pre-trained on text they might offer the best of both worlds. However, we observe that for our task, the models produce bibtex and Python code with relevant comments as part of the generated outputs. As shown in Table 12, CodeLlama (34B Instruct) is good at following instructions and at generating natural language (ROUGE2 of 5.8 and 5.02 on Multi-XScience and RollingEval-Aug dataset). With a plan, CodeLlama even surpasses vanilla 0-shot Llama 2 70B (Table 7).

Model	ROUGE1 ↑	ROUGE2 ↑	ROUGEL ↑
StarCoder	12.485	1.104	6.532
Lemur-70B	15.172	2.136	7.411
CodeLlama 34B-Instruct	25.482	5.814	13.573

Table 12: 0-shot results using code-based models on Multi-XScience dataset. CodeLlama performs reasonably well in generating natural language compared to the other code-based counterparts.

D More implementation details

D.1 Normalized Recall v/s Standard Recall: A Worked-out Example

Consider a query paper with the following statistics:

$$|\text{Ground Truth}| = n_{\text{gt}} = 84$$
$$|\text{Retrieved}| = 100$$
$$\text{Relevant Retrieved papers} = |\text{Retrieved} \cap \text{Ground Truth}| = c = 10$$
$$\text{Relevant papers in top-40} = n_{\text{rel}} = 4$$

Using these values, we compute the metrics at $k = 40$:

$$\text{Precision@40} = \frac{n_{\text{rel}}}{40} = \frac{4}{40} = 0.010; \quad \text{Normalized Recall@40} = \frac{n_{\text{rel}}}{c} = \frac{1}{10} = 0.100; \quad \text{Recall} = \frac{n_{rel}}{n_{gt}} = \frac{4}{84} = 0.048$$

This example illustrates how Normalized Recall@k differs from standard recall. Instead of being limited by the total number of ground truth citations, it evaluates how well the

[8] https://github.com/huggingface/text-generation-inference

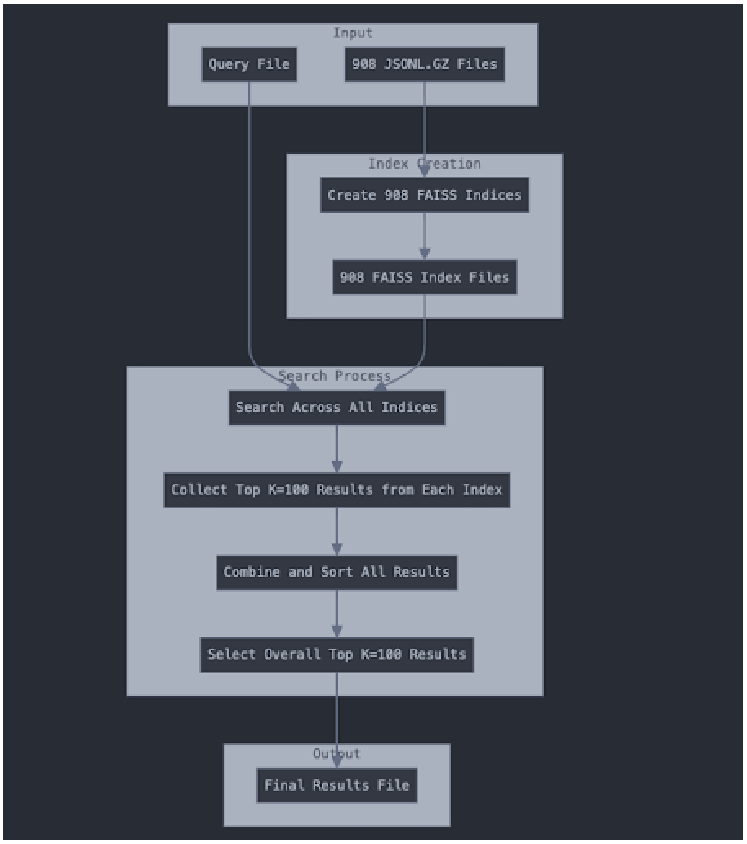


Figure 5: Pipeline for creating FAISS indexes for 150M SPECTER2 embeddings.

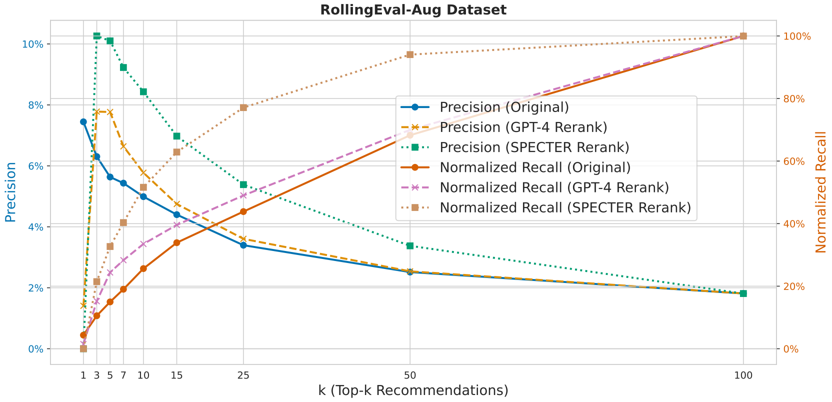


Figure 6: Effect of re-ranking strategies on the RollingEval-Aug dataset. We evaluate the Precision and Normalized Recall of the re-ranked results contrasting LLM-based based re-ranking with embedding-based ranker. We find a similar pattern as the RollingEval-Aug dataset.

method ranks the retrievable relevant papers. In this case, despite a low precision, the normalized recall is relatively high, indicating that the method effectively ranks the relevant papers it does retrieve.

D.2 Generation Implementation

We use HuggingFace Transformers and PyTorch (Paszke et al., 2017) for our experiments.[9] We calculate ROUGE scores (Lin, 2004) using the Huggingface (Wolf et al., 2019) evaluate library[10]. To split sentences, we use 'en_core_web_sm' model from SpaCy[11]. Additionally, we use Anyscale endpoints[12] to generate 0-shot Llama 2 results and OpenAI API[13] to generate results for GPT-3.5-turbo and GPT-4.

D.3 Demo implementation

We build our system using the ReactJS framework, which provides a nice interface to build system demos quickly and efficiently.

We query the Semantic Scholar or OpenAlex API to search for the relevant papers. Specifically, we use the Academic Graph[14] and Recommendations[15] API endpoint from Semantic-Scholar. We use OpenAI API to generate results for LLM using GPT-3.5-turbo and GPT-4 models. At the same time, our modular pipeline allows using any LLM (proprietary or open-sourced) for different components. We also allow the end-user to sort the retrieved papers by relevance (default S2 results), citation count, or year. More details about the demo system can be found in our system paper.

