A Hierarchical Context Augmentation Method to Improve Retrieval-Augmented LLMs on Scientific Papers

Tian-Yi Che
Beijing Institute of Technology
Beijing, China
ccty@bit.edu.cn

Xian-Ling Mao*
Beijing Institute of Technology
Beijing, China
maoxl@bit.edu.cn

Tian Lan
Beijing Institute of Technology
Beijing, China
lantiangmftby@gmail.com

Heyan Huang
Beijing Institute of Technology
Beijing, China
hhy63@bit.edu.cn

Abstract

Scientific papers of a large scale on the Internet encompass a wealth of data and knowledge, attracting the attention of numerous researchers. To fully utilize these knowledge, Retrieval-Augmented Large Language Models (LLMs) usually leverage large-scale scientific corpus to train and then retrieve relevant passages from external memory to improve generation, which have demonstrated outstanding performance. However, existing methods can only capture one-dimension fragmented textual information without incorporating hierarchical structural knowledge, eg. the deduction relationship of abstract and main body, which makes it difficult to grasp the central thought of papers. To tackle this problem, we propose a hierarchical context augmentation method, which helps Retrieval-Augmented LLMs to autoregressively learn the structure knowledge of scientific papers. Specifically, we utilize the document tree to represent the hierarchical relationship of a paper and enhance the structure information of scientific context from three aspects: scale, format and global information. First, we think each top-bottom path of document tree is a logical independent context, which can be used to largely increase the scale of extracted structural corpus. Second, we propose a novel label-based format to represent the structure of context in textual sequences, unified between training and inference. Third, we introduce the global information of retrieved passages to further enhance the structure of context. Extensive experiments on three scientific tasks show that the proposed method significantly improves the performance of Retrieval-Augmented LLMs on all tasks. Besides, our method achieves start-of-art performance in Question Answer task and outperforms ChatGPT. Moreover, it also brings considerate gains with irrelevant retrieval passages, illustrating its effectiveness on practical application scenarios.

*Corresponding author

KDD '24, August 25–29, 2024, Barcelona, Spain.

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671847

CCS Concepts

• **Computing methodologies** → Natural language generation; • **Information systems** → **Language models**; **Document structure**.

Keywords

Retrieval-Augmented LLMs, Context Augmentation, Scientific Papers, Structure Information

ACM Reference Format:

Tian-Yi Che, Xian-Ling Mao, Tian Lan, and Heyan Huang. 2024. A Hierarchical Context Augmentation Method to Improve Retrieval-Augmented LLMs on Scientific Papers. In *Proceedings of Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671847

1 Introduction

Scientific papers are the main carrier for storing, disseminating and learning professional knowledge, and there have been a lot of meaningful researches on this domain, such as SciBERT[5], SciERC [23] and so on. Recently, Ho et al. [16] make a comprehensive survey of pretrained language models on scientific domain, systematically illustrating the features of scientific text, development of scientific domain and effectiveness of pretrained language models. However, the release of GPT-4 [30] has changed the research directions in many NLP tasks on scientific domain [22, 44]. It is proved that LLMs boast formidable zero-shot learning abilities, achieving the start-of-art performance in most scientific downstream tasks [12]. But general LLMs also suffer from serious hallucination problems [17], especially in scientific domain, because the implicit knowledge learned from corpus is limited and the high training cost prevents real-time update of the latest information. A number of works demonstrate that retrieval can alleviate this problem by importing external knowledge [13, 20, 39]. The framework contains a retriever and a LLM as generator, called Retrieval-Augmented LLMs, which becomes the mainstream method for scientific tasks nowadays.

Recently, there are some interesting attempts to improve performance of Retrieval-Augmented LLMs. For example, Ma et al. [25] introduce a new rewrite-retrieval-generation framework from the perspective of query augmentation, rewriting the query to adapt to the retriever. Lyu et al. [24] propose an algorithm to reweight the data importance of retrieval corpus and improve generation effect from the perspective of retrieval quality. Shao et al. [38]

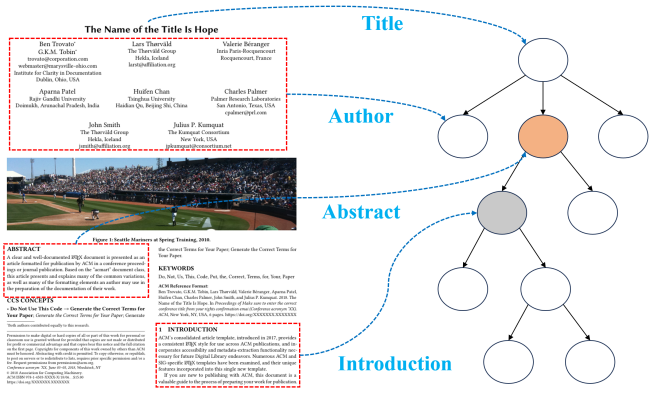

Figure 1: The example of a document tree. The tree nodes indicate the components of a scientific paper and the leaf nodes of main body are paragraphs under sections, subsections or subsubsections. The relationship of the parent node and child node represents a global-local relationship on the content, such as the abstract and introduction.

show the iterative manner can further enhance the retrieved documents and these augmented documents help to generate better results. However, existing works on Retrieval-Augmented LLMs lack the study on structure of retrieved context, which is important for scientific papers. In fact, these autoregressive language models can only capture joint probability distribution of text sequences, lacking the multi-dimensional structural modeling. By contrast, previous masked language models, such as BERT, usually utilize graph network to capture global structural information [29], which isn't applicable to LLMs. Therefore, we consider a novel direction, how to represent structure of papers in text sequences and make LLMs autoregressively learn this hierarchical relationship.

In this paper, we propose a hierarchical context augmentation method, which can help Retrieval-Augmented LLMs to learn the structural distribution of scientific papers and leverage the structural information of retrieved context to enhance generation. The basic step is to represent a single scientific paper as a document tree, as Figure 1 shows. We have extracted each component from unsupervised scientific corpus and built the tree based on the nested relationship of different components.

Then we design methods to enhance context from three aspects: scale, format and global information. First, due to the high cost of the existing extraction technology and problem of noise transmission, we propose an unsupervised method to increase data scale without structure loss. We sample and research a batch of papers, finding that the dependency relationship of section contents is mainly reflected in the parent-child sections, while the parallel section contents are relatively independent of each other. Thus, we leverage the root-leaf traversal algorithm to extract each path from root node to leaf nodes of a document tree, which contains the most important context information of papers. The dataset to continue pretraining LLMs is mixed from the scientific corpus, augmented scientific corpus and general corpus. Second, we unify the textual format of context among all stages. We consider three kinds of formats: machine-oriented LaTeX, human-oriented plain text and proposed label-based format. Intuitively, LaTeX is far from natural language and brings substantial noise, while plain text is more aligned with natural language but lacks explicit structural information, struggling to introduce the relationship between sections. Therefore, we proposed a novel label-based format, which can not only reflect the hierarchical relationship but minimize the loss of fluency as much as possible. Third, we leverage the parent nodes of retrieved passages to introduce global information. In practice, the ground-truth evidences are not provided and we need a retrieval model to recall relevant passages as evidences. When the top-k evidences are obtained by retriever, we will locate their positions in the document tree and then upwardly propagate to gain the global information related to retrieved context. Besides, we have trained an efficient dense retrieval model on scientific corpus, achieving competitive performance.

