AcademicEval: Live Long-Context LLM Benchmark

Anonymous authors
Paper under double-blind review

Abstract

Large Language Models (LLMs) have recently achieved remarkable performance in long-context understanding. However, current long-context LLM benchmarks are limited by rigid context length, labor-intensive annotation, and the pressing challenge of label leakage issues during LLM training. Therefore, we propose AcademicEval, a live benchmark for evaluating LLMs over long-context generation tasks. AcademicEval adopts papers on arXiv to introduce several academic writing tasks with long-context inputs, *i.e.*, Title, Abstract, Introduction, and Related Work, which cover a wide range of abstraction levels and require no manual labeling. Moreover, AcademicEval integrates high-quality and expert-curated few-shot demonstrations from a collected co-author graph to enable flexible context length. Especially, AcademicEval features an efficient live evaluation, ensuring no label leakage. We conduct a holistic evaluation on AcademicEval, and the results illustrate that LLMs perform poorly on tasks with hierarchical abstraction levels and tend to struggle with long few-shot demonstrations, highlighting the challenge of our benchmark. Through experimental analysis, we also reveal some insights for enhancing LLMs' long-context modeling capabilities. We will release the source code and data upon publication.

1 Introduction

Large Language Models (LLMs) have recently achieved tremendous success in natural language processing (NLP) tasks (Achiam et al., 2023; AI@Meta, 2024). However, when facing long context inputs, LLMs show a sharp decline in performance, which poses a pressing challenge to LLMs in understanding and capturing key information in long texts (Li et al., 2024; Liu et al., 2024). Therefore, several long-context LLM benchmarks are spawned to evaluate LLMs in various settings, including question answering, summarizing, and reasoning (Shaham et al., 2023; An et al., 2023; Dong et al., 2023; Bai et al., 2023b; Li et al., 2023; Zhang et al., 2024). Despite their success, these benchmarks still suffer from concerns of rigid context length, saturated performance, and being leaked in LLM training.

We envision that the *next-generation long-context LLM benchmarks* should ideally possess three key features. (1) *Flexible* and potentially *unlimited* context length: existing benchmarks fix the context for each long-context problem; ideally, the format and length of the context could be flexibly set based on the LLM's capability, especially given the release of long-context LLMs (Reid et al., 2024) and their capabilities in ingesting multi-modal information, *e.g.*, graphs (Dong et al., 2024). (2) High-quality labels derived from *real-world data*, *minimizing* human labeling efforts: existing long-context benchmarks often require human labeling (Bai et al., 2023b; An et al., 2023; Li et al., 2023; Dong et al., 2023; Zhang et al., 2024), which is costly and limits the size of the benchmarks to about 2000 samples (Xu et al., 2023) (3) Live updates to mitigate information leakage during LLM pretraining and fine-tuning: benchmark data contamination in LLM has gradually become a severe issue (Sainz et al., 2023; Ye et al., 2024; Zhu et al., 2024b;a; Xu et al., 2024); we argue that holding out future data as the val/test set is one of the most effective approaches for open benchmarks.

Based on these principles, we propose AcademicEval, a live benchmark to evaluate LLMs over long-context generation tasks. AcademicEval adopts arXiv as its data source and features a suite of academic writing tasks on each paper without labor-intensive annotation: Title, Abstract, Introduction, and

Table 1: **Comparison with Existing Long-context LLM Benchmarks**. Each column indicates the average input length, whether the annotation is human-assisted, whether there are tasks with hierarchical abstraction levels, whether it contains few-shot demonstrations, and whether the benchmark is lively updated, respectively.

Benchmark	Avg Len	Automatic Annotation	Hierarchical Abstraction	Few-shot Demos	Live Update
ZeroSCROLLS (Shaham et al., 2023)	∼10K	✓	✗	✗	✗
L-Eval (An et al., 2023)	∼8K	✗	✗	✗	✗
BAMBOO (Dong et al., 2023)	∼16K	✗	✗	✗	✗
LongBench (Bai et al., 2023b)	∼8K	✗	✗	✓	✗
LooGLE (Li et al., 2023)	∼20K	✗	✗	✗	✗
∞Bench (Zhang et al., 2024)	∼200K	✗	✗	✗	✗
AcademicEval (ours)	**Flexible**	✓	✓	✓	✓

Related Work, each of which has long-context input and hierarchical abstraction levels. In particular, we construct a co-author graph via arXiv API to conveniently obtain co-author papers as high-quality and expert-curated few-shot demonstrations, which also possess AcademicEval flexible context length. Furthermore, AcademicEval introduces efficient live evaluation based on the co-author graph, which utilizes the latest papers on arXiv to update the benchmark data periodically and ensures no label leakage. Moreover, AcademicEval provides in-context few-shot demonstrations for each sample, which is neglected by most existing long-context LLM benchmarks (Liu et al., 2024; Li et al., 2024). In the experiment, we evaluate three types of baselines on AcademicEval: standard LLMs, long-context LLMs, and retrieval-augmented language models (RALM). Experimental results show that current LLMs cannot deal with long-context context tasks well at diverse abstraction levels, while RALM is generally the best-performing one among the three types of baselines. Additionally, as the input length increases, noticeable performance degradation can be seen on most LLMs, whereas RALM maintains a more stable performance despite variations in the input length. Although we find that few-shot demonstrations from co-author papers can slightly strengthen the performance over some tasks, it is still limited by the long context modeling capabilities of LLMs. In general, the experimental findings indicate that AcademicEval is a challenging long-context LLM benchmark.

We illustrate the comparison with existing long-context LLM benchmarks in Table 1. Our contributions are summarized as follows:

- We propose a live benchmark, AcademicEval, to evaluate LLMs over long-context generation tasks. AcademicEval features four academic writing tasks with hierarchical abstraction levels and requires no manual annotation.
- We construct a co-author graph via the arXiv API and draw on the co-author papers as informative few-shot demonstrations, making the context length of AcademicEval flexible and scalable. Especially, AcademicEval conducts periodic data updates on the co-author graph to enable efficient live evaluation, which ensures no label leakage and fair evaluation.
- We conduct comprehensive experiments on AcademicEval, and the results demonstrate its challenges and yield potential insights for improving LLMs in long-context modeling.

2 Related Work

Long-context Modeling and LLM Benchmarks. LLMs are known to be powerful in language modeling tasks (Achiam et al., 2023; AI@Meta, 2024). However, when it comes to long-context inputs, LLMs show a sharp decline in performance, posing a pressing challenge when benchmarking their long-context modeling

capabilities (Liu et al., 2024; Li et al., 2024; 2025). Currently, there are two mainstream technologies for long-context modeling tasks: retrieval-augmented language models (RALM)(Ram et al., 2023; Yu et al., 2023; Trivedi et al., 2022; Jiang et al., 2023; Asai et al., 2023) and long-context LLMs (Bai et al., 2023a; Jiang et al., 2024; Teknium et al.). RALM equips LLMs with a retriever(Robertson et al., 2009; Ramos et al., 2003; Karpukhin et al., 2020; Izacard et al., 2021) to perform information retrieval on short text chunks, which are then fed to LLMs together with the input query to generate the final output. As a retrieval system, RALM is usually evaluated over retrieval-based benchmarks, including STARK (Wu et al., 2024), RGB (Chen et al., 2024), ARES (Saad-Falcon et al., 2023), etc. In comparison, long-context LLMs expand their context window length to accommodate longer inputs and are benchmarked over various tasks, which include long-context QA, summarization, conversations, reasoning, etc (Shaham et al., 2023; An et al., 2023; Dong et al., 2023; Bai et al., 2023b; Li et al., 2023; Zhang et al., 2024; Li et al., 2025).

Label Leakage in LLM Benchmarks. Label leakage has always been a severe issue that benchmarks must attempt to avoid during data collection. However, recent researches (Xu et al., 2024; Zhu et al., 2024b;a; Ye et al., 2024) point out that most LLM benchmarks are composed of statically collected data, which may be inevitably included in the large amount of training data of LLMs, causing label leakage. Therefore, some works attempt to measure or detect the extent of label leakage in LLM benchmarks. Benbench (Xu et al., 2024) leverages perplexity and N-gram accuracy to quantify potential label leakage, while PAC (Ye et al., 2024) detects contaminated data by comparing the polarized distance of samples before and after augmentation. Even though these approaches propose to measure or detect label leakage, there is little work on mitigating and solving this issue (Zhu et al., 2024b). Dynabench (Kiela et al., 2021) and Dynaboard (Ma et al., 2021) feature dynamic human-in-the-loop dataset creation while avoiding leakage, which is very labor-intensive. DyVal (Zhu et al., 2024b) leverages pre-set constraints and directed acyclic graphs (DAG) to dynamically generate test cases with diverse complexities, reducing the risk of label leakage. FreshBench (Zhu et al., 2024a) and StackMIA (Ye et al., 2024) collect the latest data from public websites periodically and simply rely on the chronological split to build a dynamic benchmark.

Long-context Summarization Benchmarks. Solving AcademicEval requires LLM's long-context summarization capability (Liu et al., 2024). Existing works include (1) query-based summarization tasks, focusing on the capability of models to position and capture local key information in long texts given a specific query (Litvak & Vanetik, 2017; Wang et al., 2022); (2) single-document or multi-document summarization tasks concentrate on evaluating the ability of models to understand long texts holistically (Cohan et al., 2018; Meng et al., 2021; Huang et al., 2021; Kryściński et al., 2021; Cachola et al., 2020). These long-context summarization benchmarks suffer from the above-mentioned limitations, including requiring human-assisted labeling and concerns about data leakage; moreover, these summarization tasks focus on one-level summarization, failing to consider the summarizations at different abstraction levels.

