Read the Docs Before Rewriting: Equip Rewriter with Domain Knowledge via Continual Pre-training

Anonymous ACL submission

Abstract

A Retrieval-Augmented Generation (RAG)-based question-answering (QA) system enhances a large language model's knowledge by retrieving relevant documents based on user queries. Discrepancies between user queries and document phrasings often necessitate query rewriting. However, in specialized domains, the rewriter model may struggle due to limited domain-specific knowledge. To resolve this, we propose the R&R (Read the doc before Rewriting) rewriter, which involves continual pre-training on professional documents, akin to how students prepare for open-book exams by reviewing textbooks. Additionally, it can be combined with supervised fine-tuning for improved results. Experiments on multiple datasets demonstrate that R&R excels in professional QA, effectively bridging the query-document gap, while maintaining good performance in general scenarios, thus advancing the application of RAG-based QA systems in specialized fields.

1 Introduction

In recent years, the development of Large Language Models (LLMs) has accelerated the widespread adoption of question-answering (QA) systems. However, in professional scenarios, LLMs' limited internal parametric knowledge often necessitates the use of Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). RAG retrieves documents relevant to the user's query to serve as contextual knowledge, and both the documents and the query are input into the LLM to generate an answer.

This paper aims to enhance the retrieval process in RAG. Accurate external knowledge is crucial for generating correct answers, yet retrieving it from the corpus is challenging. A key issue is "*Query-Document Discrepancy*," which refers to the differences between user query phrasing and the language used in target documents. Figure 1 illustrates

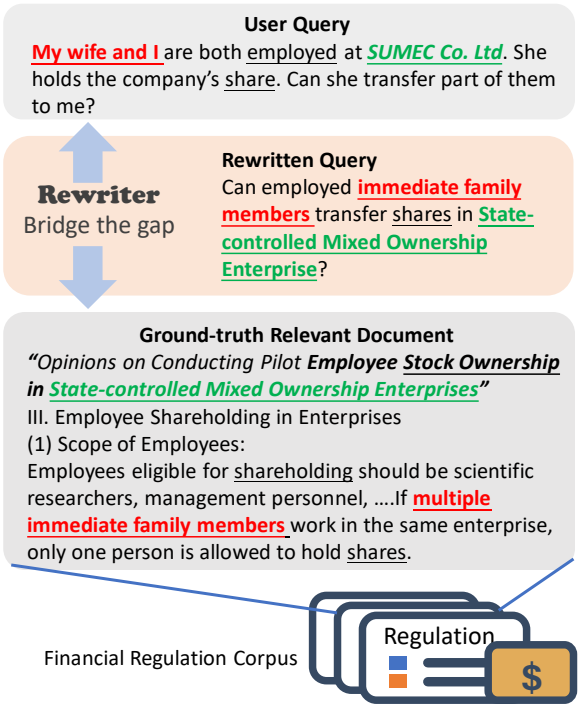

Figure 1: In professional QA, rewriting the query to retrieve relevant documents requires domain-specific knowledge from the corpus.

this challenge, showcasing a user's query alongside the relevant document. The query details a specific personal situation with informal language, like "my wife and I" and "SUMEC Co. Ltd.," while the document uses more formal and abstract terms such as "immediate family members" and "State-controlled Mixed Ownership Enterprises." This leads to a significant *Query-Document Discrepancy*.

Rewriting the query into another query that is more suitable for retrieval may mitigate this issue, as shown in the middle of Figure 1. However, bridging the gap between the wording of user queries and documents in specialized domains often requires domain-specific knowledge. For example, generating the keywords in the rewritten query in Figure 1 requires knowledge of the document cor-

pus like the regulations typically use more formal terms such as "immediate family" or "spouse." Additionally, financial regulations differ for various types of companies, so converting "SUMEC Co. Ltd." into "State-controlled Mixed Ownership Enterprises" facilitates more accurate retrieval.

The requirement of domain knowledge during rewriting creates a paradox: retrieval is intended to provide external knowledge for the large model, but the rewriter in retrieval itself demands external knowledge. Although there are some existing query rewriting methods (Liu et al., 2024; Wang et al., 2024, 2023), we have not found research addressing the *Lack of Domain Knowledge in Rewriters*.

To solve this, we propose R&R (**R**ead the doc before **R**ewriting) method, which involves Continual Pre-Training (CPT) the rewriter LLM on domain-specific corpora to enhance its professional knowledge. Specifically, given a document corpus, we convert the documents into pretraining data and then train an existing LLM using next-token prediction loss. Then, the LLM can be used to rewrite queries in RAG. This is analogous to a student reviewing a textbook before an open-book exam—familiarity with the material helps the student quickly identify relevant knowledge points and locate the right keywords in the textbook.

Moreover, we explore Supervised Fine Tuning (SFT) after CPT to enhance task-following for query rewriting. Since no publicly available training data exists for rewriting in specialized domains and hiring domain experts for annotation is costly, we propose a method to generate rewriting SFT data using advanced LLMs like GPT-4o. We refer to the CPT+SFT process as an implementation of our R&R method for comparison in experiments. We also explore integrating CPT to existing rewriting methods like query2doc (Wang et al., 2023).

We collect a new dataset to test the performance of QA systems in highly specialized scenarios. It is based on the Sponsor Representative Competency Examination (SRC), a key professional qualification exam in China's securities sector. Individuals who pass this exam are eligible to become sponsor representatives. Any stock and bond issuance applications require the signature of at least two sponsor representatives to be submitted to the China Securities Regulatory Commission, which oversees the securities market in China. The CRT exam covers fields such as securities regulations, financial accounting, corporate governance, and risk management, which makes it very challenging. During the past 20 years, only about 400 people qualify annually.

The experiments were conducted on three professional QA datasets: SRCQA, SyllabusQA, and FintextQA, as well as one general QA dataset: AmbigQA. Experimental results indicate that our proposed method is primarily suited for professional QA, particularly in cases with significant *Query-Document Discrepancy*. However, our method does not compromise the performance in broader scenarios. It is noteworthy that our method passes the SRC exam (accuracy > 0.6), indicating that LLMs have achieved expert-level performance in highly specialized domain QA.

