KaPQA: Knowledge-Augmented Product Question-Answering

Swetha Eppalapally[1], Daksh Dangi[1], Chaithra Bhat[1], Ankita Gupta[1], Ruiyi Zhang[2], Shubham Agarwal[2], Karishma Bagga[2], Seunghyun Yoon[2], Nedim Lipka[2], Ryan A. Rossi[2], Franck Dernoncourt[2]

[1]University of Massachusetts Amherst, [2]Adobe Research

Abstract

Question-answering for domain-specific applications has recently attracted much interest due to the latest advancements in large language models (LLMs). However, accurately assessing the performance of these applications remains a challenge, mainly due to the lack of suitable benchmarks that effectively simulate real-world scenarios. To address this challenge, we introduce two product question-answering (QA) datasets focused on Adobe Acrobat and Photoshop products to help evaluate the performance of existing models on domain-specific product QA tasks. Additionally, we propose a novel knowledge-driven RAG-QA framework to enhance the performance of the models in the product QA task. Our experiments demonstrated that inducing domain knowledge through query reformulation allowed for increased retrieval and generative performance when compared to standard RAG-QA methods. This improvement, however, is slight, and thus illustrates the challenge posed by the datasets introduced.

1 Introduction

The advancements in large language models (LLMs) led to exponential growth in domain-specific applications. Question Answering has emerged as one of the prominent domain-specific applications. As the demand for accurate and reliable QA systems increases, generic RAG-QA approaches often struggle to deliver satisfactory results within the specialized domains. This challenge has spurred active exploration in this area, with researchers employing various novel methodologies (Nguyen et al., 2024; Setty et al., 2024; Jiang et al., 2024; Rackauckas, 2024) to improve QA systems.

Additionally, training and evaluating these systems rigorously remains crucial. This trend underscores the critical need for domain-specific QA datasets to facilitate the training and evaluation of such systems. Notably, while efforts have been directed towards releasing datasets across prominent and expansive domains such as Medicine (Pal et al., 2022; Pampari et al., 2018), Finance (Chen et al., 2021; Zhu et al., 2021), and Legal (Zhong et al., 2019; Chen et al., 2023a), there remains an apparent scarcity of such datasets in the area of software products.

To address this gap, our work investigates such industry-specific QA datasets, namely the Adobe HelpX datasets, and releases them for others to benchmark against and further improve their QA systems. These datasets comprise of user queries and their corresponding answers pertaining to Adobe products, specifically Acrobat and Photoshop. By providing these benchmark datasets, we aim to offer valuable resources for assessing the performance of domain-specific RAG-QA systems. The datasets will be released after obtaining relevant permissions from Adobe.

Furthermore, we introduce a novel LLM-based Knowledge-Driven RAG-QA framework designed to seamlessly accommodate domain knowledge into RAG-QA systems. This framework leverages comprehensive knowledge bases for query expansion, thereby enhancing both retrieval and generation in domain-specific QA tasks.

Through extensive experimentation, we've determined that performing accurate retrieval over these datasets poses a unique challenge. Even introducing this concept of query augmentation using knowledge directly from the corpora only helped improve the model so much. This illustrates how these datasets are challenging ones - as even complex frameworks such as the one we've proposed, could only result in so much improvement.

By contributing these datasets and proposing an innovative framework, we aim to advance LLM technology and its application in domain-specific QA tasks, ultimately improving user experiences and operational efficiency across various industries.

2 Related Work

2.1 Domain-specific question answering

Several research efforts have been made to curate domain-specific question-answering benchmarks and training datasets, spanning domains like biomedical (Pal et al., 2022; Pampari et al., 2018; Li et al., 2021), finance (Chen et al., 2021; Zhu et al., 2021), and legal (Zhong et al., 2019; Chen et al., 2023a). In contrast, our work focuses on product question-answering, which is valuable in many enterprise settings. Furthermore, unlike many of these existing datasets that provide a simpler multiple-choice question-answer format, our work focuses on generative question-answering.

Among the research efforts in product-specific question-answering, Yang et al. (2023) also provides a dataset focused on answering user queries about Microsoft products. However, many of the question-answer pairs in this dataset require a yes/no answer, with only a small portion requiring more complex answers. The PhotoshopQuiA (Dulceanu et al., 2018) dataset is more closely related to our work in terms of domain, as it is also based on the Adobe Photoshop product. However, it specifically focuses on *why* questions. In contrast, our work centers on *how-to* queries, necessitating the model to generate a detailed sequence of steps to complete an operation. These answers are considerably more challenging to generate because their usefulness hinges on the accuracy of each individual step. If even one step in the generated answer is incorrect, the overall utility of the answer is compromised.

2.2 Augmenting LLMs

The Retrieval Augmented Generative (RAG) framework has been extensively worked on for years and (Gao et al., 2024; Zhao et al., 2024; Li et al., 2022) present a detailed examination of the progression of RAG paradigms (Ma et al., 2023; Ilin, 2024; Shao et al., 2023; Yu et al., 2023), and introduce the metrics and benchmarks for assessing RAG models (Chen et al., 2023b; Lyu et al., 2024). One suggested future direction involves identifying methods to fully harness the potential of Large Language Models (LLMs) to enhance domain-specific RAG systems, aligning with our aim of leveraging LLMs to answer queries related to Adobe products.

Recent efforts aim to enhance LLMs' contextual generation in specific domains by incorporating external knowledge (Mialon et al., 2023). Fatehkia et al. (2024) propose Tree-RAG where they utilize a tree structure to depict entity hierarchies in organizational documents and supplement context with textual descriptions for user queries. However, their approach is ineffective for documents lacking hierarchical organization such as ours. Another method to incorporate industry domain-specific information is presented by Yang et al. (2023). It involves getting a domain-specific language model with aligned knowledge and then feeding it to an LLM to generate enriched answers. We propose an alternative method to solve domain-specific RAG-QA through the construction of a comprehensive knowledge base consisting of triples and a multi-stage query reformulation pipeline. Zhu et al. (2024) evaluates LLMs for Knowledge Graph (KG) tasks across diverse datasets, highlighting their suitability as inference assistants. Additionally, Jagerman et al. (2023) explore query expansion by prompting LLMs through zero-shot, few-shot and Chain-of-Thought (CoT) learning. Our work takes it a step further by incorporating knowledge base tuples in query expansion.

