Federated Retrieval Augmented Generation for Multi-Product Question Answering

Parshin Shojaee[1*], **Sai Sree Harsha**[2], **Dan Luo**[2], **Akash Maharaj**[2], **Tong Yu**[2], **Yunyao Li**[2]
[1]Virginia Tech [2]Adobe
parshinshojaee@vt.edu, {ssree, dluo, maharaj, tyu, yunyaol}@adobe.com

Abstract

Recent advancements in Large Language Models and Retrieval-Augmented Generation have boosted interest in domain-specific question-answering for enterprise products. However, AI Assistants often face challenges in multi-product QA settings, requiring accurate responses across diverse domains. Existing multi-domain RAG-QA approaches either query all domains indiscriminately, increasing computational costs and LLM hallucinations, or rely on rigid resource selection, which can limit search results. We introduce MKP-QA , a novel multi-product knowledge-augmented QA framework with probabilistic federated search across domains and relevant knowledge. This method enhances multi-domain search quality by aggregating query-domain and query-passage probabilistic relevance. To address the lack of suitable benchmarks for multi-product QAs, we also present new datasets focused on three Adobe products: Adobe Experience Platform, Target, and Customer Journey Analytics. Our experiments show that MKP-QA significantly boosts multi-product RAG-QA performance in terms of both retrieval accuracy and response quality.

1 Introduction

The rapid advancement of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) has sparked significant interest in question-answering (QA) systems for domain-specific applications and products. This technology has significantly enhanced enterprise product support (Sharma et al., 2024), offering users more efficient and accurate ways to access information about complex product ecosystems with specific details, terminologies, usage procedures as well as related use cases. However, as the complexity of enterprise software suites grows, so does the challenge of providing accurate and comprehensive answers to user queries that may span multiple products or require cross-product knowledge.

In the context of enterprise product-related QA tasks, users often need to navigate multiple products and understand how they can be integrated to address specific use cases. This multi-product and cross-product nature of queries presents unique challenges for traditional RAG-QA approaches, particularly in context augmentation from diverse knowledge resources. These challenges are especially pronounced in industrial settings, where the accuracy of information retrieval and response generation directly impact customer satisfaction.

Current approaches to multi-domain search in RAG-QA systems typically fall into two main categories: (1) querying all product domains indiscriminately (Wu et al., 2024), or (2) employing resource selection techniques (Wang et al., 2024a,b). Both methods have significant drawbacks. The first approach, while comprehensive, can lead to increased computational costs and even potentially compromise answer quality due to the higher likelihood of LLM hallucination or inaccurate responses when presented with diverse and irrelevant concepts from different domains. The second approach, which attempts to narrow the search to specific domains, risks propagating selection errors that can limit the scope of the search and potentially miss crucial cross-product information, leading to incomplete or misleading answers in complex enterprise scenarios.

To address these challenges, we propose **MKP-QA** , a novel **M**ulti-domain **K**nowledge-augmented **P**roduct RAG-**QA** framework that optimizes multi-domain question answering. MKP-QA is designed to meet the specific needs of enterprise software ecosystems, where accurate cross-product information retrieval is essential. The core of MKP-QA employs a federated search mechanism (Shokouhi and Si, 2011) that intelligently navigates across multiple product

[*]Work done while interning at Adobe

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 387–397
January 19–24, 2025.

domains and their associated relevant corpus. This approach allows MKP-QA to search across a diverse range of enterprise products and their associated documentation without the need to centralize all information into a single, monolithic database – a significant advantage in large-scale enterprise deployments.

The complex nature of multi-product QA in enterprise settings necessitates a more nuanced approach than simple federated search. Cross-product queries often require information from multiple domains with overlapping terminologies, including less obvious ones. To address these practical needs, we enhance MKP-QA 's federated search with a probabilistic gating mechanism, serving three crucial functions: (i) *Exploration-Exploitation Balance*, enabling both exploitation of known relevant domains and exploration of less obvious ones, crucial for cross-product queries; (ii) *Error Mitigation*, using likelihoods in domain selection to safeguard against misclassification and missed information in domain router; and (iii) *Adaptive Query Processing*, allowing flexible and context-aware searching. By aggregating query-domain and query-document relevance scores through this mechanism, MKP-QA enhances multi-domain document retrieval for RAG-QA systems. This adaptation of federated learning techniques (Ashman et al., 2022; Huang et al., 2022) to our specific challenges enables more accurate, cross-domain product knowledge integration, particularly valuable in complex enterprise software ecosystems.

To address the lack of suitable multi-product QA benchmarks, we also introduce new benchmark datasets focused on three Adobe products: Adobe Experience Platform (AEP), Adobe Target, and Adobe Customer Journey Analytics (CJA). These datasets, which we intend to release publicly, consist of user queries and corresponding documents from Adobe product documentation. They serve as valuable resources for evaluating domain-specific and cross-domain RAG-QA systems across product domains. The datasets will be made available pending Adobe's approval.

Our experimental findings demonstrate significant improvements in the accuracy of multi-product question answering. By introducing new benchmark datasets and proposing an innovative framework, we seek to push the boundaries of AI Assistants in product-related QA. Importantly, MKP-QA achieves this without requiring separate domain-specific LLM fine-tuning or the training of adaptive modules across various product domains.

