Refiner: Restructure Retrieval Content Efficiently to Advance Question-Answering Capabilities

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are limited by their parametric knowledge, leading to hallucinations in knowledge-extensive tasks. To address this, Retrieval-Augmented Generation (RAG) incorporates external document chunks to expand LLM knowledge. Furthermore, compressing information from document chunks through extraction or summarization can improve LLM performance. Nonetheless, LLMs still struggle to notice and utilize scattered key information, a problem known as the "lost-in-the-middle" syndrome. Therefore, we typically need to restructure the content for LLM to recognize the key information. We propose *Refiner*, an end-to-end extract-and-restructure paradigm that operates in the post-retrieval process of RAG. *Refiner* leverages a single decoder-only LLM to adaptively extract query-relevant contents verbatim along with the necessary context, and section them based on their interconnectedness, thereby highlights information distinction, and aligns downstream LLMs with the original context effectively. Experiments show that a trained *Refiner* (with 7B parameters) exhibits significant gain to downstream LLM in improving answer accuracy, and outperforms other state-of-the-art advanced RAG and concurrent compressing approaches in various single-hop and multi-hop QA tasks. Notably, *Refiner* achieves a 80.5% tokens reduction and a 1.6-7.0% improvement margin in multi-hop tasks compared to the next best solution. *Refiner* is a plug-and-play solution that can be seamlessly integrated with RAG systems, facilitating its application across diverse open-source frameworks.

1 Introduction

Recent Large Language Models (LLMs) (OpenAI et al., 2024) have demonstrated their remarkable abilities in understanding query (Brown et al., 2020) (Ouyang et al., 2022) and generating human language texts. Nonetheless, LLMs still face significant limitations, particularly in domain-specific (Zhang et al., 2024) or knowledge-intensive (Kandpal et al., 2023) tasks, where they often hallucinate (Zhang et al., 2023) when handling queries beyond their parametric knowledge (Muhlgay et al., 2024). To address this issue, Retrieval-augmented generation (RAG) (Lewis et al., 2021) has been rapidly developed (Gao et al., 2024), leveraging external knowledge base to retrieve document chunks semantically similar to the query and incorporate them into LLMs, thereby generating more faithful (Khandelwal et al., 2020) and generalizable (Kamalloo et al., 2023) outputs.

However, generating answers from the top retrieved document chunks (Karpukhin et al., 2020) indiscriminately has various shortcomings. Due to the potential ambiguity or insufficiency of information in the query, the chunks could become contradictory and distracting, leading to downstream LLMs (Izacard et al., 2022) to hallucinate (Zhang et al., 2023) (Ji et al., 2023). To mitigate this issue, researchers have proposed advanced RAG approaches, such as adopting query rewriting mechanism (Chan et al., 2024) or self-reflection tokens (Asai et al., 2023) during the generation process, which enhances the performance of RAG by enabling on-demand retrieval and selecting optimal answer generated from each document chunk. Meanwhile, others focus on compressing input prompt by employing a language model (LM) (Anderson et al., 2022) (Yang et al., 2023) (Jiang et al., 2023a) to exclude irrelevant content, thereby reduces computational costs and alleviate "lost-in-the-middle" (Wang et al., 2023) syndrome.

Despite these advancements, LMs are still predisposed to neglect key information scattered across multiple chunks. Moreover, the document chunks, retrieved based on semantic proximity to a query, often pertain to identical subjects and occasionally contain similar yet distinguished information. Failing to maintain context and naively concatenat-

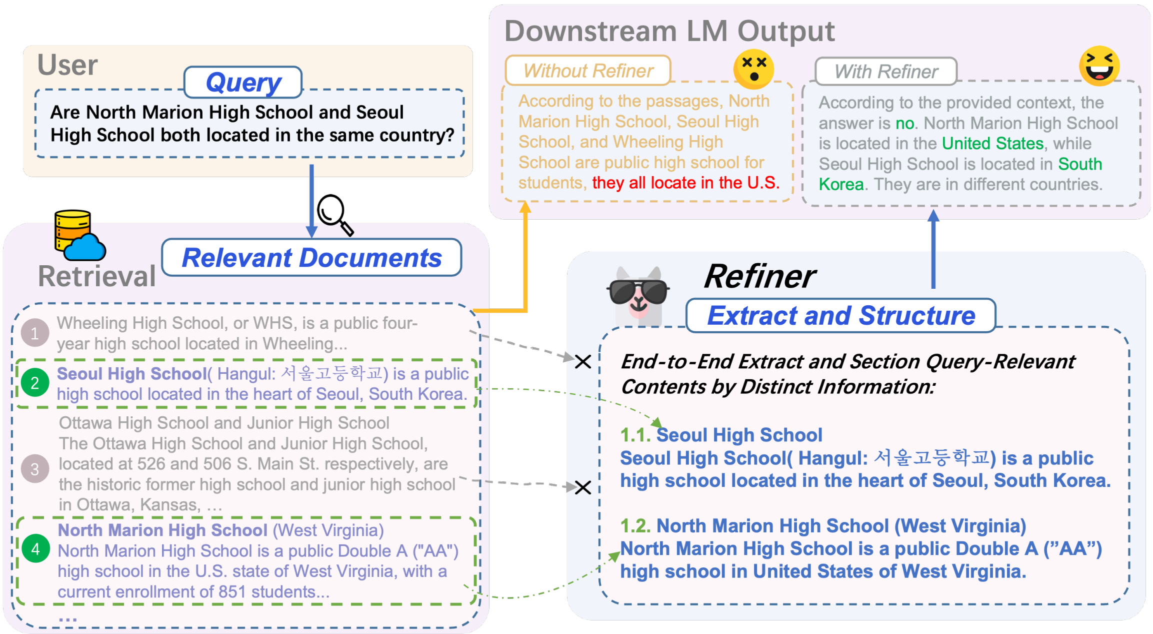

Downstream LM Output
Without Refiner
According to the passages, North Marion High School, Seoul High School, and Wheeling High School are public high school for students, they all locate in the U.S.
With Refiner
According to the provided context, the answer is no. North Marion High School is located in the United States, while Seoul High School is located in South Korea. They are in different countries.
User
Query
Are North Marion High School and Seoul High School both located in the same country?
Retrieval
Relevant Documents
1 Wheeling High School, or WHS, is a public four-year high school located in Wheeling...
2 Seoul High School(Hangul: 서울고등학교) is a public high school located in the heart of Seoul, South Korea.
3 Ottawa High School and Junior High School The Ottawa High School and Junior High School, located at 526 and 506 S. Main St. respectively, are the historic former high school and junior high school in Ottawa, Kansas, ...
4 North Marion High School (West Virginia) North Marion High School is a public Double A ("AA") high school in the U.S. state of West Virginia, with a current enrollment of 851 students...
...
Refiner
Extract and Structure
End-to-End Extract and Section Query-Relevant Contents by Distinct Information:
1.1. Seoul High School
Seoul High School(Hangul: 서울고등학교) is a public high school located in the heart of Seoul, South Korea.
1.2. North Marion High School (West Virginia)
North Marion High School is a public Double A ("AA") high school in United States of West Virginia.

Figure 1: The example shows that misleading document chunks are prone to impeding downstream LM from comprehension and leading to hallucination. While our *Refiner* mitigates the problem by extracting and structuring query-relevant contents by sectioning distinct information, making downstream LM disseminate them effortlessly.

ing these text chunks, can induce confusion and biased comprehension in downstream LM (See examples in Appendix F), exacerbating "lost-in-the-middle" syndrome. This is particularly significant in multi-hop question answering (QA)(Yang et al., 2018) (Ho et al., 2020) task, which requires LM to synthesize information from multiple contents to arrive at a correct answer. Therefore, we typically need to structure the information for downstream LM to differentiate the key information.

We propose *Refiner*, a novel end-to-end extract-and-restructure paradigm that takes into account the relatedness of information across document chunks. The proposal is realized from two approaches: (1). In order for downstream LMs to align with the original context, we propose to keep query-relevant contents verbatim, while also maintain context around those contents on-demand. (2). To emphasize information relatedness among document chunks, we propose to structure the output, as illustrated in the bottom right of Figure 1: The *Refiner* servers as grouping extractive contents by different information using hierarchical section, wherein contents with similar information are grouped in the same section, while contents with disparate information are segregated into separate sections. This approach aims at further clarifying chunk-level information relatedness for downstream LMs to easily comprehend. A case study of how *Refiner* resolves the relatedness information in practice, is illustrated in Table 17.

The *Refiner* leverages an arbitrary pre-trained LLM to generate targeted content, on the assumption that LLM possesses a broad foundation of world knowledge (Zhou et al., 2024) (Li et al., 2024), enabling it to effectively section the intricate, nuanced content-wise relationships. The *Refiner* is trained with knowledge distillation technique (Jung et al., 2024) to adaptively determine the amount of context to retain around key contents, and output with the proposed format.

Empirical results on several QA tasks, including single-hop and multi-hop tasks, demonstrate that *Refiner* significantly improves the answer accuracy of downstream LMs, surpassing the previous state-of-the-art (SOTA) advanced RAG solution and concurrent prompt compressing work by a margin of 2.2%-7.0% on two multi-hop QA datasets, and attains comparable accuracy on three single-hop QA datasets with a token reduction of 77.8% on average, compared to the second-best solution.

The *Refiner*'s plug-and-play nature make it ideal for API-based models without parameter access. Once trained, the *Refiner* can be applied across different upstream retrieval systems and downstream LLMs. The structured output, containing sections, titles and contents, could be further disassembled, leading to a easier source document retrospection or citation integration for industry application.

2 Related Work

Retrieval Augmented Generation The Retrieval Augmented Generation (RAG) (Lewis et al., 2021) (Gao et al., 2024) fundamentally enhances the capabilities of LLMs by retrieving pertinent documents from an external knowledge base, which is made possible through the calculation of semantic similarity between user's query and documents. Notably, the de facto approach is to segment documents into chunks and construct text representation embedding upon these chunks. By leveraging external knowledge sources like Wikipedia, RAG effectively mitigates the issue of generating factually inaccurate content, thereby greatly enhancing the reliability of generated text. RAG typically involves a retrieval system for query-relevant knowledge indexing from a knowledge base, and a LM that serves as question answering interface.