3 Dataset Creation

3.1 Data Pre-processing

The corpus is sourced from the publicly available Adobe HelpX[1] web pages for Acrobat and Photoshop products, which explain how to use the functionalities present in these products.

A crawling script is employed to extract the content from the web pages, segmenting them into distinct sections based on H2 headings. Each of these sections typically represents a specific topic or task within the respective product. The resulting sections tend to be non-overlapping, facilitating targeted analysis.

Throughout the process, all clickable and in-section links within the web pages are transformed into plain text to maintain consistency, while images are omitted to ensure that the corpus comprises solely textual content.

3.2 Question-Answer Pairs with URLs Creation

Gold question-answer (QA) pairs are meticulously crafted for analysis. Product experts, recruited through Upwork for Adobe Acrobat and Telus International, are instructed to write how-to questions

[1] https://helpx.adobe.com

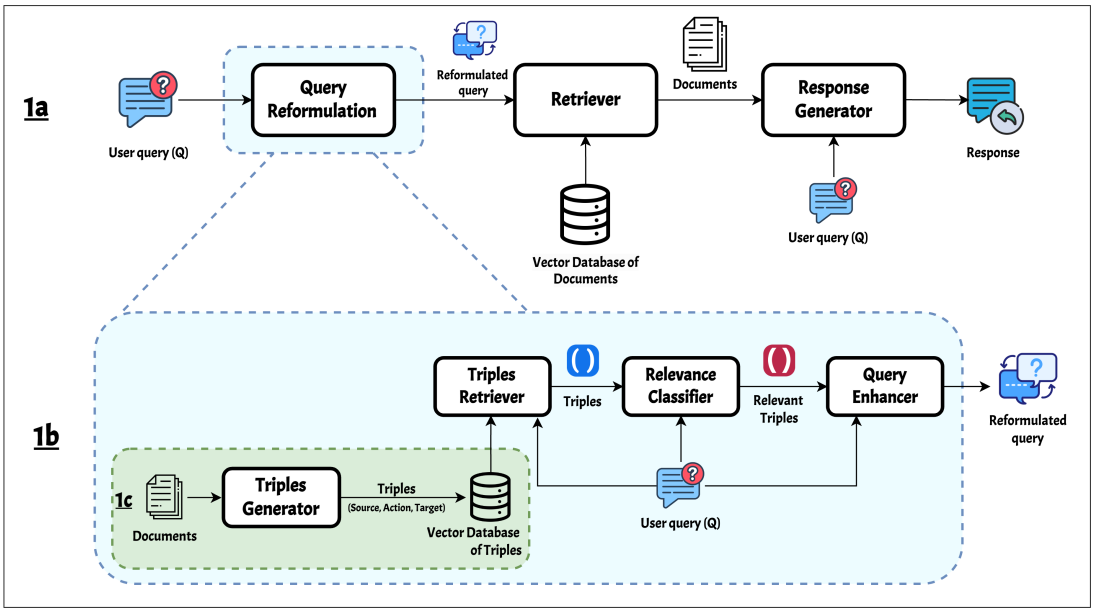

Figure 1: The figure represents our proposed framework. **1a** depicts the main RAG-QA pipeline consisting of a retriever and a generator, along with our proposed query reformulation sub-pipeline. **1b** gives a detailed view of the various components in our sub-pipeline. The process starts with the generation of knowledge base triples using the Triples Generator. Next, all matching triples to the user query are retrieved using the Triple Retriever, classified based on their relevance to the original query using the Relevance Classifier and finally reformulated using the Query Enhancer.

and answers that provide procedural steps to accomplish specific tasks using the software. Furthermore, each QA pair is mapped with its respective source web page.

The experts manually created question-answer pairs based on the respective HelpX web pages for Adobe Acrobat. Conversely, for Adobe Photoshop, GPT-4 initially generated question-answer pairs based on the web pages, which are subsequently reviewed and corrected by product experts to guarantee accuracy and relevance to the question and the answer.

This systematic approach of the question-answer pair generation ensures the integrity and usability of the dataset for evaluation and research in the domain of software products and support.

4 Data Analysis and Statistics

The Adobe Acrobat and the Photoshop datasets contain questions, answers, and corresponding source web page URLs. All questions in this dataset are how-to type asking steps to perform operations such as changing text font, editing text in a PDF, and creating certificate-based signatures. The gold answers to these questions provide procedural steps to accomplish these operations and the URLs of the web pages allow for independent verification of the answer. Table 1 provides insights into various metrics, such as the count of question-answer pairs, the average length of questions and answers, and the number of web pages and sections.

Answering these how-to queries presents significant challenges due to the critical nature of every step involved. It's essential to emphasize that for an answer to the "How to" query to be entirely accurate, each step must be precise, and all steps must be in the correct sequence. Even a minor mistake in the explanation of a step or its order can invalidate the entire utility of the answer. Moreover, the average number of steps per response for a query in the Adobe Acrobat dataset is 4.71, indicating that most queries

Metric	Acrobat	Photoshop
No. of QA pairs	131[a]	96
Avg. length of questions (words)	8.80	12.74
Avg. length of answers (words)	118.71	98.71
Total no. of web pages	146	349
Total no. of sections	1281	2478
Avg. no. of sections per web page	8.78	7.1
Avg. length of the section (words)	135.75	121.09

Table 1: Statistics for the Adobe Acrobat and Photoshop datasets.

[a]The count includes 22 composite questions. We focus on non-composite questions in this work (data analysis and experimentation). Composite Questions are an area for future exploration.

requires multi-step solutions. Below is an example highlighting the necessity for accurate and detailed instructions within every step of the response.

Sample Q&A from the dataset

Q: How to insert Images into a PDF?
A: 1. Open the PDF in Acrobat and go to the `Edit` menu.
2. Select `Image` from the `Add Content` submenu.
3. In the dialog box, choose the image file you want to insert.
4. Select the location where you want to insert the image or use the drag option to resize it as you insert.
5. A copy of the image file will appear on the page with the same resolution as the original file.