D.4 SPECTER Implementation

We build an index of 150M SPECTER2 embeddings that we can use as an alternative to both a search engine and a prompting-based ranking module. Figure 5 shows our pipeline for creating the index. Specifically, the SPECTER2 database comes with 908 json.gz files containing compressed embeddings. For each json.gz file, we construct a FAISS index that we can query for the nearest neighbors of a given query embedding. We perform index construction in a multi-threaded manner to speed up the process. Upon constructing a FAISS index for all the json.gz files, we iterate over each query paper, search for the top 100 relevant papers using the SPECTER embeddings in *each* FAISS index, and then finally merge the results to get the top 1000 papers for each query paper.

D.5 Comparative analysis of the computational costs

We compare the costs of different LLMs for both stages in Table 13.

Ranking: We explore two types of LLM-based reranking mechanisms: permutation and debate ranking. For n query papers (=500 for our RollingEval datasets) and top-k candidates retrieved from S2 per query paper (k=100 in our experiments), permutation ranking would require n API calls, whereas debate ranking would require $n * k$ API calls. Debate ranking needs more API calls as it involves one additional API call per candidate paper to generate

[9] Code will be released at github.com

[10] https://huggingface.co/spaces/evaluate-metric/rouge Since it is a known issue in the NLG community of different implementations producing different results, we stick to evaluate==0.4.0 for reporting all the results, reproducing the ROUGE scores for baselines from Multi-XScience model outputs.

[11] https://spacy.io/usage/linguistic-features

[12] https://app.endpoints.anyscale.com/

[13] https://platform.openai.com/docs/guides/gpt

[14] https://api.semanticscholar.org/api-docs/graph

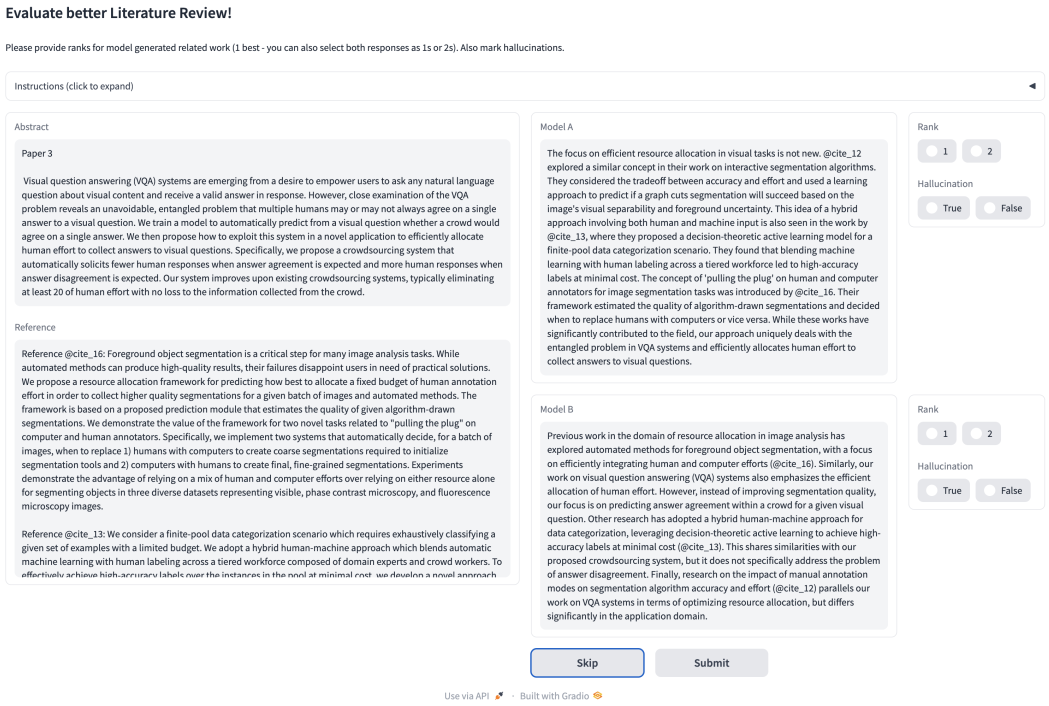
[15] https://api.semanticscholar.org/api-docs/recommendations

the citation probability score and reasoning. Therefore, there are k additional API calls per query paper compared to permutation ranking, where we prompt the LLM to directly rank relevance for all the candidate papers. We refer the reader to Figure 14for The exact prompt used for debate ranking.

Generation: There was only one request per query abstract in the RollingEval dataset, so 500 requests in total for each experiment (as n = 500 in RollingEval). The table below summarizes the API analysis for the two stages of the pipeline for the RollingEval experiments.

Experiment	Method	Requests	Tokens	Cost
Ranking	GPT-4 Permutation Reranking	500	~20M input + ~0.25M output tokens	$50
	Llama-3.1 Debate Ranking (w/o attribution)	500 x 100	~33M input + ~0.25M output tokens	$0
	Llama-3.1 Debate Ranking (w/ attribution)	500 x 100	~33M input + ~15M output tokens	$0
Generation	Llama 2 70B (using Anyscale Endpoint)	500	~0.75M input + ~0.15M output tokens	$3.84
	GPT-3.5-turbo	500	~0.75M input + ~0.15M output tokens	$4.2
	GPT-4	500	~0.75M input + ~0.15M output tokens	$22
	GPT-4 (Plan)	500	~0.75M input + ~0.15M output tokens	$25

Table 13: Computational costs for different experiments on the RollingEval dataset. Costs for generation experiments on the Multi-XScience are approximately 10 times that of the RollingEval dataset.



Evaluate better Literature Review!

Please provide ranks for model generated related work (1 best - you can also select both responses as 1s or 2s). Also mark hallucinations.

Abstract

Paper 3

Visual question answering (VQA) systems are emerging from a desire to empower users to ask any natural language question about visual content and receive a valid answer in response. However, close examination of the VQA problem reveals an unavoidable, entangled problem that multiple humans may or may not always agree on a single answer to a visual question. We train a model to automatically predict from a visual question whether a crowd would agree on a single answer. We then propose how to exploit this system in a novel application to efficiently allocate human effort to collect answers to visual questions. Specifically, we propose a crowdsourcing system that automatically solicits fewer human responses when answer agreement is expected and more human responses when answer disagreement is expected. Our system improves upon existing crowdsourcing systems, typically eliminating at least 20 of human effort with no loss to the information collected from the crowd.

Reference

Reference @cite_16: Foreground object segmentation is a critical step for many image analysis tasks. While automated methods can produce high-quality results, their failures disappoint users in need of practical solutions. We propose a resource allocation framework for predicting how best to allocate a fixed budget of human annotation effort in order to collect higher quality segmentations for a given batch of images and automated methods. The framework is based on a proposed prediction module that estimates the quality of given algorithm-drawn segmentations. We demonstrate the value of the framework for two novel tasks related to "pulling the plug" on computer and human annotators. Specifically, we implement two systems that automatically decide, for a batch of images, when to replace 1) humans with computers to create coarse segmentations required to initialize segmentation tools and 2) computers with humans to create final, fine-grained segmentations. Experiments demonstrate the advantage of relying on a mix of human and computer efforts over relying on either resource alone for segmenting objects in three diverse datasets representing visible, phase contrast microscopy, and fluorescence microscopy images.