Advanced RAG Advanced RAG model incorporates multiple optimization strategies that span both pre-retrieval and post-retrieval processes. For example, Self-RAG (Asai et al., 2023) introduces a selective knowledge retrieval mechanism, which utilizes a retrieve token to determine the necessity of retrieval and employs multiple critic tokens to assess answer quality. Building upon this, RQ-RAG (Chan et al., 2024) is trained to dynamically improve query by integrating rewriting, decomposition, and disambiguation functionalities. Furthermore, chain-of-thought (COT) (Wang et al., 2024) and Tree-of-thought (TOT) (Yao et al., 2023) are incorporated to RAG to tackle certain logical-demanding tasks, which proved to have significantly enhanced the performance of LLMs.

Concurrent Post-Retrieval Work Previous research has explored the application of language model-based approaches to further distill or extract query-relevant key information from retrieved document segments (Lewis et al., 2021). Some studies have developed summarization models (Lennox et al., 2023) (Jung et al., 2024) to derive crucial query-relevant information from retrieved document blocks, RECOMP (Xu et al., 2023) has designed a dual-encoder extractive compressor and a encoder-decoder abstractive model to expunge irrelevant context and summarize upon the remainder. Additionally, LongLLMLingua (Jiang et al., 2023a) proposes a question-aware compression method with a configurable compression rate, allowing them to reduce computational cost and latency, while LLMLingua-2 (Pan et al., 2024) formulates prompt compression as a token classification problem, thereby retains the faithfulness of the compressed prompt.

Compared with recent studies (Xu et al., 2023) (Yoran et al., 2024) (Jiang et al., 2023a) that are similar to our *Refiner*, major differences are worth mentioning: These methodologies aim at effective and efficient summarization/extraction without considering the semantic relatedness among them, while we dedicate to retain relatedness information by explicitly sectioning them to form a more structured content. Furthermore, we take contextual completeness into account for downstream LMs to align with the original context effortlessly.

3 Problem Formulation

Following previous work (Xu et al., 2023), we use $\mathcal{X} = \{q_1, \mathcal{D}_1, ..., q_\text{i}, \mathcal{D}_\text{i}\}$ to represent *Refiner*'s input sequence set, where q_i denotes query and $\mathcal{D}_\text{i} = \{d_\text{i}^1, ..., d_\text{i}^{\mathcal{K}}\}$ denotes the corresponding top-$\mathcal{K}$ retrieved documents at step i. The *Refiner* is designated to extract from $\mathcal{D}_\text{i}$ with respect to q_i to generate output sequence y_i.

The output sequences of the model y_i should satisfy the following principles: (1) **Concise**: The output should be verbatim and context-completed w.r.t retrieved documents, such that $y_\text{i} \subseteq \mathcal{D}_\text{i}$. (2) **Sectioned**: contents extracted from different document chunks should all be numeric sectioned. Contextually related contents should be grouped into the same section. Conversely, contents with different information should be grouped to different sections. (3) **Succinct**: The output length should be significantly shorter than total length of the retrieved documents $\mathcal{D}_\text{i}$, such that $|y_\text{i}| \ll |D_\text{i}|$. All contents should be helpful to response to the query. If none of the documents d is relevant, then $|y_\text{i}| = 0$.

4 Proposed Method: *Refiner*

The primary objective of our research is to develop a model that leverages intrinsic knowledge to structure content relevant to the question in the post-retrieval process. Empirical study (Hu et al., 2024) shows that models with larger parameters tend to better conform their output format to instructions and exhibit enhanced capabilities in comprehension. However, larger models are accompanied by a quadratic increase in computational cost pro-

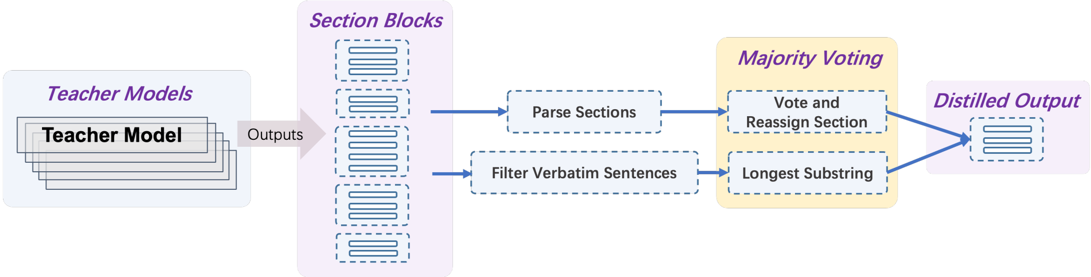


Figure 2: Knowledge distillation for *Refiner*. The curated training data is generated by multiple teacher models, which undergoes parsing, filtering, voting to improve section and context expediency.

portional to the length of the prompt. In our case, where the model is required to respond with brief content from a discursive, lengthy context, the computational cost necessitates particular attention. Therefore, we propose to fine-tune a smaller model in order to reign the computational costs and enhance its capabilities to conform to our proposed structured output format.

4.1 Knowledge Distillation

Accurate extraction of query-relevant content with pertinent context is crucial for the *Refiner* model. To achieve this, we develop a training dataset that teaches *Refiner* to determine the optimal scope of context. While a simple approach to maintain context might involve selecting a few sentences surrounding the key content from the original document chunk, this risks introducing irrelevant context. Instead, we rely on larger LLMs to adaptively determine the extent of context to retain. Specifically, we employ knowledge distillation (Jung et al., 2024) technique to train *Refiner*, where a simpler student model approximates the behavior of more intricate teacher models (Zuchniak, 2023).

However, single teacher model can be imperfect, occasionally producing unstructured output, forgetting to provide context or grouping with erroneous section. To address this, we utilize multi-teacher knowledge distillation (Zuchniak, 2023), where multiple teacher models generate outputs from the same input, and the optimal output segment is selected to create a curated dataset.

As depicted in Figure 2, we utilize five teacher models, which are prompted to section context-completed query-relevant content from provided document chunks. The generated sentences are rigorously verified against the original document chunks to ensure verbatimness. The section parts are voted on and rearranged, and only the sentences generated by a majority of the models are considered relevant context. The *Refiner* is subsequently trained on concatenated sentences with section restoration. We design diverse prompts on various QA datasets for teacher models to consistently provide the proposed extraction. See detailed data preparation procedure in Figure 4, composition of teacher model and hyperparameters in Table 6, and task-specific prompt templates in Table 8.

4.2 Model Training

Now that we have constructed the training dataset, we use Supervised Fine-Tuning (SFT) (Sun, 2024) to train a pre-trained Llama-2-7B-Chat (Touvron et al., 2023) model. In particular, given a curated high-quality train dataset $\mathcal{S}_{\mathcal{SFT}} = \{\mathcal{X}_i, D_\mathrm{i}, \mathcal{Y}_i\}_{i=1}^N$, where N denotes number of samples in the dataset, the goal is to gain a model parameters $\mathcal{M}_\theta$ with the following training objective function:

$$\underset{\theta}{minimize}\mathcal{L}(\theta;\mathcal{S}_{SFT}) = -\mathbb{E}_{\mathcal{X},\mathcal{Y}\sim\mathcal{S}_{SFT}}[\log \mathcal{M}_\theta(\mathcal{Y}|\mathcal{X})] \tag{1}$$

where $\mathcal{L}$ denotes the negative likelihood objective function to minimize, $\mathcal{M}_\theta$ represents the model parameters. The expectation $\mathbb{E}_{\mathcal{X},\mathcal{Y}\sim\mathcal{S}_{SFT}}$ averages over our dataset. Note that negative likelihood is adopted to formulate the loss function. Following previous works (Lewis et al., 2021) (Asai et al., 2023) (Yan et al., 2024) (Xu et al., 2023), we train *Refiner* $\mathcal{M}_\theta$ to generate a textual sequence $\{y_\mathrm{i}\}_{i=1}^N$ that extracts core information and context in $\mathcal{D}$ relevant to q with fewer tokens than $\mathcal{D}$. To ensure our model can effectively handle training datasets of varying sizes, we employ a progressive training approach (Wu et al., 2024) for the *Refiner*. More training details are in the Appendix B.

5 Experiments

We assess the effectiveness of our paradigm on two distinct criteria: its potential to enhance the performance of downstream LMs and its relative efficacy compared to other RAG approaches. We integrate it with various downstream LMs and open-domain question answering datasets, building upon the methodologies established in prior researches (Asai et al., 2023) (Yan et al., 2024) (Xu et al., 2023).

5.1 Evaluate Tasks and Datasets

We evaluate our *Refiner* and downstream LMs on a diverse open-domain QA tasks, including short-form QA, long-form QA and multi-hop QA tasks. Throughout these experiments, we conduct zero-shot evaluations, where the prompts only provide instructions on tasks and outputs in contrast to few-shot examples (Wei et al., 2022) (Sanh et al., 2022). Detailed experiment environments, including prompt templates for *Refiner* and downstream LMs, are listed in the Appendix C.

Following previous work (Asai et al., 2023), we use accuracy as evaluation metric on all test dataset. Notably, the applied accuracy metric is reported to not always reflects the performance of the RAG system, as it looks up the existence of answer string in the generated output, omitting the semantic coherence which results in biased evaluation. In ARC Challenge task, we construct a regular expression-based calculation, which may influence the results compared with other reported scores from advanced RAG solutions. See subsection C.6 for our motivation and methodology for evaluating ARC Challenge task specifically.

Single-Hop QA Dataset includes three datasets, PopQA (Mallen et al., 2023), TriviaQA-unfiltered (Joshi et al., 2017) and Reasoning Challenge (ARC-Challenge) (Clark et al., 2018). The RAG system needs to answer arbitrary questions about factual knowledge for PopQA and TriviaQA, and multiple-choice questions task about scientific event that occurs in daily life for ARC Challenge. For PopQA, we utilize long-tail subset, consisting of 1,399 rare entity queries with monthly Wikipedia page views of less than 100. The ARC Challenge consists of 1,172 QA samples. For TriviaQA, we employ the established validation and test split methodology adopted in prior research (Asai et al., 2023) (Min et al., 2019) (Guu et al., 2020), using 11,313 test queries for evaluation. We employ Contriever-MS MARCO (Izacard et al., 2021) as our retriever and retrieve ten documents for each query by default.

Multi-hop QA Dataset includes HotpotQA-dev-distractor (Yang et al., 2018) dataset with 7,405 samples, and 2Wiki-dev (Ho et al., 2020) dataset with 12,576 samples. In contrast to the single-hop datasets, where document chunks are retrieved using our default retriever, we directly utilize the relevant content provided by these datasets.