Adding to the complexity, a substantial portion of the Acrobat dataset, over 50%, is comprised of implicit questions, defined by their brevity and conversational tone, lacking clear indications of user intent, as shown in Table 2. Furthermore, close to a quarter of the questions of the Acrobat dataset are ambiguous, lacking clear context and leading to multiple potential interpretations of the ask. Our proposed QA framework has been designed considering these challenges and it effectively interprets the user ask.

To assess the generalizability of our proposed QA framework, we curated an additional synthetic dataset focused around Adobe Photoshop product, closely resembling Adobe Acrobat in terms of question type, question and answer lengths, as shown in Table 1. Additionally, the average number of steps per answer for Photoshop dataset is 4.6. However, since these are synthetically framed queries, they are well-formed, explicit, and unambiguous. By contrasting the characteristics of the synthetic dataset with those of Adobe Acrobat, we aim to evaluate the adaptability of our approach.

Moreover, both datasets serve as evaluation benchmarks, representing real-world user queries (both implicit and ambiguous) in Adobe Acrobat and controlled questions in the synthetic Adobe Photoshop dataset. They offer question-answer pairs for diverse scenarios, making them valuable resources for research in the software product domain.

5 Methodology

As summarized by (Zhao et al., 2024; Li et al., 2022; Gao et al., 2024; Lewis et al., 2021), in a standard RAG-QA process, upon receiving an input query, the retriever identifies and retrieves pertinent data sources, which are subsequently utilized by the response generator to enrich the overall generation process. To enable Adobe domain-specific QA, we add an initial query reformulation stage which enhances the user query using knowledge base triples. Query reformulation or rewriting (Anand et al., 2023; Ma et al., 2023) encompasses a set of techniques to transform a user's original query into one that's better aligned with the user's intent, thereby enhancing the retrieval outcomes. Our proposed query reformulation pipeline consists of multiple steps as shown in Fig.1. It starts with the generation of knowledge base triples using the Triples Generator. Next, all matching triples to the user query are retrieved using the Triple Retriever, classified based on their relevance to the original query using the Relevance Classifier and finally reformulated using the Query Enhancer. Refer to Appendix E for the LLM prompts used at various stages. Given below is a detailed outline of the different components in our pipeline:

Step 1: Triples generation. The goal is to represent each document as a collection of triples, each of which represents the key information contained within a document. Each triple is of the form (Source, Action, Target) to mimic what might be asked through a query i.e. assistance to act on a target. E.g., one of the generated triples from a document about editing text and text boxes is (rotation handle, rotate, text box) - the source of the action (rotate) is the rotation handle, which acts on a text box. Similarly, for each document, the LLM contextualizes input text using its vast knowledge base, identifying entities, actions, and relationships, before generating triples by selecting relevant phrases as sources, actions, and targets, informed by inferred context and linguistic patterns. The number of triples for each document varies (approximately 1 to 35 triples) based on the document's content and the model's comprehension. Each triple is then encoded into a numerical vector representation using a pre-trained sentence encoder model which converts the textual elements of the triple into dense vec-

Category	Count of questions (%)	Question Example
Explicit	48 (42.10%)	I need to increase image in PDF towards right direction, how to do that in Acrobat?
Implicit	66 (57.89%)	resize jpg in Acrobat
Ambiguous	28 (24.56%)	Unable to delete PDF content need help.

Table 2: Question Categories with examples in the Acrobat dataset.

tors. These are organized into a high-dimensional index structure, enabling efficient similarity search.

Step 2: Triples Retrieval. This stage accepts the user query as input and then searches the vector store to retrieve all triples related to the query by calculating the similarity scores between the query vector and the vectors of stored triples, utilizing a similarity search algorithm. For every user query, it over-retrieves numerous triples.

Step 3: Relevance Classification. Through the previous step, we obtain numerous triples that have some relevance to the user query. In this stage, we use the capabilities of an LLM to identify only those triples that are the most relevant to the user query. The content of the document along with the list of the triples retrieved in Step 2 are passed to the LLM as a prompt with the instruction that it identifies and return only those triples that are the most relevant to the user's query. Only the triples that are classified as relevant are considered in the subsequent steps.

Step 4: Query Enhancement. Here the user query is reformulated to ensure that it has all the necessary information within it that can help the retriever fetch the correct associated documents. This reformulation is a form of query enhancement where the user query is augmented with words that are used interchangeably in the Adobe products domain. This gives more information for the retriever to use through which it can perform a more accurate search over its vector store and return the documents that are more likely to be relevant. The relevant triples along with the original user query are passed as the prompt to the LLM which rephrases the query.

6 Experiments

We conducted a variety of experiments (as shown in Tables 3 and 4) on the Adobe datasets, where the main datasets formed the corpus of texts to be retrieved from, a selected few of which would be passed in as part of the prompt to the LLM; and the gold set was used for measuring performance. They ranged from using different retrievers to incorporating multiple components and techniques into the RAG-QA pipeline to achieve a performance gain. Throughout the processes we used GPT-3.5 0301 and GPT-4 0314 from Azure OpenAI.

6.1 Baselines

BM25 retriever + LLM: We utilize a BM25 retriever to scan the document corpus and select the top k (k=3) most relevant documents based on the user's query. The relevance of each document is calculated by considering the frequency of query terms within the document and the document's length relative to the entire corpus. These selected documents then serve as contextual input for subsequent response generation tasks.

DPR + LLM: The Dense Passage Retrieval (DPR) method utilizes embeddings to represent passages and queries as vectors, which are then indexed using a similarity search algorithm. When a query is received, DPR compares the vector representation of the query with those of passages in the store, selecting the top k documents with the highest similarity scores as context for answer generation.

General purpose LLM (or QR with no Triples): We add an intermediate step of query reformulation using an LLM to the second baseline. The LLM is instructed to improve the original user query directly without any additional information, i.e. without the addition of domain knowledge into the query.