We evaluate the proposed method on three scientific tasks, including Question Answer, Keyphrase Generation and Abstract Summary. The experimental results show our method can significantly improve the performance of Retrieval-Augmented LLMs on all tasks. In Question Answer, the comparison with task-specific models is presented in Table 7 and our trained models achieve the start-of-art performance. The ablation experiments demonstrate the effectiveness of every module. Besides, we compare the effect of different formats and find the proposed label-based format outperforms previous LaTeX format and plain text, illustrating Retrieval-Augmented LLMs could learn more diverse structure information by the novel context textual format. Moreover, we consider some practical application scenarios, eg. retriever could recall some irrelevant passages and generator need to answer on these suboptimal context. The experiments show our method can bring considerate gains for low retrieval quality and zero-shot setting.

To summarize, the contributions of our work are as follows:

- We propose an effective hierarchical context augmentation method, which makes Retrieval-Augmented LLMs to autoregressively learn hierarchical structure relationship of scientific papers and generate more reliable responses.
- We study the impact of context textual format on Retrieval-Augmented LLMs and propose a novel representation format based on tree labels, which is more effective than LaTeX and plain text.
- The experiments on three scientific tasks show our method significantly improves the performance of Retrieval-Augmented LLMs and it's also effective in actual retrieval scenarios.

2 Related Works

2.1 Data Representation of Scientific Papers

Scientific domain has been attracting widespread attention and there are a lot of meaningful works on the data representation of scientific papers. At first, the scientific papers on Internet are usually stored in layout-oriented data formats, like PDF, which is friendly to humans. To mine the explicit structural and semantic information, Ronzano and Saggion [36] present a novel framework to extract structured information from plain text or PDF format. Based on the extracted text and entities, some researchers strive to explicitly represent the structure of papers. Ammar et al. [3]

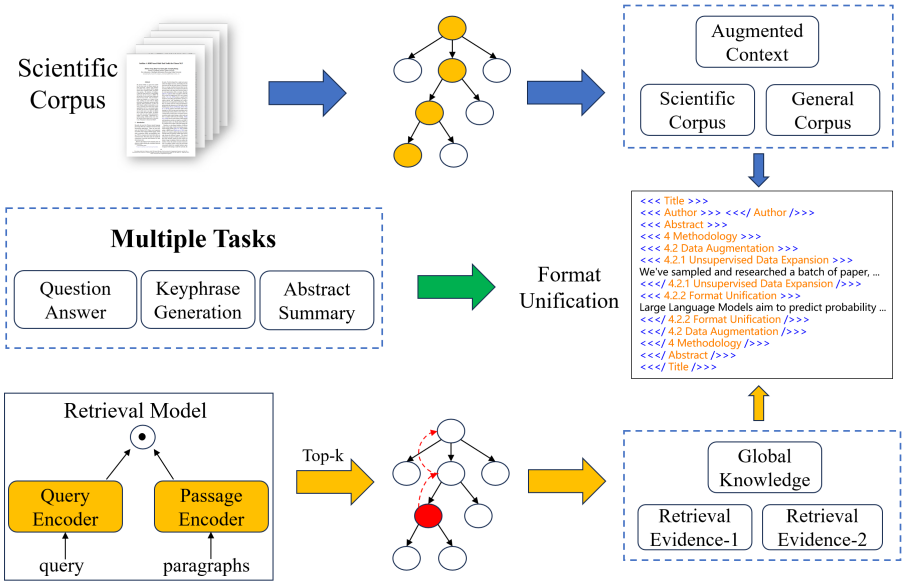

Figure 2: The overview of the proposed hierarchical context argumentation method. The blue, green and orange arrows indicate the stage of continuing pretraining, fine-tuning and inference respectively. The orange and red tree nodes indicate the context used to expand data and retrieved paragraphs respectively. There is an example of the proposed format based on tree labels in the Format Unification.

propose the literature graph of semantic scholar to represent papers, authors, entities and various interactions, serving a series NLP tasks. Furthermore, Meloni et al. [26] introduce the knowledge graphs to conversational agents, to produce factual and relevant answers in a specific domain. Besides, other works focus on the document-level representation of scientific papers. For example, Beltagy et al. [5] leverage large-scale scientific corpus to pretrain masked language models and gain document-level vector embedding. Cohan et al. [8] introduce citation-graph as a powerful signal of document-level relatedness to enhance pretrained transformer. Moreover, Wang et al. [43] propose an author contributed representation to consider different authors' contribution for modeling scholarly network. However, existing data representation is not applicable to autogressive language modeling, and it remains a challenge to make LLMs learn the structure of scientific papers.

2.2 Retrieval-Augmented LLMs

Typically, Large Language Models (LLMs) refer to Transformer language models that contain hundreds of billions (or more) of parameters, which are trained on massive text data [9, 32, 37]. Retrieval-Augmented LLMs mean the combination of a retriever and a LLM as generator, utilizing retrieved external knowledge as context to enhance LLM [13, 18, 20]. Since the release of GPT-4 [30], there have been a bunch of excellent LLMs emerging. PALM2 [4], Claude2, ChatGLM [47] are the representatives of closed source LLMs, achieving outstanding performance on multiple NLP tasks. LLaMA [42], Baichuan [46] and Falcon [2] are open source LLMs and also reach competitive performance. Considering the excellent ecology and effect, We choose the LLaMA and Baichuan as the foundational LLMs. As for the retriever, the popular methods are divided into sparse retrieval [35, 48] and dense retrieval [19, 33]. Many studies demonstrate that dense retrieval is more effective and efficient [51], and therefore we choose bi-encoder dense retrieval architecture and leverage scientific corpus to train specific retriever. Moreover, there are many works aiming to improve the performance of the Retrieval-Augmented LLMs. Ma et al. [25] propose the rewrite strategy to enhance the fitness between query and retriever. Shao et al. [38] illustrate the iterative manner can improve the quality of retrieved passages to enhance generation. Ram et al. [34] first demonstrate the in-context learning can further boost performance of Retrieval-Augmented LLMs. However, these methods ignore the impact of structure of retrieved context, which is important for understanding some knowledge-intensive documents, especially scientific papers.

3 METHODOLOGY

In this section, we introduce the overview of the proposed method, as Figure 2 shows. Firstly, we find that the document tree is suitable to represent the structure of scientific papers. The detailed construction process and representation format of the document tree will be described in Section 3.1. Secondly, we introduce the three modules of hierarchical context augmentation in Section 3.2. Thirdly, the complete training process and detail will be presented in Section 3.3, both continuing pretraining and fine-tuning. Besides, we train a novel scientific dense retrieval model for practical applications.

3.1 Document Tree

Different from the literature graph, we introduce the tree structure to represent scientific papers. As is shown in Figure 1, we define the title as the global identification of the single paper, which is

placed at the position of root node. The author information, abstract and references are parallel concrete components, with abstract as the concise summary of the main body. Then, the dependency relationships between sections in the main content are linked to the subtree of the abstract node.