5.2 Advanced RAG and Concurrent Work

We compare *Refiner*-augmented LMs with other advanced RAG solutions, including Self-RAG (Asai et al., 2023) and RQ-RAG (Chan et al., 2024). Additionally, we assess the performance of concurrent compressors, including RECOMP abstractive compressor (Xu et al., 2023) and LongLLMLingua (Jiang et al., 2023a). Hyperparameter settings are detailed in the subsection C.1 and subsection C.2. To facilitate a fair comparison, we maintain consistency with the experimental frameworks, utilizing the same retriever framework and downstream LMs as those used in *Refiner*.

6 Results

6.1 *Refiner* consistently outperforms other approaches on multi-hop QA task

As shown in Table 1, our *Refiner* significantly outperforms existing advanced RAG variants and concurrent compressors on multi-hop QA tasks, while demonstrating competitive performance on single-hop QA tasks. Compared to the next best solutions, our *Refiner*-augmented system registers a performance margin of 7.0% on HotpotQA task, and 6.4% on 2WikiMultihop task. Furthermore, the *Refiner*-augmented LMs bridges the performance gap between small-scale and extreme-scale LLM on TriviaQA and HotpotQA tasks, surpasses GPT-3.5-Turbo model on PopQA, HotpotQA and 2WikiMultihop task, highlighting the substantial contribution of our *Refiner* component to RAG. As anticipated, the performance gains fall short on single-hop tasks compared to LongLLMLingua, revealing that the structured output that distinguishes information disparity produced by *Refiner* has a less pronounced impact on single-hop tasks than on multi-hop tasks.

6.2 *Refiner*'s output is highly utilized by LMs

To quantify the efficiency among compressors, we compare their compression rates by dividing the average number of output tokens generated by the compressors by the average number of input tokens in the document chunks, then subtracting the value from one. (see Table 2). We use Llama2-7B Chat tokenizer throughout the experiment. The results reveal that the RECOMP abstractive compressor

	Single-Hop			Multi-Hop	
LM	**POPQA** (acc)	**TQA** (acc)	**ARC-C** (acc)	**HOTPOT** (acc)	**2WIKI** (acc)
Proprietary LLM					
GPT-3.5-Turbo∗	29.3	74.3	75.3	57.3	54.3
Advanced RAG					
Self-RAG$_{7B}$∗	54.9	66.4	47.3	41.6	39.1
Self-RAG$_{13B}$∗	55.8	69.3	53.1	47.5	45.7
RQ-RAG∗	57.1	-	57.4	62.6	44.8
***With Downstream LM:* Llama2-Chat$_{7B}$**					
No Compressor	55.4	67.5	50.5	60.7	57.9
RECOMP$_{abstractive}$	52.6	65.0	18.7	47.3	58.5
LongLLMLingua	**60.3**	**70.5**	51.6	61.3	55.3
Refiner	59.3	68.9	**51.8**	**65.1**	**61.0**
***With Downstream LM:* Llama2-Chat$_{13B}$**					
No Compressor	54.8	71.8	57.3	63.8	58.4
RECOMP$_{abstractive}$	53.4	65.8	49.2	49.2	58.7
LongLLMLingua	58.3	**71.6**	**57.8**	60.1	56.2
Refiner	**59.4**	70.6	55.1	**67.1**	**61.6**
***With Downstream LM:* Meta-Llama3-Instruct$_{8B}$**					
No Compressor	64.2	74.4	77.1	70.4	67.3
RECOMP$_{abstractive}$	50.1	63.3	48.0	46.2	56.1
LongLLMLingua	59.3	**71.2**	**72.0**	64.9	54.2
Refiner	**59.7**	69.3	71.1	**67.2**	**63.7**

Table 1: Accuracy on single-hop and multi-hop tasks. We report results of proprietary LLM, advanced RAG that works without *Refiner*, compared to off-the-shelf Llama2-(7,13)B-Chat and Meta-Llama3-Instruct-8B augmented by *Refiner*. **Bold** numbers indicate the best performance among compressors. * indicates results reported by concurrent work. - indicates the corresponding task is not reported by the original paper.

Task	RECOMP (avg #token)	LongLLMLingua (avg #token)	*Refiner* (avg #token)
POPQA	21.0	764.6	143.1
TQA	144.1	1,049.0	206.7
ARC-C	20.7	1,022.2	280.2
HOTPOT	28.0	757.9	130.0
2WIKI	64.6	578.6	131.2
Compress Rate	96.6%	48.9%	89.1%

Table 2: Average number of output tokens and compress rates among tasks and compressors.

achieves the highest compression rate, at 96.6%, followed by *Refiner*, which compresses 89.1% of input tokens. LongLLMLingua's compression rate, set as a hyperparameter to 50%, results in an actual compression rate of approximately 48.9%.

Our *Refiner* achieves a comparable performance to LongLLMLingua in single-hop QA tasks, with providing only 22.2% of the average number of tokens, and surpasses it in multi-hop QA tasks with 19.5% of tokens, demonstrating high utilization of *Refiner*'s output by downstream LMs, attributable to its structured and context-completed output.

6.3 *Refiner*-augmented system is less susceptible to noisy and lengthy content

Recent study (Xu et al., 2023) reveals that appending irrelevant documents can negatively impact the performance of RAG, our *Refiner* has successfully helped downstream LMs to mitigate the limitation by significantly reducing prompt length while keeping the performance of downstream LMs. Notably, our experimental results show that even when five additional irrelevant document chunks are appended, the performance of downstream LMs remains steady. In fact, as illustrated in Table 3, the accuracy of downstream LMs in both single-hop and multi-hop tasks has actually experienced an upswing. This content length tolerance exhibited by *Refiner*-augmented system could alleviate the lost-in-the-middle (Wang et al., 2023) phenomenon, commonly observed in downstream LMs. Such tolerance can be attributed to our training data, which deliberately organizes input document chunks in random order, thereby forcing the model to search and extract key information from full context.

6.4 The *Refiner* is resilient to RAG system

To conduct a thorough investigation of the effectiveness coherency of *Refiner* within different RAG settings, we undertook evaluation tasks using a range of *Refiner*-augmented LMs and varied top retrieval settings, as illustrated in Table 3. Notwithstanding the variations in content lengths and downstream LMs, the in-task performance exhibits its stability. As shown in Table 3, the standard deviation of in-task accuracy remains relatively constant,

	Single-Hop			Multi-Hop	
	POPQA	TQA	ARC-C	Hotpot	2Wiki
LM	(acc)	(acc)	(acc)	(acc)	(acc)
***Refiner* of top 5 document chunks**					
Refiner + Llama2-Chat$_{\text{7B}}$	59.3	68.9	51.1	47.4	46.0
Refiner + Llama2-Chat$_{\text{13B}}$	59.0	70.5	54.2	49.7	45.7
Refiner + Llama2-Chat$_{\text{70B}}$	58.3	72.2	55.9	52.3	50.1
Refiner + Meta-Llama3-Instruct$_{\text{8B}}$	58.3	69.1	71.8	49.0	48.0
Refiner + Meta-Llama3-Instruct$_{\text{70B}}$	57.9	70.9	84.1	49.2	46.7
STD	*0.57*	*1.36*	*14.06*	*1.77*	*1.80*
***Refiner* of top 10 document chunks**					
Refiner + Llama2-Chat$_{\text{7B}}$	59.3	68.9	51.8	65.1	61.0
Refiner + Llama2-Chat$_{\text{13B}}$	59.4	70.6	55.1	67.1	61.6
Refiner + Llama2-Chat$_{\text{70B}}$	60.8	72.1	57.0	69.8	68.3
Refiner + Meta-Llama3-Instruct$_{\text{8B}}$	59.7	69.3	71.1	67.2	63.7
Refiner + Meta-Llama3-Instruct$_{\text{70B}}$	59.8	71.1	84.7	68.6	61.8
STD	*0.61*	*1.31*	*13.75*	*1.77*	*2.98*

Table 3: Results of various downstream LMs in Meta Llama family augmented by *Refiner* with top 5 and top 10 retrieved document chunks. A low standard deviation of accuracy score across downstream LMs in-task, indicating that the *Refiner* is agnostic to both the downstream LM and the upstream retrieval content length.

demonstrating its ubiquitous effectiveness, with the exception of the ARC Challenge task. This is due to the retrieved content's limited assistance with challenging questions, which necessitates downstream LMs to utilize intrinsic knowledge for deduction and extrapolation, eventually highlights the superiority of more sophisticated models.

7 Ablation Study and Analysis

To assess the efficacy of the proposed structured output, an ablation study is implemented by eliminating the structural components from the *Refiner*'s output, resulting in naively concatenating the extracted content without any structural organization. As illustrated in Table 4, we evaluate the impact of the removed structure on downstream tasks by testing models equipped with the expunged outputs, This includes using Llama2-(7B, 13B, 70B)-Chat, Meta-Llama3-(8B, 70B)-Instruct models, with both the top 5 and top 10 retrieval configurations. More detailed table from which data in Table 4 is derived, are presented in Table 12 and Table 13. Furthermore, we have conducted various revisions of structure format, the format and corresponding performance statistics are attached in Appendix E.

7.1 Task-Level Analysis

Our analysis of the decay in accuracy resulting from the removal of sections reveals significant variations across tasks. As shown in Table 4, the mean accuracy decay calculated for the top 5 and top 10 retrieval blocks indicates that sections make a substantial contribution to accuracy in most tasks, with most accuracy decays being positive. One notable exception is the ARC Challenge task, which experiences a decline in accuracy when sections are added. Further investigation reveals that this phenomenon stems from the retrieval system's propensity to retrieve documents related to incorrect options, leading to confusion for *Refiner* and a bias towards incorrect information. Encouragingly, the importance of sections varies significantly across tasks, with the overall trend remaining consistent between top retrieval settings.

7.2 Model-Level Analysis

Another phenomenon has caught our attention: all downstream LMs under the top 10 retrieval setting have experienced a greater decay in accuracy compared to those under the top 5 retrieval settings. Our explanation is that *Refiner* tends to extract more content from the extra document chunks that are remotely correlated to the query, and separates them from those that are closely related using sections. Without these sections, downstream LMs struggle to disseminate crucial information from the lengthy contents, leading to a degraded accuracy. More detailed statistical data in the Table 4, are attached in the Appendix D.