6.2 Evaluation Methods

We use different metrics to evaluate the performance of the two main components of the RAG-QA pipeline. For retrieval, we employ the Hit Rate

	Retriever	Query reformulation method	Retriever Hit Rate	Answer Similarity Score		
				ROUGE-L	BERTScore	G-Eval
Baselines	BM25	None	41.2%	0.301	0.835	0.412
	DPR	None	73.6%	0.409	0.859	0.574
Ours	DPR	**augmentation with domain-specific triples**	**74.7%**	**0.416**	**0.860**	**0.578**
Ablation	DPR	via general purpose LLM (w/o triple retriever and relevance classifier)	70.2%	0.393	0.855	0.562
	DPR	w/o triple relevance classifier	65.7%	0.385	0.852	0.557
Oracle	DPR	w/o triple retriever (triples obtained from gold documents)	80.7%	0.455	0.870	0.649

Table 3: Performance of baseline methods, our proposed method, and the ablation experiments over Acrobat test set. In all the above experiments, we use GPT3.5 as the LLM for triple generation and answer generation. The semantic similarity scores are computed between the gold and the generated answer. Among ablations, in *w/o triple relevance classifier* setting, we provide all retrieved triples to the query expansion model; in *via general purpose LLM* setting, we only use LLM to reformulate query without including any triples. In the Oracle setting, *w/o triple retriever*, we use gold documents to generate triples and provide them to the relevance classifier. This setting gives us an upper bound on the performance when correct knowledge triples are known. The reported hit rate is the top-3 hit rate.

or the number of times the Gold document was correctly retrieved as the metric. To measure the quality of the generated answers, multiple metrics are explored. As supported by Yang et al. (2023), ROUGE-L (Lin, 2004) was utilized for measuring the lexical overlap between the generated answers and Gold answers, and BERTScore (Zhang et al., 2020) for calculating the semantic overlap by measuring the distances of embeddings between both the answers. Additionally, we incorporate G-Eval with GPT-4 (Liu et al., 2023) as an LLM-based metric to measure the similarity, leveraging its capability to provide a highly adaptable and versatile evaluation with human-like accuracy, enhancing the comprehensiveness of our evaluation approach. Apart from these, we also perform human evaluation to ensure correctness since there is a lack of good metrics for long-form answers (Yang et al., 2023; Fan et al., 2019).

7 Results

7.1 Performance on the Adobe dataset

Table 3 presents the results for baseline methods, our proposed method, and the ablations of different components in our proposed method. We observe that our proposed method outperforms (Hit Rate: 74.7%; GPTEval: 0.58) the baselines without any query reformulation when using BM25 (Hit Rate: 41.2%; GPTEval: 0.41) and DPR retrievers (Hit Rate: 73.6%; GPTEval: 0.57). Among the baseline retrievers, the DPR-based method performs better than the BM25 retrieval. Therefore, we present other results using the DPR retriever.

Our method also performs better than the baseline using simple LLM prompting (i.e., without including any domain-specific knowledge) to reformulate the query (Hit Rate: 70.2%; GPTEval: 0.56). On qualitative examination of reformulated queries, we observe that while query reformulation using a general-purpose LLM can make queries more grammatical or well-formed, it still cannot augment the queries with domain-specific knowledge. On the other hand, our method of using LLM-generated triples can help link entities that have similar meanings within the domain. For instance, for the query "How to convert word docs to pdf," our method retrieves triples that aid in reformulating the query to "How to convert files to pdf," thereby assisting in retrieving the correct document.

7.2 Ablation Studies

Next, we perform a series of ablations on our framework to evaluate the functions of various components in our pipeline.

Ablation on relevance classifier: We directly provide all the triples retrieved by the triples retriever to the query expansion model without filtering out any retrieved triples via the relevance classifier. This approach performs worse than our proposed model (Table 3), since the retrieved triples may include numerous noisy ones, which are then integrated into the query through the query expan-

	Retriever	Query reformulation method	Retriever Hit Rate	Answer Similarity Score		
				ROUGE-L	BERTScore	G-Eval
Baselines	BM25	None	79.17%	0.336	0.822	0.494
	DPR	None	**92.70%**	0.470	0.879	**0.828**
Ours	DPR	**augmentation with domain-specific triples**	**92.70%**	**0.480**	**0.880**	0.776
Ablation	DPR	via general purpose LLM (w/o triple retriever and relevance classifier)	83.33%	0.406	0.859	0.692
	DPR	w/o triple relevance classifier	91.67%	0.447	0.873	0.755

Table 4: Performance of baseline methods, our proposed method, and the ablation experiments over the Photoshop test set. In all the above experiments, we use GPT3.5 as the LLM for triple generation and answer generation. The semantic similarity scores are computed between the gold and the generated answer. The reported hit rate is the top-3 hit rate.

sion model, resulting in a noisy query. Thus, our results highlight the necessity for the relevance classifier to provide only highly relevant triples to the query expansion model.

Ablation on Triple Retriever: In this experiment, we only consider gold documents annotated for a given query in the Adobe dataset and generate triples from them. The generated triples are filtered via the relevance classifier and then fed to the query enhancer. As shown in Table 3, this setting performs the best due to the use of highly relevant triples for query reformulation. Our results highlight the importance of building an efficient triple retriever to improve overall performance.

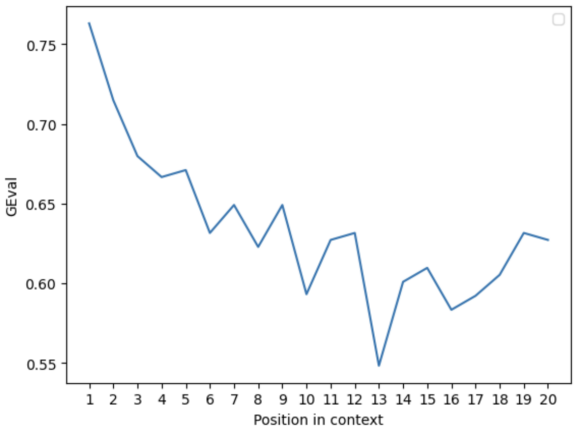

Figure 2: GEval score relative to the position the gold document is passed in as context over Acrobat test set.

7.3 Performance on the Photoshop dataset

Table 4 presents the results over the Photoshop test set. Once again, we observe that our proposed method is on par with the baseline, with a slight decrease in the GEval score. We attribute this to the fact that the Photoshop test set queries are already properly formulated; and reformulating these queries results in a slight deviation, hence the decrease in the G-Eval score. This illustrates that the performance of our proposed method is not dataset-specific. The pipeline is flexible enough to incorporate multiple different corpora - having created the respective knowledge bases.