3 AcademicEval Benchmark

In this section, we propose AcademicEval (Figure 1) for live evaluation over long-context generation tasks with hierarchical abstraction levels. We first describe data collection and preprocessing in Section 3.1. Then, in Section 3.2, four academic writing tasks with diverse abstraction levels are introduced, and we also integrate few-shot demonstrations to make the context length flexible and scalable. Finally, Section 3.3 elucidates the live evaluation with periodic data updates.

3.1 Data Curation

Co-author Graph Construction via arXiv. As a public paper preprint platform, arXiv[1] has always been favored by researchers. It archives a huge amount of papers and updates the latest ones daily, which serves as an excellent data source and also lays the foundation for the live update of our benchmark. Thanks to the arXiv API[2], paper files can be obtained in batch without much manual effort. We first collect and construct a co-author graph (*i.e.*, edges are established between two co-author nodes) using the arXiv API through

[1]https://arxiv.org/
[2]https://info.arxiv.org/help/api/index.html

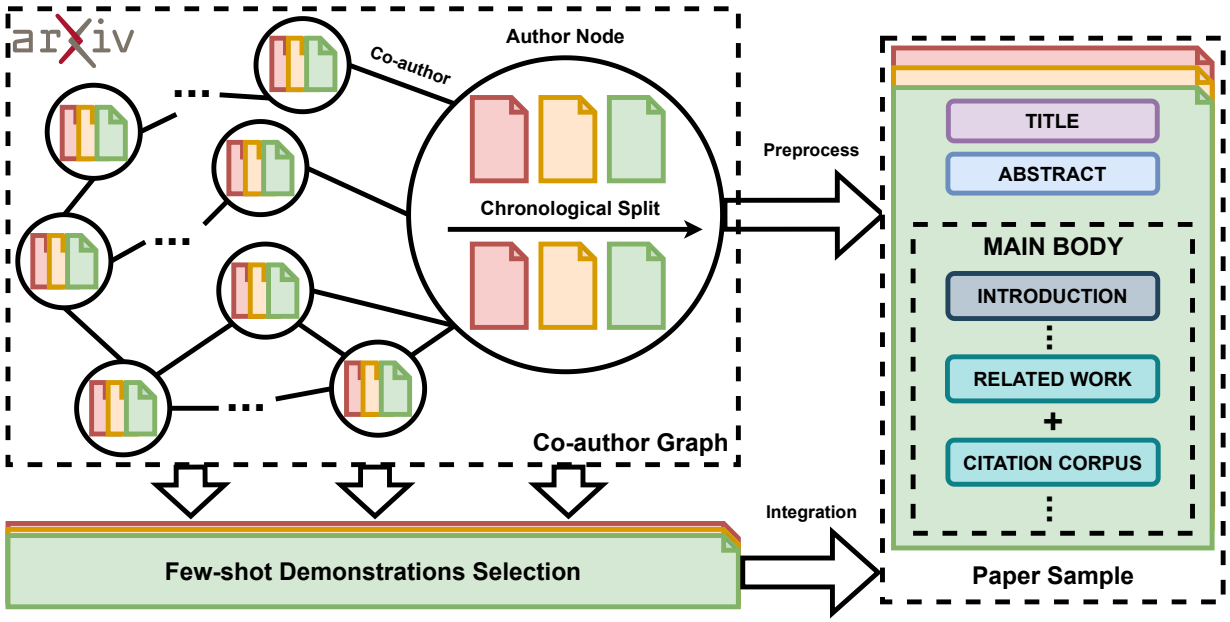

Figure 1: **AcademicEval Benchmark.** We construct a co-author graph via arXiv and conduct a chronological split on all paper samples (training, validation, and test samples are represented by red, orange, and green, respectively). Each paper sample is preprocessed into separate sections and can be integrated with few-shot demonstrations from co-author papers.

breadth-first search (BFS), where the features of each author node include the published first-author papers. By making the co-author graph the carrier of papers, we can form an interconnected whole of scattered articles, which provides valuable structural information to be exploited for our benchmark. Furthermore, we can enable efficient live updates on the co-author graph, which will be introduced in Section 3.3.

Academic Data Gathering and Preprocessing. After the co-author graph is collected, we remove authors who have not published independent first-author papers (*i.e.*, appear only as co-authors in the author list) and then prune it to obtain the maximum connected component. For each paper (*i.e.*, node features), we collect essential metadata via the arXiv API, including author information, publication timestamp, etc., and download the PDF file simultaneously, which further goes through a series of pipelines to split and extract the text of several sections in it. In detail, we leverage PyMuPDF[3] to detect section headings (*e.g.*, "Introduction") and extract the paper content by sections. Especially, for the "Related Work" section, we extract each cited paper's abstract and title via arXiv API to form an additional citation corpus. All these processed data constitute the node feature of each author node. We will further describe in Section 3.2 how to use these data to design long-context academic writing tasks.

3.2 Benchmarking LLMs over Long-context Generation Tasks with Hierarchical Abstraction

Task Description Employing machine learning approaches to automate academic writing has always been a research hotspot with significant practical application value (Chen et al., 2022; 2021). Therefore, inspired by the leave-one-out validation, we introduce four academic writing tasks with ultra-long context to evaluate the generation capability of LLMs under different abstraction levels, as shown below:

- Title Writing. This task takes a paper's main body and abstract, along with a specific task prompt as inputs, and then asks LLMs to output a predicted title.

[3]https://github.com/pymupdf/PyMuPDF

- Abstract Writing. Similar to the above, this task takes a paper's main body (with the "Conclusion" section removed) and title, along with a specific task prompt as inputs, and then asks LLMs to output a predicted abstract.

- Introduction Writing. This task takes a paper's main body (with the "Introduction" section removed), title, and abstract, along with a specific task prompt as inputs, and then asks LLMs to output a predicted introduction.

- Related Work Writing. This task takes a paper's main body (with the "Related Work" section removed), title, abstract, and citation corpus (introduced in Section 3.1), along with a specific task prompt as inputs, and then asks LLMs to output a predicted related work.

Based on the above task descriptions, we can generate four basic benchmark settings with different abstraction levels, namely Title-10K, Abs-9K, Intro-8K and Related-34K, with suffixes indicating their input context length[4]. Intuitively, the paper content itself can be considered as a kind of original, expert-curated, and high-quality labeled data without manual annotation. Therefore, for evaluation, we directly adopt the corresponding paper section as the ground truth for each benchmark setting, minimizing human labeling efforts.

Integration of Few-shot Demonstrations Given the rigid context length of current long-context LLM benchmarks and the general effectiveness of in-context learning in LLMs (Dong et al., 2022; Wei et al., 2022a;b; Kojima et al., 2022), we propose to integrate long few-shot demonstrations to enable flexible and scalable context length, and we have two selection options for each sample in the above four basic benchmark settings: *(1) Randomly select papers under the same category.* According to the paper categories provided by the arXiv API, we can randomly select several non-duplicate papers under the same category. *(2) Randomly Select co-author papers.* The motivation is straightforward: the similarity of research directions between co-author papers is more fine-grained. Thanks to the co-author graph, it is convenient to obtain co-author papers of each original paper sample. These selected papers serve as few-shot demonstrations and are utilized as input-output pairs to enrich the input context of the original samples, providing potentially insightful and relevant content while enabling flexible and scalable context length.

Consequently, we have completed the construction of benchmark settings, and the data statistics in the initial collection round are shown in Table 2.

Data Statistics As shown in Table 2, AcademicEval has four academic writing tasks with hierarchical abstraction levels, and each task features four settings with diverse input context lengths, some of which are obtained by integrating few-shot demonstrations. For instance, each sample in Title-10K consists of a single paper sample. Title-30K and Title-31K-G are obtained by integrating with two few-shot demonstrations from random papers and co-author papers, respectively, while Title-50K-M is obtained by using both of the above integration options. Actually, we can scale context length by increasing the number of few-shot demonstrations to provide more informative references, enhancing task performance.

Furthermore, we present the text compression rate (defined as the number of input tokens divided by the number of output tokens) for each benchmark setting in Table 2 to illustrate the diverse abstraction levels in AcademicEval. Across the four tasks, a higher compression rate means a higher level of text abstraction in this task. Among several settings within each task, a higher compression rate makes it tougher to exploit information holistically but more likely to produce better outputs (since more references are integrated). These different tasks and settings increase the diversity of the AcademicEval benchmark.

As for data splitting, we perform a chronological split in AcademicEval, which means that the test set always contains the latest papers collected in each collection round, ensuring no label leakage. Note that Table 2 shows only the data collected in the initial round, which will be updated periodically as described in the next section.

[4]We use BERT (Devlin et al., 2018) tokenizer by default to count the number of input tokens (output tokens are not included).

Table 2: **Data Statistics of AcademicEval (Initial Round).** It includes 4 writing tasks and provides four settings of different context length for each task. For each setting, we list their Comp. Rate, Samples of Each, Chronological Split, and Timespan of Test Data.