Reference @cite_13: We consider a finite-pool data categorization scenario which requires exhaustively classifying a given set of examples with a limited budget. We adopt a hybrid human-machine approach which blends automatic machine learning with human labeling across a tiered workforce composed of domain experts and crowd workers. To effectively achieve high-accuracy labels over the instances in the pool at minimal cost, we develop a novel approach

Model A

The focus on efficient resource allocation in visual tasks is not new. @cite_12 explored a similar concept in their work on interactive segmentation algorithms. They considered the tradeoff between accuracy and effort and used a learning approach to predict if a graph cuts segmentation will succeed based on the image's visual separability and foreground uncertainty. This idea of a hybrid approach involving both human and machine input is also seen in the work by @cite_13, where they proposed a decision-theoretic active learning model for a finite-pool data categorization scenario. They found that blending machine learning with human labeling across a tiered workforce led to high-accuracy labels at minimal cost. The concept of 'pulling the plug' on human and computer annotators for image segmentation tasks was introduced by @cite_16. Their framework estimated the quality of algorithm-drawn segmentations and decided when to replace humans with computers or vice versa. While these works have significantly contributed to the field, our approach uniquely deals with the entangled problem in VQA systems and efficiently allocates human effort to collect answers to visual questions.

Model B

Previous work in the domain of resource allocation in image analysis has explored automated methods for foreground object segmentation, with a focus on efficiently integrating human and computer efforts (@cite_16). Similarly, our work on visual question answering (VQA) systems also emphasizes the efficient allocation of human effort. However, instead of improving segmentation quality, our focus is on predicting answer agreement within a crowd for a given visual question. Other research has adopted a hybrid human-machine approach for data categorization, leveraging decision-theoretic active learning to achieve high-accuracy labels at minimal cost (@cite_13). This shares similarities with our proposed crowdsourcing system, but it does not specifically address the problem of answer disagreement. Finally, research on the impact of manual annotation modes on segmentation algorithm accuracy and effort (@cite_12) parallels our work on VQA systems in terms of optimizing resource allocation, but differs significantly in the application domain.

Figure 7: Interface of our human evaluation setup.

Abstract of Multi-XScience paper (Lu et al., 2020)
Reference @cite_1: Multi-document summarization is a challenging task for which there exists little large-scale datasets. We propose Multi-XScience, a large-scale multi-document summarization dataset created from scientific articles. MultiXScience introduces a challenging multi-document summarization task: writing the related-work section of a paper based on its abstract and the articles it references. Our work is inspired by extreme summarization, a dataset construction protocol that favours abstractive modeling approaches. Descriptive statistics and empirical results—using several state-of-the-art models trained on the MultiXScience dataset—reveal that Multi-XScience is well suited for abstractive models.
Abstract of Extractive and Abstractive Summarization paper (Pilault et al., 2020)
Reference @cite_2: We present a method to produce abstractive summaries of long documents that exceed several thousand words via neural abstractive summarization. We perform a simple extractive step before generating a summary, which is then used to condition the transformer language model on relevant information before being tasked with generating a summary. We show that this extractive step significantly improves summarization results. We also show that this approach produces more abstractive summaries compared to prior work that employs a copy mechanism while still achieving higher rouge scores. Note: The abstract above was not written by the authors, it was generated by one of the models presented in this paper.
Abstract of Galactica paper (Taylor et al., 2022)
Reference @cite_3: Information overload is a major obstacle to scientific progress. The explosive growth in scientific literature and data has made it ever harder to discover useful insights in a large mass of information. Today scientific knowledge is accessed through search engines, but they are unable to organize scientific knowledge alone. In this paper we introduce Galactica: a large language model that can store, combine and reason about scientific knowledge. We train on a large scientific corpus of papers, reference material, knowledge bases and many other sources. We outperform existing models on a range of scientific tasks. On technical knowledge probes such as LaTeX equations, Galactica outperforms the latest GPT-3 by 68.2% versus 49.0%. Galactica also performs well on reasoning, outperforming Chinchilla on mathematical MMLU by 41.3% to 35.7%, and PaLM 540B on MATH with a score of 20.4% versus 8.8%. It also sets a new state-of-the-art on downstream tasks such as PubMedQA and MedMCQA dev of 77.6% and 52.9%. And despite not being trained on a general corpus, Galactica outperforms BLOOM and OPT-175B on BIG-bench. We believe these results demonstrate the potential for language models as a new interface for science. We open source the model for the benefit of the scientific community.
Plan for Table 15
Please generate 5 sentences in 120 words. Cite @cite_1 at line 1, 3 and 5.
Plan for Table 15
Please generate 5 sentences in 120 words. Cite @cite_1 at line 1 and 3. Cite @cite_2 at line 2 and 5. Cite @cite_3 at line 4 and 5.

Table 14: Abstracts of papers which are reference citations in Tables 15 and 16.

GPT-4 0-shot

The field of multi-document summarization has been a challenging one due to the lack of large-scale datasets (@cite_1). One significant attempt to address this gap is the Multi-XScience dataset, a large-scale collection of scientific articles designed to simulate the task of writing the related-work section of a research paper (@cite_1). This approach was inspired by extreme summarization, a methodology that promotes abstractive modeling approaches. However, the current study offers a novel strategy for literature review generation using large language models (LLMs), which goes beyond the scope of the Multi-XScience dataset. The proposed method involves the creation of an intermediate plan, which is then used to generate the actual text. Empirical evidence suggests that this approach improves the quality of the generated literature reviews compared to the conventional zero-shot generation. Additionally, this work expands the Multi-XScience corpus to include the full text of research papers, further enhancing the depth and breadth of information available for literature review generation. This study also ensures the validity of zero-shot experiments by creating a new test corpus from recent arXiv papers, thus avoiding test set contamination. This reflects a more comprehensive and rigorous approach to the generation of literature reviews using LLMs.

GPT-3.5 Plan

The task of generating literature reviews using large language models (LLMs) has gained attention recently. One approach is to condition the LLM on the abstracts of the papers to be cited and generate the literature review directly. However, the authors propose a novel strategy where they first generate a plan for the review and then use it to generate the actual text. This intermediate planning step is found to improve the quality of the generated literature reviews compared to vanilla zero-shot generation. The authors extend the Multi-XScience corpus, previously containing only abstracts, to include the full text of research papers. Additionally, they create a new test corpus consisting of recent arXiv papers, ensuring that their zero-shot experiments are not affected by contamination from the test set. The Multi-XScience dataset, used in their work, is particularly suitable for abstractive models due to its challenging multi-document summarization task (@cite_1). The empirical results from several state-of-the-art models trained on this dataset demonstrate its effectiveness for abstractive modeling approaches.