7.4 Error analysis

While our method is able to link entities that are related in a particular domain, we also observe some cases of errors. For instance, the query 'Create PDFs of specific size by cutting a large PDF into a smaller file size.' was reformulated to 'How can I reduce the size of a PDF file?' as a result of retrieving the following triples :
1. (Reduce File Size command, reduces, size of PDF), 2. (PDF, reduce, size), 3. (PDF Optimizer, reduces, size of PDF files). These triples seemed to focus more on the keyword "size" and seemed to attribute the word "cutting" to reducing, while the intention of the original query was more towards splitting a PDF into multiple PDFs. This reformulation seemed to mis-interpret the original question, which resulted in incorrect retrieval. We attribute the source of such error to the noise originating from the retrieved triples that are irrelevant to the user query. Our analysis further highlights the importance of a high-precision triple retriever, as also suggested by our ablation study, which excludes the triple retriever and relies solely on triples from gold documents. Furthermore, another cause for concern was the similarity score metrics only slightly increasing when compared to their retrieval metric counterparts. We realized that hit rate may

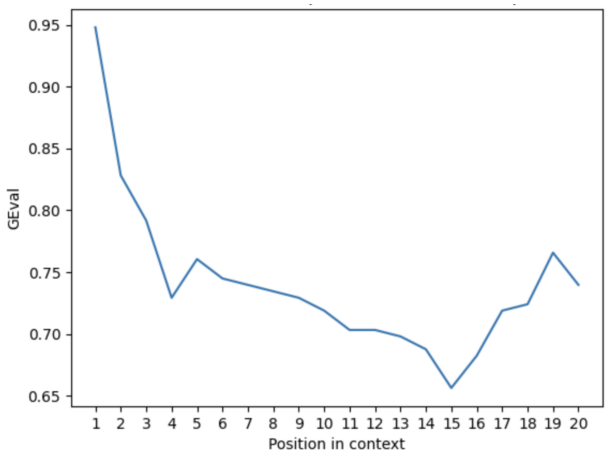

Figure 3: GEval score relative to the position the gold document is passed in as context over Photoshop test set.

not offer a holistic understanding of our retrievers performance. (Ravaut et al., 2024) suggests that LLM's are sensitive and exhibit different utilization of input tokens depending on their position within the context provided, which we also found to be the case as shown in Figure 2 3. It illustrates that the rank, or position, of the retrieved gold document impacts generation to a large extent, and is thus more indicative of proper retrieval than just hit rate alone. Following this, we decided to also consider NDCG as an evaluation metric, and the results are shown in Table 5.

Model	NDCG
Query Reformulation w/ Triples(Proposed Model)	0.447
Query Reformulation w/o Triples	0.453
No Query Reformulation (DPR Baseline)	0.507

Table 5: NDCG values for different models.

7.5 Performance using GPT-4o

Finally, we test our framework using a state-of-the-art model, GPT-4o, as well as different embeddings for the retriever to see how the performance would translate. In this experiment, we pass in a varying number of retrieved documents, and observe the corresponding NDCG and GEval values. Figures 4 and 5 present these results over the Acrobat test set. While the proposed model still outperforms the baseline in both of these metrics, vanilla query reformulation without triples seems to greatly outperform the aforementioned. Upon further qualitative analysis it was inferred that while GPT3.5 was more lax on introducing information from the triples into the query; GPT-4o tried to incorporate all of the information, which resulted in a far nosier query, thus leading to poorer retrieval.

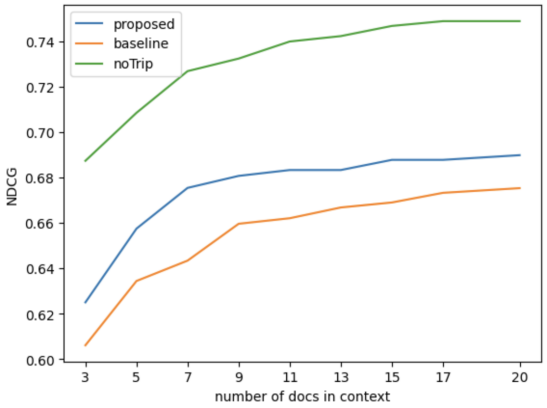

Figure 4: NDCG scores for our proposed model, the DPR baseline, and query reformulation without triples using an LLM (noTrip) over the Acrobat test set.

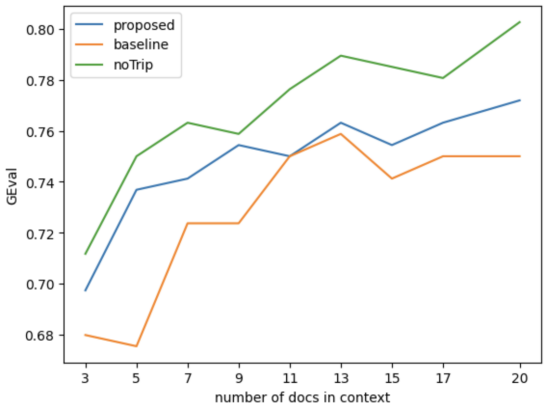

Figure 5: GEval scores for our proposed model, the DPR baseline, and query reformulation without triples using an LLM (noTrip) over the Acrobat test set.

Conversely, we observe the opposite results over the Photoshop dataset, as shown in Figures 6 and 7, which indicate that the baseline outperforms the proposed models in this setting. We attribute this to the fact that the Photoshop test set was synthesized using LLMs, which resulted in accurately formulated queries. In such scenarios, the proposed model under performs the baseline, as it tries to reformulate the query and deviates too much from the original meaning in doing so, thus resulting in poorer retrieval.

8 Conclusion

In this paper, we introduce two QA datasets focused on Adobe Acrobat and Photoshop products, that serve as benchmarks to evaluate an RAG-QA framework tailored for domain-specific procedural

long-form QA datasets. We also introduce a novel and detailed pipeline with components directed to improve the retrieval and generation metrics in such QA datasets. It equips LLMs with domain-specific knowledge through the use of Knowledge Base triples to bridge the gap between general RAG-QA methods and industry demands. Through various experiments we showcase the effectiveness and limitations of our proposed pipeline in standard metrics.