Further investigation showed that CPT does not enhance the direct question-answering process. While CPT helps the model acquire some domain knowledge, it does not retain that knowledge effectively. This parametric knowledge is more beneficial for query rewriting than for answering questions directly.

Our method is also resource-efficient. As domain-specific corpora are generally limited in size, pre-training a 7B model on a document with 100k tokens takes only 43.9 seconds using a single NVIDIA 4090 GPU.

This paper makes the following contributions:

1. We propose incorporating professional knowledge into LLMs through continual pre-training to enhance domain expertise in query rewriters.

2. We introduce a cost-effective method for generating supervised fine-tuning data for rewriting from query-answer pairs.

3. Experiments indicate that our method is particularly effective for professional QA while maintaining performance in general QA.

2 Preliminary: Retrieval Augmented Generation

This paper studies the rewrite step in the Retrieval Augmented Generation (RAG) pipeline. So, we briefly introduce the RAG pipeline (Ma et al., 2023) here. Usually, we have a document corpus $\mathcal{D}$ for retrieval. When a user proposes a query q, the pipeline works as follows.

The rewriter model M_W rewrites q into a rewritten query set Q^* which is better for retrieving relevant documents to answer q:

$$Q^* = M_W(q). \tag{1}$$

In some implementations, there is only one element

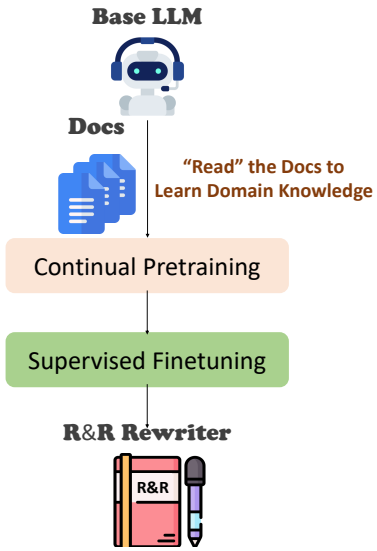

Figure 2: The base LLM undergoes continued pretraining (CPT) on documents and supervised fine-tuning (SFT) on SFT data pairs to enhance query rewriting

in Q^*, i.e. it only generates one rewritten query for each user query.

Then, the retriever M_T retrieves a set of documents D based on Q^*:

$$D = M_T(Q^*). \tag{2}$$

Finally, an LLM M_R produce the final answer a based on q, D:

$$a = M_R(q \circ D), \tag{3}$$

where $\circ$ represents concatenation.

3 R&R: Read the doc Before Rewriting

We introduce our proposed R&R in this section. We first introduce how R&R rewrites a query during inference in the RAG pipeline in section 3.1. Then, we introduce the training process of R&R in section 3.2. An overview of the R&R is shown in Figure 2.

3.1 Inference using R&R

Our query rewriter is an LLM designed to identify knowledge points through in-context learning. The input to M_W consists of an instruction, demonstration, and question as shown in Figure 8.

Given Q^*, we retrieve relevant documents as follows. Each $q_i^* \in Q^*$ is transformed into an embedding vector v_i. Then, v_i is compared against all document embedding vectors to determine their similarity and obtain the most similar k documents as a set $D(q_i^*)$. Finally, we select the top-k documents in $\bigcup_i D(q_i^*)$ among all rewritten queries.

3.2 Training R&R

Our method focuses on improving rewriter M_W using continual pertaining and finetuning.

3.2.1 Continual Pretraining of R&R

We want to inject domain knowledge into the Rewriter to produce new queries that can bridge the lexical and knowledge gaps between the user query and the document. So, we employ continual pretraining on the Rewriter with document data. The format of the document data used for pretraining is provided in the Appendix.

3.2.2 Supervised Finetuning of R&R

The finetuning process includes two steps: SFT data pair collection and supervised training.

Data Collection To fine-tune the rewriter, we need a training dataset comprising questions and their corresponding rewrites, namely $\mathcal{F} = \{(q, Q^*)\}$. Manually annotating such a dataset in knowledge-intensive domains requires professional annotators, which is costly. While there are no datasets in professional domains on query rewriting, there exist datasets that contain the final answer to user questions. We find advanced LLMs, such as GPT-4o, can be utilized to automatically generate rewrites if we can provide both question and answer.

Figure 3 illustrates the annotation process using GPT prompts. We feed a question and its answer into GPT, and ask GPT to generate a step-by-step analysis on how to derive the answer and then summarize what rewrites are important in this analysis.

Finetuning The annotated data $\mathcal{F}$ is used for supervised fine-tuning, with questions serving as input and rewrites acting as supervision signals.

4 Experiments

4.1 Datasets

We evaluate our method on three specialized and one general-domain datasets. The professional QA datasets include SRCQA, SyllabusQA(Fernandez et al., 2024), and FintextQA(Chen et al., 2024). SRCQA, which we created, consists of 2,742 multiple-choice questions from the Chinese Sponsor Representative Competency Exam, focusing on accounting, finance, and regulatory knowledge, with documents taken from the exam preparation guide. An example is shown in Figure 7. We will release this dataset after publication.