7.3 How Verbatim is *Refiner*'s Output?

Like many other decoder-only models, *Refiner* is a probabilistic LM (Operationnelle et al., 2001) that sequentially generates the probability distribution of the next token given the preceding tokens. Although our objective is to train the model to produce output that selectively replicates the source content, there is always a possibility of non-conformity, wherein the model fails to reproduce the content verbatim from the given document

	Single-Hop			Multi-Hop	
	POPQA	TQA	ARC-C	Hotpot	2Wiki
LM	(acc decay w/o. structure)				
Top 5 document chunks					
Refiner + Llama2-Chat$_{7B}$	1.6	2.4	-0.6	0.1	-0.3
Refiner + Llama2-Chat$_{13B}$	0.5	1.1	-0.2	0.2	-1.5
Refiner + Llama2-Chat$_{70B}$	0.3	0.8	-2.8	0.1	-0.1
Refiner + Meta-Llama3-Instruct$_{8B}$	0.4	2.3	-2.6	-0.2	0.0
Refiner + Meta-Llama3-Instruct$_{70B}$	0.3	2.0	-1.4	0.5	-0.1
MEAN	*0.6*	*1.7*	*-1.5*	*0.1*	*-0.3*
Top 10 document chunks					
Refiner + Llama2-Chat$_{7B}$	0.9	2.7	1.4	1.9	3.4
Refiner + Llama2-Chat$_{13B}$	1.5	2.3	0.1	0.8	0.2
Refiner + Llama2-Chat$_{70B}$	2.6	1.2	-2.8	0.8	0.2
Refiner + Meta-Llama3$_{Instruct8B}$	1.9	2.1	-2.2	-0.2	0.1
Refiner + Meta-Llama3$_{Instruct70B}$	1.3	1.7	0.3	0.3	0.0
MEAN	*1.6*	*2.0*	*-0.6*	*0.7*	*0.8*

Table 4: Ablation analysis on various downstream LMs in Meta Llama family augmented by *Refiner* with top 5 and top 10 retrieved document chunks. Accuracy decay and task-level statistic are presented by removing section parts from *Refiner*'s structured output.

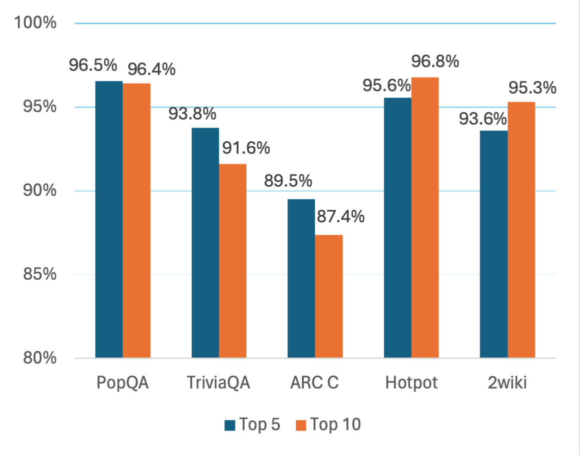


Figure 3: The ratio of contents extracted by *Refiner* that are exactly originate from respective document chunks under various top retrieval settings and tasks.

chunks. The ratio of authentic output generated by *Refiner* for each top retrieval setting and evaluation task dataset is calculated and illustrated in Figure 3. Overall, the model exhibits a range of conformity from 87.4% to 96.8%, whereupon we proceed to investigate its malformed output and discuss the underlying causes of this phenomenon.

Temptation to Answering The major predicament that *Refiner* encounters, is its compulsion to answer to the query rather than extracting relevant information. This is often exacerbated when the query lacks challenge and the information in the document chunk directly answers to the query. Under such circumstances, the model is inclined to repeat the query and the information, rather than engaging in meaningful information extraction.

Temptation to Extract from Query Additionally, *Refiner* is also prone to extracting informative content from the query, a problem that is particularly evident in the ARC-Challenge task. This task is characterized by a query comprising a question and four or more potential answers. The model is more inclined to extract content from these options, which, although potentially incorrect, are nonetheless relevant to the question at hand, thus tempting the *Refiner* model to extract.

8 Conclusion

This work presents a novel paradigm for Retrieval-Augmented Generation (RAG), introducing *Refiner*, a novel end-to-end paradigm that extracts and restructures document chunks, organizing query-relevant context-completed content into sections, thereby enhancing the performance of downstream language models. Thorough evaluations on five open-domain question-answering tasks conclusively demonstrate that *Refiner* outperforms state-of-the-art RAG solutions that employ more complex enhancements, rivaling and surpassing concurrent prompt compress approaches with a significantly higher token compress rate. Our *Refiner* has been tested in collaboration with various downstream language models and retrieval settings, proving to be resilient to the challenges posed by lengthy and noisy document chunks and consistently effective. Finally, *Refiner* is a plug-and-play solution that can be seamlessly integrated with modular RAG systems, facilitating its application across diverse open-source frameworks.

9 Limitations

This study seeks to enhance the quality of Large Language Model (LLM) outputs, a deficiency that continues to have significant real-world consequences, including spread of misinformation and provision of incorrect and dangerous advice. Notwithstanding our method and evaluation have demonstrated efficacy, they are conducted in a relatively indirect way, as how to evaluate the correctness of generated structural output from the model still requires further study. Moreover, our model is trained to extract and section information from open domain document chunks, its robustness has not been tested on alternative input structures, such as table data or domain-specific documents, which raises our concerns about potential misuse. Additionally, due to the nature of *Refiner* being a probabilistic model, there is always a probability of extracting content that does not exactly derive from the document chunks. Therefore, when using them, it is still necessary to exercise caution when distinguishing and verifying the output.

10 Broader Impact

Current RAG system has been found to hinder LLMs from producing misleading (Kwon et al., 2023) or incorrect information (Yan et al., 2024). Although there exist concurrent studies that adopt extractive and summarize models to distill information from retrieved documents in post-retrieval process, none have considered the relatedness of document-wise information, potentially resulting in a loss of contextual information that sometimes critical for downstream LMs to arrive at a correct answer. We believe our *Refiner* is an important step to fill this gap. Furthermore, in contrast to the pre-retrieval and retrieval processes, the post-retrieval process has not been fully explored (Gao et al., 2024) by academia and, in our opinion, holds significant potential. Therefore, we hope that our work will shed light on this process and inspire further research in this area. The *Refiner* dataset, code and model files will be released under the CC BY-NC 4.0 license.

References

Nathan Anderson, Caleb Wilson, and Stephen D. Richardson. 2022. Lingua: Addressing scenarios for live interpretation and automatic dubbing. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 202–209, Orlando, USA. Association for Machine Translation in the Americas.

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

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, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey 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. *Preprint*, arXiv:2005.14165.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *Preprint*, arXiv:2404.00610.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *Preprint*, arXiv:2002.08909.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Hanxu Hu, Pinzhen Chen, and Edoardo M. Ponti. 2024. Fine-tuning large language models with sequential instructions. *Preprint*, arXiv:2403.07794.

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

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Atlas: Few-shot learning with retrieval augmented language models. *Preprint*, arXiv:2208.03299.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Huiqiang Jiang, Qianhui Wu, , Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. LongLLMLingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *ArXiv preprint*, abs/2310.06839.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Preprint*, arXiv:1705.03551.

Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. 2024. Impossible distillation: from low-quality model to high-quality dataset & model for summarization and paraphrasing. *Preprint*, arXiv:2305.16635.

Ehsan Kamalloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. *Preprint*, arXiv:2305.06984.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. *Preprint*, arXiv:2211.08411.

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. *Preprint*, arXiv:2004.04906.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. *Preprint*, arXiv:1911.00172.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. *Preprint*, arXiv:2309.06180.

Connor Lennox, Sumanta Kashyapi, and Laura Dietz. 2023. Retrieve-cluster-summarize: An alternative to end-to-end training for query-specific article generation. *Preprint*, arXiv:2310.12361.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Guozheng Li, Peng Wang, Wenjun Ke, Yikai Guo, Ke Ji, Ziyu Shang, Jiajun Liu, and Zijie Xu. 2024. Recall, retrieve and reason: Towards better in-context relation extraction. *Preprint*, arXiv:2404.17809.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. 2020. Pytorch distributed: Experiences on accelerating data parallel training. *Preprint*, arXiv:2006.15704.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2024. Generating benchmarks for factuality evaluation of language models. *Preprint*, arXiv:2307.06908.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Departement Operationnelle, Y. Bengio, R Ducharme, Pascal Vincent, and Centre Mathematiques. 2001. A neural probabilistic language model.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Ruhle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *ArXiv preprint*, abs/2403.12968.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. *Preprint*, arXiv:2110.08207.

Hao Sun. 2024. Supervised fine-tuning as inverse reinforcement learning. *Preprint*, arXiv:2403.12017.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *Preprint*, arXiv:2311.08377.

Zihao Wang, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024. Rat: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *Preprint*, arXiv:2403.05313.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *Preprint*, arXiv:2402.01364.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *Preprint*, arXiv:2310.04408.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *Preprint*, arXiv:2401.15884.

Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5364–5375, Singapore. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. *Preprint*, arXiv:2310.01558.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *Preprint*, arXiv:2403.10131.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Sizhe Zhou, Yu Meng, Bowen Jin, and Jiawei Han. 2024. Grasping the essentials: Tailoring large language models for zero-shot relation extraction. *Preprint*, arXiv:2402.11142.

Konrad Zuchniak. 2023. Multi-teacher knowledge distillation as an effective method for compressing ensembles of neural networks. *Preprint*, arXiv:2302.07215.

Part I

Appendix

Table of Contents

A Model Cards **14**

B Training Details **15**
- B.1 Dataset 15
- B.2 Teacher Models 15
- B.3 Structure Correction 15
- B.4 Environment and Hyperparameters 15
- B.5 Training Process 16

C Evaluation Details **18**
- C.1 Hyperparameters for Adavanced RAG 18
- C.2 Hyperparameters for Concurrent Compressor 18
- C.3 Environment and Hyperparameters 18
- C.4 Prompt Template for *Refiner* 18
- C.5 Prompt Template for Downstream LM 18
- C.6 Evaluation Metric for ARC Challenge 20

D Ablation Analysis Details **22**

E More Analysis on Output Structure **24**

F Case Study **26**

A Model Cards

Below illustrates models inclusive in our work, all usages of models are under the compliance of their intended use.