9 Limitations

This research presents multiple opportunities for improvement that can be explored further. To enhance the versatility and applicability of our method, it would be advantageous to explore its effectiveness across a broader range of larger industry-specific datasets beyond those provided by Adobe. Additionally, although our RAG-QA framework exhibits advancements over the baselines and offers valuable insights, the noise introduced by the LLM during query reformulation makes room for enhancement in the retrieval process. Moreover, expanding the application of our framework beyond text-based QA scenarios to include multi-modal capabilities opens up exciting new possibilities and broadens its potential impact. Lastly, it's important to note that assessing long-form QA remains an ongoing area of research, highlighting the necessity for a well-defined and automated metric to ensure accurate evaluation. Addressing these limitations is crucial for advancing the efficiency and scope of future research endeavors.

10 Acknowledgments

The authors of this paper would like to thank Hamed Zamani and more generally the entire UMass COMPSCI 696DS course staff for their insightful feedback, as well as the anonymous reviewers for their constructive comments.

References

Abhijit Anand, Venktesh V, Vinay Setty, and Avishek Anand. 2023. Context aware query rewriting for text rankers using llm. *Preprint*, arXiv:2308.16753.

Andong Chen, Feng Yao, Xinyan Zhao, Yating Zhang, Changlong Sun, Yun Liu, and Weixing Shen. 2023a. Equals: A real-world dataset for legal question answering via reading chinese laws. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 71–80.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2023b. Benchmarking large language models in retrieval-augmented generation. *Preprint*, arXiv:2309.01431.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Andrei Dulceanu, Thang Le Dinh, Walter Chang, Trung Bui, Doo Soon Kim, Manh Chien Vu, and Seokhwan Kim. 2018. Photoshopquia: A corpus of non-factoid questions and answers for why-question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Masoomali Fatehkia, Ji Kim Lucas, and Sanjay Chawla. 2024. T-rag: Lessons from the llm trenches. *Preprint*, arXiv:2402.07483.

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

Ivan Ilin. 2024. Advanced rag techniques: an illustrated overview - towards ai. *Medium*.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *Preprint*, arXiv:2305.03653.

Feihu Jiang, Chuan Qin, Kaichun Yao, Chuyu Fang, Fuzhen Zhuang, Hengshu Zhu, and Hui Xiong. 2024. Enhancing question answering for enterprise knowledge bases using large language models. *arXiv preprint arXiv:2404.08695*.

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.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *Preprint*, arXiv:2202.01110.

Jing Li, Shangping Zhong, and Kaizhi Chen. 2021. MLEC-QA: A Chinese Multi-Choice Biomedical Question Answering Dataset. In *Proceedings of the*

2021 Conference on Empirical Methods in Natural Language Processing, pages 8862–8874, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, Enhong Chen, Yi Luo, Peng Cheng, Haiying Deng, Zhonghao Wang, and Zijia Lu. 2024. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *Preprint*, arXiv:2401.17043.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *Preprint*, arXiv:2305.14283.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Preprint*, arXiv:2302.07842.

Zooey Nguyen, Anthony Annunziata, Vinh Luong, Sang Dinh, Quynh Le, Anh Hai Ha, Chanh Le, Hong An Phan, Shruti Raghavan, and Christopher Nguyen. 2024. Enhancing q&a with domain-specific fine-tuning and iterative reasoning: A comparative study. *arXiv preprint arXiv:2404.11792*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa : A large-scale multi-subject multi-choice dataset for medical domain question answering. *Preprint*, arXiv:2203.14371.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. *Preprint*, arXiv:1809.00732.

Zackary Rackauckas. 2024. Rag-fusion: a new take on retrieval-augmented generation. *arXiv preprint arXiv:2402.03367*.

Mathieu Ravaut, Aixin Sun, Nancy F. Chen, and Shafiq Joty. 2024. On context utilization in summarization with large language models. *Preprint*, arXiv:2310.10570.

Spurthi Setty, Katherine Jijo, Eden Chung, and Natan Vidra. 2024. Improving retrieval for rag based question answering models on financial documents. *arXiv preprint arXiv:2404.07221*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *Preprint*, arXiv:2305.15294.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Empower large language model to perform better on industrial domain-specific question answering. *Preprint*, arXiv:2305.11541.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. *Preprint*, arXiv:2209.10063.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *Preprint*, arXiv:2402.19473.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2019. Jec-qa: A legal-domain question answering dataset. *Preprint*, arXiv:1911.12011.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2024. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *Preprint*, arXiv:2305.13168.

A Code Release

We will release the code along with the data at `https://github.com/daksh-dangi/KaPQA` upon getting the required permission from Adobe.

B Prompt Structures

In this section we have listed the high-level structure of all the prompts used for the components of our pipeline.

Triples Generator LLM prompt

System:
You are an assistant for the Adobe Acrobat application that helps create tuples of the form (source, action, target) based on the information given to you.

User:
You are given a section from an adobe help document. Extrapolate the most relevant relationships you can from the context and generate tuples of the form (source, action, target). Ensure that the sources, actions, and targets are directly present in the provided context.\n
You must use only the provided data in variable 'Context' to identify relationships.\n
You must not use any other information from any other source or from previous knowledge beyond the provided 'Context'.\n
Example: If the document contained the phrase "To edit the image, first click on the triple line menu", one relevant tuple would be (triple line menu, edit, image). Here, the source of the action (editing the image) is the triple line menu. The direct effect of this action is on the image, hence that is the target. In a similar manner, create tuples for the provided context.
Context: <CONTEXT>
Constraints:\n
1. The created tuples must form the same format as the example provided.\n
2. The source and target in the tuple must only reference objects or menu items, and no actions\n
3. Ensure that the tuples generated are all related to the title of the section: <SECTION HEADER>\n
4. Only generate the most relevant tuples for the provided document with the given section header\n
5. You must make the contents of the tuples short and concise\n
6. Ensure the words "Adobe", "PDF", and "Acrobat" are not in the generated tuples.\n

Relevance Classifier LLM prompt

System:
You are an assistant for the Adobe Acrobat application. You are designed to filter out the information provided and classify what is most relevant to the given query.