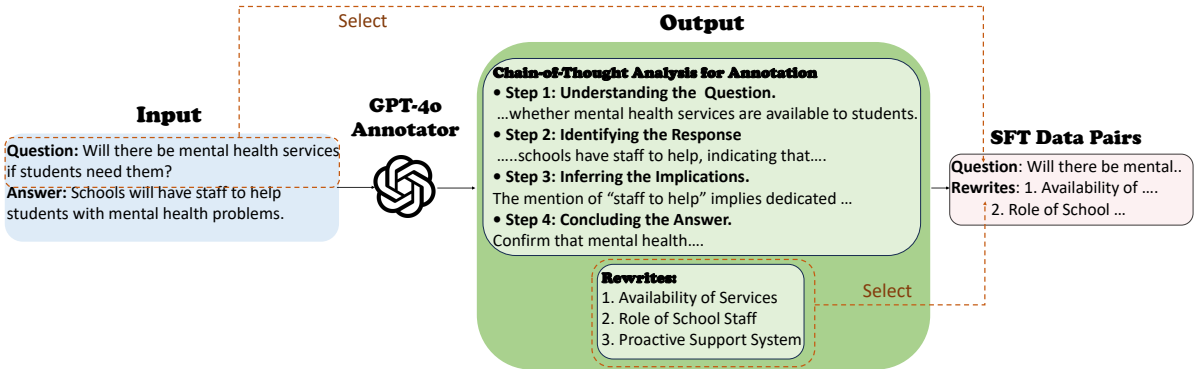

Figure 3: SFT data annotation is illustrated with a mental health service example, where a GPT-4 annotator performs a step-by-step problem-solving review, generating rewrites.

For the general-domain dataset, we use AmbigQA(Min et al., 2020), which tests models on query ambiguity using documents from various web-based sources and Wikipedia.

The number of document tokens for CPT in each dataset is shown in Table 1. The scale of general domain datasets is much larger than that of specialized domain datasets.

Dataset	Document Token Count
SRCQA	4.226M
SyllabusQA	0.243M
FintextQA	6.893M
AmbigQA	2.320B

Table 1: Number of document tokens for each dataset

4.2 Evaluation Metrics

Due to the absence of gold rewrite and document retrieval annotations in these datasets, we cannot directly evaluate the impact of rewritten queries on retrieval performance. Instead, we opted for an end-to-end evaluation of the final answers. For the SRCQA dataset, we assess QA performance based on accuracy in answering questions. For SyllabusQA, we employ Fact-QA, a GPT-based evaluation method, to measure the precision, recall, and F1 score of the LLM Reader's answers. In the case of FintextQA, we use Accuracy, ROUGH, and BLEU to gauge text similarity, while for AmbigQA, we focus solely on F1. We ensure that the selected metrics align with those used in the corresponding paper's tests for each dataset.

4.3 Baselines

We evaluated our model against four baselines. Below is a brief description of each baseline model:

Query2Doc (Wang et al., 2023) prompts the LLM to generate pseudo-documents to address the lexical gap between queries and documents.

TOC (Kim et al., 2023) disambiguates queries using LLMs. To adapt TOC for closed-domain QA tasks, we replaced the web search engine with QA-specific documents, utilizing the LLM for query rewriting.

RL (Ma et al., 2023) finetunes LLM using reinforcement learning techniques, where the discrepancy between model predictions and ground truth answer acts as a reward signal.

RaFe (Mao et al., 2024) fintunes the rewriter based on feedback from a reranker that scores rewritten queries, using a threshold value to optimize the clarity of sentences.

In order to ensure a fair comparison between our R&R model and the baseline models, the prompt template of baselines is also in instruction-demonstration-question format, and the instruction is the same as the task instruction in Figure 8. The demonstration part uses the demonstration mentioned in the original baseline paper. For our evaluation, we reproduced RL, RaFe, and Query2Doc models on closed-domain QA datasets. More details are provided in the Appendix.

4.4 Experimental Setup

RAG Pipeline. We employ LangChain (Chase, 2022) to implement the RAG pipeline. The rewriter models encompass baselines and our proposed R&R, each utilizing different foundational models. Note that every rewriter tested adheres to the same instruction prompt template, which includes the knowledge domain and the motivation for rewriting the query. This uniformity helps mitigate the impact of variations in prompts across different rewriters. The retriever component comprises dense vector retrievers with OpenAI text embeddings and

FAISS (Douze et al., 2024) vector stores. We have set $k = 4$ as the number of top relevant documents to be retrieved. We utilize GPT-4o-mini to generate the final answer.

Data Partitioning. This study utilizes question-answer pairs and reference documents. Reference documents train the CPT model. Question-answer pairs are split into training data for SFT and testing data for end-to-end evaluation, following the train/test splits outlined in the original data set papers: SRCQA (90/10, simulated and real exam questions), Syllabus and AmbigQA (80/20, random split), and FintextQA (80/20, financial textbooks and regulatory policies).

Training Details. The training process was facilitated by LLaMA-Factory (Zheng et al., 2024) for open-source LLMs and OpenAI Platform for ChatGPTs. Query rewriters on three datasets were trained respectively and tested separately. Both continual pretraining and supervised finetuning were based on the LoRA technique, with parameters tuned to $\alpha = 16$, $rank = 8$, and $dropout = 0$, applied uniformly across all target layers. For optimization, we employed AdamW with a maximum gradient norm of 1.0. The experiments are conducted on a single NVIDIA 4090 24GB GPU.

We utilize bf16 precision to improve performance, establishing a cutoff length of 512 tokens per sample for CPT and 2048 for SFT. The learning rate is set at 5e-5, with the model trained for 3 epochs using a batch size of 8 for CPT and 2 for SFT, on up to 100,000 samples. Additionally, we optimize memory and computation using flash-attention and bitsandbytes quantization.

5 Results and Analysis

5.1 Main Results

Comparison Against Method w/o Rewriter The results in Table 2 indicate that our selected rewriting method significantly boosts performance on SRCQA and FinTextQA compared to the retrieval method without rewriting. However, on SyllabusQA, the performance without rewriting surpasses the three baseline methods—TOC, RL, and RaFe—and is comparable to Query2doc. We hypothesize that this outcome is due to the logical multi-hop nature of SyllabusQA, which demands precise retrieval, making unnecessary rewriting prone to retrieval errors.