Model Name	Description
GPT-3.5-Turbo	Developed by OpenAI, GPT-3.5-Turbo, or ChatGPT, is an advanced NLP model in the GPT series, featuring enhanced context understanding and text generation capabilities. Trained on a vast array of internet text data, it excels in tasks such as text generation, question answering, translation, and programming assistance, finding use in sectors like customer service, content creation, and education (URL: `https://openai.com/chatgpt`).
Meta-Llama3-Instruct (8B, 70B)	Meta-Llama3-Instruct, a cutting-edge language model designed to excel in understanding and responding to natural language instructions. This innovative AI model is capable of comprehending complex directives and generating human-like responses that are both accurate and informative. By leveraging advanced machine learning algorithms and large-scale training datasets, Meta-Llama3-Instruct has achieved unprecedented levels of language understanding and generation capabilities, making it an invaluable tool for a wide range of applications.
Llama2-Chat (7, 13, 70B)	Llama2-Chat (Touvron et al., 2023), developed and publicly released by Meta, is a refined version of LLMs, optimized specifically for conversational purposes. The Llama2-Chat belongs to the Llama 2 family, a series of pre-trained and fine-tuned generative text models with 7 to 70 billion parameters. Being a fine-tuned LLM, Llama2-Chat excels in dialogue scenarios.
Alpaca (7B)	Alpaca (Taori et al., 2023) is a language model that has been fine-tuned from Meta's LLaMA 7B model for instruction-following tasks. Using 52,000 self-instruct style demonstrations, it was trained with OpenAI's text-davinci-003 model for instruction-following tasks. In evaluations, Alpaca has shown behaviors akin to OpenAI's text-davinci-003. Notably, Alpaca is distinguished by its small size, ease of replication, and low cost, making it an efficient, accessible model.
RECOMP Abstractive Compressor	RECOMP (Xu et al., 2023) Abstractive Compressor is a part of the RECOMP (Retrieve, Compress, Prepend) method, which involving retrieving documents and compressing them into textual summaries before they are integrated into the context for the language model. This compression not only reduces computational costs but also makes it easier for the language model to identify relevant information in long retrieved documents. The RECOMP Abstractive Compressor is one of the two types of compressors used in this method, which generates summaries by synthesizing information from multiple documents. This allows the compressor to create a concise summary that contains the key information from the retrieved documents. The Abstractive Compressor is trained to improve the performance of the language model on end tasks when the generated summaries are prepended to the language model's input, while keeping the summary concise. If the retrieved documents are irrelevant to the input or offer no additional information to the language model, the compressor can return an empty string.
LongLLMLingua	LongLLMLingua (Jiang et al., 2023a) is a method designed to compress prompts, thereby enhances the performance of LLMs, reduce computational and financial costs, and decrease latency. By compressing the prompt on token level, it reduces costs and boosts efficiency, improving performance by up to 21.4% using only 1/4 of the tokens. Built upon LLMLingua (Jiang et al., 2023b), the contributions of this paradigm are: (1) a question-aware coarse-to-fine compression method to enhance key information density in the prompt; (2) a document reordering mechanism to minimize information loss; (3) dynamic compression ratios to integrate coarse-grained and fine-grained compression for adaptive granular control; (4) a post-compression subsequence recovery strategy to improve key information integrity.
Self-RAG	Self-Reflective Retrieval-Augmented Generation, or Self-RAG (Asai et al., 2023), is a framework designed to enhance the capabilities of LLMs. It does this by integrating retrieval and self-critique mechanisms into the model's generation process. The framework trains a single arbitrary language model to adaptively retrieve passages on-demand, generate text, and reflect on retrieved passages and its own generations using special tokens, called reflection tokens. Reflection tokens are categorized into retrieval and critique tokens. Retrieval tokens indicate the need for retrieval, while critique tokens assess the quality of the generation1. Generating reflection tokens makes the language model controllable during the inference phase, enabling it to tailor its behavior to diverse task requirements.
RQ-RAG	Following Self-RAG, RQ-RAG, or Learning to Refine Queries for Retrieval Augmented Generation (Chan et al., 2024), is a paradigm to refine queries with capabilities for explicit rewriting, decomposition, and disambiguation. By regenerating responses based on search results from external search engine, RQ-RAG surpasses the previous state-of-the-art by an average of 1.9% across three single-hop QA datasets, and also demonstrates enhanced performance in handling complex, multi-hop QA datasets.

Table 5: Model Cards.

B Training Details

B.1 Dataset

Single-hop QA Dataset consists of ARC Challenge (Clark et al., 2018) train, a set of natural, grade-school science questions, PubHealth (Kotonya and Toni, 2020) train, a fact-checking for claims provided contents that support or refute to the claims, and TriviaQA (Joshi et al., 2017) train, a challenging reading comprehension QA dataset. Since ARC Challenge train and TriviaQA train dataset do not contain document candidates, we use off-the-shelf Contriever-MS MARCO to retrieve top document chunks from 2020 English Wikipedia using official Wikipedia embeddings.

Multi-hop QA Dataset includes HotpotQA (Yang et al., 2018) train dataset. The dataset contains 90447 pairs of QA and a set of supporting contents randomly mingled with irrelevant or misleading contents.

We further clean datasets by removing answers that are absent in the contents or in the outputs of teacher models. Please refer to Table 7 for dataset statistics.

B.2 Teacher Models

We employ five teacher models to generate the proposed structured output for each query-document pair, including Llama2-70B-Chat (Touvron et al., 2023), Meta-Llama3-(8B,70B)-Instruct and Alpaca-7B (Taori et al., 2023). We design four prompts to tackle with diverse QA tasks. For each output generated by each teacher model, we utilize regular expressions to parse and resolve sections, titles, and contents. We then exclude any contents that are not present in the provided document chunks. The remaining contents are subject to a voting process among the teacher models, whereby only those contents that receive a majority vote from the teacher models are retained. Finally, the filtered contents are concatenated to formulate the final training dataset. The detailed task-specific prompts are illustrated in Table 8, and hyperparameters of the teacher models are listed in Table 6.

Hyperparameters	**Meta-Llama3-70B-Instruct**	**Meta-Llama3-8B-Instruct**	
temperature	None	None	
repetition-penalty	1.0	1.0	
max-length	2048	2048	
do-sample	False	False	
Hyperparameters	**Llama2-70B-Chat**	**Llama2-13B-Chat**	**Alpaca-7B**
temperature	None	None	None
repetition-penalty	1.0	1.0	1.0
max-length	2048	2048	2048
do-sample	False	False	False

Table 6: Hyper-parameters of teacher models.

B.3 Structure Correction

Once outputs are generated by teacher models, we make a series of cleaning and filtering procedures so as to satisfy the proposed principles. Firstly, we filter out contents that are absent in retrieved document chunks, store legitimate sections voted by teacher models, then remove sentences that are vetoed by teacher models. Finally, the selected sections are unified by re-aligning section numbers, titles and extractive contents to form the *Refiner*'s training data as illustrated in Figure 4.

B.4 Environment and Hyperparameters

We use 32GB memory and 4 Nvidia A100 with 80GB memory to train our model. We adopt distributed data parallel (Li et al., 2020) and Deepspeed stage 3 (Rajbhandari et al., 2020) for multi-GPU distributed training. We apply parameter-efficient funetuning with LoRA (Hu et al., 2021) rank of 64, alpha of 16 on Llama2-7B-Chat. AdamW (Loshchilov and Hutter, 2019) is adopted as optimizer. All seeds are set to 633. The total train steps reaches 6 hours in the specified environment.

Input: Given n Teacher LM $\{M_j\}_1^n$, *Refiner* $\mathcal{M}_\theta$, query set $\{q_i\}_1^N$, where q represents the user's query, R denotes retrieval system by default, C denotes corpus set, $\mathcal{D}$ is a set of corresponding retrieved documents such that $\mathcal{D} \in C$, $\mathbf{V}$ is a collection of parsed contents in sentence level generated by teacher LM. $\mathcal{T}$ denotes temporary content collection.
Output: An updated *Refiner* $\mathcal{M}_\theta$

```
$\mathcal{S} \leftarrow \emptyset$
for $i \in \{1, \ldots, N\}$ do
    $\mathbf{D}_i$ = TopN($R(q_i, C)$)
    for $j \in \{1, \ldots, n\}$ do
        $\mathbf{s_j}$ = Generate($M_j$, $[q_i, \mathbf{D}_i]$)
        $\mathbf{V}$ = ParseGeneratedContent($\mathbf{s_j}$)
        for content $\in \mathbf{V}$ do
            if content $\notin \mathbf{D}_i$ then
                $\mathbf{V}$.remove(content)
    $\mathcal{T} \leftarrow \emptyset$
    $c$ = Counter($\mathbf{V}$)
    for content, count $\in c$ do
        // Iterate over result from counter to get
        // number of occurrence for each content
        if count $> n/2$ then
            content = assignSection(content)
            $\mathcal{T} \leftarrow \mathcal{T} \cup$ content
    $\mathcal{Y}_i \leftarrow$ Concatenate($\mathcal{T}$)
    $S \leftarrow S \cup [q_i, \mathbf{D}_i; \mathcal{Y}_i]$
$\mathcal{M}_\theta$ = Finetune($\mathcal{M}_\theta$, $S$)
```

Figure 4: Prepare curated train dataset for *Refiner*.

B.5 Training Process

As shown in Table 7, the cleaned datasets are highly imbalanced in number of samples. Thus, in order to train the model effectively, we practice progressive training: First, we form training dataset with TriviaQA and HotpotQA dataset combined, the model is trained for 3 epochs with a batch size of 128, a peak learning rate of 4e-5 with 3% warmup steps, and linear decay afterward. Subsequently, we continue training the model combining Arc Challenge and PubHealth datasets, with 5 epoch and a batch size of 128, a peak learning rate of 2e-5 with 5% warmup steps.

Dataset	Initial	Cleaned
Arc Challenge	1119	1119
PubHealth	9513	9513
TriviaQA	61888	56655
HotpotQA	90447	79157

Table 7: Training data statistics.