Because there are a large number of scientific papers on the Internet but the extracted and annotated datasets are lacking, we design a pipeline to automatically convert the scientific corpus to document trees. First, we collect many urls of open-source arxiv papers from RedPajama corpus [1], which is the formal pretraining corpus of LLaMA [42]. Then, we design a web crawling framework based on Scrapy [2] to download research papers to our local storage. This framework boasts features such as multi-processing, resume-from-breakpoint crawling, load-balancing and so on. Next, we leverage the mature tool Grobid [3] to convert pdf files to xml files. The converted files possess very complex context, which is due to the flexibility of pdf format. Thus, the data cleaning is necessary when the xml files are obtained. During data cleaning, we first delete the duplicate data and then remove the useless labels and content in xml files. Besides, we filter out the wrong data extracted by Grobid and leverage regular expression to replace the labels in content, including <<<, >>>, <<< / and />>>, which are the special vocabularies we define. These vocabularies are selected because they are unique and similar to HTML syntax, making them as similar as possible to general corpus to avoid conflicting reactions. Finally, we store the structure of a paper in Json format and then construct the document tree from the Json file.

When constructing document trees, we will extract the title, publications, authors, abstract, references and main body of an entire paper. The key step is to extract the nested structure between the sections of main body. Because it is a very challenging task to extract pdf files and Grobid maybe treat the paragraph labels as section titles bringing some unpredictable noise, we design an algorithm to refresh the structure of main body. The *id*, *content*, *body* indicate the index, title and body content of a section respectively. We utilize regular expression to unify the expression and leverage the inclusion relationship of *id* to indicate they are parent-child sections. When the section is abnormal such as *id* is -1 or *content* is empty, it will be marked as exception and merged to the body of the last normal section. We use the recursive thinking to achieve this algorithm.

3.2 Context Augmentation

To enhance the models' understanding on the structure of scientific papers, we propose a novel context augmentation method based on the document tree for Retrieval-Augmented LLMs. First, due to the limited efficiency of existing pdf extraction technology, such as Grobid, we propose an unsupervised data expansion method to effectively increase the scale of corpus. Then, we define the unified textual format among the stages of continuing pretraining, fine-tuning and inference. It has been proven that language models can learn the implicit knowledge through pretraining and the different data formats of scientific datasets could be conflicting among the stages of continuing pretraining, fine-tuning and inference. The aim of format unification is to leverage the learned implicit knowledge of context in practice. Finally, in the stage of inference, we introduce the global information of retrieved context based on the document tree, which can provide the central thought of context.

3.2.1 Unsupervised Data Expansion. We sample and research a batch of papers, finding that the dependency relationship of section content is mainly reflected in the parent-child sections, while the parallel section contents are relatively independent of each other. Thus, we assume that the content of a section is just related with the content of its parent sections and child sections. In the document tree, each corresponding path from root node to leaf nodes indicate an independent context. We extract each path from root node to leaf nodes, containing the main body, to expand data and enhance the global-local relationships between sections. The processed corpus will be used to continue pretraining the model.

$$C_{aug} = \sum_{i}^{N} \sum_{j}^{M_i} f(root_i, leaf_{ij}) \tag{1}$$

where N, M_i is the number of scientific papers and the leaf nodes of the i_{th} paper. f indicates the list of text from the path from the root node to j_{th} leaf node in i_{th} paper.

$$C_{pretrain} = C_{sci} \cup C_{aug} \cup C_{general} \tag{2}$$

where C_{sci} is the previous scientific papers and $C_{general}$ is the general corpus such as CommonCrawl. The mixed dataset will be used to continue pretraining the language models.

Next, we will introduce the detail of the root-leaf algorithm, which extract each paths from root node to leaf nodes of a document tree. Similar to the deep-first search algorithm, we insert the content of leaf nodes into the labels. Then, for each non-leaf nodes, we gain the templates of each child and use the label and content of this section to nest them and return. The pseudo code is shown in Algorithm 2 and the example is shown in Figure 2.

3.2.2 Format Unification. Large Language Models aim to predict the probability of next token and different language distribution will largely affect the models' understanding and generation. Meanwhile, the language distribution is associated with the data format of context. For example, there are many special tokens with LaTeX format, which is different from the general textual expression. Thus, we consider that unified data format will make models leverage learned implicit knowledge better. The unified format is used in every stage, including continuing pretraining, fine-tuning and inference.

To decide which format can balance the naturalness of the text with the structural features of the paper, we consider three options. First, we consider the machine-oriented LaTeX format, which is widely used to pretrain LLMs. We clean the useless labels and delete the figures, tables and algorithms. This format possesses some structural features but is insufficiently natural. There are also many other machine-oriented formats, such as JSON and XML. They are similar to LaTeX in property, so we don't consider them. Second, we consider the human-oriented plain text, of which there are a number of scientific datasets. Then we add some prompts to introduce the index and function of sections. But the structural

[1] https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T
[2] https://github.com/scrapy/scrapy
[3] https://github.com/kermitt2/grobid

features are still difficult to be captured by models. Third, we design a format based on tree labels. The example is shown in Figure 2. This format reflects the relationship between global and local sections better and minimizes the loss of fluency as much as possible.

3.2.3 Global Information Augmentation. In general application, we will first retrieve evidences on query and then input query and evidences to trained language models to generate responses. Obviously, the retrieval performance will greatly affect the generation quality. We conduct experiments on the common retrieval methods, including sparse retrieval and dense retrieval. To improve the retrieval performance on scientific papers, we've trained a dual-encoder retrieval model, which achieve the competitive effect and ensure the inference speed. Besides, we propose the global information augmentation method, which introduce more global information on the retrieved passages. This method can compensate for the poor retrieval situation and play a good role in practical applications.

Compared to sparse retrieval that only focuses on word overlap, dense retrieval can identify the semantic similarity, which maps the query and passages to vector space. Followed by DPR [19], we train a dense retrieval model, using the dual-encoder architecture. Thanks to the care-fully annotated pairs of questions and ground-truth evidences, we could fine-tune the general dense retrieval model with an in-batch negative sampling method.

Specifically, given the question q, ground-truth evidences $\{e_i^+\}_{i=0}^k$ and a bunch of random sampled paragraphs in the paper $\{e_i^-\}_{i=0}^m$, the InfoNCE loss is used to optimize the dense retrieval model:

$$\mathcal{L} = -\log \frac{\sum_{j=0}^{k} e^{E_q(q)\cdot E_a(e_j^+)}}{\sum_{j=0}^{k} e^{E_q(q)\cdot E_a(e_j^+)} + \sum_{j=0}^{m} e^{E_q(q)\cdot E_a(e_j^-)}} \tag{3}$$

where E_q and E_a are the question encoder and the answer encoder in the dense retrieval model.

Next, we propose the global information augmentation method. When the top-k evidences are obtained by Evidence Retrieval, we will first find the their positions in the document tree and then upwardly propagate to gain the global knowledge related the retrieved context. As is shown in Figure 2, we seek the parent nodes of retrieved passages gradually, leveraging these nodes to provide more consistent external knowledge. The selected subtree will be converted to context evidence with unified format to enhance generation.