Comparison against baselines on Qwen2.5-7B Under the condition that the Foundation With the Foundation LLM being Qwen2.5-7B, our method outperforms baselines across all three datasets. This demonstrates that enhancing knowledge for query rewriters can significantly boost performance in specialized retrieval question-answering systems. Notably, our method's Precision on SyllabusQA is slightly lower than that of Query2Doc. This is likely due to the smaller knowledge gap between questions and documents in SyllabusQA compared to the other datasets, making it less challenging for baseline methods to perform well. A more detailed analysis of this behavior is provided in Section 5.7.

Limitations of General QA. Our approach is less effective than Professional QA and is similar to the non-rewriting method, whereas TOC effectively addresses vague general-domain queries. Nevertheless, our method did not negatively impact the overall performance of the question-answering system. ***Consequently, the upcoming experiments will concentrate on performance on professional QA.***

Performance on other foundation rewriting LLMs. To demonstrate the versatility of our method across various foundation LLMs, we evaluated R&R using Gemma2-2B and Llama3-8B as base models, labeled R&R-2B and R&R-8B, respectively. Both R&R-2B and R&R-8B consistently outperform the Qwen2.5-7B baseline models. Notably, R&R-2B performs similarly to R&R-8B on SyllabusQA and FintextQA, indicating that our method is effective for both small and medium-sized LLMs.

5.2 Evaluation of Query-Document Discrepancy

We assess Query-Document Discrepancy by measuring the semantic similarity between the query and the document, where higher similarity indicates lower discrepancy. We evaluated the impact of rewriting and CPT on discrepancies across three professional QAs. As illustrated in Figure 4, SRCQA exhibits greater discrepancy than FintextQA, which in turn has more than SyllabusQA. Query rewriting effectively decreases discrepancy, and CPT further reduces it by integrating knowledge. In datasets with smaller discrepancies, like SyllabusQA, the effects of rewriting and CPT are less significant. Intuitively, lower Query-Document Discrepancy means less knowledge needs to be supplemented.

Rewriting Method	Rewriting LLM	Professional							General
		SRCQA	SyllabusQA			FinTextQA			AmbigQA
		Acc	P	R	F1	Acc	ROUGE-L	BLEU	F1
w/o rewriter	/	0.286	0.438	0.554	0.489	0.458	0.227	0.049	0.623
TOC	Qwen2.5-7B	0.428	0.405	0.468	0.434	0.489	0.253	0.066	**0.658**
RL	Qwen2.5-7B	0.404	0.447	0.398	0.421	0.467	0.245	0.060	0.597
RaFe	Qwen2.5-7B	0.444	0.421	0.517	0.464	0.479	0.266	0.058	0.583
Query2Doc	Qwen2.5-7B	0.381	**0.470**	0.521	0.494	0.493	0.254	0.062	0.512
R&R-7B	Qwen2.5-7B	**0.622**	0.463	**0.584**	**0.517**	**0.505**	**0.285**	**0.081**	0.625
R&R-2B	Gemma2-2B	0.515	0.459	0.578	0.511	0.498	0.274	0.073	0.617
R&R-8B	Llama3-8B	0.600	0.461	0.577	0.512	0.509	0.280	0.077	0.629

Table 2: Performance comparison among different query rewriters. The best and second-best results of the same rewriting LLM are **bolded** and underlined.

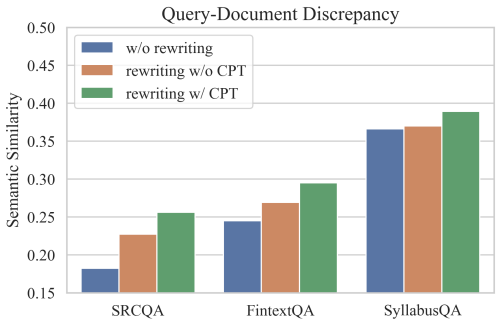

Figure 4: Evaluation of Query-Document Discrepancy in Professional QAs, assessed by measuring the semantic similarity between queries and documents.

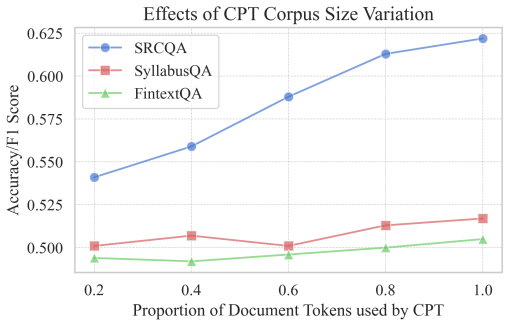

Figure 5: Impact of Corpus Size on R&R: We adjust the corpus size by varying the proportion of document tokens in three professional QAs.

5.3 Influence of Corpus Size

To further validate the role of CPT, we examined how corpus size impacts rewriting performance. We proportionally sampled three sets of document tokens from professional QA for CPT. Results in Figure 5 show that a larger document token count generally enhances performance. This effect is particularly significant for SRCQA compared to SyllabusQA and FintextQA, as SRCQA has a greater Query-Document Discrepancy, enabling CPT to offer more knowledge enhancement.

5.4 Influence of Model Scale

We investigated how model scale affects training duration and performance by testing Qwen2.5 models with parameters ranging from 0.5B to 7B. We recorded the continual pretraining time for each scale. As shown in Figure 6, both model performance and training duration increase at a slower rate with larger models, confirming that our rewriting model adheres to the scaling law of LLMs(Zhang et al., 2024). In particular, training a 7B model with 100k tokens takes only 43.9 seconds, highlighting the efficiency and time-saving nature of our approach. *This indicates that our approach is source-efficient.*

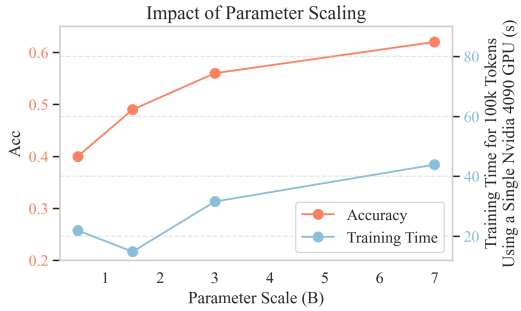

Figure 6: Experiment Results: Impact of Model Scale on Training Duration and Scores. Evaluation is done on SRCQA

5.5 Impact of the CPT on Direct Question-Answering

We conducted an expansion experiment to assess the effect of CPT on the direct question-answering process, which exclusively utilized the LLM reader for answering questions without query rewriting or retrieval.