GPT-3.5 0-shot

Prior work in the field of generating literature reviews has primarily focused on the use of large language models (LLMs) and multi-document summarization. Multi-document summarization is a challenging task due to the lack of large-scale datasets (@cite_1). The Multi-XScience dataset was proposed as a solution to this problem, specifically designed for generating the related-work section of a paper based on its abstract and the articles it references. This dataset introduced a challenging multidocument summarization task that favors abstractive modeling approaches (@cite_1).

Our work builds upon these prior efforts by exploring the zero-shot abilities of recent LLMs for the task of generating literature reviews. We propose a novel strategy that involves generating an intermediate plan for the review, which is then used to generate the actual text. While modern LLMs can generate literature reviews without such intermediate plans, our empirical study shows that the use of these plans improves the quality of the generated reviews compared to vanilla zero-shot generation.

In addition to extending the Multi-XScience corpus to include the full text of research papers, we also create a new test corpus consisting of recent arXiv papers. This allows us to ensure that our zero-shot experiments are not affected by test set contamination.

The prior work on multi-document summarization and the Multi-XScience dataset provides a valuable foundation for our study. However, they do not specifically address the use of intermediate plans for literature review generation, which is a key contribution of our work. By incorporating these intermediate plans, we aim to enhance the quality and coherence of the generated literature reviews.

Llama 2 Plan

The task of generating literature reviews for scientific research papers has gained significant attention in recent years, with a focus on using large language models (LLMs) for zero-shot generation. Inspired by extreme summarization, Multi-XScience was introduced as a large-scale dataset for multi-document summarization, including the challenging task of writing related-work sections based on abstracts and referenced articles. Our work builds upon this foundation by exploring the effectiveness of intermediate plans for literature review generation. Our empirical study shows that these plans improve the quality of generated literature reviews, and we contribute to the extension of the Multi-XScience corpus to include full-text papers. Additionally, we create a new test corpus to ensure zero-shot experiments are free from test set contamination. (@cite_1)

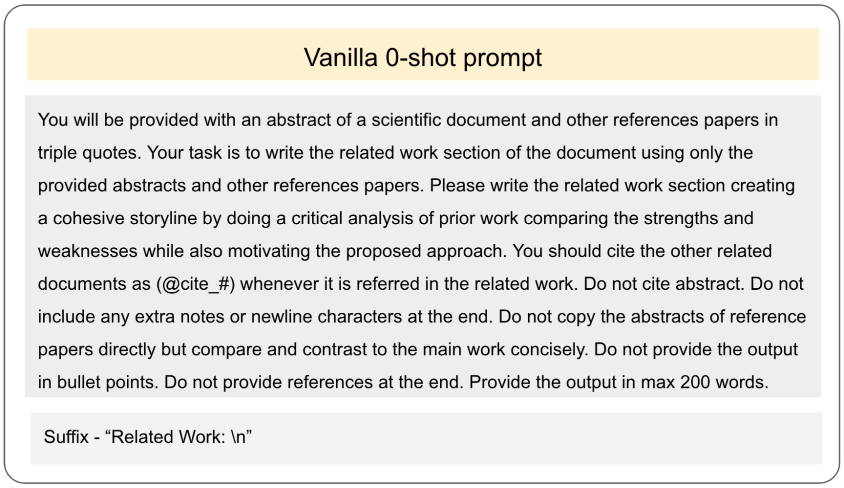
Llama 2 0-shot

In recent years, there has been growing interest in using large language models (LLMs) for zero-shot literature review generation. However, most studies have relied on training or prompting LLMs to condition on all abstracts of papers to be cited, which can be limiting in terms of quality and efficiency. In contrast, our proposed approach involves generating a plan for the literature review and then using it to generate the actual text. We evaluate the effectiveness of this approach using a novel extended version of the Multi-XScience corpus, which now includes the full text of research papers. Our results show that intermediate plans improve the quality of generated literature reviews over vanilla zero-shot generation. One of the key challenges in multi-document summarization is the lack of large-scale datasets. To address this, we have created a new test corpus consisting of recent arXiv papers with full content, ensuring that our zero-shot experiments are not contaminated by the release of the LLMs used in our study. Our work builds on the concept of extreme summarization, which favors abstractive modeling approaches. We have employed several state-of-the-art models trained on the Multi-XScience dataset to evaluate the effectiveness of our approach. The results show that our dataset is well-suited for abstractive models, providing a valuable resource for researchers working in this area. In summary, our contribution includes a novel approach to literature review generation using LLMs, an extended version of the Multi-XScience corpus, and a new test corpus. Our results demonstrate the effectiveness of our approach and highlight the potential of using LLMs for zero-shot literature review generation. References: * @cite_1: Multi-XScience: A Large-Scale Multi-Document Summarization Dataset. * @cite_2: Extreme Summarization: A Dataset Construction Protocol for Abstractive Modeling.

Table 15: We show outputs of different models corresponding to the same plan and prompt as paragraph of our related work (by GPT-4 Plan). Llama 2 70B 0-shot has the tendency to hallucinate citation (in red) which is remedied in plan based generation. Abstract of reference paper shown in Table 14 while the main abstract is a slight modified version of our current abstract. We also find that the 0-shot results are more verbose, copying the exact text verbatim, while the generated literature review using plans are more representative of the way humans write literature review.