3.3 Training and Inference

We first introduce the overview of the training strategy and detail, including the continuing pretraining and fine-tuning. Then, we simply explain the process of inference. Due to the limited hardware, we decide to continue pretraining on the widely used foundational LLMs.

3.3.1 Continue Pretraining. We first obtain the processed corpus by the unsupervised data expansion, as described in Section 3.2.1. The corpus are converted to the unified textual format with structural labels. Then, we use the new corpus to continue pretraining the general LLMs, LLaMA and Baichuan, thereby adapting it to the data distribution of scientific papers' structure and content. Specifically, we utilize the recently proposed QLoRA [11] optimization method combined with the DeepSpeed toolkit[4] to directly optimize LoRA parameters on an 8-card 3090 server. The additional LoRA weights are optimized upon the LLMs by:

$$\mathcal{L} = \Pi_{t=0}^{T} f(y_t|\theta_{\text{LLM}}, \theta_{\text{LoRA}}, y_{<t}) \tag{4}$$

where θ_{LLMs} is the parameters of LLM f, which is frozen during pre-training. y is a chunk of the sequential tokens tokenized from the scientific paper data samples. θ_{LoRA} is the trainable LoRA parameters.

3.3.2 Fine-tune. We select three well-known scientific tasks to fine-tune the language models, including Question Answer, Keyphrase Extraction and Abstract Summary. We build the instruction prompts to combine datasets of the three tasks and unify the context format. The prompt is shown in the Appendix C.3. Then, we further supervised fine-tune the LoRA parameters of LLMs, enabling it to understand the task and query, and generate accurate responses to solve user problems. In this stage, we use the ground-truth evidences provided by datasets to fine-tune the model and the LoRA weights θ_{LoRA} are further optimized by:

$$\mathcal{L} = \Pi_{t=0}^{T} f(y_t|\theta_{\text{LLM}}, \theta_{\text{LoRA}}, y_{<t}, q, \{e_i\}_{i=0}^{k}) \tag{5}$$

where a is the ground-truth answer related to the query q. Note that we directly optimize the LoRA weights added during pre-training process, and the model parameters of LLMs are still frozen during supervised fine-tuning.

3.3.3 Inference. In Question Answer, we need to retrieve the relevant paragraphs on query as evidences to input the language models. In Keyphrase Generation and Abstract Summary, the components of papers is fixed and we just need to extract the papers. In this section, we mainly introduce the generation based on retrieval.

We first train a novel dense retrieval model to retrieve evidences on users' query. Then, we extract the global information on retrieved context from the document tree, as Figure 2 shows. Next, we utilize the same prompt format as the fine-tuning, to input the LLMs. The probability of next token in the inference will be calculated by:

$$y_{t,i} = \underset{t \in vocab}{\arg\max}\, P(y_t|\theta_{LLM}, q, e_{retrieval}, e_{global}, y_{<i}) \tag{6}$$

where θ_{LLM} is the parameters of LLM and q is the query. $e_{retrieval}$ and e_{global} are the retrieved evidences and augmented global evidences. y is a chunk of the sequential tokens and i means the index of currently generated token.

4 Experimental Setup

In this sections, we introduce the detail and cause of the experimental setup from three aspects: models, datasets and evaluation metrics. We focus on the scientific tasks on the single document.

4.1 Models

In this study, we choose LLaMA [42] and Baichuan [46] as the foundational LLMs, which have been used for various supervised fine-tuning processes. Then we choose three models fine-tuned on

[4]https://github.com/microsoft/DeepSpeed

Table 1: Statistics of datasets to fine-tune and evaluate models. The context type indicates the components of scientific papers provided by datasets. Question Answer, Keyphrase Generation and Abstract Summary aim to train and evaluate the models' capacity for different provided context of scientific papers. We focus on the single document in all tasks.

Task	Dataset	Train	Dev	Test	Context Type
Question Answer	QASPER	2,314	1,601	1,268	Full Paper
	SciMRC	3,974	484	1,099	
Keyphrase Generation	KP20k	527,830	20,000	20,000	Title and Abstract
Abstract Summary	Arxiv-Summary	203,037	6,436	6,440	Main Body

LLaMA as baselines to evaluate the performance between our models and other excellent open-source fine-tuned models. They are: (1) Alpaca [41] further fine-tunes the LLaMA on 52K instruction following data generated by the Self-Instruct [45], behaving similarly to the GPT-3 [6]. (2) OpenAlpaca [40] is an instruction-following model based on OpenLLaMA [14], a permissively licensed open source reproduction of LLaMA. (3) Vicuna [53] fine-tunes LLaMA on user-shared conversations collected from ShareGPT. Besides, to eliminate the influence of continuing pretraining and fine-tuning on the experimental results, we process the foundational LLMs using the same training procedure and steps without context augmentation, called $\text{LLAMA}_{\text{sft}}$ and $\text{Baichuan}_{\text{sft}}$. We evaluate the effectiveness of proposed method by comparing with these models. All the models have almost 7 Billion parameters.

4.2 Datasets

As Table 1 shows, we choose three scientific Natural Language Process tasks: Question Answer, Keyphrase Generation and Abstract Summary, to evaluate the capacity for processing different context components. In Question Answer, we use the evidence-based scientific dataset, QASPER [10] and SciMRC [50], which provide the title, abstract and all sections with their paragraphs. In Keyphrase Generation, KP20k [27] is widely used in the Keyphrase Extraction and Generation tasks, which aims to obtain the keyphrases of scientific papers based on the title and abstract. In Abstract Summary, we use the dataset Arxiv-Summary [7] that generates the abstract on the main body of scientific papers from the Arxiv. To improve the test efficiency, we sample separately 1k test instances from KP20k and Arxiv-Summary, which is close to the size of QASPER testset. We mix different datasets with the same number of instances and build the instructions containing the description of different tasks, which aims to enhance the models' performance on multiple downstream tasks simultaneously.

Moreover, to show the effect of data expansion, we make a statistics on scientific corpus used to continue pretraining LLMs. Limited by the performance of PDF extractor, we have to filter those mistaken documents and the high quality data consumes a lot of time and computational costs. But as Table 2 shows, the unsupervised data expansion can significantly increase the data scale, with the item count increased by nearly 10 times and token count increased by nearly 2.5 times. It can provide more diverse and concise data. Note that we guarantee the total number of tokens for training is consistent to fairly compare the effect of different formats.

Table 2: The statistics of datasets to continue pretraining LLMs. Corpus indicates whether the corpus build the document tree and do data expansion. Format indicates the format of textual language. The expanded structural corpus utilize the unsupervised data expansion, significantly increasing the scale.

Corpus	Format	Item Count	Token Count
Raw Corpus	LaTeX	61,961	0.89B
	Plain-text	47,183	0.21B
Structural Corpus	Tree Label	50,832	0.40B
Expanded Structural Corpus		493,487	0.99B

4.3 Evaluation Metrics

In this paper, we choose the popular machine metrics to evaluate models. In Natural Language Generation, BLEU [31], Rouge [21] and F1 are usually used to evaluate the performance of models by assessing the word overlap between generated sequences and reference sequences. But in Keyphrase Generation, we utilize the keyphrases instead of spans as units to calculate F1. Besides, considering the semantic similarity, we also use the BERTScore [49] that calculates the inner product of the semantic vectors encoded by BERT. Note that the responses for paper-ground instructions are deterministic, and the above word-overlap based evaluation metrics are good enough to examine their performance.