Table 3 shows that all models' performance has significantly declined due to insufficient retrieval.

LLM	Dataset (Metric)		
	SRCQA (Acc)	SyllabusQA (F1)	FinTextQA (Acc)
Qwen2.5-14B	0.196	0.174	0.377
Qwen2.5-14B+CPT	**0.203**	**0.179**	**0.380**
GPT-4o-mini	0.201	0.177	0.396
GPT-4o-mini+CPT	0.198	**0.180**	0.394

Table 3: Impact of CPT on Direct Question-Answering without retrieval. When CPT improves the original performance, data is bolded. Otherwise, it is highlighted in red.

The LLM with CPT performs similarly to the one without CPT across all three datasets. This indicates that *CPT is only helpful for document retrieval and has no significant assistance in directly answering questions.*

5.6 Ablation Study

Table 4 shows the results of the ablation studies on R&R-7B. It compares the performance of models under different configurations: directly prompting the foundation LLM to generate rewrites without any training (**Vanilla**), only continual pretraining (**CPT**), only finetuning (**SFT**), and CPT combined with supervised finetuning (**CPT + SFT**).

In Table 4, while continual pretraining (CPT) and supervised fine-tuning (SFT) can improve the performance separately, the combination of CPT and SFT consistently yields the best performance. This indicates that these methods may complement each other. A detailed analysis of this phenomenon in section 7 shows that CPT may be good at providing domain knowledge, while SFT enhances task generation style.

5.7 Case Study

Figure 7 presents three examples: example 1 from SRCQA, example 2 from SyllabusQA, and example 3 from FintextQA. Example 1 compares our R&R to R&R without CPT or SFT, while examples 2 and 3 are used to compare our R&R with baseline methods. From these examples, we draw two conclusions.

c1. Both CPT and SFT are crucial for R&R. In Example 1, R&R output without CPT is concise but lacks domain-specific details, such as the connection between financial reports and securities issuance, and contains inaccuracies like conflating "ChiNext" with "ChiNext listed companies," leading to retrieval errors. Without SFT, R&R may showcase bond issuance knowledge but often generates excessive and disorganized content, neglecting relevancy. In contrast, R&R which incorporates both CPT and SFT concisely covers all key aspects, including ChiNext's financial report requirements and the impact of audit opinions, resulting in a professional, precise output that directly addresses user queries without redundancy.

Datasets	Metric	Vanilla	CPT	SFT	CPT + SFT
SRCQA	Acc	0.428	$0.489_{(\uparrow 0.061)}$	$0.526_{(\uparrow 0.098)}$	$0.622_{(\uparrow 0.194)}$
SyllabausQA	F1	0.462	$0.475_{(\uparrow 0.013)}$	$0.492_{(\uparrow 0.030)}$	$0.517_{(\uparrow 0.055)}$
FintextQA	Acc	0.468	$0.481_{(\uparrow 0.013)}$	$0.494_{(\uparrow 0.026)}$	$0.505_{(\uparrow 0.037)}$

Table 4: Ablation experiment results on the R&R-7B model across different datasets.

c2. R&R is better on highly specialized datasets. On datasets with lower specialization, like SyllabusQA, R&R performs similarly to Query2doc. However, on highly specialized datasets, R&R has the greatest advantage. In example 2, the knowledge point to rewrite the query is relatively straightforward to analyze. However, in example 3, the query is more complex and nuanced, with a significant gap between the query and the underlying knowledge point. This complexity poses a greater challenge for Query2Doc, which tends to focus on generating pseudo-document sections rather than deeply interpreting the knowledge points. In contrast, R&R excels at bridging this gap by capturing the knowledge points behind such intricate queries, leading to more accurate and contextually appropriate responses.

6 Related Works

6.1 Retrieval Augmented Generation

Early sparse retrievers were untrained, term-based systems (Chen et al., 2017; Ramos et al., 2003; Trotman et al., 2014), while more recent dense retrievers leverage pretrained models to generate embeddings for comparing query and document similarities (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Su et al., 2023). Enhancements to these methods include pretraining for relevance matching (Izacard and Grave, 2021), joint representation learning (Ge et al., 2023), and the use of contrastive learning to improve retriever performance (Rubin et al., 2022). Unlike these approaches, our work focuses on pretraining the query rewriter, optimizing the alignment between user queries and the retriever before the retrieval process begins, thereby improving the overall relevance of the retrieved information.