Dataset	Prompts
TriviaQA & HotpotQA	***System***: You are an expert research assistant. Your job is to find the quotes from the markdown documents that are relevant to a question.`\n` Please mark quotes with sections and titles of documents to group quotes by different information only from relevant documents.`\n` Rule of labelling sections: if the fact of the first dummy quote "aaa" from a document with title "## AAA" is consistent with that of the second dummy quote "bbb" from a document with title "## BBB", and they contradict to that of the third dummy quote "ccc" from a document with title "## CCC", then label them as:`\n` “‘`\n1.1.` ## AAA`\n` aaa`\n\n1.2.` ## BBB`\n` bbb`\n\n2.1.` ## CCC`\n` ccc`\n\n`“‘`\n` Quotes should be verbatim and context completed.`\n` Please respond without any explanation. ***User***: Question: `{query}\n` Document (multiple documents are separated by "—"):`\n {context}`
ARC Challenge	***System***: You are an expert research assistant. Your job is to find the quotes from the markdown documents that are relevant to a question.`\n` Please mark quotes with sections and titles of documents to group quotes by different information only from relevant documents that either support or contradict to the question.`\n` Rule of labelling sections: if the fact of the first dummy quote "aaa" from a document with title "## AAA" is consistent with that of the second dummy quote "bbb" from a document with title "## BBB", and they contradict to that of the third dummy quote "ccc" from a document with title "## CCC", then label them as:`\n` “‘`\n1.1.` ## AAA`\n` aaa`\n\n1.2.` ## BBB`\n` bbb`\n\n2.1.` ## CCC`\n` ccc`\n\n`“‘`\n` Quotes should be verbatim and context completed.`\n` Please respond without any explanation. ***User***: Question: `{query}\n` Document (multiple documents are separated by "—"):`\n {context}`
PubHealth	***System***: You are an expert research assistant. Your job is to find the quotes from the markdown documents that either support or contradict to a statement.`\n` Please mark quotes with sections and titles of documents to group quotes by different information only from relevant documents that are helpful with answering to the statement.`\n` Rule of labelling sections: if the fact of the first dummy quote "aaa" from a document with title "## AAA" is consistent with that of the second dummy quote "bbb" from a document with title "## BBB", and they contradict to that of the third dummy quote "ccc" from a document with title "## CCC", then label them as:`\n` “‘`\n1.1.` ## AAA`\n` aaa`\n\n1.2.` ## BBB`\n` bbb`\n\n2.1.` ## CCC`\n` ccc`\n\n`“‘`\n` Quotes should be verbatim and context completed.`\n` Please respond without any explanation. ***User***: Statement: `{query}\n` Document (multiple documents are separated by "—"):`\n {context}`

Table 8: Prompts used to generated structured outputs from teacher models `{query}` and `{context}` represent the actual input query and retrieved documents.

C Evaluation Details

C.1 Hyperparameters for Adavanced RAG

For Self-RAG, we adopt the default inference settings. Specifically, we set the weight terms **IsRel**, **IsSup**, **IsUse** values of 1.0, 1.0 and 0.5, respectively. The retrieval threshold is set to 0.2 to encourage frequent retrieval.

C.2 Hyperparameters for Concurrent Compressor

We assess the performance of concurrent compressors encompassing RECOMP abstractive compressor (Xu et al., 2023) and LongLLMLingua (Jiang et al., 2023a). Notably, the RECOMP abstractive compressor has two versions, one trained on TriviaQA dataset and the other on HotpotQA dataset. We evaluate the former version on the PopQA, TriviaQA, and ARC Challenge datasets, while the latter version was tested on the HotpotQA and 2WikiMultihop datasets. Regarding the hyperparameters of LongLLMLingua, we utilized a default compression rate of 0.5 and a dynamic context compression ratio of 0.3.

C.3 Environment and Hyperparameters

We use 32GB memory and 4 Nvidia A100 with 80GB memory for inference. we adopt vllm (Kwon et al., 2023) to speed up inference. All models, including downstream LMs and *Refiner*, use greedy algorithm to sample their next tokens, and max new tokens is set to 2048.

C.4 Prompt Template for *Refiner*

To efficiently train *Refiner* and accelerate convergence speed, we revise its base model, Llama2-Chat's chat template, which is illustrated in Table 9.

Prompt Template for *Refiner*	[INST]«SYS»[MONITOR]`{documents}`«/SYS»`{query}`[/INST]
Prompt Template for `{documents}`	## `{title}` `{content}` ## `{title}` `{content}` ## `{title}` `{content}` ...

Table 9: Prompt template for *refiner* where `{query}` represents the actual user input query, `{documents}` represents document chunks retrieved for extraction, `{content}` represents content in document and `{title}` represents title of the document from which the content derives.

C.5 Prompt Template for Downstream LM

To simulate real-world application of *Refiner*, we prompt downstream LMs to work with *Refiner*'s output as illustrated in the Table 10.

Prompt
System: You are an AI assistant backboned by selective content from different documents, answer user's question helpfully and precisely, with the guidance of the following steps: * If there are no content provided: determine whether it is still possible to answer precisely to the question. * If is possible, offer a helpful answer. Otherwise, offer the reason of impossibility. * If there exists contents: determine whether the necessary information to answer the question is either directly mentioned or can be inferred from the documents. * When there exists different information that can answer to the question, determine whether it is the question too opaque that causes the problem. * If not, answer with a summarized information. Otherwise, also provide advice or ask question to disambiguate. * When summarizing, ensure to include contents that are relevant to the question. Here is the content: `{refiner}` ***User***: `{query}`

Table 10: Prompts used to generated structured outputs from teacher models `{query}` represent the actual input query, and `{refiner}` represent the actual generated output from *Refiner*.

C.6 Evaluation Metric for ARC Challenge

C.6.1 Motivation

To deliver our motivation to revising evaluation metric for ARC Challenge task, we provide a case from the ARC Challenge test subset.In this case, the model is prompted to answer question based on the A, B, C and D choices given, while through reading the context we can easily find out that the model is answering with the choice B. However, the generated content contains all of the four choices, as previous work does not take such cases into account, the performance of the model could be over-evaluated.

Method	Case in Arc Challenge test subset	Result	Comment
Previous Work	**Question**: The morning temperature in a city is 41°F. If a sunny, mild day is forecast, which temperature is most likely for 2:00 p.m.? A. 32° F B. 41° F C. 78° F D. 98° F **Ground Truth**: C **LLM outputs**: Based on the given information, the best answer choice is (B) 41°F. The given information states that the morning temperature is 41°F, which suggests that the day will be sunny and mild. As the temperature drops gradually throughout the day, it is unlikely that the temperature at 2:00 p.m. will be above 78°F or 98°F, as these temperatures are much higher than the morning temperature. Therefore, the most likely temperature for 2:00 p.m. is 41°F, which is the same as the morning temperature. Answer choice (A) 32°F is too low, as it is below the morning temperature and unlikely to be the temperature at 2:00 p.m. in a sunny and mild day. Answer choice (C) 78°F is also unlikely, as it is higher than the morning temperature and may not be reached in the afternoon. Answer choice (D) 98°F is the highest temperature mentioned in the passage and is therefore unlikely to be the temperature at 2:00 p.m.	**Matches ground truth C in generated sentence:** "Answer choice (C) 78°F is also unlikely"	Wrong evaluation result
Ours		**Ground truth C mismatches with predicted B in sentence**: "Based on the given information, the best answer choice is (B) 41°F."	Correct evaluation result

Table 11: The result of regular-expression-based accuracy compared to previous work.

C.6.2 Code Comparison

For a concise convey of our evaluation metric, we disclose previous work and our pseudo-python code in the following list.

```python
# Previous Methods
def match(prediction: str, ground_truth: list[str]):
    # Argument 'prediction' represents generated output from downstream LM,
    # and 'ground_truth' represents a list of answer key strings.
    # For example, if the ground truth choice for a question is A,
    # then ['A'] will be passed to ground_truth.

    for gt in ground truth:
        if gt in prediction:
            return 1

    return 0
```

```python
# Our Methods
import re


def arc_c_acc(prediction: str, ground_truth: list[str]):
    # The presence of 'is' preceding the letter 'A' to 'E'
    # is a necessary condition for a match to occur.
    regex_answer_key = re.compile(r"is[^\w]+([A-E])[^\w]+", re.DOTALL)

    # re.search only looks for the first match occurence.
    if len(ground_truth) == 0:
        return re.search(regex_answer_key, prediction) is None

    lst_res = []
    for key in ground_truth:
        search = re.search(regex_answer_key, prediction)
        if search is None:
            lst_res.append(False)
            continue

        if key.lower() == search.groups()[0].lower():
            lst_res.append(True)
        else:
            lst_res.append(False)

    # only when all ground truths are matched should we return True.
    return all(lst_res)
```

D Ablation Analysis Details

We present a comprehensive disclosure of all ablation analysis results, with specific sections omitted from the *Refiner*'s output. The accuracy outcomes under the top 5 retrieval setting are detailed in Table 12, while the accuracy outcomes under the top 10 retrieval setting are presented in Table 13.

Task	*Refiner*	acc. w/o. Section	acc. Decay
Llama2-7B-Chat			
PopQA	59.3	57.7	1.6
TriviaQA	68.9	66.5	2.4
ARC Challenge	51.1	51.7	-0.6
HotpotQA	47.4	47.3	0.1
2WikiMultihop	46.0	46.3	-0.2
Llama2-13B-Chat			
PopQA	59.0	58.5	0.5
TriviaQA	70.5	69.4	1.1
ARC Challenge	54.2	54.4	-0.2
HotpotQA	49.7	49.5	0.2
2WikiMultihop	49.7	47.0	-1.3
Llama2-70B-Chat			
PopQA	58.3	58.0	0.3
TriviaQA	72.2	71.4	0.8
ARC Challenge	55.9	58.7	-2.8
HotpotQA	52.3	52.2	0.1
2WikiMultihop	50.1	50.2	-0.1
Meta-Llama3-8B-Instruct			
PopQA	58.3	57.9	0.4
TriviaQA	69.1	66.8	2.3
ARC Challenge	71.8	74.4	-2.6
HotpotQA	49.0	49.2	-0.2
2WikiMultihop	48.0	48.0	0.0
Meta-Llama3-70B-Instruct			
PopQA	57.9	57.6	0.3
TriviaQA	70.9	68.9	2.0
ARC Challenge	84.1	85.5	-1.4
HotpotQA	49.2	48.7	0.5
2WikiMultihop	46.7	46.8	-0.1

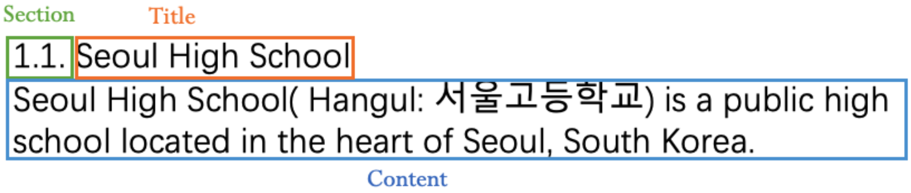
Table 12: *Refiner*-augmented model accuracy under top 5 retrieval setting.