5 Experiments

5.1 Main Result

5.1.1 Question Answer. The comparison between our method and baselines on the evidence-based Question Answer dataset is presented in Table 3. As is shown, our method significantly improves the performance of Retrieval-Augmented LLMs, which is effective for both LLaMA and Baichuan. Specifically, $\text{LLaMA}_{\text{aug}}$ improves BLEU by 47.2%, ROUGE by 30.7% and F1 by 31.9% compared to $\text{LLaMA}_{\text{sft}}$ and $\text{Baichuan}_{\text{aug}}$ improves BLEU by 49.3%, ROUGE by 28.4% and F1 by 31.7% compared to $\text{Baichuan}_{\text{sft}}$. This illustrates the proposed context augmentation method can further enhance the retrieval-based generation of LLMs under traditional fine-tuning. Meanwhile, our trained models prominently outperform the popular open-source LLMs, showing the potential of our models as

excellent products on scientific domain. Besides, the comparison with task-specific models is presented in Table 7 and our models achieve the start-of-art performance. Note that the evidences are ground-truth and we mainly examine their capacity to understand standard scientific context in this section. The researches on different retrieval quality are shown in 5.3.3.

Table 3: Experimental results on QASPER [10] test set with ground-truth evidences. sft **and** aug **undergo the same training steps. What's different is that** aug **utilizes the context augmentation method but** sft **didn't.**

Models	BLEU	$\text{ROUGE}_{\text{sum}}$	$\text{BERTScore}_{\text{F1}}$	F1
LLaMA	2.70	10.83	80.06	14.43
Baichuan	-	8.46	76.15	8.02
Alpaca	5.30	20.51	84.69	25.60
OpenAlpaca	5.52	21.87	85.02	25.30
Vicuna	7.38	21.18	86.12	26.74
$\text{LLaMA}_{\text{sft}}$	29.71	39.84	88.36	48.14
$\text{Baichuan}_{\text{sft}}$	29.57	40.24	88.85	49.33
$\text{LLaMA}_{\text{aug}}$	43.73	**52.06**	91.37	63.49
$\text{Baichuan}_{\text{aug}}$	**44.15**	51.65	**92.25**	**64.96**

5.1.2 Keyphrase Generation. We further evaluate the capacity of short-range understanding and condensing on the popular scientific dataset, KP20k [27]. In this task, the third module, global information augmentation, will not work because the models don't need retrieval in generation, with only title and abstract as input. As Table 4 shows, our method improves the performance by about 2 points compared to $\text{LLaMA}_{\text{sft}}$ and $\text{Baichuan}_{\text{sft}}$. It illustrates that even with the simple structural information with title and abstract, our context augmentation method can also bring considerable improvement, showing the generality of proposed method.

Table 4: Experimental results on kp20k [27] test set for Keyphrase Generation with title and abstract.

Models	F1	$\text{BertScore}_{\text{F1}}$
LLaMA	-	79.91
Baichuan	0.50	75.07
Alpaca	14.29	83.4
OpenAlpaca	5.11	83.33
Vicuna	11.98	81.89
$\text{LLaMA}_{\text{sft}}$	22.28	87.76
$\text{Baichuan}_{\text{sft}}$	21.44	86.96
$\text{LLaMA}_{\text{aug}}$	**24.75**	**89.03**
$\text{Baichuan}_{\text{aug}}$	24.56	88.23

5.1.3 Abstract Summary. We further evaluate the capacity of long-range understanding on the main body of scientific papers. In this task, we mainly utilize the explanatory effect of section titles on content. As Table 5 shows, our method improves the performance by 5-7 points compared to $\text{LLaMA}_{\text{sft}}$ and $\text{Baichuan}_{\text{sft}}$. It illustrates the relationship of section titles and content is also helpful for generation of LLMs, which is an important component of structure of papers. Moreover, the experimental results show our method improves the understanding capacity on long document of Retrieval-Augmented LLMs.

Table 5: Experimental results on Arxiv-Summary [7] test set for Abstract Summary with long-range main body.

Models	BLEU	$\text{ROUGE}_{\text{sum}}$	$\text{BERTScore}_{\text{F1}}$	F1
LLaMA	0.33	4.11	74.91	10.20
Baichuan	-	0.83	73.42	-
Alpaca	0.40	7.93	77.66	11.36
OpenAlpaca	0.49	10.26	75.87	13.99
Vicuna	0.27	8.52	72.61	12.84
$\text{LLaMA}_{\text{sft}}$	4.81	22.09	84.19	35.17
$\text{Baichuan}_{\text{sft}}$	4.15	20.93	84.03	33.78
$\text{LLaMA}_{\text{aug}}$	**10.66**	**27.74**	90.85	**41.30**
$\text{Baichuan}_{\text{aug}}$	9.65	26.45	**91.15**	40.36

Table 6: The results on ablation experiments and different formats on QASPER. We use de, fu and ca to refer to the unsupervised data expansion, format unification and global information augmentation respectively. LaTeX and plain text indicate the different textual format with the complete data augmentation.

Models	BLEU	$\text{ROUGE}_{\text{sum}}$	$\text{BERTScore}_{\text{F1}}$	F1
$\text{LLaMA}_{\text{sft}}$	29.71	39.84	88.36	48.14
w/o de	37.22	47.65	90.56	58.26
w/o fu	32.90	45.27	89.92	55.33
w/o ca	40.21	50.07	90.74	60.27
w LaTeX	39.07	50.23	88.88	59.63
w plain text	41.40	50.25	90.99	61.17
Our Method	**43.73**	**52.06**	**91.37**	**63.49**

5.2 Ablation

5.2.1 Effect of Unsupervised Data Expansion. In this study, we continue pretraining using the raw structural corpus with format based on tree labels. As Table 2 shows, there are about 50k instances and 0.4 billion tokens. We use the same training steps and config to ensure fairness. The experimental results are presented in Table 6, which decrease the main metrics BLEU, ROUGE and F1 by about 5 points. It indicates that the unsupervised data expansion improves the performance of Retrieval-Augmented LLMs, demonstrating the high quality of expanded structural corpus with diversity and efficiency.

5.2.2 Effect of Format Unification. In this study, we use inconsistent textual context format to train LLMs, where the pretraining corpus are plain text and fine-tuning datasets are label-based format. As is shown in Table 6, the F1 decreases almost 8 points and

Table 7: The performance of Evidence F1 and Answer F1 on the test set of QASPER. "SciBERT" and "LED Encoder" are backbone models for evidence selection methods. The last three rows are the upperbounds of the task, where the gold evidence is used or the human expert is engaged [28].