Setting	Comp. Rate (In-Len. / Out-Len.)	#Samples of Each.	Chronological Split (Train-Val-Test)	Timespan of Test Data
Title Writing				
Title-10K	587	5098	72%-19%-9%	2024.06-2024.07
Title-30K	1773			
Title-31K-G	1807			
Title-50K-M	2968			
Abstract Writing				
Abs-9K	36	5098	72%-19%-9%	2024.06-2024.07
Abs-28K	108			
Abs-29K-G	112			
Abs-48K-M	185			
Introduction Writing				
Intro-8K	6	4665	71%-20%-9%	2024.06-2024.07
Intro-28K	21			
Intro-28K-G	22			
Intro-48K-M	37			
Related Work Writing				
Related-34K	34	2240	72%-20%-8%	2024.06-2024.07
Related-53K	53			
Related-53K-G	53			
Related-72K-M	72			

Note: We use the BERT tokenizer by default to count the number of tokens.

3.3 Live Evaluation with Periodic Data Updates on the Co-author Graph

The daily updates of arXiv provide the basis for the live evaluation of AcademicEval: we can periodically update the benchmark with the latest papers on arXiv. By setting a reasonable update cycle (*e.g.*, monthly or quarterly), we can ensure that the data in the benchmark is not contaminated so that it can be used to evaluate LLMs fairly in a live manner. Therefore, we proposed an efficient incremental update procedure on the co-author graph:

(1) Node Update For each author on the co-author graph, check whether the author has a newly published first-author paper through the arXiv API. If so, add it to the corresponding node feature on the co-author graph.

(2) Node and Edge Update During the traversal of Node Update, each author's new co-authors are added to a candidate list, and the number of new papers (including first-author and non-first-author papers) when searching for the author is used as the priority of the co-authors (co-authors of active authors tend to be active as well, and we can efficiently collect the latest papers from active authors). Then, we use the prioritized candidate list to conduct BFS to update nodes and edges until a specific number of incremental update papers is met.

(3) Graph Pruning As the benchmark is updated, we will remove some outdated papers and inactive authors (defined as those who have not published new first-author or non-first-author papers for a long time) from the co-author graph.

In this way, the latest papers can be obtained sufficiently and efficiently while ensuring connectivity and a smaller graph size.

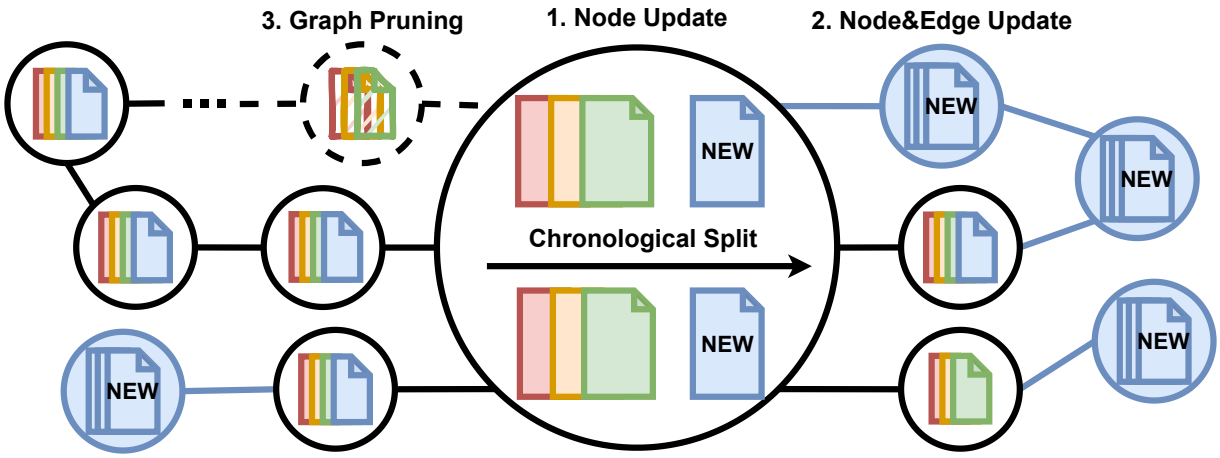

Figure 2: **Live Evaluation of AcademicEval Benchmark.** To support continual benchmarking, AcademicEval incrementally updates the co-author graph using daily arXiv data. The procedure includes: (1) **Node Update** – augmenting node features for authors with newly published first-author papers; (2) **Node and Edge Update** – identifying and prioritizing new co-authors via BFS to expand the graph with recent publications; and (3) **Graph Pruning** – removing outdated papers and inactive authors to maintain graph connectivity and efficiency.

4 Experiments

4.1 Baselines

We adopt the following three types of baselines to conduct a holistic evaluation of AcademicEval.

Standard LLMs We choose Gemma Instruct (7B) (Team et al., 2024) and LLaMA-3 Chat (70B) (AI@Meta, 2024) as standard LLM baselines, each with a context length of 8K.

Long-context LLMs We choose Qwen 1.5 Chat (72B) (Bai et al., 2023a), Mixtral-8x7B Instruct (46.7B) (Jiang et al., 2024), and Nous Hermes 2 - Mixtral 8x7B-DPO (46.7B) (Teknium et al.) as long-context LLM baselines, each with a context length of 32K.

Retrieval-augmented language models (RALM) First, we consider two sparse retrievers: (1) **BM25** (Robertson et al., 2009): This is a widely used retrieval model that ranks documents based on the frequency of query terms in each document. (2) **TF-IDF** (Ramos et al., 2003): It scores documents by multiplying the term frequency of each query term by the inverse document frequency. Second, we also consider three dense retrievers: (3) **DPR** (Karpukhin et al., 2020): It uses a bi-encoder to retrieve relevant documents based on dense embeddings. (4) **Contriever** (Izacard et al., 2021): It leverages unsupervised contrastive learning to learn high-quality dense representations. (5) **Dragon** (Lin et al., 2023): It enhances retriever training by employing data augmentation, including query and label augmentation.

4.2 Settings

API Access. In this paper, we conduct a comprehensive evaluation over AcademicEval benchmark using the LLM API provided by together.ai[5]. For each API call, we fix the temperature parameter to 0 (*i.e.*, greedy decoding).

[5]https://www.together.ai/

Input Truncation. By default, we use a BERT tokenizer to calculate the number of input tokens for AcademicEval. However, since the tokenizer of each LLM is usually different, it will cause some inputs to exceed the context length limit of the LLM. Therefore, for the evaluation of each LLM, we additionally download its tokenizer configuration file from the official website at Hugging Face, which is utilized to ensure correct and accurate truncation of input tokens.

Refinement of LLM Responses. For the Title Writing task, the responses of LLMs are relatively short. If the response contains some extra redundant information, it will have a greater impact on the evaluation metric score (although we have given LLM instructions not to generate irrelevant information). Therefore, for the Title Writing task, we additionally refine the LLM responses, for example, removing irrelevant information such as "here is the title". For other tasks, since LLM's responses are relatively long, occasional small amounts of irrelevant information will not have a significant impact on the evaluation, so we do not perform any refinement on LLM's responses in this case.

Details of the implementation of RALM. We use the inputs of AcademicEval as the external corpus of RALM (such as Target Content and Reference Content introduced in Section E). For text split, we use the RecursiveCharacterTextSplitter from LangChain[6] and set chunk size and chunk overlap to 512 and 64, respectively. For each retrieval, we recall up to 12 text chunks (limited by the context length of standard LLMs) based on text similarity (semantic similarity based on inner product for dense retrievers or similarity based on word frequency for sparse retrievers).

4.3 Evaluation Metrics

For evaluation metrics, we adopt (1) **BERTScore**[7] (Zhang et al., 2019): This metric leverages BERT-based embedding to measure semantic similarity between predicted and reference texts. (2) **RougeL** (Lin, 2004): This metric evaluates the longest common subsequence between the generated and reference texts, providing a measure of similarity in terms of sequential matching. For both metrics, higher scores indicate a better match between the predicted and the reference text.

4.4 Main Result Analysis

We conduct comprehensive experiments on the four academic writing tasks, and the results w.r.t. BERTScore and RougeL are presented in Table 3 and 4, respectively. Note that we do not conduct experiments on -M settings because its context length is too long for most of our selected baselines.

Diverse Task Difficulties and Abstractions The four tasks we proposed are designed to challenge LLMs over long-context generation tasks with different abstraction levels. From Table 3 and 4, we can clearly observe that it provides different difficulties for LLMs to perform well from Title Writing to Related Work Writing tasks, and the results of all baselines on these four tasks have a relatively obvious trend. For example, the Title Writing task tends to have a higher score than the Abstract Writing task, which may indicate that the Title Writing task is easier than the Abstract Writing task. Since a title only has a few words, LLMs only need to generate a roughly related theme to achieve a high semantic similarity, while an abstract requires a more detailed description generated to achieve it.

Baseline Performance Comparison Among different baselines, RALM with LLaMA frequently delivers the highest scores across various tasks and context lengths, with only a context length of 8K. Standard LLMs also achieve competitive performance, which is slightly inferior to long-context LLMs. This exposes the shortcomings of long-context LLMs' generation capabilities, which is well revealed by AcademicEval. Among long-context LLMs, Hermes performs best overall, but is still slightly inferior to RALM with LLaMA. This shows that although the current long-context LLMs have a longer context window size, they still have great deficiencies in processing long text information. In contrast, RALM-based methods generally outperform other baselines. This is primarily due to the retrieval mechanisms of RALM, which retrieves and processes information in a few relevant shorter chunks, enabling it to focus on key information.