User:
A user has provided you with the following query: <USER QUERY>\n
Use the data given in the variable 'Context' to classify which of the data elements are most relevant to the user's query.\n
Data in the 'Context' variable is of the form (source, action, target), where the first value contains the source of the action that is directed towards the target.
Example: One data element could be (triple line menu, edit, image). If the query was asking about how to edit an image, this element would be relevant. However, if the query was instead asking how to edit a video, it would be very irrelevant, and hence should not be included. The presence of a query word in the tuple does not make it relevant, look at the meaning as well when you are considering relevance. In a similar manner, filter out the context for the provided provided document.
Context: <CONTEXT>
Constraints:\n
1. Retrieve the data elements that are most relevant to the action that the user is trying to do in the query provided.
2. Ensure that the source and target of the data elements retrieved are similar to what is present in the query. \n
4. Give the most relevant data elements in a numbered list, and only provide the data elements themselves. No explanation.\n

Query Enhancer LLM prompt

System:
You are an assistant that is designed to only enhance user queries.

User:
You are given a query by the user and you must enhance the query by only using the data provided in variable 'Tuples'. The 'Tuples' variable is of the form (source, action, target).\n
Constraints:\n
1. Rephrase the query using the provided tuples, but do not change the meaning of the initial query.\n
2. Only use information from the tuples that are relevant to the query to reform the query.\n
3. Make the rewritten query one sentence at most.\n
4. Re-write the query in a manner similar to how a human might search for an answer on a help page. Keep the query short.\n
5. Only reformulate the given query, without answering it.\n
6. You must not use any other information from any other source or from previous knowledge beyond the provided 'Tuples'.\n
7. Ensure the words "Adobe" and "Acrobat" are not in the query.\n
8. Only answer with one reformulated query.
Example:\n
Given Query: 'how to remove letters from a text box'\n
Tuples: (text, delete key, remove),(page thumbnail, delete key, remove), (text, font item, edit), (text, font item, remove)\n
Reformulated Query: how to delete text
In a similar manner, reformulate the query below.
Given Query: <USER QUERY>
Tuples: <Tuples>
Reformulated Query:

DPR + General purpose LLM Prompt

System:
You are an assistant for the Adobe application that is designed to only enhance user queries. When asked about anything that does not relate to Adobe, only reply with 'Content not found'

User:
You are asked a question by the user and you must enhance the query. Do not answer the query, only change it's wording.\n
You must not use any other information from any other source or from previous knowledge beyond the query provided.\n
Understand what might be the cause of confusion, and rewrite the query by trying to model what the user could have been asking.\n
Ensure that the reformulated query is bound by the given constraints.\n
Query to be enhanced: "query" Constraints:\n
1. Make the rewritten query one sentence at most.\n
2. Make sure that the rewritten query does not have any excessive adjectives, and is short and to the point.\n
3. Only reformulate it, without answer it.\n
4. Only answer with the reformulated query.

C Graphs of Evaluation Metrics across models using GPT-4o

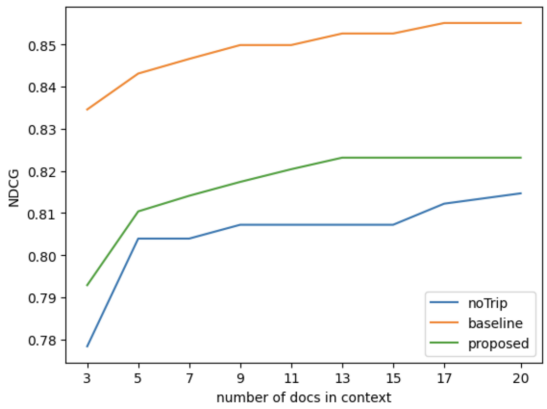

Figure 6: NDCG scores for our proposed model, the DPR baseline, and query reformulation without triples using an LLM (noTrip) over the Photoshop test set.

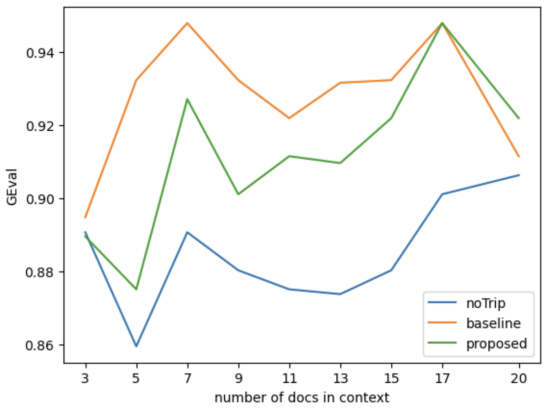

Figure 7: GEval scores for our proposed model, the DPR baseline, and query reformulation without triples using an LLM (noTrip) over the Photoshop test set.

D Examples of proposed method outperforming DPR + GPT3.5.

We have listed some examples in Table 6 where the proposed model using query reformulation outperforms the DPR + GPT3.5 baseline.

E Examples of proposed method under performing DPR + GPT3.5

We have listed some examples in Table 7 where the proposed model using query reformulation under performs the DPR + GPT3.5 baseline.