While much work has focused on improving the

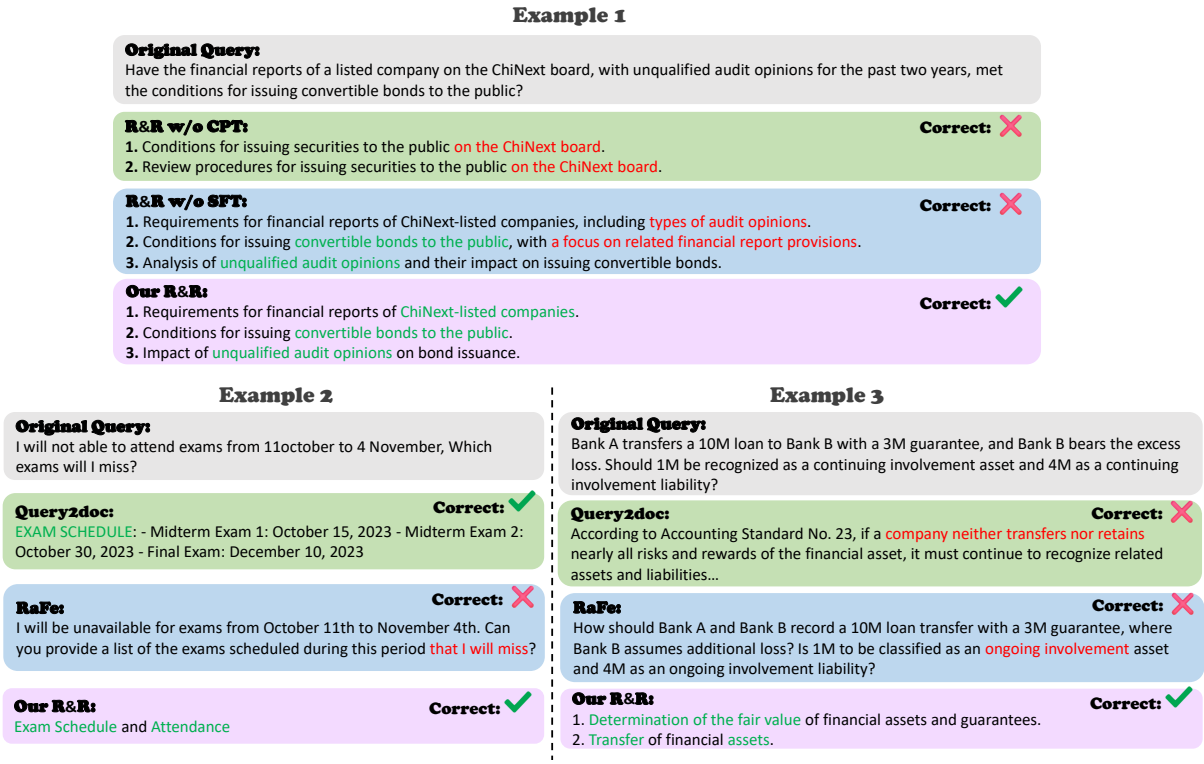

Figure 7: Three examples of query rewriting, illustrating the original queries and their rewrites generated by baseline models compared to our R&R approach (all based on Qwen-7B). "Correct" indicates whether the rewritten query leads to a correct answer. Keywords that lead to incorrect answers are highlighted in red, while those contributing to correct answers are highlighted in green.

retriever, recent efforts are expanding to the entire RAG pipeline, which includes query rewriting (Ma et al., 2023), retrieval, and the LLM reader(Yu et al., 2023). Dual instruction tuning between the retriever and the LLM reader (Lin et al., 2023) contributes to improving both the retriever and the LLM reader. Additionally, reranking retrieved documents (Zhuang et al., 2024) or employing natural language inference models for robustness (Yoran et al., 2023) can further enhance the LLM reader's ability to generate accurate results.

6.2 Query Rewriting with LLMs

Prior research on LLM-based query rewriting has addressed several key challenges, such as handling unclear user query intentions (Liu et al., 2024), interpreting the multifaceted meanings of queries (Wang et al., 2024), and incorporating historical context in dialogue-based queries (Jang et al., 2024; Wu et al., 2022). Various methods have been proposed to address these challenges, including query expansion with LLM feedback (Mackie et al., 2023), pseudo-document generation (Wang et al., 2023; Gao et al., 2023), query decomposition (Chan et al., 2024; Kim et al., 2023), and leveraging LLM reader feedback for reinforcement learning (Ma et al., 2023). These approaches aim to expand, refine, or restructure the information within queries to improve retrieval accuracy.

Our query rewriting algorithm specifically focuses on scenarios where the rewritten queries contain dense domain-specific knowledge. This places higher demands on the rewriter's ability to not only understand but also effectively utilize complex domain knowledge to generate accurate and relevant rewrites.

7 Conclusion

This paper presents R&R, a novel query rewriting method that integrates continual pre-training and supervised fine-tuning to align rewritten queries with domain-specific documents. Experiments across multiple datasets demonstrate that our method outperforms existing methods, especially in knowledge-intensive tasks. Our work highlights the importance of domain-aware query rewriting for retrieval-augmented QA and offers a practical approach for integrating trainable components into RAG pipelines using LLMs.

8 Limitations

The proposed method in this paper requires continual pre-training, which entails training a dedicated rewriter model for each corpus. Although this approach improves answer quality, it requires users to train and deploy their own LLMs, limiting the potential application scope of the proposed method primarily to scenarios where high response accuracy is needed. Furthermore, larger corpora contain more extensive knowledge, and retaining such knowledge may require either larger models or full fine-tuning.

References

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. *arXiv preprint arXiv:2404.00610*.

Harrison Chase. 2022. LangChain.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Jian Chen, Peilin Zhou, Yining Hua, Yingxin Loh, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024. Fintextqa: A dataset for long-form financial question answering. *arXiv preprint arXiv:2405.09980*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Nigel Fernandez, Alexander Scarlatos, and Andrew Lan. 2024. Syllabusqa: A course logistics question answering dataset. *arXiv preprint arXiv:2403.14666*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Suyu Ge, Chenyan Xiong, Corby Rosset, Arnold Overwijk, Jiawei Han, and Paul Bennett. 2023. Augmenting zero-shot dense retrievers with plug-in mixture-of-memories. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1796–1812.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. Itercqr: Iterative conversational query reformulation with retrieval guidance. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8114–8131.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*.

Yushan Liu, Zili Wang, and Ruifeng Yuan. 2024. Querysum: A multi-document query-focused summarization dataset augmented with similar query clusters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18725–18732.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2026–2031.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. Rafe: Ranking feedback improves query rewriting for rag. *arXiv preprint arXiv:2405.14431*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*.

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

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

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

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, pages 58–65.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303.07678*.

Shuting Wang, Xin Xu, Mang Wang, Weipeng Chen, Yutao Zhu, and Zhicheng Dou. 2024. Richrag: Crafting rich responses for multi-faceted queries in retrieval-augmented generation. *arXiv preprint arXiv:2406.12566*.

Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. 2022. CONQRR: Conversational query rewriting for retrieval with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10000–10014, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, S Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference on Learning Representations*.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024. When scaling meets llm finetuning: The effect of data, model and finetuning method. *arXiv preprint arXiv:2402.17193*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47.

A Prompt Template for R&R

Task instruction:
Please rewrite this query to overcome the limitations of vector-distance-based retrieval systems.
Demos:
Input: What is the grading scale to get an A in this course?
Reasoning: The primary task is to extract the knowledge points from the input, identifying ..... Next, connect these concepts to the broader knowledge point.... Finally, rewrite the input to concisely....
Output: Academic Assessment (Grading)
Question
Input: xxxxxxxxx

Figure 8: The prompt template of R&R, including task instruction, demonstration, and question. The demonstration consists of the example input, reasoning process, and example output.

Our R&R was tested on a total of 4 datasets. During the test, the task instructions in the prompt template were the same, and the demos were selected from samples in the dataset to be tested, which were manually verified. Figure 8 shows the demo on SyllabusQA, followed by part of demonstrations from other datasets.

Input:
The correct statement regarding joint venture arrangements is ( ). A joint venturer increases its interest in a joint operation that constitutes a business; if the joint venturer still has joint control over the joint operation, the interest in the joint operation previously held by the joint venturer should be measured at fair value on the date of the additional investment···
Reasoning:
The question do not require reasoning.
Output:
To answer option A, in the textbook for sponsored representatives, what I need to look up the most are: 1. The definition and classification of joint arrangements. 2. The specific criteria for determining joint control in joint arrangements. 3. The accounting treatment principles for joint operations.

Figure 9: Demo for SRCQA

B Additional Explanation on Training Data

B.1 SRCQA Data collection and processing

We acquired supplementary educational materials, primarily comprising:

c1. The most up-to-date textbooks as of the end of 2023;

c2. Authentic examination questions from 2017 to 2023;

c3. Specific knowledge points categorize Simulated questions developed by educational institutions.

Input:
Are transactions signed between 12 July 2023 and 12 October 2023 (and implemented on 12 October 2023 or later) subject to mandatory notification under Regulation EU 2022/2560?
Reasoning:
First, we need to understand the specific provisions of Regulation EU 2022/2560. We have to check whether there are specific time limits and conditions stipulated in this regulation regarding the mandatory notification of transactions.
Then, we need to determine whether the time range from 12 July 2023 to 12 October 2023 (and transactions implemented on 12 October 2023 or later) falls within the scope of the situations that require mandatory notification as defined by the regulation. This may involve analyzing factors such as the nature of the transactions, the parties involved, and the specific circumstances of the transactions.
Finally, based on the above analysis and understanding of the regulation, we can make a judgment on whether these transactions are subject to mandatory notification.
Output:
The understanding and application of the time limits and mandatory notification requirements of transactions in Regulation EU 2022/2560.

Figure 10: Demo for FintextQA

Input: Who made the play *The Crucible*?
Reasoning:
The original question "Who made the play *The Crucible*?" is ambiguous because the verb "made" can have multiple meanings. In the context of a play, "made" could potentially refer to the act of writing (the most common interpretation, as in who created the text of the play)
Output:
Who wrote the play *The Crucible*?

Figure 11: Demo for AmbigQA

These materials were digitized by scanning them and subsequently converted into editable text format using OCR software. All graphical elements within the documents were removed during this process.

We manually annotated all titles and hierarchical levels for each textbook document to preserve the document structure. Considering that these textbook documents would be utilized for continued pre-training of LLMs, they needed to be segmented into multiple textual data entries. In this process, we employed the following strategy to maintain the integrity of information in each data entry:

s1. We constructed a document title tree based on the annotated data, where each node corresponds to a chapter or section (specifically, leaf nodes correspond to the smallest indivisible subsections). We assumed the input length limit of the LLM to be n.

s2. We traversed the title tree using a depth-first approach. If the length of the chapter corresponding to the current node i does not exceed n, we sequentially examined the lengths of chapters corresponding to its sibling nodes until their cumulative length surpassed n. Denoting these nodes as $i, i+1, ..., i+k$, we merged the chapters corresponding to $i, i+1, ..., i+k-1$ into a single data

entry. The next traversal would then commence from node $i + k$.

s3. If the length of the chapter corresponding to the current node i exceeded n, we continued the downward traversal until reaching a node with a chapter length less than n, then repeated step 2.

s4. If, upon reaching a leaf node, the chapter length still exceeded n, we segmented it based on natural paragraphs, striving to make the length of each data entry as close to n as possible.

For the authentic examination questions and simulated questions, we employed regular expressions to extract the following components for each item: the stem of the question, the options (in the case of multiple-choice questions), the correct answer, and the accompanying explanation.

B.2 Document Data Format

The format of our document data conforms to the document's directory structure. We'll also split the full data of the document according to this structure, using the approach outlined in Section B.1. Taking a page from the SRCQA document as an example, we'll show our splitting results.

As shown in Figure 12, a document is split while preserving its original structure in Markdown format. The example document, titled "Sponsor Representative Competency Exam Guide," contains hierarchical sections, such as chapters and subsections.

The document is divided into two parts. Split Doc 1 covers contingent liabilities, including definitions, accounting treatment, disclosure, and conversion. Split Doc 2 addresses contingent assets with similar elements. Each split maintains the same headings and structure as the original to ensure readability and consistency.

C Details of Baselines Reproduction

TOC: TOC is designed to clarify ambiguous queries using LLMs. It integrates disambiguated candidates with web search results. Subsequently, these candidates undergo a factual verification process based on a tree data structure. To adapt TOC for closed-domain QA datasets, the web search engine is replaced with QA-specific documents, and the LLM is utilized for disambiguation as a query rewriter. The underlying LLM for TOC is Qwen2.5-7B.