GPT-4 Plan
The concept of generating the related-work section from abstracts and referenced articles is not entirely new, as evidenced by the Multi-XScience dataset proposed by @cite_1. The approach of extending abstractive summarization to lengthy documents via an extractive step for conditioning, as described by @cite_2, aligns with the proposed methodology here. However, the strategy introduced in this study differentiates itself by creating an iterative plan prior to the generation of the review, a step not present in the work of @cite_1. The LLM used in this study also builds upon the capabilities of models like Galactica @cite_3, which have demonstrated exceptional performance on scientific tasks. The results of @cite_2 and @cite_3 support the potential of large language models in scientific literature review generation, while the proposed approach aims to further enhance the quality of these outputs through structured planning.
GPT-4 0-shot
The literature review generation task, a form of multi-document summarization, has been tackled through different strategies. @cite_1 proposed a dataset, MultiXScience, to foster abstractive approaches, while @cite_2 used an extractive step to enhance neural summarization. Our work complements these by introducing intermediate plans to improve generation quality. Meanwhile, @cite_3's Galactica demonstrated potential for language models in scientific tasks, a direction we explore with zero-shot experiments.
GPT-3.5 Plan
The task of generating a literature review using large language models (LLMs) has gained attention recently. One approach is to condition the LLM on the abstract and content of related papers to generate the review (@cite_1). However, it has been observed that generating an intermediate plan for the review improves its quality compared to vanilla zero-shot generation. This motivated the proposal and examination of a novel strategy where a plan is first generated and then used to generate the actual text (@cite_1). Another method to improve abstractive summarization is by performing an extractive step before generating the summary. This step conditions the transformer language model on relevant information, leading to better summarization results compared to prior work that employs a copy mechanism (@cite_2). Additionally, the use of large language models like Galactica has shown promising results in organizing and reasoning about scientific knowledge (@cite_3). Galactica outperforms existing models on technical knowledge probes and downstream tasks, demonstrating the potential for language models as a new interface for science (@cite_3).
GPT-3.5 0-shot
Prior work has explored the use of large language models (LLMs) for various natural language processing tasks. Multi-document summarization is one such task, and the Multi-XScience dataset (@cite_1) has been created specifically for this purpose. This dataset focuses on writing the related-work section of a scientific paper based on its abstract and the articles it references. The dataset construction protocol favors abstractive modeling approaches, making it suitable for LLMs. Another approach to abstractive summarization is presented in a paper (@cite_2), where the authors propose a method to generate summaries of long documents. They employ an extractive step to gather relevant information before conditioning the transformer language model for summary generation. This approach yields more abstractive summaries compared to prior work, while still achieving higher rouge scores. Furthermore, the concept of using LLMs for scientific knowledge organization and reasoning is introduced in the Galactica model (@cite_3). This model is trained on a large scientific corpus and outperforms existing models on various scientific tasks. It demonstrates superior performance on technical knowledge probes and reasoning tasks, surpassing GPT-3 and other models. Galactica also achieves state-of-the-art results on downstream tasks such as PubMedQA and MedMCQA, showcasing the potential of LLMs as a new interface for science. Based on these prior works, we propose a novel strategy for literature review generation using LLMs. We first generate a plan for the review and then use it to generate the actual text. While modern LLMs can generate literature reviews without intermediate plans, our empirical study indicates that these plans improve the quality of generated reviews. Additionally, we create a new test corpus consisting of recent arXiv papers, ensuring that our zero-shot experiments are not affected by test set contamination.
Llama 2 Plan
In recent years, there has been a growing interest in using large language models (LLMs) for scientific literature review generation (@cite_1). One approach is to use a novel strategy that involves generating a plan for the review and then using it to generate the actual text (@cite_1). This approach has been shown to improve the quality of generated literature reviews over vanilla zero-shot generation (@cite_1). Another approach is to use a simple extractive step before generating a summary, which is then used to condition the transformer language model on relevant information (@cite_2). This approach has been shown to produce more abstractive summaries compared to prior work that employs a copy mechanism while still achieving higher ROUGE scores (@cite_2). Additionally, there have been efforts to train LLMs on large scientific corpora to improve their ability to store, combine, and reason about scientific knowledge (@cite_3). For example, Galactica, a large language model, has been trained on a range of scientific tasks and has outperformed existing models on technical knowledge probes, reasoning, and downstream tasks (@cite_3). These advancements in LLMs have the potential to revolutionize the way we interact with scientific literature and accelerate scientific progress.
Llama 2 0-shot
In this section, we discuss the related work in the field of scientific literature review generation using large language models (LLMs). Recent works have explored the use of LLMs for multi-document summarization tasks, such as generating the related work section of a paper based on its abstract and the articles it references (@cite_1). These works have shown that LLMs can produce high-quality summaries, but may suffer from the lack of relevant information and context (@cite_2). To address this issue, we propose a novel strategy for literature review generation that involves generating a plan for the review before generating the actual text. This approach has been shown to improve the quality of generated literature reviews over vanilla zero-shot generation (@cite_3). Additionally, we create a new test corpus consisting of recent arXiv papers to ensure that our zero-shot experiments do not suffer from test set contamination. Our work builds upon the recent advances in LLMs and their ability to store, combine, and reason about scientific knowledge (@cite_3). We believe that our approach has the potential to revolutionize the way scientific literature is reviewed and organized, and we open-source our model for the benefit of the scientific community. In summary, our related work demonstrates the potential of LLMs for scientific literature review generation and highlights the importance of context and information in producing high-quality summaries. Our proposed approach aims to address these limitations and provide a more effective way of generating literature reviews using LLMs.

Table 16: We show outputs of different models corresponding to the reference cited abstracts and plan mentioned in Table 14 while the main abstract is a slightly modified version of our current abstract. In this example, though, we have all the citations covered by all the models, we can see GPT-4 (Plan) output to be concise and closely following the plan. Moreover, GPT-3.5 and Llama 0-shot outputs are excessively elaborate, making them unsuitable for inclusion in the literature review. Our findings indicate that while LLMs can help in certain aspects, the final output still heavily relies on inputs from a human researcher.

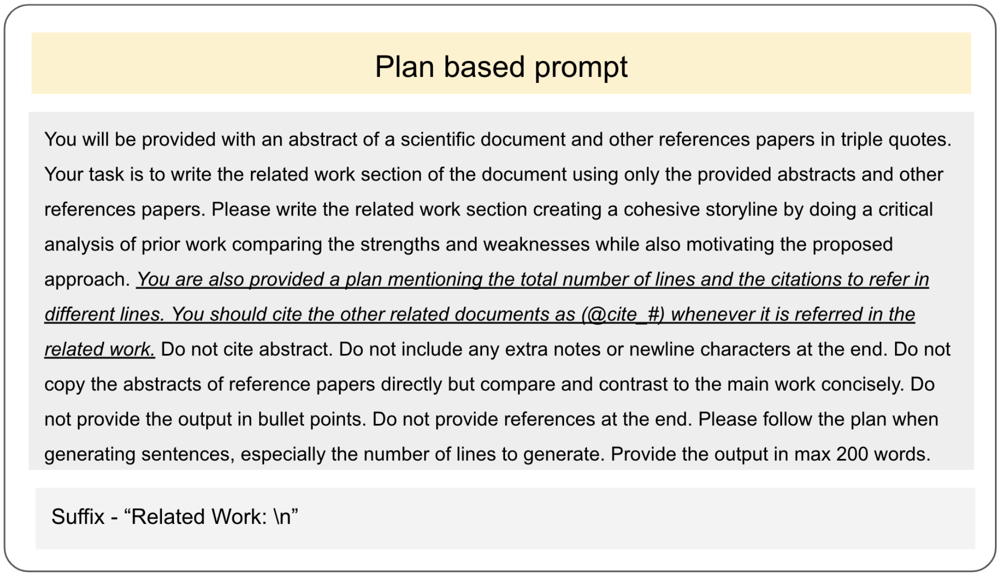


Vanilla 0-shot prompt

You will be provided with an abstract of a scientific document and other references papers in triple quotes. Your task is to write the related work section of the document using only the provided abstracts and other references papers. Please write the related work section creating a cohesive storyline by doing a critical analysis of prior work comparing the strengths and weaknesses while also motivating the proposed approach. You should cite the other related documents as (@cite_#) whenever it is referred in the related work. Do not cite abstract. Do not include any extra notes or newline characters at the end. Do not copy the abstracts of reference papers directly but compare and contrast to the main work concisely. Do not provide the output in bullet points. Do not provide references at the end. Provide the output in max 200 words.

Suffix - “Related Work: \n”

Figure 8: Prompt used for Vanilla 0-shot generation.