Task	*Refiner*	acc. w/o. Section	acc. Decay
Llama2-7B-Chat			
PopQA	59.3	58.4	0.9
TriviaQA	68.9	66.2	2.7
ARC Challenge	51.8	50.4	1.4
HotpotQA	65.1	63.2	1.9
2WikiMultihop	61.0	57.6	3.4
Llama2-13B-Chat			
PopQA	59.4	57.8	1.5
TriviaQA	70.6	68.3	2.3
ARC Challenge	55.1	55.0	0.1
HotpotQA	67.1	66.3	0.8
2WikiMultihop	61.6	61.4	0.2
Llama2-70B-Chat			
PopQA	60.8	58.2	2.6
TriviaQA	72.1	70.9	1.2
ARC Challenge	57.0	59.8	-2.8
HotpotQA	69.8	69.0	0.8
2WikiMultihop	68.3	68.1	0.2
Meta-Llama3-8B-Instruct			
PopQA	59.7	57.8	1.9
TriviaQA	69.3	67.2	2.1
ARC Challenge	71.1	73.3	-2.2
HotpotQA	67.2	67.4	-0.2
2WikiMultihop	63.7	63.6	0.1
Meta-Llama3-70B-Instruct			
PopQA	59.8	58.5	1.3
TriviaQA	71.1	69.4	1.7
ARC Challenge	84.7	84.4	0.3
HotpotQA	68.6	68.3	0.3
2WikiMultihop	61.8	61.8	0.0

Table 13: *Refiner*-augmented model accuracy under top 10 retrieval setting.

E More Analysis on Output Structure

Our *Refiner* comes with a structured output, this allows for seamless dissemble of section, titles and contents (see Figure 5) using regular expressions, enables us to easily convert them into numerous alternative structures. In this context, we will explore a total of 16 additional different structure options, as illustrated in Table 14.



Seoul High School(Hangul: 서울고등학교) is a public high school located in the heart of Seoul, South Korea.

Figure 5: The structure of output generated by *Refiner* consists of three parts: section, Title and content, whereSection elucidates relatedness information among extracted contents, Title represents the Title of the Original document, content represents the question-relevant content extracted from that document chunk.

Content Structure	Section Structure	Title Structure	Template
Original	**Hierarchy**	**Original**	1.1. [*Title*]\n[*Content*]
		Markdown	1.1. ## [*Title*]\n[*Content*]
		Quote	1.1. ”[*Title*]”\n[*Content*]
	Numbered	**Original**	1. [*Title*]\n[*Content*]
		Markdown	1. ## [*Title*]\n[*Content*]
		Quote	1. ”[*Title*]”\n[*Content*]
	Star	**Original**	* [*Title*]\n[*Content*]
		Markdown	* ## [*Title*]\n[*Content*]
		Quote	* ”[*Title*]”\n[*Content*]
Quote	**Hierarchy**	**Original**	1.1. [*Title*]\n”[*Content*]”
		Markdown	1.1. ## [*Title*]\n”[*Content*]”
		Quote	1.1. ”[*Title*]”\n”[*Content*]”
	Numbered	**Original**	1. [*Title*]\n”[*Content*]”
		Markdown	1. ## [*Title*]\n”[*Content*]”
		Quote	1. ”[*Title*]”\n”[*Content*]”
	Star	**Original**	* [*Title*]\n”[*Content*]”
		Markdown	* ## [*Title*]\n”[*Content*]”
		Quote	* ”[*Title*]”\n”[*Content*]”

Table 14: Several restructure templates are utilized to assess the impact on the performance of downstream LMs. In these templates, $[Title]$ denotes the actual Title of the Original document, while $[Content]$ represents the actual content extracted from the corresponding document chunk. A comparative analysis is conducted between the Original hierarchicalSection structure and two alternative structures: a Numbered edSection structure and a Star Section structure. Furthermore, the Original Title is compared in its Original form and when formatted using markdown or quotation marks, while the Original content is compared in its Original form and when decorated with quotation marks.

Output Structure	**POPQA**		**TQA**		**ARC-C**	
	mean	std	mean	std	mean	std
Original Content						
Numbered Section Markdown Title	1.0%	0.9%	1.7%	0.5%	-0.8%	2.5%
Numbered Section Original Title	1.2%	0.8%	1.9%	0.5%	-0.1%	1.4%
Numbered Section Quote Title	1.1%	0.8%	1.9%	0.6%	-0.4%	2.2%
Hierarchy Section Markdown Title	1.1%	0.7%	1.7%	0.5%	-1.2%	2.6%
Hierarchy Section Quote Title	1.0%	0.9%	1.8%	0.7%	-0.5%	2.8%
Star Section Markdown Title	1.4%	0.7%	1.7%	0.6%	-0.9%	2.5%
Star Section Original Title	1.2%	0.7%	1.9%	0.5%	0.1%	1.4%
Star Section Quote Title	0.9%	0.9%	1.9%	0.7%	-0.5%	1.9%
Quote Content						
Numbered Section Markdown Title	1.3%	0.7%	1.7%	0.6%	-0.9%	2.6%
Numbered Section Original Title	1.2%	0.6%	1.8%	0.5%	-0.3%	2.0%
Numbered Section Quote Title	1.4%	0.7%	1.8%	0.6%	-0.9%	2.9%
Hierarchy Section Markdown Title	1.3%	0.7%	1.6%	0.7%	-1.4%	3.2%
Hierarchy Section Quote Title	1.1%	0.7%	1.7%	0.7%	-0.5%	2.9%
Star Section Markdown Title	1.2%	0.6%	1.7%	0.6%	-1.2%	1.9%
Star Section Original Title	1.2%	0.6%	1.8%	0.6%	-0.6%	1.7%
Star Section Quote Title	1.1%	0.7%	1.8%	0.7%	-0.6%	2.4%

Table 15: Statistics of improved accuracy margin from various downstream LMs augmented by restructured outputs from *Refiner*, compared with no structured output in single-hop tasks.

Output Structure	**HOTPOT**		**2WIKI**	
	mean	std	mean	std
Original Content				
Numbered Section Markdown Title	0.5%	0.8%	-0.5%	1.2%
Numbered Section Original Title	0.5%	0.8%	-0.3%	1.2%
Numbered Section Quote Title	0.7%	1.1%	-0.1%	1.5%
Hierarchy Section Markdown Title	0.5%	0.9%	-0.4%	1.6%
Hierarchy Section Quote Title	0.7%	0.9%	-0.3%	1.5%
Star Section Markdown Title	0.6%	0.7%	-0.2%	1.4%
Star Section Original Title	0.7%	0.8%	-0.1%	1.4%
Star Section Quote Title	0.7%	1.0%	0.2%	1.5%
Quote Content				
Numbered Section Markdown Title	0.9%	1.2%	0.0%	1.5%
Numbered Section Original Title	1.0%	1.2%	0.1%	1.5%
Numbered Section Quote Title	0.9%	1.3%	0.5%	2.0%
Hierarchy Section Markdown Title	0.7%	1.2%	-0.2%	1.8%
Hierarchy Section Quote Title	1.0%	1.4%	0.2%	1.7%
Star Section Markdown Title	0.8%	1.2%	0.2%	1.8%
Star Section Original Title	1.1%	1.2%	0.2%	1.7%
Star Section Quote Title	1.1%	1.5%	0.7%	2.1%

Table 16: Statistics of improved accuracy margin from various downstream LMs augmented by restructured outputs from *Refiner*, compared with no structured output in multi-hop tasks.

F Case Study

Below, we will provide several exemplary cases illustrating how our model assists downstream models in further extracting information to enhance accuracy. We select one case from PopQA and two cases from TriviaQA subset under top 5 retrieval setting. For case under PopQA, *Refiner* successfully disseminates similar yet different information using section, improves the quality of answer of downstream LM. While in both of the cases under TriviaQA, the *Refiner*-augmented models successfully answer with ground truths which are only mentioned indirectly.

Question	**What is George Lewis's occupation?**
Ground Truths	**'politician', 'political leader', 'political figure', 'polit.', 'pol'**
Retrieved Document Chunks	**Title**: George Lewis (footballer, born 1913) **Content**: Thomas George Lewis (20 October 1913 – 6 August 1981) was a Welsh footballer who played as a centre-forward for Watford in the late 1930s, before joining Southampton for a brief period after World War II....Lewis spent the 1947–48 season back in the Third Division South with Brighton & Hove Albion before joining Dartford, where he remained until 1952, when he took up a position as groundsman. **Title**: George Lewis (footballer, born 2000) **Content**: George Lewis Igaba-Ishimwe Maniraguha (born 16 June 2000) is a professional footballer who plays as a forward for club Arsenal. Born in Kigali, Rwanda, Lewis moved to Tanzania at the age of one. He then moved to Norway the age of four, starting his career at local side Stakkevollan IF, before moving to Tromsdalen, and later Tromsø in 2015. He mostly played for the reserve teams of both clubs, notably notching 32 goals in 35 appearances over three seasons for Tromsdalen's reserve team. **Title**: George Robert Lewis **Content**: George Robert Lewis (1782–1871) was a versatile English painter of landscapes and portraits....Attribution **Title**: George E. Lewis **Content**: George Emanuel Lewis (born July 14, 1952) is an American composer, performer, and scholar of experimental music. He has been a member of the Association for the Advancement of Creative Musicians (AACM) since 1971, when he joined the organization at the age of 19. He is renowned for his work as an improvising trombonist and considered a pioneer of computer music, which he began pursuing in the late 1970s; in the 1980s he created Voyager, an improvising software he has used in interactive performances. Lewis's many honors include a MacArthur Fellowship and a Guggenheim Fellowship, and his book A Power Stronger Than Itself: The AACM and American Experimental Music received the American Book Award. Lewis is the Edwin H. Case Professor of American Music, Composition & Historical Musicology at Columbia University. **Title**: George Lewis (rugby) **Content**: George Lewis (9 August 1897 – death unknown) was a Welsh rugby union and professional rugby league footballer who played in the 1920s and 1930s. He played club level rugby union (RU) for Pontypool RFC, as a centre, i.e. number 12 or 13, and representative level rugby league (RL) for Wales and Monmouthshire, and at club level for St. Helens, as a, i.e. number 1, 3 or 4, 6, or 7....George Lewis was the brother of the rugby union footballer for Pontypool RFC, and the rugby league for St. Helens; Stanley "Stan" Lewis.
Refiner	1.1. George Lewis (footballer, born 1913) Thomas George Lewis (20 October 1913 – 6 August 1981) was a Welsh footballer... 1.2. George Lewis (footballer, born 2000) George Lewis Igaba-Ishimwe Maniraguha (born 16 June 2000) is a professional footballer... 2.1. George E. Lewis George Emanuel Lewis (born July 14, 1952) is an American composer, performer, and scholar of experimental music.
Llama2-7B-Chat	George Lewis's occupation is footballer. He played as a centre-forward for Watford in the late 1930s and joined Southampton for a brief period after World War II.
Refiner + Llama2-7B-Chat	Based on the content provided, there are three individuals with the name George Lewis: 1. George Lewis (footballer, born 1913) - The content mentions that he was a Welsh footballer. Therefore, his occupation was a footballer. 2. George Lewis (footballer, born 2000) - The content mentions that he is a professional footballer. Therefore, his occupation is also a footballer. 3. George E. Lewis - The content does not mention his occupation directly, but based on the fact that he is a composer, performer, and scholar of experimental music, it can be inferred that his occupation is related to music, possibly as a composer or musician. Therefore, the answer to the question "What is George Lewis's occupation?" would be: * For George Lewis (footballer, born 1913) and George Lewis (footballer, born 2000): footballer. * For George E. Lewis: composer, performer, and scholar of experimental music."