Original query	Reformulated Query	Generated Answer without QR	Generated Answer with QR	Gold Answer
test box resizing in acrobat	How to resize a text box or text in a PDF using the Edit panel in Acrobat?	To resize a box in Acrobat, follow these steps: 1. Open the PDF file in Acrobat.2. Choose the hamburger Menu (Windows), or the File menu (macOS) > Print. 3. Select Size, and choose one of the following options: - Scales small pages up and large pages down to fit the paper. [...] 4. If you don't see a Page Scaling options, click the arrow next to the Printer menu to expose more controls in the Print dialog box.	To resize a text box in Acrobat, follow these steps: 1. Select the text box you want to resize by clicking on it. 2. Drag one of the selection handles on the edge of the box to resize it. 3. To maintain alignment with other list items, press Shift as you drag. That's it! You have successfully resized the text box in Acrobat.	1. Choose 'Edit' from the mega verb bar. The PDF switched is to edit mode and the Edit PDF panel will be displayed. 2. Click on an edge of the text box to select it. 4. Drag any of the handles that appear on the corners or edges of the text box to resize it.
adding image on a pdf file	What are the different ways to add images to a PDF file?	To add an image to a PDF file, follow these steps: 1. Open the PDF file in Acrobat. 2. Choose Edit from the menu and select the image or object you want to edit. 3. If you want to edit multiple items together in the same file, select them all. 4. Under Adjust Objects in the Edit panel, select Edit using, and then choose an editor. 5. Make the desired changes in the editor. 6. Choose File > Save in the editor to save the changes. 7. The object is automatically updated and displayed in the PDF when you bring Acrobat to the foreground. If you want to copy and paste an image [...] file or a clipboard.	To add an image to a PDF file, follow these steps: 1. Open the PDF file in Acrobat. 2. Select Edit PDF̈from the right-hand pane. 3. Click on Image in the toolbar. 4. Select the image file you want to insert. 5. Choose where you want to insert the image or drag to resize it as you insert. 6. Once the image is added, you can move or resize it by hovering over the image and using the move or resize icons. 7. To adjust the image further, select Image from the left panel, [...] select File > Save.	1.Open the PDF in Acrobat and go to the 'Edit' mega verb. 2. Select 'Image' from the 'Add Content'. 3. In the dialog box that appears, choose the image file you want to insert. 4. Select the location where you want to insert the image or use the drag option to resize it as you insert. 5. A copy of the image file will appear on the page with the same resolution as the original file.
Are we allowed to crop images in a pdf file?	Can images be edited and added to a PDF file?	Yes, you can crop images in a PDF file using Adobe Acrobat. Here are the steps to do so: 1. From the top tools bar, select Edit. Alternatively, select the All tools menu > Edit a PDF.2. It displays the Edit toolset in the left panel. 3. From the Edit tools menu, select Crop Pages. 4. Drag a rectangle on the section that you want to crop. If necessary, drag the corner handles of the cropping rectangle until the page is the size you want.5. Double-click [...] 8. Select OK.	Yes, you can crop images in a PDF file using Adobe Acrobat. Here are the steps to do so: 1. Open the PDF in Acrobat and select Edit. 2. Choose the image you want to crop. 3. Under Adjust Objects in the Edit panel, select Crop Image. 4. Drag the cropping handles to adjust the size of the image. 5. Click outside the image to apply the crop. 6. Save the PDF.	1.Open the PDF in Acrobat and go to the Edit mega verb.2. select the image you want to crop.3. In the Adjust Objects section of the Edit panel, choose the Crop image tool.4. A selection handle will appear around the image. Drag the selection handles to adjust the cropping area.5. Once you have selected the desired area to crop, click outside the selection to apply the crop.The image will be cropped to the selected area.

Table 6: Examples of proposed method outperforming DPR + GPT3.5.

Original query	Reformulated Query	Generated Answer without QR	Generated Answer with QR	Gold Answer
How can I create PDFs from all non-pdfs stored in a directory? Is it possible to do it in one go.	Is it possible to convert all non-PDF files in a folder to PDFs with electronic features using Adobe Acrobat?	To create PDFs from all non-PDFs stored in a directory in one go, follow these steps:1. In Acrobat, select Create from the global bar.2. The Create a PDF tool opens. On the left rail, select Multiple files > Create multiple PDF files, and then select Next.3. Choose Add Files > Add Files or Add Folders, and then select the files or folder.4. Select OK. The Output options dialog box appears.5. In the Output Options dialog box, specify your target folder and filename preferences, and then select OK.[...]	To create PDFs from all non-PDFs stored in a directory, you can follow these steps:1. In Acrobat, select Create from the global toolbar.2. The Create a PDF tool opens. On the left rail, Single file is selected by default. Choose the Select a file button.3. In the Open dialog box, navigate to the directory where your non-PDF files are stored.4. Select all the non-PDF files you want to convert to PDF.5. Select Create to convert the files to PDF.6. Depending on the type [...] using this method.	1. You can create multiple PDFs from multiple native files, including files of different supported formats, in one operation. [...] 1. In Acrobat, select Create from the global bar.2. The Create a PDF tool opens. On the left rail, select Multiple files > Create multiple PDF files, and then select Next.3. Choose Add Files > Add Files or Add Folders, and then select the files or folder.4. Select OK. The Output options dialog box appears.5. Specify your target folder and filename preferences, and then select OK.
I need to create a link on a PDF.	How can I add a link to a PDF document?	To create a link on a PDF, follow these steps:1. In the source document, choose Edit > Link > Add or edit a link.2. Drag a rectangle to specify a location for the link.3. In the Create Link dialog box, set the link appearance.4. Select Go to a page view, and then select Next.5. In the target document, [...] create a link.	To create a link on a PDF, follow these steps:1. Choose the Select tool.2. Position the pointer over the linked area on the page until the pointer changes to the hand with a pointing finger. A plus sign (+) or a w appears within the hand if the link points to the web. Then click the link.3. In the Create Link [...] for a link to work correctly.	1. Choose Edit > Link > Add or edit a link. The pointer becomes a crosshair, and any existing links in the document, including invisible links, are temporarily visible.2. Drag a rectangle where you want to create a link. This is the area in which the link is active.3. In the Create Link dialog, choose the options you want for the link appearance.4. Select one of the following link actions:- Go to a page viewSelect Next [...] associated with the link.
How to remove password from the PDF	How to remove password protection from a PDF?	To remove password from a PDF, follow these steps:1. Open the PDF in Acrobat.2. Go to All tools > Protect a PDF > Remove security.3. If prompted, type the permissions password. If you don't know the permissions password, contact the author of the PDF.4. Acrobat will remove the security from the PDF and you will be able to access it without a password.	To remove password from a PDF, follow these steps:1. Go to All tools > Protect a PDF > Protect with password from the Protect a PDF toolset in the left pane.2. Choose Editing.3. Type and retype your password. Your password must be at least six characters long. The password strength is displayed next to your password [...] using the password.	You can remove security from an open PDF if you have the permissions to do so. If the PDF is secured with a server-based security policy, only the policy author or a server administrator can change it.1. Open the PDF, then select All tools > Protect a PDF > Set security properties.2. In the Document Properties window, select the Security tab and then select Change settings.3. Your options vary [...] Select OK again to confirm the action.

Table 7: Examples of proposed method under performing DPR + GPT3.5.