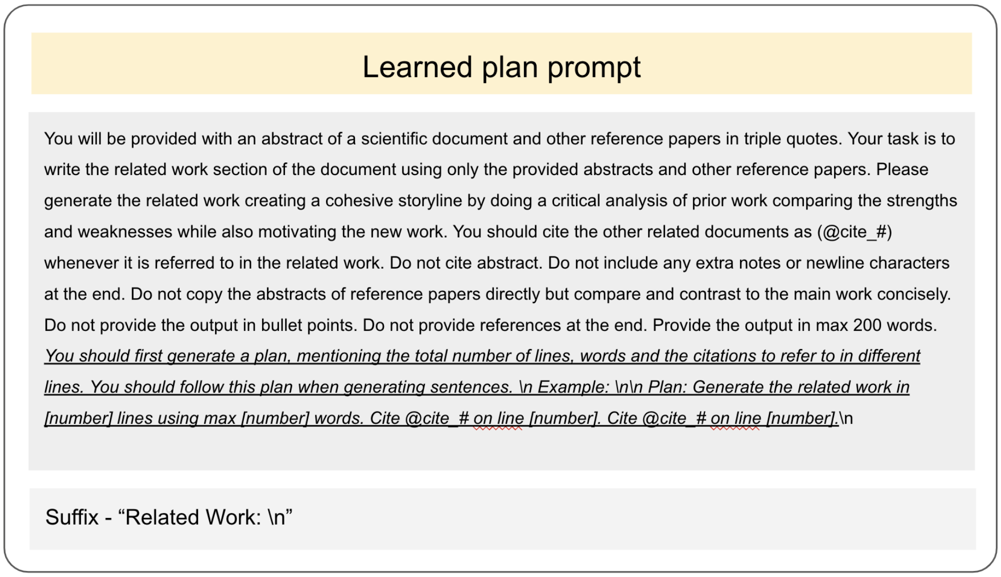


Plan based prompt

You will be provided with an abstract of a scientific document and other references papers in triple quotes. Your task is to write the related work section of the document using only the provided abstracts and other references papers. Please write the related work section creating a cohesive storyline by doing a critical analysis of prior work comparing the strengths and weaknesses while also motivating the proposed approach. *You are also provided a plan mentioning the total number of lines and the citations to refer in different lines. You should cite the other related documents as (@cite_#) whenever it is referred in the related work.* Do not cite abstract. Do not include any extra notes or newline characters at the end. Do not copy the abstracts of reference papers directly but compare and contrast to the main work concisely. Do not provide the output in bullet points. Do not provide references at the end. Please follow the plan when generating sentences, especially the number of lines to generate. Provide the output in max 200 words.

Suffix - “Related Work: \n”

Figure 9: Prompt used for plan-based generation. Underlined text shows the variation compared to the vanilla 0-shot prompting, where the user provides a structure of the expected paragraph.

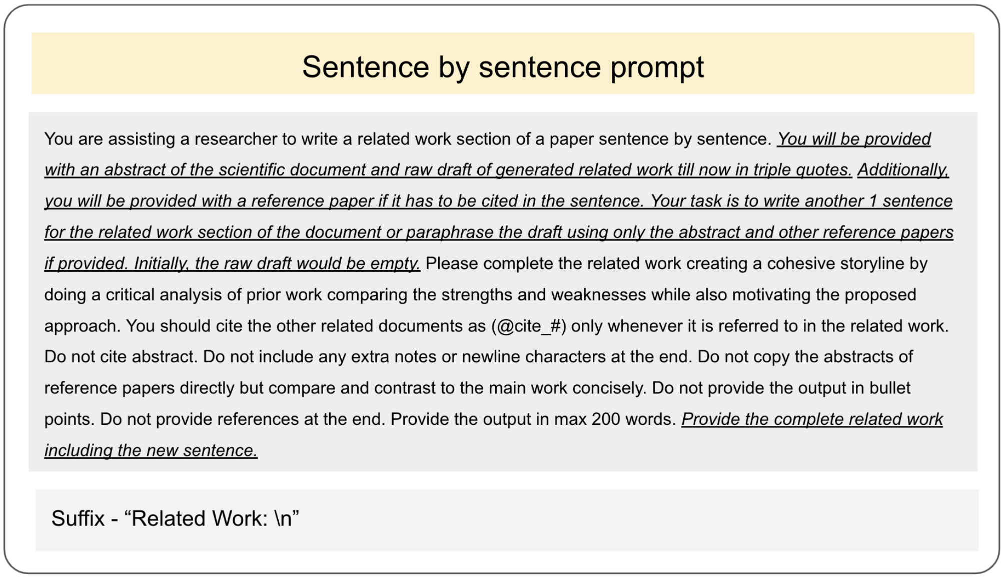


Learned plan prompt

You will be provided with an abstract of a scientific document and other reference papers in triple quotes. Your task is to write the related work section of the document using only the provided abstracts and other reference papers. Please generate the related work creating a cohesive storyline by doing a critical analysis of prior work comparing the strengths and weaknesses while also motivating the new work. You should cite the other related documents as (@cite_#) whenever it is referred to in the related work. Do not cite abstract. Do not include any extra notes or newline characters at the end. Do not copy the abstracts of reference papers directly but compare and contrast to the main work concisely. Do not provide the output in bullet points. Do not provide references at the end. Provide the output in max 200 words. *You should first generate a plan, mentioning the total number of lines, words and the citations to refer to in different lines. You should follow this plan when generating sentences. \n Example: \n\n Plan: Generate the related work in [number] lines using max [number] words. Cite @cite_# on line [number]. Cite @cite_# on line [number].*\n

Suffix - “Related Work: \n”

Figure 10: Prompt used when the plan is learned during generation. The model first generates a plan of sentences and citations which it would then condition upon to generate the final related work text, which can be considered as an extension of CoT style thinking step by step.