Models	Evidence-F1	Answer-F1
LED[10]	29.20	33.12
ETC[1]	51.17	35.37
DPR[19]	36.98	30.22
AISO[54]	42.74	32.52
RCM[15]	47.28	35.99
BERT-DM[52]	48.09	36.52
CGSN[29]	49.55	37.21
AISO(SciBERT)	42.74	32.52
RCM(SciBERT)	47.28	35.99
BERT-DM(SciBERT)	48.09	36.52
CGSN(SciBERT)	**53.98**	39.44
LLaMA$_{\text{aug}}$	40.67	**57.23**
LLaMA$_{\text{aug}}$(Gold Evidence)	-	63.49
LED(Gold Evidence)	-	52.87
gpt-3.5-turbo(Gold Evidence)	-	57.99
Human	71.62	60.92

BLEU decreases almost 10 points, proving the importance of format unification on Retrieval-Augmented LLMs. We think the reason is that the LLMs gain implicit knowledge by learning the probability distribution of next token. There are a gap of the probability distribution between different textual formats, leading to learned world knowledge struggling to be fully utilized in fine-tuning stage.

5.2.3 Effect of Global Information Augmentation. In this study, we just import retrieved passages to LLMs without additional information. As Table 6 shows, it decreases the main metrics BLEU, ROUGE and F1 by 2-3 points, illustrating that introducing global knowledge can enhance the performance of Retrieval-Augmented LLMs. Meanwhile, due to the fact that the length of global information is usually not too long, this module won't bring new context-length problem of LLMs.

5.3 Analysis

5.3.1 Analysis of Context Format. We study the effect of different textual formats that are unified among all stages, including the LaTeX, plain text and tree labels. Taking LaTeX for example, we convert the document tree to LaTeX corpus to continue pretrain and transfer the datasets of downstream tasks to LaTeX-formatted context. As Table 6 shows, our proposed format outperforms the other two formats, which indicates that the proposed format is more suitable to represent scientific papers for LLMs, which can balance document structure and language fluency.

5.3.2 Alalysis of Task-Specific Models. Although the motivation of this paper is to improve the performance of Retrieval-Augmented LLMs, we also want to make a comparison with specifical models to evaluate their capacity. We choose the long-document Question Answer, because the retriever is needed in this task. Followed Nie et al. [29], the experimental results are presented in Table 7. It shows our method achieves the start-of-art answer generation performance on scientific long-document Question Answer. Moreover, our method on retrieved evidences can reach competitive effect with gpt-3.5. If the ground-truth evidences are given, our method can gain 63.49 Answer-F1, outperforming the gpt-3.5 and human. It illustrates the strong capacity of our models on processing scientific papers.

Table 8: Experimental results of zero-shot generation, where models are fine-tuned on QASPER train set and tested on SciMRC [50] test set.

Models	BLEU	ROUGE	BERTScore	F1
LLaMA$_{\text{sft}}$	7.80	30.66	85.64	31.97
Baichuan$_{\text{sft}}$	7.13	30.40	84.91	31.29
LLaMA$_{\text{aug}}$	10.82	**37.49**	**87.54**	**38.56**
Baichuan$_{\text{aug}}$	**11.71**	36.70	86.91	37.64

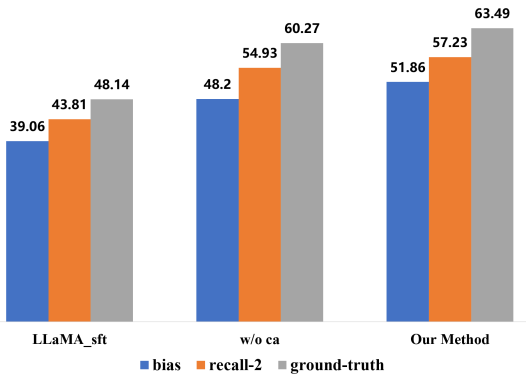

Figure 3: Experimental Results of different retrieval quality on QASPER in terms of F1. The ground-truth means the evidences provided by datasets. The recall-2 means the evidences retrieved by our model. The bias means the remaining evidences to remove ground-truth evidences from recall-2 evidences.

5.3.3 Analysis of Low Retrieval Quality. To demonstrate the effect of proposed method on practical applications where LLMs could need generate responses on retrieved irrelevant context, we define three situations of different retrieval quality: ground-truth, recall-2 and bias. The ground-truth refers to the evidences provided by datasets. The recall-2 refers to the evidences retrieved by our model. The bias refers to the remaining evidences to remove ground-truth evidences from recall-2 evidences. As Figure 3 shows, the performance on recall-2 evidences decreases by 6.26 compared to the ground-truth evidences and the performance on bias evidences decreases by 11.63. Obviously, the retrieval quality will affect the performance of Retrieval-Augmented LLMs. But even if the ground-truth evidences aren't retrieved, our method can still generate better answers that LLaMA$_{\text{sft}}$, shown as 48.14 and 51.86 in

the figure. Besides, the global information augmentation, as one of modules, contributes to the overall improvement. The experiments show the practical application ability of our method.

5.3.4 Analysis of zero-shot scenario. Because the fine-tuning datasets are limited, we design the experiment to evaluate the performance on unseen datasets. The experimental results are presented in table 8. As is shown, our method increases the F1 metric by about 21.6% compared to $\text{LLaMA}_{\text{sft}}$ and 23.2% compared to $\text{Baichuan}_{\text{sft}}$. It indicates that the proposed method possesses stronger generalization ability than existing Retrieval-Augmented LLMs. We think the cause is that the structure layout of scientific papers is very similar in different datasets and profession, and our method succeeds to make LLMs learn the structure knowledge. Therefore, it can enhance the understanding capacity on unseen research fields.

6 Conclusion and Future Work

In this paper, we propose a hierarchical context augmentation method for Retrieval-Augmented LLMs on scientific papers. We represent the structure and content of a paper as a document tree and try enhancing context from three aspect: scale, format and global information, which aims to make Retrieval-Augmented LLMs learn the hierarchical relationship of papers and understand the central thought of papers better. The experiments show that our method significantly improve the performance of Retrieval-Augmented LLMs on multiple scientific tasks. Moreover, we also demonstrate that the proposed method is effective for low retrieval quality and zero-shot scenario, and achieves start-of-art performance on Question Answer task. The augmented datasets and trained models will be released to support the study and application on scientific domain. Apparently, this method can be easily extended to other domains whose documents possess hierarchical structure, such as Law and Finance. In future, we will train more models on other vertical domains and evaluate their performance.

Acknowledgments

The work is supported by National Natural Science Foundation of China (No. 62172039U21B2009 and 62276110) and MIIT Program(CEIEC-2022-ZM02-0247).

References

[1] Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Kenneth Fisher, Philip Pham, Anirudh Ravula, Sumit K. Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding Long and Structured Inputs in Transformers. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:221845203

[2] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra-Aimée Cojocaru, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. *ArXiv* abs/2311.16867 (2023). https://api.semanticscholar.org/CorpusID:265466629

[3] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu A. Ha, Rodney Michael Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler C. Murray, Hsu-Han Ooi, Matthew E. Peters, Joanna L. Power, Sam Skjonsberg, Lucy Lu Wang, Christopher Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the Literature Graph in Semantic Scholar. In *North American Chapter of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:19170988

[4] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403* (2023).