Table 17: Case study of how *Refiner* helps downstream LM to disseminate ambiguous information from PopQA evaluation set.

Question	**For which team did Babe Ruth blast his last Major League home run?**
Ground Truths	**’boston braves’, ’boston braves disambiguation’, ’Boston Braves (disambiguation)’, ’Boston Braves’**
Retrieved Document Chunks	**Title**: Lou Chiozza **Content**: Braves, had been through for years but was still being exploited for his name. Just a few days before the Philadelphia series, the Babe hit three home runs in Pittsburgh. Chiozza, who had just been brought up from the Memphis Chicks, hit a short fly ball down the left field line that would have ordinarily at best been a double. Due to his advanced age and decreased mobility, Ruth stumbled after the ball in the outfield. The shortstop ran out, retrieved the ball and threw the ball home to barely stop Chiozza from an inside-the-park home run. Many in the park believed that the umpire felt sorry for the aging star and called an obviously safe Chiozza out at home plate. After Ruth realized that he was so slow that Lou almost made a home run on an ordinary base hit, he stood for a minute, folded his glove and walked off the field into the clubhouse. The Babe knew he was done and he officially retired a few days later on June 2, 1935. Chiozza recalled in his later life that he had wished Ruth had retired on a high note after hitting the three home runs in Pittsburgh **Title**: Guy Bush **Content**: #714, and was a mammoth of a shot. It was the first home run to clear the right field grandstands at Forbes Field and was reportedly measured as a bomb. It was the last home run – in fact, the last hit of any kind – in Ruth’s major league career, which ended less than a week later. Despite Ruth’s goliath performance, Waite Hoyt, Ruth’s former Yankee teammate, came in relief for Bush and won the game 11-7, albeit after initially blowing the save. The Pirates sent the 34-year-old Bush to the bullpen for the season. The transition did not **Title**: Babe Ruth **Content**: As part of the Yankees’ vaunted "Murderers’ Row" lineup of 1927, Ruth hit 60 home runs, which extended his MLB single-season record by a single home run. Ruth’s last season with the Yankees was 1934; he retired from the game the following year, after a short stint with the Boston Braves. During his career, Ruth led the AL in home runs during a season twelve times. Ruth’s legendary power and charismatic personality made him a larger-than-life figure during the Roaring Twenties. During his career, he was the target of intense press and public attention for his baseball exploits and off-field...a long fly caught by Paul Waner probably would have been a home run in any other ballpark besides Forbes Field. Ruth played in the third game of the Pittsburgh series on May 25, 1935, and added one more tale to his playing legend. Ruth went 4-for-4, including three home runs, though the Braves lost the game 11–7. The last two were off Ruth’s old Cubs nemesis, Guy Bush. The final home run, both of the game and of Ruth’s career, sailed out of the park over the right field upper deck–the first time anyone had hit a fair ball **Title**: Atlanta Braves **Content**: Brave before asking to be traded to the Milwaukee Brewers, while Ruth finished his career as a Boston Brave. In fact, until Barry Bonds eclipsed the 714 home runs hit by Babe Ruth in 2006, the top two home run hitters in Major League history had at one time been Braves. In 1976, the team was purchased by media magnate Ted Turner, owner of superstation WTBS, as a means to keep the team (and one of his main programming staples) in Atlanta. The financially strapped Turner used money already paid to the team for their broadcast rights as a down-payment. **Title**: 1935 Boston Braves season **Content**: the team due to the attention he commanded. He also discovered that rather than give him a share of the Braves’ profits, Fuchs expected him to invest some of "his" money in the team. Seeing a team in utter collapse and realizing he was finished even as a part-time player, Ruth retired on June 1, six days after he had what remains one of the most memorable afternoons in baseball history. He clouted what turned out to be the last three home runs of his career in a game at Forbes Field while playing the Pittsburgh Pirates. He’d wanted to
Refiner	1.1. Babe Ruth Ruth played in the third game of the Pittsburgh series on May 25, 1935, and added one more tale to his playing legend. Ruth went 4-for-4, including three home runs, though the Braves lost the game 11–7. The last two were off Ruth’s old Cubs nemesis, Guy Bush. 1.2. Atlanta Braves Brave before asking to be traded to the Milwaukee Brewers, while Ruth finished his career as a Boston Brave.
Llama2-13B-Chat	According to the text, Babe Ruth’s last Major League home run was hit off Guy Bush of the Pittsburgh Pirates at Forbes Field on May 25, 1935.
Refiner + Llama2-13B-Chat	Sure, I’d be happy to help! Based on the content provided, Babe Ruth played for the Boston Braves when he hit his last Major League home run. The content states that Ruth finished his career as a Boston Brave. Therefore, the answer to your question is the Boston Braves.

Question	**Apimania is an obsession with which creatures?**
Ground Truths	**’Ground bees’, ’bees’, ’Solitary bee’, ’Apiformes’, ’Bee flight’, ’ground bee’, ’bee flight’, ’anthophila’, ’solitary bee’, ’apiforme’, ’Solitary bees’, ’ground bees’, ’Anthophila’, ’Bee’, ’Bees’, ’apiformes’, ’Apiforme’, ’Ground bee’, ’solitary bees’, ’bee’**
Retrieved Document Chunks	**Title**: Pedro Reyes (artist) **Content**: Assembly of "pUN" took place in December 2015 at the 21st Century Museum of Contemporary Art in Kanazawa, Japan. Entomofagia (Spanish for entomophagy–the practice of eating insects) is a series of culinary experiments. The first happened in 2013 at Casa do Vidro in São Paulo, where a small snack cart offered visitors a Brazilian street food specialty: giant ants. The project focuses on the enormous ecological advantages of shifting protein consumption to insects. As an alternative to the ubiquitous fast-food staple, the beef hamburger, Reyes created the Grasswhopper, a burger with a patty made of crickets, a common snack in **Title**: Insects in literature **Content**: Grasshopper" by Samson Samsonov (1955) and Jerry Paris (1970) based on that story. The Ancient Greek playwright Aeschylus has a gadfly pursue and torment Io, a maiden associated with the moon, watched constantly by the eyes of the herdsman Argus, associated with all the stars: "Io: Ah! Hah! Again the prick, the stab of gadfly-sting! O earth, earth, hide, the hollow shape—Argus—that evil thing—the hundred-eyed." William Shakespeare, inspired by Aeschylus, has Tom o’Bedlam in "King Lear", "Whom the foul fiend hath led through fire and through flame, through ford and whirlpool, o’er bog and quagmire", driven mad by the constant **Title**: Dromomania **Content**: sometimes been clinical and pathologizing, and other times been descriptive of unusual enthusiasm without negative or medicalizing connotations, reflecting the diverse uses of the term mania itself. In the 17th century, that the term "mania" came to be used to describe any show of great enthusiasm for a specific activity or object. Later, it came to be used as a suffix for Greek words to refer to an irrational obsession, such as in the words "guanaikomania", "hippomania", and "nymphomania". At the same time emerged the French , such as in which was borrowed in English as "bibliomania". The original sense — **Title**: Erich Hoyt **Content**: entomologist Ted Schultz) broke new ground. In Hoyt’s "delightful…multi-layered" "The Earth Dwellers", the "ant’s eye view of life works spectacularly" as Hoyt "fashions the ants into enchanting characters" charting "an insect’s course through sex, aggression and foreign policy". "Insect Lives", an American Library Association "Outstanding Book for the College Bound" is a "potpourri of fascinating excerpts written by some of the finest insect biologists and naturalists spanning many centuries." A deep sea book called "Creatures of the Deep" (Firefly, 2001) won the American Society of Journalists & Authors, Inc. Outstanding Book Award, General Nonfiction. A second deep sea book, "Weird **Title**: Obsession (novel) **Content**: Obsession (novel) Obsession is a mystery novel by American author Jonathan Kellerman. It is the 21st novel in the Alex Delaware series. Tanya Bigelow, a former patient, comes to Alex requesting help. She wants him and his friend Milo to investigate something her mother said on her death bed. Her mother told her that she did something terrible. No one believes that Tanya’s mother Patty, who worked with Milo’s partner at the hospital as a nurse, could have done anything terrible, but as the reader learns her past, it contains dark secrets. Alex has a new dog, Blanche, that Robin bought him after Spike died. Robin is living with Alex again.
Refiner	1.1. Erich Hoyt entomologist Ted Schultz) broke new ground. In Hoyt’s "delightful…multi-layered" "The Earth Dwellers", the "ant’s eye view of life works spectacularly" as Hoyt "fashions the ants into enchanting characters" charting "an insect’s course through sex, aggression and foreign policy". 1.2. Insects in literature "Io: Ah! Hah! Again the prick, the stab of gadfly-sting! O earth, earth, hide, the hollow shape—Argus—that evil thing—the hundred-eyed." 2.1. Obsession (novel) Her mother told her that she did something terrible. No one believes that Tanya’s mother Patty, who worked with Milo’s partner at the hospital as a nurse, could have done anything terrible, but as the reader learns her past, it contains dark secrets.
Llama2-13B-Chat	The content provided does not directly mention an obsession with a specific creature related to the term "Apimania". However, it is possible to infer that the term "entomophagy" mentioned in the first paragraph refers to an interest or obsession with insects as a food source. The term "entomophagy" is derived from the Greek words "entomon" meaning insect and "phagein" meaning to eat. This practice has been explored in Pedro Reyes’ art project "Entomofagia" which focuses on the advantages of shifting protein consumption to insects. Therefore, Apimania could refer to an obsession with insects.
Refiner + Llama2-13B-Chat	Based on the content provided, I can answer your question precisely. According to the content, Apimania is an obsession with bees. This information can be inferred from the text "Obsession (novel)" and the quote "Io: Ah! Hah! Again the prick, the stab of gadfly-sting! O earth, earth, hide, the hollow shape—Argus—that evil thing—the hundred-eyed." which mentions bees and their sting. Therefore, the answer to your question is: Bees.

Table 18: Two cases from TriviaQA evaluation set illustrating how *Refiner* helps downstream LM to highlight the correct information.