[6]https://www.langchain.com/

[7]We use deberta-xlarge-mnli (He et al., 2021) instead of the default roberta-large (Liu et al., 2019) as the backbone model to have the best correlation with human evaluation.

Table 3: **Main Results on AcademicEval w.r.t. BERTScore.**

Models	Standard LLMs		Long-context LLMs			RALM	
	Gemma	LLaMA	Qwen	Mixtral	Hermes	Gemma†	LLaMA†
#Params.	7B	70B	72B	8x7B	8x7B	7B	70B
Context Length	8K	8K	32K	32K	32K	8K	8K
Setting: Title Writing							
Title-10K	66.1	74.1	73.9	73.4	**74.2**	65.8	73.9
Title-30K	-	-	73.0	72.9	73.4	65.7	**73.9**
Title-31K-G	-	-	72.8	72.8	73.3	65.7	**73.8**
Setting: Abstract Writing							
Abs-9K	59.9	62.4	**62.5**	61.4	62.2	60.3	61.5
Abs-28K	-	-	61.3	61.2	**62.6**	60.1	61.4
Abs-29K-G	-	-	61.3	61.4	**62.5**	60.2	61.3
Setting: Introduction Writing							
Intro-8K	54.8	**55.8**	55.4	54.6	55.2	55.0	55.2
Intro-28K	-	-	54.8	54.0	54.8	55.0	**55.2**
Intro-28K-G	-	-	54.9	54.1	54.7	55.0	**55.3**
Setting: Related Work Writing							
Related-34K	52.0	56.2	**58.5**	55.3	57.8	52.4	54.7
Related-53K	-	-	-	-	-	52.4	**54.7**
Related-53K-G	-	-	-	-	-	52.4	**54.8**

Bold indicates the highest score in each row.
† denotes augmentation with a retriever (Default: Contriever).
"-" means that the context length is too long to be fed into LLMs.

Impact of Context Length The impact of context length on performance is evident across all task settings and both metrics, with baselines generally performing worse as the context length increases. For example, the Title Writing task shows a noticeable drop in scores as the context length extends from 10K to 31K tokens. This trend is also apparent in Abstract Writing and Introduction Writing, where longer contexts correlate with decreased model performance, showing that our benchmark challenges LLMs in effectively processing ultra-long inputs.

Impact of Few-shot Demonstrations From Table 3 and 4, we can observe that the integration of few-shot demonstrations generally degrades the performance of baselines, except for a few tasks where the results are slightly improved. This shows that current LLMs cannot exploit long few-shot demonstrations to benefit the target tasks well, emphasizing the importance of evaluating long in-context learning in LLM benchmarks. In addition, we can also find that few-shot demonstrations from co-author papers generally have a more positive impact on task performance than randomly selected ones.

4.5 Additional analysis on RALM

We conduct extensive experiments on RALM on the Abs-9K setting using standard LLMs Gemma Instruct (7B) and LLaMA-3 Chat (70B), and the results are presented in Figure 3. We can find that the performance of dense retrievers consistently outperforms sparse retrievers, among which contriever achieves the best results. This is because the summary generation task emphasizes semantic similarity, which can be well measured by the similarity of dense embeddings. However, the sparse retrievers perform text chunk recall based on sparse embeddings, and the results are significantly worse than those of the dense retrievers.

Table 4: **Main Results on AcademicEval w.r.t. RougeL.**

Models	Standard LLMs		Long-context LLMs			RALM	
	Gemma	LLaMA	Qwen	Mixtral	Hermes	Gemma†	LLaMA†
#Params.	7B	70B	72B	8x7B	8x7B	7B	70B
Context Length	8K	8K	32K	32K	32K	8K	8K
Setting: Title Writing							
TITLE-10K	44.5	47.1	44.2	45.2	46.2	42.7	**47.3**
TITLE-30K	-	-	44.5	44.6	45.9	42.6	**47.3**
TITLE-31K-G	-	-	44.2	44.4	45.3	42.5	**47.0**
Setting: Abstract Writing							
ABS-9K	22.4	25.0	24.3	24.1	**26.1**	23.4	24.2
ABS-28K	-	-	23.3	24.7	**26.6**	23.1	24.1
ABS-29K-G	-	-	23.3	24.9	**26.6**	23.2	24.0
Setting: Introduction Writing							
INTRO-8K	14.9	**18.1**	16.2	17.2	17.8	15.4	17.9
INTRO-28K	-	-	16.3	17.5	17.5	15.3	**17.8**
INTRO-28K-G	-	-	16.3	17.5	17.5	15.4	**17.8**
Setting: Related Work Writing							
RELATED-34K	13.5	14.9	**16.0**	13.4	15.1	14.1	15.3
RELATED-53K	-	-	-	-	-	14.0	**15.3**
RELATED-53K-G	-	-	-	-	-	14.0	**15.2**

Bold indicates the highest score in each row.
† denotes augmentation with a retriever (Default: Contriever).
"-" means that the context length is too long to be fed into LLMs.

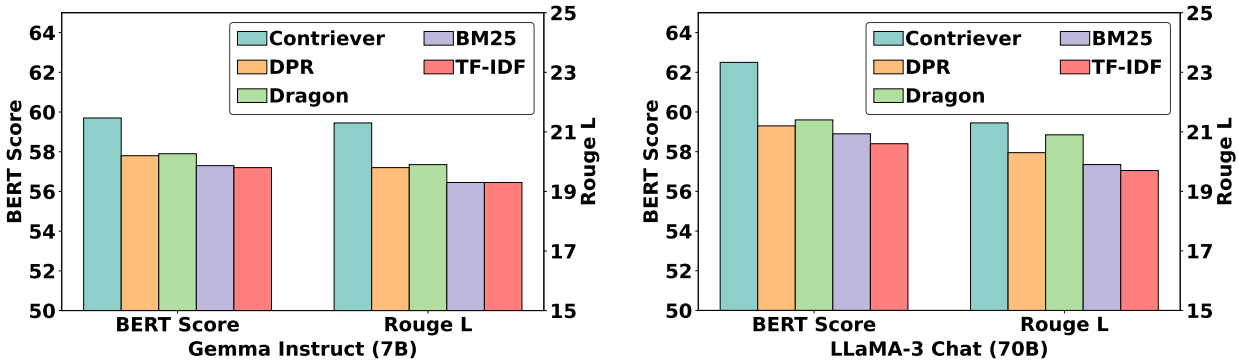

Figure 3: **Analysis of RALM on Abs-9K.** The left figure shows results with Gemma Instruct (7B), while the right one shows results with LLaMA-3 Chat (70B).

4.6 Additional analysis on LLM-as-a-Judge

To complement the automatic metrics used in the main evaluation, we introduce an *LLM-as-a-Judge* framework to provide a more comprehensive assessment of the generated outputs.

Table 5: **Additional results on AcademicEval w.r.t. LLM-as-a-Judge win rate (%).**

Models	Standard LLMs		Long-context LLMs			RALM	
	Gemma	LLaMA	Qwen	Mixtral	Hermes	Gemma†	LLaMA†
#Params.	7B	70B	72B	8x7B	8x7B	7B	70B
Context Length	8K	8K	32K	32K	32K	8K	8K
Setting: Title Writing							
Title-10K	45.7	42.7	63.2	43.1	**72.0**	50.0	43.9
Title-30K	-	-	54.5	45.6	**62.5**	47.5	44.9
Title-31K-G	-	-	52.4	**62.7**	45.4	47.7	43.7
Setting: Abstract Writing							
Abs-9K	12.0	55.5	**77.0**	70.0	61.1	14.3	43.2
Abs-28K	-	-	**72.7**	66.1	41.2	12.7	42.0
Abs-29K-G	-	-	**71.0**	65.9	40.7	12.0	43.9
Setting: Introduction Writing							
Intro-8K	34.6	63.2	**79.3**	61.5	58.0	48.8	64.1
Intro-28K	-	-	**70.3**	60.1	56.9	46.5	62.9
Intro-28K-G	-	-	**70.9**	61.9	59.3	48.2	63.9
Setting: Related Work Writing							
Related-34K	55.9	**91.9**	91.2	65.6	88.6	71.3	89.8
Related-53K	-	-	-	-	-	72.5	**90.7**
Related-53K-G	-	-	-	-	-	71.7	**90.2**

Bold indicates the highest score in each row.
† denotes augmentation with a retriever (Default: Contriever).
"-" means that the context length is too long to be fed into LLMs.

Evaluation setup. We adopt the open-source `Mixtral-8x22B-Instruct-v0.1` model as the evaluation agent. This choice is motivated by two considerations: (1) it enables cost-efficient large-scale evaluation given the limited computational budget; and (2) it ensures full reproducibility for the research community. For each generated output, the LLM judge is prompted to compare it with the corresponding ground truth across five dimensions: (i) **Novelty** — the degree to which the content introduces new and meaningful ideas; (ii) **Feasibility** — the plausibility and practicality of the described methods or claims; (iii) **Consistency** — the internal logical coherence of the output; (iv) **Factuality** — the correctness of factual statements; and (v) **Academic Style** — the alignment with conventions of scholarly writing. The model then provides an overall preference decision between the generated output and the ground truth, which is recorded as a win rate for the generated output. We report results for the overall preference in Table 5.