[5] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *Conference on Empirical Methods in Natural Language Processing* (2019).

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *In Proceedings of NeurIPS* (2020).

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

[8] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. *ArXiv* abs/2004.07180 (2020). https://api.semanticscholar.org/CorpusID:215768677

[9] Together Computer. 2023. *RedPajama: An Open Source Recipe to Reproduce LLaMA training dataset.* https://github.com/togethercomputer/RedPajama-Data

[10] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011* (2021).

[11] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314* (2023).

[12] Luca Foppiano, Guillaume Lambard, Toshiyuki Amagasa, and Masashi Ishii. 2024. Mining experimental data from Materials Science literature with Large Language Models. https://api.semanticscholar.org/CorpusID:267068955

[13] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *ArXiv* abs/2312.10997 (2023). https://api.semanticscholar.org/CorpusID:266359151

[14] Xinyang Geng and Hao Liu. 2023. *OpenLLaMA: An Open Reproduction of LLaMA.* https://github.com/openlm-research/open_llama

[15] Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent Chunking Mechanisms for Long-Text Machine Reading Comprehension. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:218674216

[16] Xanh Ho, Anh Khoa Duong Nguyen, An Tuan Dao, Junfeng Jiang, Yuki Chida, Kaito Sugimoto, Huy Quoc To, Florian Boudin, and Akiko Aizawa. 2024. A Survey of Pre-trained Language Models for Processing Scientific Text. https://api.semanticscholar.org/CorpusID:267335082

[17] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Isuhii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55 (2022), 1 – 38. https://api.semanticscholar.org/CorpusID:246652372

[18] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. *ArXiv* abs/2305.06983 (2023). https://api.semanticscholar.org/CorpusID:258615731

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

[20] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Neural Information Processing Systems* (2020).

[21] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:964287

[22] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *Comput. Surveys* 55 (2021), 1 – 35. https://api.semanticscholar.org/CorpusID:236493269

[23] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *Conference on Empirical Methods in Natural Language Processing* (2018).

[24] Xiaozhong Lyu, Stefan Grafberger, Samantha Biegel, Shaopeng Wei, Meng Cao, Sebastian Schelter, and Ce Zhang. 2023. Improving Retrieval-Augmented Large Language Models via Data Importance Learning. *ArXiv* abs/2307.03027 (2023). https://api.semanticscholar.org/CorpusID:259360590

[25] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting for Retrieval-Augmented Large Language Models. *ArXiv* abs/2305.14283 (2023). https://api.semanticscholar.org/CorpusID:258841283

[26] Antonello Meloni, Simone Angioni, Angelo Salatino, Francesco Osborne, Diego Reforgiato Recupero, and Enrico Motta. 2023. Integrating Conversational Agents and Knowledge Graphs Within the Scholarly Domain. *IEEE Access* 11 (2023), 22468–22489. https://api.semanticscholar.org/CorpusID:257382593
[27] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. *arXiv preprint arXiv:1704.06879* (2017).
[28] Inderjeet Nair, Shwetha Somasundaram, Apoorv Saxena, and Koustava Goswami. 2023. Drilling Down into the Discourse Structure with LLMs for Long Document Question Answering. *ArXiv* abs/2311.13565 (2023). https://api.semanticscholar.org/CorpusID:265351878
[29] Yuxiang Nie, Heyan Huang, Wei Wei, and Xian ling Mao. 2022. Capturing Global Structural Information in Long Document Question Answering with Compressive Graph Selector Network. *ArXiv* abs/2210.05499 (2022). https://api.semanticscholar.org/CorpusID:252815949
[30] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).
[31] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:11080756
[32] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116* (2023). arXiv:2306.01116 https://arxiv.org/abs/2306.01116
[33] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *North American Chapter of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:231815627
[34] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *ArXiv* abs/2302.00083 (2023). https://api.semanticscholar.org/CorpusID:256459451
[35] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3 (2009), 333–389. https://api.semanticscholar.org/CorpusID:207178704
[36] Francesco Ronzano and Horacio Saggion. 2015. Dr. Inventor Framework: Extracting Structured Information from Scientific Publications. In *IFIP Working Conference on Database Semantics*. https://api.semanticscholar.org/CorpusID:42116649
[37] Murray Shanahan. 2022. Talking about large language models. *arXiv preprint arXiv:2212.03551* (2022).
[38] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. *ArXiv* abs/2305.15294 (2023). https://api.semanticscholar.org/CorpusID:258866037
[39] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:233240939
[40] Yixuan Su, Tian Lan, and Deng Cai. 2023. OpenAlpaca: A Fully Open-Source Instruction-Following Model Based On OpenLLaMA. https://github.com/yxuansu/OpenAlpaca.
[41] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.
[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).
[43] Binglei Wang, Tong Xu, Hao Wang, Yanmin Chen, Le Zhang, Lintao Fang, Guiquan Liu, and Enhong Chen. 2020. Author Contributed Representation for Scholarly Network. In *APWeb/WAIM*. https://api.semanticscholar.org/CorpusID:224771817
[44] Haifeng Wang, Jiwei Li, Hua Wu, Eduard H. Hovy, and Yu Sun. 2022. Pre-Trained Language Models and Their Applications. *Engineering* (2022). https://api.semanticscholar.org/CorpusID:252129834
[45] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-Instruct: Aligning Language Model with Self Generated Instructions. *ArXiv* abs/2212.10560 (2022).
[46] Ai Ming Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Hai Zhao, Hang Xu, Hao-Lun Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, Juntao Dai, Kuncheng Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Pei Guo, Ruiyang Sun, Zhang Tao, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yan-Bin Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open Large-scale Language Models. *ArXiv* abs/2309.10305 (2023). https://api.semanticscholar.org/CorpusID:261951743
[47] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414* (2022).
[48] ChengXiang Zhai. 2008. Statistical Language Models for Information Retrieval: A Critical Review. *Found. Trends Inf. Retr.* 2 (2008), 137–213. https://api.semanticscholar.org/CorpusID:61572040
[49] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. *ArXiv* abs/1904.09675 (2019).
[50] Xiao Zhang, Heqi Zheng, Yuxiang Nie, Heyan Huang, and Xian-Ling Mao. 2023. SciMRC: Multi-perspective Scientific Machine Reading Comprehension. *arXiv preprint arXiv:2306.14149* (2023).
[51] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. *ArXiv* abs/2211.14876 (2022). https://api.semanticscholar.org/CorpusID:254044526
[52] Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. Document Modeling with Graph Attention Networks for Multi-grained Machine Reading Comprehension. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:218595722
[53] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *ArXiv* abs/2306.05685 (2023). https://api.semanticscholar.org/CorpusID:259129398
[54] Yunchang Zhu, Liang Pang, Yanyan Lan, Huawei Shen, and Xueqi Cheng. 2021. Adaptive Information Seeking for Open-Domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:237502990

A ALGORITHMS

A.1 Build Document Tree

This algorithm reflects how to build the document tree on extracted list of sections.