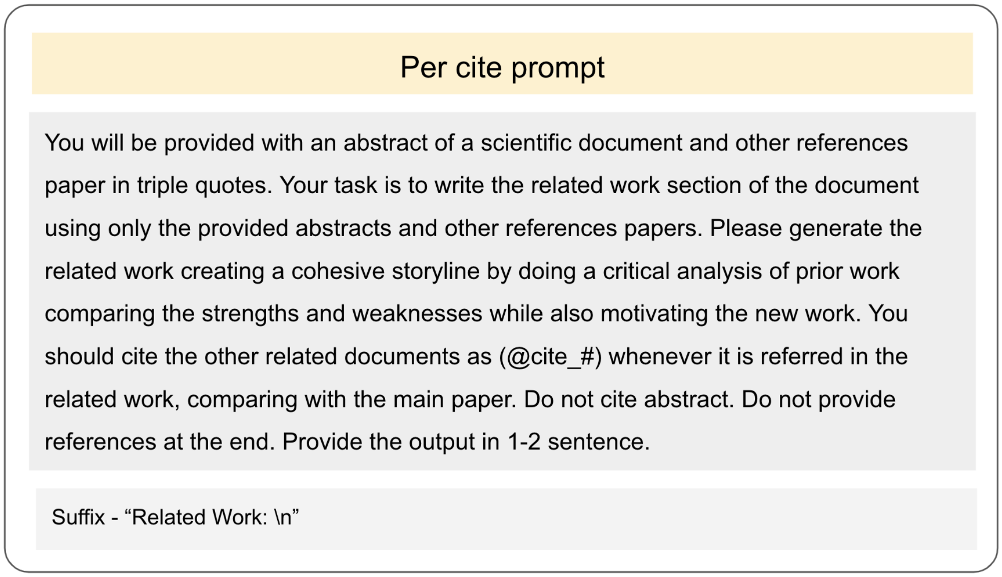


Sentence by sentence prompt

You are assisting a researcher to write a related work section of a paper sentence by sentence. *You will be provided with an abstract of the scientific document and raw draft of generated related work till now in triple quotes. Additionally, you will be provided with a reference paper if it has to be cited in the sentence. Your task is to write another 1 sentence for the related work section of the document or paraphrase the draft using only the abstract and other reference papers if provided. Initially, the raw draft would be empty.* Please complete the related work creating a cohesive storyline by doing a critical analysis of prior work comparing the strengths and weaknesses while also motivating the proposed approach. You should cite the other related documents as (@cite_#) only whenever it is referred to in the related work. Do not cite abstract. Do not include any extra notes or newline characters at the end. Do not copy the abstracts of reference papers directly but compare and contrast to the main work concisely. Do not provide the output in bullet points. Do not provide references at the end. Provide the output in max 200 words. *Provide the complete related work including the new sentence.*

Suffix - “Related Work: \n”

Figure 11: Prompt used for sentence-by-sentence generation. In this scenario, we prompt the model to generate one sentence for each citation individually.

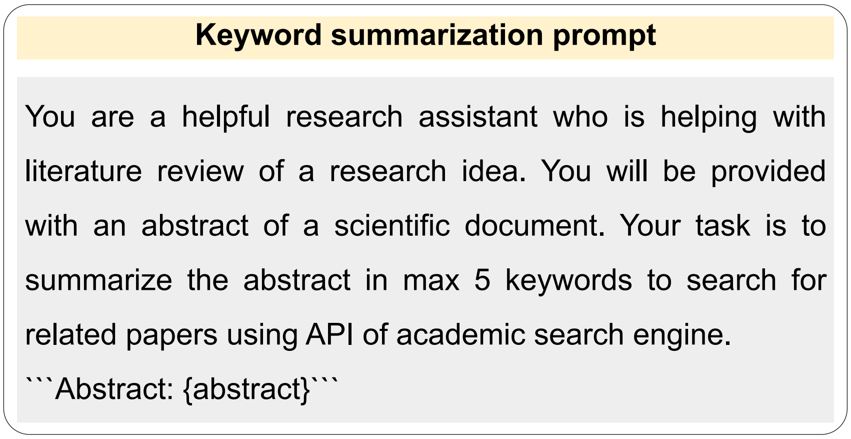


Per cite prompt

You will be provided with an abstract of a scientific document and other references paper in triple quotes. Your task is to write the related work section of the document using only the provided abstracts and other references papers. Please generate the related work creating a cohesive storyline by doing a critical analysis of prior work comparing the strengths and weaknesses while also motivating the new work. You should cite the other related documents as (@cite_#) whenever it is referred in the related work, comparing with the main paper. Do not cite abstract. Do not provide references at the end. Provide the output in 1-2 sentence.

Suffix - “Related Work: \n”

Figure 12: Prompt used for generating output per citation.



Keyword summarization prompt

You are a helpful research assistant who is helping with literature review of a research idea. You will be provided with an abstract of a scientific document. Your task is to summarize the abstract in max 5 keywords to search for related papers using API of academic search engine.
```Abstract: {abstract}```

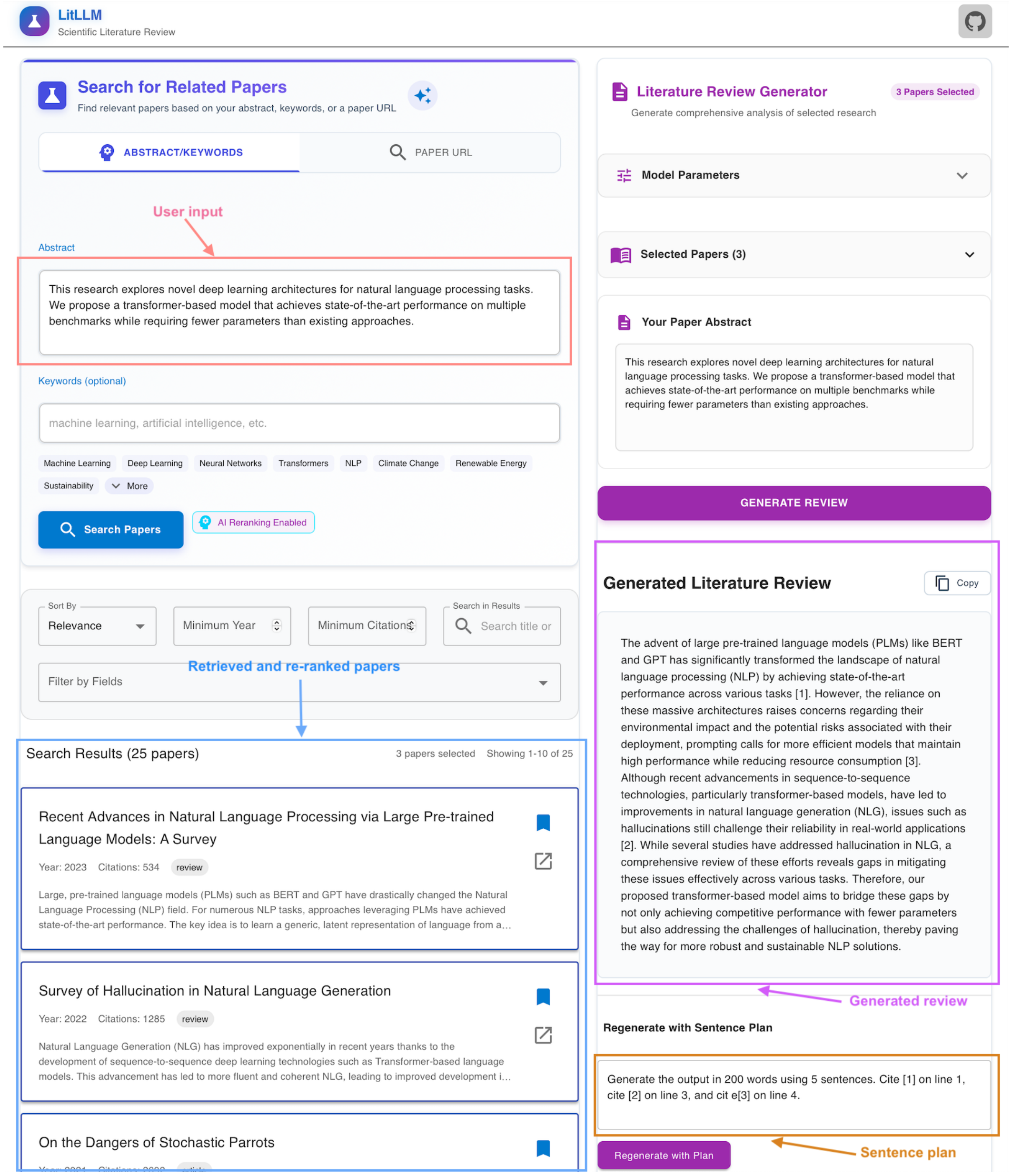
Figure 13: Prompt used to summarize the research idea by LLM to search an academic engine