Findings. The LLM-as-a-Judge results reveal patterns that differ from those observed with BERTScore and ROUGE-L. In the automatic metrics, RALM variants generally achieve the highest scores across most settings, whereas in the LLM-as-a-Judge evaluation, this dominance is less consistent. This divergence highlights that while RALM excels in surface-level semantic similarity, its outputs are not always preferred when broader qualitative aspects are considered.

Notably, in the Related Work generation setting, RALM models still achieve competitive or even top win rates in the LLM-as-a-Judge evaluation. This suggests that retrieval-augmented generation is particularly beneficial for this task, as the retrieved content often contains relevant citations, prior studies, and domain-specific terminology that align well with what human evaluators (or LLM judges) consider high-quality

related work sections. In contrast, for tasks such as title or abstract generation, where brevity, originality, and style play a larger role, non-retrieval long-context models sometimes receive higher preference scores.

5 Conclusion

In this paper, we propose AcademicEval, a live long-context LLM benchmark for evaluating long-context generation tasks with hierarchical abstraction levels. AcademicEval adopts arXiv as the data source and introduces several long-context academic writing tasks without manual annotation since the papers on arXiv can be regarded as original, high-quality, and expert-curated labels. Moreover, we integrate few-shot demonstrations from a collected co-author graph to make the context length of our benchmark flexible and scalable. An efficient live evaluation is also designed to make AcademicEval immune to the label leakage issue and move toward a more fair evaluation. In the experiments, we conduct a comprehensive analysis on AcademicEval using several LLM baselines, and the results show that AcademicEval is a challenging long-context LLM benchmark. Insightful findings are also elucidated for potentially strengthening the long-context modeling capabilities of LLMs and inspiring future long-context LLM benchmarks to evaluate LLMs more flexibly and holistically.

References

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

AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Chenxin An, Shansan Gong, Ming Zhong, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. L-eval: Instituting standardized evaluation for long context language models. *arXiv preprint arXiv:2307.11088*, 2023.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023a.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*, 2023b.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. Tldr: Extreme summarization of scientific documents. *arXiv preprint arXiv:2004.15011*, 2020.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17754–17762, 2024.

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. Association for Computational Linguistics, 2021.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. Target-aware abstractive related work generation with contrastive learning. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 373–383, 2022.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F Yang, and Anton Tsitsulin. Don't forget to connect! improving rag with graph-based reranking. *arXiv preprint arXiv:2405.18414*, 2024.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. *arXiv preprint arXiv:2309.13345*, 2023.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=XPZIaotutsD`.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*, 2021.

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

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*, 2023.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in nlp, 2021.

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.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*, 2021.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*, 2023.

Jiaxi Li, Xingxing Zhang, Xun Wang, Xiaolong Huang, Li Dong, Liang Wang, Si-Qing Chen, Wei Lu, and Furu Wei. Wildlong: Synthesizing realistic long-context instruction data at scale. *arXiv preprint arXiv:2502.16684*, 2025.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*, 2023.

Marina Litvak and Natalia Vanetik. Query-based summarization using mdl principle. In *Proceedings of the multiling 2017 workshop on summarization and summary evaluation across source types and genres*, pp. 22–31, 2017.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL `http://arxiv.org/abs/1907.11692`.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking, 2021.

Rui Meng, Khushboo Thaker, Lei Zhang, Yue Dong, Xingdi Yuan, Tong Wang, and Daqing He. Bringing structure into summaries: a faceted summarization dataset for long scientific documents. *arXiv preprint arXiv:2106.00130*, 2021.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.

Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pp. 29–48. Citeseer, 2003.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

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

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*, 2023.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*, 2023.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. Zeroscrolls: A zero-shot benchmark for long text understanding. *arXiv preprint arXiv:2305.14196*, 2023.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Teknium, theemozilla, karan4d, and art huemin. Nous hermes 2 mixtral 8x7b dpo. URL `[https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO](https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO)`.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R Bowman. Squality: Building a long-document summarization dataset the hard way. *arXiv preprint arXiv:2205.11465*, 2022.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, et al. Autosurvey: Large language models can automatically write surveys. *Advances in neural information processing systems*, 37:115119–115145, 2024.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022b.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. *arXiv preprint arXiv:2411.00816*, 2024.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N. Ioannidis, Karthik Subbian, James Zou, and Jure Leskovec. Stark: Benchmarking llm retrieval on textual and relational knowledge bases, 2024.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*, 2023.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. Benchmarking benchmark leakage in large language models, 2024.

Wentao Ye, Jiaqi Hu, Liyao Li, Haobo Wang, Gang Chen, and Junbo Zhao. Data contamination calibration for black-box llms, 2024.

Haofei Yu, Zhaochen Hong, Zirui Cheng, Kunlun Zhu, Keyang Xuan, Jinwei Yao, Tao Feng, and Jiaxuan You. Researchtown: Simulator of human research community. *arXiv preprint arXiv:2412.17767*, 2024.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*, 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, et al. Extending long context evaluation beyond 100k tokens. *arXiv preprint arXiv:2402.13718*, 2024.

Chenghao Zhu, Nuo Chen, Yufei Gao, and Benyou Wang. Evaluating llms at evaluating temporal generalization, 2024a.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. In *The Twelfth International Conference on Learning Representations*, 2024b. URL `https://openreview.net/forum?id=gjfOL9z5Xr`.

A API cost

We adopt LLM API provided by together.ai[8] to conduct experiments in this paper. API costs mainly come from evaluating the test set of AcademicEval, which are estimated to be around $300.

B Additional Related Work

ResearchTown (Yu et al., 2024) is a recent multi-agent simulation framework that models the dynamics of a research community through message-passing on an agent–data graph, simulating tasks such as paper and review writing. Its focus lies in collaborative activity simulation and the realism of generated outputs under controlled settings. In contrast, AcademicEval is a live long-context LLM benchmark grounded in real-world academic papers, targeting hierarchical generation tasks (title, abstract, introduction, related work) under evolving, leakage-resistant conditions.

While both works leverage graph structures, their purposes differ: in ResearchTown, the graph enables simulation of researcher–paper interactions, whereas in AcademicEval, the co-author graph is used to retrieve high-quality few-shot demonstrations, support scalable context lengths, and enable periodic dataset updates. Furthermore, evaluation methodologies diverge: ResearchTown operates in a synthetic environment with similarity-based metrics on simulated outputs, while AcademicEval evaluates on authentic, expert-authored content using established automatic metrics (BERTScore, ROUGE-L) to stress-test long-context modeling, retrieval integration, and robustness.

These differences in goals, data design, and evaluation make the two approaches complementary: ResearchTown advances research community simulation, whereas AcademicEval addresses the need for a real-world, graph-enabled long-context benchmark.

Another work that shares some similarity with AcademicEval is WildLong (Li et al., 2025), which introduces a scalable framework for synthesizing realistic long-context instruction data. It extracts meta-information from real user queries, builds co-occurrence graphs, and employs adaptive generation to create 150K instruction–response pairs that support complex reasoning tasks such as cross-document comparison and aggregation.

While WildLong focuses on data synthesis for long-context instruction tuning—leveraging graph-based methods to generate large-scale, realistic multi-document instruction-response pairs, our work AcademicEval instead builds a real-world benchmark for long-context evaluation using authentic academic papers from arXiv. WildLong aims to improve training data diversity and enable long-context finetuning, whereas our goal is to provide a live, leakage-resistant, and automatically updated evaluation framework that measures LLM performance on hierarchical academic writing tasks (title, abstract, introduction, related work). In this sense, WildLong and AcademicEval are also complementary: one addresses the supply of long-context instruction training data, the other the rigorous evaluation of long-context reasoning abilities.

C Limitation and Future Improvement

AcademicEval is a live benchmark without label leakage, which leverages co-author papers from a collected co-author graph as few-shot demonstrations to make the context length flexible and scalable. AcademicEval adopts arXiv as its data source without the need for manual labeling, and the content of the papers on it can naturally serve as high-quality and expert-curated annotations.

However, AcademicEval still has some limitations:

- **Task Diversity.** AcademicEval currently has only four academic writing tasks, which limits the task diversity.

[8]https://www.together.ai/

- **Independent Evaluation of the Paper Section.** In AcademicEval, we independently evaluate the section content extracted from a paper, which may lack a comprehensive evaluation of the paper as a whole.

- **Popularity Bias.** AcademicEval first collects a co-author graph from arXiv, which contains a subset of all papers on arXiv. Therefore, the collected papers may have some popularity bias. For example, most of the papers may come from a few active authors, which will cause bias in the evaluation.

Based on the above limitations, our future improvements will include:

- **Introduce More Data Source.** The goal of AcademicEval is to make context length flexible and scalable by using few-shot demonstrations and high-quality labels without manual annotation, so papers on arXiv are a more suitable data source. We will consider adding other websites as data sources in the future, such as some question-answering websites (Stack Overflow or Reddit, etc.). In this case, we can use the best answers as high-quality labels. By modeling the citation relationship between posts into a graph, we can also obtain few-shot demonstrations to enrich the context length.

- **K-fold Cross-validation.** We can use k-fold cross-validation for a paper, that is, leaving a section (or fold) as the label, the remaining sections as inputs, and finally calculating the average of all leave-one-out evaluation scores.