Algorithm 1 Build Document Tree

Input: the main body list L, where each item is a section
Output: the root node Head of the main body tree

function BuildTree(L)
if the length of $L = 0$ **then**
  return None
end if
$section = L$.pop(0)
Extract id, $content$, $body$ from $section$
Initial TreeNode N with id, $content$, $body$
if id is abnormal **then**
  return None
end if
while the length of $L > 0$ **do**
  extract id_{next} from $L[0]$
  if id_{next} is abnormal or $id \in id_{next}$ **then**
    $child$ = BuildTree(L)
    N add $child$ to the children
  else
    break
  end if
end while
return N
end function

A.2 Root-Leaf Traversal

This algorithm reflects how to extract all paths from root node to leaf nodes of a document tree.

Algorithm 2 Root-Leaf Traversal

Input: the root node T of a document tree
Output: a list of the context from root node to each leaf node

```
function root_leaf_traversal(T)
if the length of T.children = 0 then
    define template1
    prompts = [template1.format(components)]
else
    define template2
    prompts = []
    for child in T.children do
        for temp in root_leaf_traversal(child) do
            prompts.append(template2.format(components, temp))
        end for
    end for
end if
return prompts
end function
```

B REPRODUCIBILITY

In this paper, we use greedy algorithm to generate responses, ensuring the reproduction. And the specific parameters are shown in Table 9.

Table 9: The hyper-parameters during continueing pretraining and fine-tuning.

Parameters	Value
Zero stage	2
Warmup ratio	0.05
Pre-train Step	122k
Fine-tune Step	2k h
Sequence Length	4096
Batch Size	64
Gradient Clipping	1.0
LoRA Rank	64
LoRA Alpha	16.0
LoRA Dropout	0.1
LoRA weights	> 0.16B

C Experiment

C.1 Dense Retrieval Model

To improve practical retrieval performance in scientific domain, we simply train an dense retrieval model based on the dual-encoder architecture, named as SciDPR. Then we choose some classical retrieval model as baselines, including TF-IDF, LED [10] and OpenAI Embedding. The experimental results are shown in Table 10, from which we could make following conclusions: (1) dense retrieval models (SciDPR and OpenAI embedding) achieves the competitive performance with the cross-encoder baselines (LED-base and LED-base$_{\text{InfoNCE}}$). In the view of the much lower inference cost, the SciDPR and OpenAI embedding are more practical in real-world scenarios; (2) Our proposed SciDPR achieves better Evidence-F1 scores than OpenAI embedding service on Dev set and test set, indicating its better performance in scientific domain. Although the performance gap between our model and OpenAI embedding serve is not significant, in the view of the relative high cost of using OpenAI APIs, our SciDPR that deployed locally is less expensive and faster.

Table 10: Expermental results of SciDPR and start-of-art retrieval models on scientific benchmark QASPER. The evidence F1 is used to evaluate the performance.

Model	Dev set	Test set
First Paragraph	0.71	0.34
Random Paragraph	2.09	1.30
TF-IDF	10.68	9.20
LED-base	23.94	29.85
LED-base$_{\text{InfoNCE}}$	24.90	30.60
OpenAI Top-1	24.41	30.00
OpenAI Top-2	26.67	31.46
OpenAI Top-3	26.37	29.71
OpenAI Best	35.04	40.17
SciDPR Top-1	25.41	30.38
SciDPR Top-2	26.74	30.98
SciDPR Top-3	26.68	29.58
SciDPR Best	36.83	**40.67**
Human$_{lb}$	-	71.62

C.2 Case Study

In this subsection, some qualitative cases of our trained model, LLaMA$_{\text{sft}}$ and ChatGPT are shown in Table 11. It can be found that our proposed model could generate response to user's queries more concisely and correctly than the widely-used LLaMA$_{\text{sft}}$, which is competitive with ChatGPT.

C.3 Prompt

The prompt used to fine-tune models and generate responses is similar to Alpaca [5]. The only difference is to change the *Input* to *Evidence*, indicating the retrieved or extracted context. In the same way, we use *Instruction* to describe the query, which is the question in Question Answer and the task description in Keyphrase Generation and Abstract Summary.

[5] https://github.com/tatsu-lab/stanford_alpaca#data-release/

Table 11: Qualitative cases of our trained model, $\text{LLaMA}_{\text{sft}}$ and ChatGPT. It can be found that our model could generate concise and accurate response according to the evidence. In contrast, the hallucination problem of $\text{LLaMA}_{\text{sft}}$ is severe. The text in bold are the problematic content.

Question: How do they define rumors? **Ground-Truth Evidence:** To increase instantaneous detection performance, we compensate for the absence of future information by consulting additional data sources. In particular, we make use of news wire articles, which are considered to be of high credibility. This is reasonable as according to Petrovic et. al (2013), in the majority of cases, news wires lead social media for reporting news. When a message arrives from a social media stream, we build features based on its novelty with respect to the confirmed information in the trusted sources. In a nutshell, the presence of information unconfirmed by the official media is construed as an indication of being a rumour. Note that this closely resembles the definition of what a rumour is.		
Our Model: The answers are shown as follows: a rumour is a piece of information that is not confirmed by the official media.	**$\text{LLaMA}_{\text{sft}}$:** The answers are shown as follows: **rumour detection is hard** because the most accurate systems operate retrospectively, only recognising rumours once they have collected repeated signals. we compensate for the absence of future information by consulting additional data sources. In particular, we make use of news wire articles, which are considered to be of high credibility. This is reasonable as according to Petrovic et. al (2013), in the majority of cases, news wires lead social media for reporting news. When a message arrives from a social media stream, we build features based on its novel.	**ChatGPT:** They define rumors as information unconfirmed by the official media.
Question: Which NER dataset do they use? **Ground-Truth Evidence:** We evaluate our model in two English NER datasets and four Chinese NER datasets. (1) CoNLL2003 is one of the most evaluated English NER datasets, which contains four different named entities: PERSON, LOCATION, ORGANIZATION, and MISC BIBREF34. (2) OntoNotes 5.0 is an English NER dataset whose corpus comes from different domains, such as telephone conversation, newswire. We exclude the New Testaments portion since there is no named entity in it BIBREF8, BIBREF7. This dataset has eleven entity names and seven value types ... (3) BIBREF35 released OntoNotes 4.0. We use the Chinese part. We adopted the same pre-process as BIBREF36. (4) The corpus of the Chinese NER dataset MSRA came from news domain BIBREF37. (5) Weibo NER was built based on text in Chinese social media Sina Weibo BIBREF38, and it contained 4 kinds of entities. (6) Resume NER was annotated by BIBREF33.		
Our Model: The answers are shown as follows: (1) CoNLL2003 (2) OntoNotes 5.0 (3) OntoNotes 4.0 (4) MSRA (5) Weibo NER (6) Resume NER.	**$\text{LLaMA}_{\text{sft}}$:** The answers are shown as follows: CoNLL2003, OntoNotes 5.0, **BIBREF35, BIBREF36, BIBREF37**, MSRA, Weibo NER. Resume NER.	**ChatGPT:** They use two English NER datasets (CoNLL2003 and OntoNotes 5.0) and four Chinese NER datasets (OntoNotes 4.0, MSRA, Weibo NER, and Resume NER).