RL: RL refers to a LLM that has been fine-tuned using reinforcement learning techniques. This model employs the discrepancy between the LLM reader's predictions and the actual ground truth results as a reward signal. First, we generate some pseudo data and do warm-up training. We rewrite the original queries using prompts for the LLM and collect the generated pseudo labels for the warm-up training. Then, we optimize the Rewriter using reinforcement learning, generating queries at each step and assessing their impact on the final predictions. The reward function is based on the correctness of the LLM's responses and a KL divergence regularization. We use policy gradient methods (like PPO) to update the model and improve the query rewriting. The warm-up training is done with a learning rate of 3e-5 over 8 training epochs. In the reinforcement learning phase, we set the sampling steps to 5120, with 512 samples per step, using a learning rate of 2e-6 and a batch size of either 8 or 16, training for 3 epochs. The reward function parameters include λ_f and λ_h set to 1.0, a KL divergence target of 0.2, and β initialized at 0.001 and adjusted dynamically.

RaFe: RaFe is a process that involves fine-tuning the LLM rewriter based on feedback from the LLM reranker. The LLM reranker evaluates the rewritten results by assigning scores and determines its preference using a threshold value, which we set at 0.5. This preference is then used as feedback. Employing Direct Policy Optimization (DPO) and KTO, the LLM rewriter can be refined to reformulate sentences more effectively, thereby enhancing their clarity and comprehensibility. For PPO, the batch size is set to 32, and it is trained for 1000 optimization steps (approximately 1.067 epochs). The clip range parameter ϵ and the coefficient β_{KL} for the KL divergence are both set to 0.2. For DPO and KTO, the offline training is carried out for 1 epoch on all the good - bad rewrite data with a learning rate of 5e-6, and the temperature parameter β is set to 0.1.

Query2Doc: Query2Doc is designed to prompt the LLM to generate pseudo documents, aiming to bridge the lexical and linguistic expression gap between queries and documents. We have tested Query2Doc on two foundational LLMs: Qwen2.5-7B and GPT-4o.

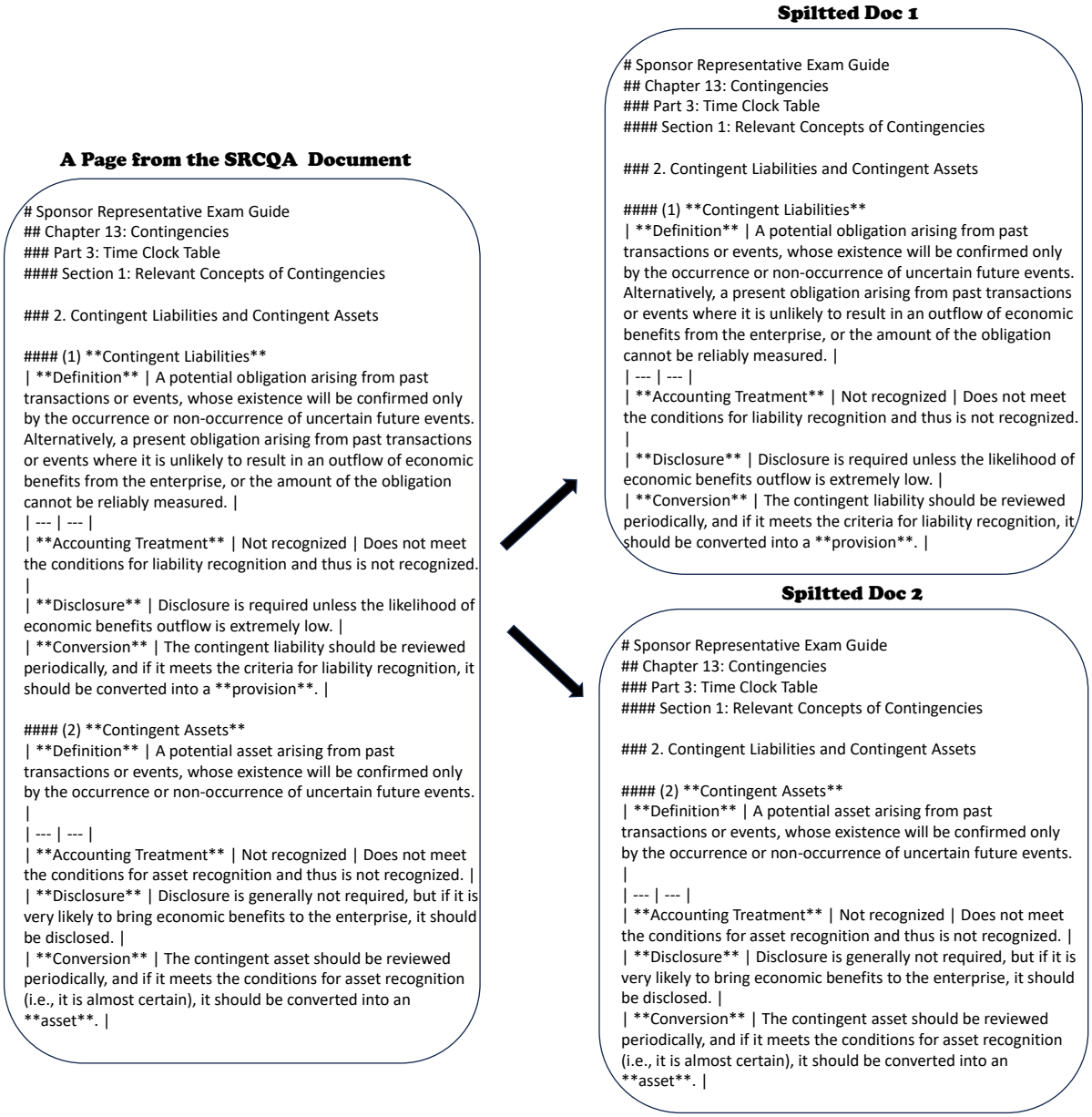

Figure 12: An Example Doc from SRCQA. The data format follows the document's directory structure in markdown format. The document splits also maintain the original directory structure.