- **Eliminate Popularity Bias.** We will perform probabilistic sampling on papers when collecting the co-author graph and give a lower sampling probability to active authors to alleviate the impact of popularity bias.

D Social Impact

The proposed benchmark AcademicEval will promote the academic community's exploration of using LLMs to automate academic writing tasks (Wang et al., 2024; Lu et al., 2024; Weng et al., 2024). Here are some key points highlighting its significance:

- **Efficiency and Productivity.** LLMs can drastically reduce the time and effort required for various academic writing tasks. These tasks include drafting papers, writing literature reviews, summarizing research articles, and generating bibliographies. By automating these processes, researchers can focus more on high-level thinking, experimentation, and analysis.

- **Enhanced Writing Quality.** LLMs have the ability to produce coherent and grammatically correct text, which can improve the overall quality of academic writing. They can assist in refining arguments, improving clarity, and ensuring consistency in style and tone, which is particularly useful for non-native English speakers.

- **Support for Multidisciplinary Research.** Given their training on diverse topics, LLMs can assist researchers in exploring interdisciplinary approaches by providing information and generating content across various fields of study. This can foster innovation and encourage collaboration between different academic disciplines.

E LLM Prompts

In this section, we present the LLM prompts used in the experiments, including Title Writing, Abstract Writing, Introduction Writing, and Related Work Writing. For each academic writing task, we provide prompts for standard LLMs, long-context LLMs, and RALM (RALM additionally includes the retrieval query).

E.1 LLM Prompts for Title Writing

Prompt for Standard and Long-context LLMs on TITLE-10K

Please read the following Target Content carefully and summarize the Target Content as required.
Target Content: {CONTENT}
Target Content Abstract: {ABSTRACT}
Please craft a title highly summarizing the main theme from the above provided Target Content. The title should be of appropriate length (strictly limited to about 10 words). The title should also include and highlight the core and most critical theme of the Target Content, ignoring minor and redundant information. Please ensure that the title captures the essence of the Target Content in a clear and concise manner. Please output the title directly without including other redundant or irrelevant text.

Prompt for Standard and Long-context LLMs on TITLE-30K and TITLE-31K-G

Please read the following Reference Content and Output carefully and summarize the Target Content as required.
Reference Content 0: {CONTENT_0}
Reference Abstract 0: {ABSTRACT_0}
Reference Output 0: {OUTPUT_0}
...
Target Content: {CONTENT}
Target Content Abstract: {ABSTRACT}
Please craft a title highly summarizing the main theme from the above provided Target Content. The Reference Content and Output provide some demonstrations, which may also contain some information that is potentially related to the Target Content. You can refer to the input and output text forms of the Reference Content and Output to assist in summarizing the Target Content and try to explore and use the information that is potentially related to the Target Content contained in the Reference Content and Output. The title should be of appropriate length (strictly limited to about 10 words). The title should also include and highlight the core and most critical theme of the Target Content, ignoring minor and redundant information. Please ensure that the title captures the essence of the Target Content in a clear and concise manner. Please output the title directly without including other redundant or irrelevant text.

Prompt for RALM on TITLE-10K, TITLE-30K, and TITLE-31K-G

Please read the following Target Content carefully and summarize the Target Content as required.
Target Content 0: {CONTENT_0}
Target Content 1: {CONTENT_1}
...
Target Content Abstract: {ABSTRACT}
Please craft a title highly summarizing the main theme from all the above provided Target Contents. The title should be of appropriate length (strictly limited to about 10 words). The title should also include and highlight the core and most critical theme of the Target Contents, ignoring minor and redundant information. Please ensure that the title captures the essence of the Target Contents in a clear and concise manner. Please output the title directly without including other redundant or irrelevant text.

Retrieval Query for RALM on Title-10K, Title-30K, and Title-31K-G

Please craft a title highly summarizing the main theme of the provided text. The abstract of the text is: {ABSTRACT}

E.2 LLM Prompts for Abstract Writing

Prompt for Standard and Long-context LLMs on Abs-9K

Please read the following Target Content carefully and summarize the Target Content as required.
Target Content: {CONTENT}
Target Content Title: {TITLE}
Please craft an abstract summarizing the key points from the above provided Target Content. The abstract should be of appropriate length (around 200 words) and include the main theme, significant findings or arguments, and conclusions of the Target Content. Please ensure that the abstract captures the essence of the Target Content in a clear, coherent, and succinct manner. Please output the abstract directly without including other redundant or irrelevant text.

Prompt for Standard and Long-context LLMs on Abs-28K and Abs-29K-G

Please read the following Reference Content and Output carefully and summarize the Target Content as required.
Reference Content 0: {CONTENT_0}
Reference Title 0: {TITLE_0}
Reference Output 0: {OUTPUT_0}
...
Target Content: {CONTENT}
Target Content Title: {TITLE}
Please craft an abstract summarizing the key points from the above provided Target Content. The Reference Content and Output provide some demonstrations, which may also contain some information that is potentially related to the Target Content. You can refer to the input and output text forms of the Reference Content and Output to assist in summarizing the Target Content and try to explore and use the information that is potentially related to the Target Content contained in the Reference Content and Output. The abstract should be of appropriate length (around 200 words) and include the main theme, significant findings or arguments, and conclusions of the Target Content. Please ensure that the abstract captures the essence of the Target Content in a clear, coherent, and succinct manner. Please output the abstract directly without including other redundant or irrelevant text.

Prompt for RALM on Abs-10K, Abs-30K, and Abs-31K-G

Please read the following Target Content carefully and summarize the Target Content as required.
Target Content 0: {CONTENT_0}
Target Content 1: {CONTENT_1}
...
Target Content Title: {TITLE}
Please craft an abstract summarizing the key points from all the above provided Target Contents. The abstract should be of appropriate length (around 200 words) and include the main theme, significant findings or arguments, and conclusions of the Target Contents. Please ensure that the abstract captures the essence of the Target Contents in a clear, coherent, and succinct manner. Please output the abstract directly without including other redundant or irrelevant text.

Retrieval Query for RALM on Abs-10K, Abs-30K, and Abs-31K-G

Please craft an abstract summarizing the key points of the provided text. The title of the text is: {TITLE}

E.3 LLM Prompts for Introduction Writing

Prompt for Standard and Long-context LLMs on Intro-8K

Please read the following Target Content carefully and summarize the Target Content as required.
Target Content: {CONTENT}
Target Content Title: {TITLE}
Target Content Abstract: {ABSTRACT}
Please craft an introduction summarizing the key points from the above provided Target Content. The introduction should be of appropriate length (about 1000 to 1500 words). The introduction should first describe the topic or main theme of the Target Content, then provide relevant background knowledge, and summarize the existing relevant research on this topic from the Target Content, point out their advantages and disadvantages, and highly summarize the specific research problem and problem statement targeted by the Target Content. Next, describe in detail the core approach or insights proposed by the Target Content on this topic and include any necessary experimental results. Then, use about 3 short paragraphs (each paragraph is about 50 words) to highly summarize the approach or insights proposed in the Target Content, as well as the experimental results. Finally, briefly give an overview of the Target Content's structure. Please ensure that the introduction captures the essence of the Target Content in a clear, coherent, and succinct manner. Please output the introduction directly without including other redundant or irrelevant text.

Prompt for Standard and Long-context LLMs on Intro-28K and Intro-28K-G

Please read the following Reference Content and Output carefully and summarize the Target Content as required.
Reference Content 0: {CONTENT_0}
Reference Title 0: {TITLE_0}
Reference Abstract 0: {ABSTRACT_0}
Reference Output 0: {OUTPUT_0}
...
Target Content: {CONTENT}
Target Content Title: {TITLE}
Target Content Abstract: {ABSTRACT}
Please craft an introduction summarizing the key points from the above provided Target Content. The Reference Content and Output provide some demonstrations, which may also contain some information that is potentially related to the Target Content. You can refer to the input and output text forms of the Reference Content and Output to assist in summarizing the Target Content and try to explore and use the information that is potentially related to the Target Content contained in the Reference Content and Output. The introduction should be of appropriate length (about 1000 to 1500 words). The introduction should first describe the topic or main theme of the Target Content, then provide relevant background knowledge, and summarize the existing relevant research on this topic from the Target Content, point out their advantages and disadvantages, and highly summarize the specific research problem and problem statement targeted by the Target Content. Next, describe in detail the core approach or insights proposed by the Target Content on this topic and include any necessary experimental results. Then, use about 3 short paragraphs (each paragraph is about 50 words) to highly summarize the approach or insights proposed in the Target Content, as well as the experimental results. Finally, briefly give an overview of the Target Content's structure. Please ensure that the introduction captures the essence of the Target Content in a clear, coherent, and succinct manner. Please output the introduction directly without including other redundant or irrelevant text.

Prompt for RALM on Intro-8K, Intro-28K, and Intro-28K-G

Please read the following Target Content carefully and summarize the Target Content as required.
Target Content 0: {CONTENT_0}
Target Content 1: {CONTENT_1}
...
Target Content Title: {TITLE}
Target Content Abstract: {ABSTRACT}
Please craft an introduction summarizing the key points from all the above provided Target Contents. The introduction should be of appropriate length (about 1000 to 1500 words). The introduction should first describe the topic or main theme of the Target Contents, then provide relevant background knowledge, summarize the existing relevant research on this topic from the Target Contents, point out their advantages and disadvantages, and highly summarize the specific research problem and problem statement targeted by the Target Contents. Next, describe in detail the core approach or insights proposed by the Target Contents on this topic and include any necessary experimental results. Then, use about 3 short paragraphs (each paragraph is about 50 words) to highly summarize the approach or insights proposed in the Target Contents, as well as the experimental results. Finally, briefly give an overview of the Target Contents' structure. Please ensure that the introduction captures the essence of the Target Contents in a clear, coherent, and succinct manner. Please output the introduction directly without including other redundant or irrelevant text.