```
"""
You are a helpful research assistant who is helping with literature
    review of a research idea. Your task is to rank some papers based on
    their relevance to the query abstract.


## Instruction:
Given the query abstract:
<query_abstract>{query_abstract}</query_abstract>

Given the candidate reference paper abstract:
<candidate_paper_abstracts>{reference_papers}</candidate_paper_abstracts>

* Given the abstract of the candidate reference papers, provide me with a
     number between 0 and 100 (upto two decimal places) that is
    proportional to the probability of a paper with the given query
    abstract including the candidate reference paper in its literature
    review.
* In addition to the probability, give me arguments for and against
    including this paper in the literature review.
* You must enclose your arguments for including the paper within <
    arguments_for> and </arguments_for> tags.
* You must enclose your arguments for including the paper within <
    arguments_against> and </arguments_against> tags.
* Extract relevant sentences from the candidate paper abstract to support
     your arguments.
* Put the extracted sentences in quotes.
* You can use the information in other candidate papers when generating
    the arguments for a candidate paper.
* You must enclose your score within <probability> and </probability>
    tags.
* Generate the arguments first then the probability score.
* Generate arguments and probabitlity for each paper separately.
* Do not generate anything else apart from the probability and the
    arguments.
* Follow this process even if a candidate paper happens to be identical
    or near-perfect match to the query abstract.

### Response Format for each paper:
<arguments_for>
[Paper ID]: [Reason for including the paper]
Extracted Sentences: "Sentence 1", "Sentence 2", ...
</arguments_for>
<arguments_against>
[Paper ID]: [Reason for not including the paper]
Extracted Sentences: "Sentence 1", "Sentence 2", ...
</arguments_against>
<probability>
[Paper ID]: [Final Probability Score Based on the Arguments]
</probability>

### Your Response:
"""
```

Figure 14: Prompt used for Debate Ranking.

```
multiple_queries_prompt = ”””You are a helpful research assistant who is
    helping with literature review of a research idea. You will be
    provided with an abstract of a scientific document. Your task is to
    frame queries that would be used to search an academic search engine
    and retrieve relevant papers.

Here is the abstract:
{abstract}

## Instruction:
* Each query should not be more than 5 keywords. Please write the query
    in a similar fashion as a human would use search engine.
* Please generate {n_keywords} search queries.
* Please make sure to generate different search queries using a variety
    of key words so as to get maximum papers that could be cited.
* Please return a JSON with a key ”queries” that has the list of queries:
”””

optimized_kw_prompt = ”””
You are a helpful research assistant who is helping with literature
    review of a research idea. You will be provided with an abstract of a
     scientific document. Your task is to frame multiple queries that
    would be used to search an academic search engine and retrieve
    relevant papers.

Here is the abstract:
{abstract}

## Instruction:
* Each query should not be more than 5 keywords. Please write the query
    in a similar fashion as a human would use search engine.
* Generate as many search queries as you need.
* Please make sure to generate mutually-exclusive search queries using a
    variety of key words so as to get maximum papers that could be cited.
* In addition to the queries, also provide a reasoning for the generated
    queries.
* Extract the relevant sentences from the abstract that justify your
    reasoning.
* Put the extracted sentences in quotes and put them at the end of each
    of your reasonings.
* Example reasoning:
”The query is framed to get papers that discuss the method proposed in
    the abstract. The sentence 'The method proposed in this paper is
    similar to the one proposed in the query abstract.' is extracted from
     the abstract.”
* Please return a JSON with a key ”queries” that has the list of queries
    and a ”reasoning” key that has the reasoning for the queries.
* Do not generate anything else apart from the JSON.

### Response:
”””
```

Figure 15: Prompts used for constructing search queries.



This research explores novel deep learning architectures for natural language processing tasks. We propose a transformer-based model that achieves state-of-the-art performance on multiple benchmarks while requiring fewer parameters than existing approaches.

Search Results (25 papers)   3 papers selected   Showing 1-10 of 25

Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey

Year: 2023   Citations: 534   review

Large, pre-trained language models (PLMs) such as BERT and GPT have drastically changed the Natural Language Processing (NLP) field. For numerous NLP tasks, approaches leveraging PLMs have achieved state-of-the-art performance. The key idea is to learn a generic, latent representation of language from a...

Survey of Hallucination in Natural Language Generation

Year: 2022   Citations: 1285   review

Natural Language Generation (NLG) has improved exponentially in recent years thanks to the development of sequence-to-sequence deep learning technologies such as Transformer-based language models. This advancement has led to more fluent and coherent NLG, leading to improved development i...

On the Dangers of Stochastic Parrots

The advent of large pre-trained language models (PLMs) like BERT and GPT has significantly transformed the landscape of natural language processing (NLP) by achieving state-of-the-art performance across various tasks [1]. However, the reliance on these massive architectures raises concerns regarding their environmental impact and the potential risks associated with their deployment, prompting calls for more efficient models that maintain high performance while reducing resource consumption [3]. Although recent advancements in sequence-to-sequence technologies, particularly transformer-based models, have led to improvements in natural language generation (NLG), issues such as hallucinations still challenge their reliability in real-world applications [2]. While several studies have addressed hallucination in NLG, a comprehensive review of these efforts reveals gaps in mitigating these issues effectively across various tasks. Therefore, our proposed transformer-based model aims to bridge these gaps by not only achieving competitive performance with fewer parameters but also addressing the challenges of hallucination, thereby paving the way for more robust and sustainable NLP solutions.

Generate the output in 200 words using 5 sentences. Cite [1] on line 1, cite [2] on line 3, and cit e[3] on line 4.

Figure 16: System interface. Our system works on the Retrieval Augmented Generation (RAG) principle to generate the literature review grounded in retrieved relevant papers. The user needs to provide the abstract in the textbox (in purple) and press send to get the generated related work (in red). First, the abstract is summarized into keywords, which are used to query a search engine. Retrieved results are re-ranked (in blue) using an LLM, which is then used as context to generate the related work. Users could also provide a sentence plan (in green) according to their preference to generate a concise, readily usable literature review.