Retrieval Query for RALM on Intro-8K, Intro-28K, and Intro-28K-G

Please craft an introduction summarizing the main theme of the provided text (including background knowledge, advantages and disadvantages of existing research and challenges, the proposed approach, experimental results, etc.). The title of the text is {TITLE}. The abstract of the text is {ABSTRACT}.

E.4 LLM Prompts for Related Work Writing

Prompt for Standard and Long-context LLMs on Related-34K

Please read the following Target Content and Target Citations carefully and summarize the Target Citations according to the topic of the Target Content as required.
Target Citation 0:
Target Citation Title: {C_TITLE_0}
Target Citation Abstract: {C_ABSTRACT_0}
...
Target Content: {CONTENT}
Target Content Title: {TITLE}
Target Content Abstract: {ABSTRACT}
Given the Target Content and its Abstract and Title, along with its Target Citations (including Target Citation Title and Abstract), please craft a related work summarizing the key points from the above provided Target Citations. There is no specific length requirement or limit for the entire related work (it is best to keep it around 500 to 1000 words), but each Target Citation that appears in the related work needs to be highly summarized in extremely concise and short sentences. You can refer to the topic or main theme described by the Target Content and its Abstract and Title to filter irrelevant information in the Target Citations and leverage relevant information. Furthermore, you can categorize the relevant Target Citations, briefly summarize the advantages and disadvantages of each categorization, and explain the advantages of the approach proposed in the Target Content. Please ensure that the related work captures all the relevant key points of the Target Citations in a clear, coherent, and succinct manner. Please output the related work directly without including other redundant or irrelevant text.

Prompt for Standard and Long-context LLMs on Related-53K and Related-53K-G

Please read the following Reference Content and Output carefully and summarize the Target Citations according to the topic of the Target Content as required.
Reference Content 0: {CONTENT_0}
Reference Title 0: {TITLE_0}
Reference Abstract 0: {ABSTRACT_0}
Reference Output 0: {OUTPUT_0}
...
Target Citation 0:
Target Citation Title: {C_TITLE_0}
Target Citation Abstract: {C_ABSTRACT_0}
...
Target Content: {CONTENT}
Target Content Title: {TITLE}
Target Content Abstract: {ABSTRACT}
Given the Target Content and its Abstract and Title, along with its Target Citations (including Target Citation Title and Abstract), please craft a related work summarizing the key points from the above provided Target Citations. The Reference Content and Output provide some demonstrations, which may also contain some information that is potentially related to the Target Content. You can refer to the input and output text forms of the Reference Content and Output to assist in summarizing the Target Citations and try to explore and use the information (e.g., related citations missing from the Target Citations) that is potentially related to the Target Content contained in the Reference Content and Output. There is no specific length requirement or limit for the entire related work (it is best to keep it around 500 to 1000 words), but each Target Citation that appears in the related work needs to be highly summarized in extremely concise and short sentences. You can refer to the topic or main theme described by the Target Content and its Abstract and Title to filter irrelevant information in the Target Citations and leverage relevant information. Furthermore, you can categorize the relevant Target Citations, briefly summarize the advantages and disadvantages of each categorization, and explain the advantages of the approach proposed in the Target Content. Please ensure that the related work captures all the relevant key points of the Target Citations in a clear, coherent, and succinct manner. Please output the related work directly without including other redundant or irrelevant text.

Prompt for RALM on Related-34K, Related-53K, and Related-53K-G

Please read the following Target Content and Target Citations carefully and summarize the Target Citations according to the topic of the Target Content as required.
Target Content 0: {CONTENT_0}
Target Content 1: {CONTENT_1}
...
Target Content Title: {TITLE}
Target Content Abstract: {ABSTRACT}
Given the Target Content Abstract and Title, please craft a related work summarizing the key points from all the above provided Target Contents. There is no specific length requirement or limit for the entire related work (it is best to keep it around 500 to 1000 words), but each Target Content that appears in the related work needs to be highly summarized in extremely concise and short sentences. You can refer to the topic or main theme described by the Target Content Abstract and Title to filter irrelevant information in the Target Contents and leverage relevant information. Furthermore, you can categorize the relevant Target Contents and briefly summarize the advantages and disadvantages of each categorization. Please ensure that the related work captures all the relevant key points of the Target Contents in a clear, coherent, and succinct manner. Please output the related work directly without including other redundant or irrelevant text.

Retrieval Query for RALM on Related-34K, Related-53K, and Related-53K-G

Please craft a related work summarizing all the relevant key points of the provided text. The title of the text is {TITLE}. The abstract of the text is {ABSTRACT}.

F Examples

In this section, we present some generation examples w.r.t. Title Writing, Abstract Writing, and Introduction Writing tasks.

F.1 Title Writing

Comparison between Generated Title and Ground Truth on Title Writing Task

Ground Truth:
When to Trust LLMs: Aligning Confidence with Response Quality

- **Qwen 1.5 Chat (72B)**

Generated Title (0-shot):
CONQORD: Reinforcement Learning for Confidence-Response Alignment in Large Language Models

Generated Title (2-shot demonstrations from random selection):
CONQORD: Enhancing LLM Confidence Alignment for Improved Trustworthiness and Retrieval

Generated Title (2-shot demonstrations from co-author papers):
CONQORD: Enhancing Confidence Calibration for Language Models via Reinforcement Learning

- **Mixtral-8x7B Instruct (46.7B)**

Generated Title (0-shot):
CONQORD: Aligning Confidence with Quality in Large Language Models

Generated Title (2-shot demonstrations from random selection):
CONQORD: Reinforcement Learning for Confidence-Quality Alignment in LLMs

Generated Title (2-shot demonstrations from co-author papers):
CONQORD: Aligning Confidence with Response Quality in LLMs

- **Nous Hermes 2 - Mixtral 8x7B-DPO (46.7B)**

Generated Title (0-shot):
Aligning Confidence with Response Quality in Large Language Models using Reinforcement Learning with Dual-Component Reward Function: CONQORD Approach

Generated Title (2-shot demonstrations from random selection):
Aligning Confidence with Response Quality in Large Language Models: Introducing CONQORD

Generated Title (2-shot demonstrations from co-author papers):
Confidence Alignment in Large Language Models via Reinforcement Learning: A CONQORD Approach for Trust

Comparison between Generated Title and Ground Truth on TITLE WRITING Task

Ground Truth:
Distillation for Multilingual Information Retrieval

- **Qwen 1.5 Chat (72B)**

Generated Title (0-shot):
Multilingual Translate-Distill: Enhancing Cross-Language Information Retrieval Models

Generated Title (2-shot demonstrations from random selection):
Multilingual Translate-Distill for Enhanced Cross-Language Information Retrieval

Generated Title (2-shot demonstrations from co-author papers):
Multilingual Translate-Distill for Effective Multilingual Information Retrieval

- **Mixtral-8x7B Instruct (46.7B)**

Generated Title (0-shot):
Multilingual Translate-Distill for Cross-Language Information Retrieval

Generated Title (2-shot demonstrations from random selection):
Multilingual Translate-Distill for Cross-Language Information Retrieval

Generated Title (2-shot demonstrations from co-author papers):
Multilingual Translate-Distill for MLIR: Extending Translate-Distill for Robust Language Mixing

- **Nous Hermes 2 - Mixtral 8x7B-DPO (46.7B)**

Generated Title (0-shot):
Multilingual Translate-Distill: Enhancing Multilingual Information Retrieval with Translation and Distillation

Generated Title (2-shot demonstrations from random selection):
Multilingual Translate-Distill: Efficient Training for Multilingual Information Retrieval

Generated Title (2-shot demonstrations from co-author papers):
Multilingual Translate-Distill: Training Dual-Encoder Models for Multilingual Information Retrieval

F.2 Abstract Writing

Comparison between Generated Abstract and Ground Truth on Abstract Writing Task

Ground Truth:
In recent years, neural ranking models (NRMs) have been shown to substantially outperform their lexical counterparts in text retrieval. In traditional search pipelines, a combination of features leads to well-defined behavior. However, as neural approaches become increasingly prevalent as the final scoring component of engines or as standalone systems, their robustness to malicious text and, more generally, semantic perturbation needs to be better understood. We posit that the transformer attention mechanism can induce exploitable defects through positional bias in search models, leading to an attack that could generalize beyond a single query or topic. We demonstrate such defects by showing that non-relevant text–such as promotional content–can be easily injected into a document without adversely affecting its position in search results. Unlike previous gradient-based attacks, we demonstrate these biases in a query-agnostic fashion. In doing so, without the knowledge of topicality, we can still reduce the negative effects of non-relevant content injection by controlling injection position. Our experiments are conducted with simulated on-topic promotional text automatically generated by prompting LLMs with topical context from target documents. We find that contextualization of a non-relevant text further reduces negative effects whilst likely circumventing existing content filtering mechanisms. In contrast, lexical models are found to be more resilient to such content injection attacks. We then investigate a simple yet effective compensation for the weaknesses of the NRMs in search, validating our hypotheses regarding transformer bias.

- **Qwen 1.5 Chat (72B)**

Generated Abstract (2-shot demonstrations from co-author papers):
The study investigates the vulnerability of Neural Ranking Models (NRMs) to adversarial attacks through the injection of promotional content into documents. The authors propose a framework for generating contextualised text using Large Language Models (LLMs) and examine the impact of position and context on the relevance estimation of injected text. They find that NRMs, particularly those based on BERT and T5 architectures, are sensitive to the position of injected text, with content placed after salient sentences having a lesser negative effect on relevance. The study also introduces a mitigation strategy using a classifier to detect promotional content, demonstrating its effectiveness in reducing the negative impact on retrieval performance. The findings highlight the potential for exploiting positional bias in NRMs and the need for robustness against such attacks in semantic search.

- **Nous Hermes 2 - Mixtral 8x7B-DPO (46.7B)**

Generated Abstract (2-shot demonstrations from co-author papers):
This study investigates the exploitation of positional bias in neural ranking models (NRMs) for query-agnostic generative content injection in search. The authors propose the concept of attention bleed-through, where positive attention scores from relevant spans propagate to non-relevant spans, affecting relevance estimation. They find that injecting non-relevant text close to document sentences with high similarity scores with the query can largely affect the rank of documents. The study also demonstrates that controlling the position and contextualization of promotional text can mitigate its negative impact on retrieval performance. A model-agnostic mitigation strategy is proposed, which improves nDCG@10 significantly under a classic evaluation setting by reducing the effect of contextualization. The findings highlight the importance of addressing positional bias in NRMs and the potential risks of generative content injection in semantic search.

Comparison between Generated Abstract and Ground Truth on Abstract Writing Task

Ground Truth:
Developing autonomous agents for mobile devices can significantly enhance user interactions by offering increased efficiency and accessibility. However, despite the growing interest in mobile device control agents, the absence of a commonly adopted benchmark makes it challenging to quantify scientific progress in this area. In this work, we introduce B-MoCA, a novel benchmark designed specifically for evaluating mobile device control agents. To create a realistic benchmark, we develop B-MoCA based on the Android operating system and define 60 common daily tasks. Importantly, we incorporate a randomization feature that changes various aspects of mobile devices, including user interface layouts and language settings, to assess generalization performance. We benchmark diverse agents, including agents employing large language models (LLMs) or multi-modalLLMs as well as agents trained from scratch using human expert demonstrations. While these agents demonstrate proficiency in executing straightforward tasks, their poor performance on complex tasks highlights significant opportunities for future research to enhance their effectiveness. Our source code is publicly available at https://b-moca.github.io.

- **Qwen 1.5 Chat (72B)**

Generated Abstract (2-shot demonstrations from co-author papers):
The Benchmark for Mobile Device Control Agents (B-MoCA) is a novel evaluation platform designed to assess the performance of agents in controlling mobile devices across diverse configurations. The benchmark features 60 practical tasks grounded in real-life scenarios and supports randomization of device attributes like icon placements, wallpapers, languages, and device types to test generalization ability. The study compares three types of agents: LLM (Large Language Model) agents, MLLM (Multimodal Large Language Model) agents, and Vision-Language-UI (VLUI) agents. Results show that while LLM and MLLM agents perform well in simple tasks, they struggle with complex tasks requiring sequential decision-making. VLUI agents, trained from scratch, demonstrate better performance in completing tasks but still face challenges with severe visual changes in unseen device configurations. The study highlights the importance of diverse training data and the need for more efficient methods to leverage multi-modal inputs. Overall, B-MoCA provides a comprehensive framework for evaluating and advancing mobile device control agents in real-world settings.

- **Nous Hermes 2 - Mixtral 8x7B-DPO (46.7B)**

Generated Abstract (2-shot demonstrations from co-author papers):
This paper introduces B-MoCA, a benchmark designed to evaluate the performance of mobile device control agents on diverse device configurations in executing common daily tasks. The benchmark includes 60 basic tasks grounded in realistic scenarios and a randomization feature that changes icon placements and sizes, wallpapers, languages, and device types. Baseline agents include LLM agents, MLLM agents, and Vision-Language-UI (VLUI) agents. Experiments reveal that LLM agents and MLLM agents show high robustness across diverse device configurations while they fall short on multiple sequential decision-making. Agents trained with BC successfully mimic expert behaviors but lack generalization ability in test environments with unseen device configurations. The study highlights the limitations of existing methods in mobile device control and calls for future research.

F.3 Introduction Writing

Comparison between Generated Introduction and Ground Truth on Introduction Writing Task

Ground Truth:
The original text is too long to be included here. Please see 2404.17196v1 via arXiv for more details.

- **Qwen 1.5 Chat (72B)**

Generated Introduction (2-shot demonstrations from co-author papers):
Retrieval poisoning is a newly identified threat to large language model (LLM)-powered applications, particularly those utilizing the retrieval augmented generation (RAG) technique. This threat arises from the vulnerability in LLM application frameworks, which allow attackers to subtly manipulate the content of documents in a way that appears benign to humans but can guide the LLMs to generate incorrect or malicious responses. The attack is executed by injecting invisible attack sequences into the documents, which remain undetected during the RAG process, leading to compromised application responses. The prevalence of LLM application frameworks, such as LangChain, has enabled the development of numerous LLM-powered applications. However, these frameworks often overlook the security risks associated with external content. Retrieval poisoning exploits this vulnerability by crafting malicious documents that, despite containing accurate information, can prompt the LLMs to produce misleading responses when used as reference sources during RAG. The success rate of such attacks is alarming, with preliminary experiments demonstrating an 88.33% success rate in misleading LLMs and a 66.67% success rate in real-world applications. To execute retrieval poisoning, attackers analyze the LLM application framework, identifying exploitable components like document parsers, text splitters, and prompt templates. They then generate an attack sequence that is invisible to human perception but influences the LLM's response generation. The crafted malicious document is designed to maintain its original appearance while containing the hidden attack sequence, which is strategically placed to remain undivided by text splitters. The proposed approach involves a two-step process: analyzing the framework to identify invisible injection points and generating the attack sequence. Attackers leverage the framework's prompt templates to create augmented requests that guide the LLMs towards generating targeted malicious responses. The effectiveness of the attack is demonstrated through experiments on different LLMs, augmented requests, and real-world applications, showcasing the potential impact of retrieval poisoning. In summary, retrieval poisoning is a stealthy attack method that exploits the vulnerabilities in LLM application frameworks, particularly in the RAG process. By injecting invisible attack sequences into documents, attackers can manipulate LLMs into generating incorrect responses, posing a significant security concern for LLM-powered applications. The proposed approach highlights the need for better understanding of LLM application frameworks and the development of more robust defense mechanisms to mitigate such threats. The paper structure delves into the problem, presents the attack methodology, demonstrates its effectiveness through experiments, and discusses potential defense strategies, ultimately contributing to the ongoing research on securing LLM-powered applications.

Comparison between Generated Introduction and Ground Truth on Introduction Writing Task

Ground Truth:
The original text is too long to be included here. Please see 2404.18465v2 via arXiv for more details.

- **Qwen 1.5 Chat (72B)**

Generated Introduction (2-shot demonstrations from co-author papers):
The M3oE framework is a pioneering solution to the challenge of multi-domain multi-task recommendation, addressing the intricate dependencies among multiple domains and tasks in a disentangled and adaptive manner. Recognizing the need for a more comprehensive approach in practical recommendation systems, M3oE integrates multi-domain information and leverages knowledge transfer across domains and tasks, optimizing for multiple objectives simultaneously. The framework consists of three mixture-of-experts modules, each dedicated to capturing common, domain-specific, and task-specific user preferences. A two-level fusion mechanism is employed to effectively extract and integrate features across diverse domains and tasks, ensuring a balanced representation of user preferences. Existing multi-domain and multi-task recommendation methods have shown promise in enhancing recommendation accuracy by exploiting shared patterns. However, they often fall short when faced with the complexity of simultaneously addressing multiple domains and tasks. M3oE stands out by introducing a self-adaptive approach, utilizing AutoML to optimize the model structure dynamically. This adaptability is crucial for real-world scenarios where data characteristics and task requirements can vary significantly. The core of M3oE lies in its three expert modules: shared, domain, and task experts. The shared expert module learns common patterns across domains and tasks, while the domain expert module focuses on domain-specific characteristics. The task expert module, on the other hand, models task-specific user preferences. The two-level fusion mechanism balances the contribution of these modules, allowing for a flexible and effective representation of user preferences. Experimental results on benchmark datasets demonstrate M3oE's superior performance compared to state-of-the-art baselines, highlighting its ability to address the multi-domain multi-task seesaw problem. In summary, M3oE is a novel and adaptive multi-domain multi-task recommendation framework that integrates multi-domain information, transfers knowledge across domains and tasks, and optimizes for multiple objectives. The framework's disentangled and fused expert modules, along with the AutoML-driven fusion weights, enable it to effectively capture user preferences in complex recommendation scenarios. The proposed approach not only surpasses existing methods in performance but also offers a more generalizable solution to the multi-domain multi-task recommendation challenge. The structure of the paper delves into the problem definition, framework overview, detailed module descriptions, experimental setup, and results, providing a comprehensive understanding of M3oE's design and effectiveness.