2 Related Work

2.1 Domain-specific Question-Answering

Domain-specific QA has seen significant advancements across various fields, addressing the unique challenges posed by specialized knowledge and terminology. Research efforts have focused on developing tailored methods and datasets for domains such as biomedical (Gu et al., 2021), physics (Chen et al., 2023), finance (Wu et al., 2023), and legal (Cui et al., 2024). These works have contributed to improving QA accuracy and relevance within their respective fields. In the context of product-related QA, which is most relevant to our work, efforts have been more limited. Notable among these is the dataset in (Liu et al., 2023) which focuses on Microsoft product queries. However, this dataset primarily consists of yes/no questions, with only a small portion requiring more complex generative text answers. Our work extends this line of research by addressing multi-product QA in enterprise software ecosystems. We focus on more complex, cross-domain queries that often require integrating knowledge from multiple products - a scenario common in enterprise settings but underexplored in current literature.

2.2 Retrieval Augmented Generation

Retrieval augmented generation (RAG) has recently emerged as a powerful approach for enhancing the performance of LLMs in knowledge-base QA tasks. RAG combines the strengths of retrieval-based and generation-based methods to produce more accurate and faithful responses. (Lewis et al., 2020) introduced the foundational RAG model, which retrieves relevant documents and conditions its output on both the retrieved information and the input query. Subsequent works have further improved it with (Guu et al., 2020) developing REALM for joint training of retriever and generator, and (Karpukhin et al., 2020) introducing dense passage retrieval for improved efficiency. Recent research has explored RAG in domain-specific contexts. (Head et al., 2021) adapted RAG for scientific literature, while (Khattab et al., 2022) investigated its application in customer support settings. Our work extends this line of research by introducing a novel multi-domain RAG framework that addresses the specific challenges of enterprise systems, where queries often span multiple products and require integration of diverse knowledge.

2.3 Multi-domain Document Retrieval

Multi-domain document retrieval presents unique challenges, particularly in enterprise product settings where information is often distributed across diverse and overlapping knowledge sources. Research in this area has focused on developing methods to accurately retrieve relevant information from multiple sources. Federated search approaches (Shokouhi and Si, 2011) enable querying multiple distributed indexes simultaneously, while domain adaptation techniques (Shi et al., 2020) handle transfer across diverse search domains. Recent work leveraging LLMs for retrieval resource selection (Wang et al., 2024b) has also shown strong zero-shot performance. Our work extends these efforts by introducing a stochastic gating mechanism combined with federated search, tailored for RAG-QA pipelines in complex environments with cross-product queries.

3 Methodology

Our MKP-QA framework, shown in Fig. 1, integrates components for domain relevance, exploration-exploitation, retrieval, and multi-domain aggregation, detailed below.

3.1 Query-Domain Router

To effectively estimate the query-domain relevance scores. we leverage a query-domain router $\mathcal{F} : Q \to [0,1]^m$, mapping from the space of queries Q to the m product domains as a multi-label classification task. We use a Transformer model (Vaswani, 2017), specifically a variant of BERT, fine-tuned for our multi-domain classification task: $\mathcal{F}(q) = \sigma(W + \text{BERT}(q) + b)$, where $\text{BERT}(q)$ is the contextualized representations of query q at `[cls]` token; W and b are learnable parameters; and σ is the sigmoid activation function. As this is a multi-label classification task, we employ a binary cross-entropy loss for each domain, summed over all domains. At inference, we estimate the query-domain relevance likelihood with the trained domain router: $\mathcal{F}(q) = [p_1, p_2, \ldots, p_m]$.

3.2 Stochastic Gating

To address the challenge of balancing exploitation of high-confidence domains with exploration of potentially relevant but less certain domains, we introduce a stochastic gating mechanism with adaptive threshold control. This approach allows for dynamic adjustment of the search space based on the Query-Domain Router's confidence and the inherent uncertainty in multi-domain RAG-QA. We define an adaptive threshold $\tau(q)$ for query q based on the entropy of the domain probability distribution p: $\tau(q) = \tau_0(1 - \frac{-\sum_j^m p_j \log(p_j)}{\log(m)})$, where τ_0 is the base threshold hyperparameter; and $[p_1, \ldots, p_m]$ is the vector of domain probabilities output by the router. We utilizie stochastic gating function $\mathcal{G} : M \times Q \to \{0, 1\}$, with M as the domain space and Q as the query space, to facilitate domain selection and introduce exploration. This function is defined as $\mathcal{G}(q, j) = \text{Bernouli}(\min(1, p_j/\tau(q)))$, where $\mathcal{G}(q, j)$ is the domain j-th selection for query q based on the Bernouli sampling.

3.3 Query-Document Retriever

To facilitate efficient and effective retrieval of relevant documents across multiple product domains, we employ a bi-encoder architecture for our Query-Document Retriever. This model generates dense vector representations for both queries and documents, enabling rapid similarity computations in the embedding space. Our retriever embedding model E is based on the Sentence-BERT (Reimers, 2019) with shared weights for query and document encodings, generating dense vector representations $E_\theta(q)$ and $E_\theta(d)$ for query q and document d.

We fine-tune the retriever model on our multi-domain dataset using a contrastive learning approach with a symmetric supervised variant of the InfoNCE loss (Oord et al., 2018), incorporating supervised relevance labels and symmetry, which we find particularly effective for query-document retrieval tasks in multi-domain settings. The retriever loss function is $\mathcal{L}_r = -(\mathcal{L}_{q2d} + \mathcal{L}_{d2q})/2$, where $\mathcal{L}_{q2d}$ and $\mathcal{L}_{d2q}$ represent the query-to-document and document-to-query directional losses. The $\mathcal{L}_{q2d}$ is computed as follows and the $\mathcal{L}_{d2q}$ can be obtained similarly.

$$\mathcal{L}_{q2d} = \sum_q \sum_{d_+ \in \mathcal{D}_+} \frac{\exp(s(q, d_+)/\tau)}{\exp(s(q, d_+)/\tau) + \sum_{d \in \mathcal{D}_-} \exp(s(q, d)/\tau)}$$

where $\mathcal{D}_+$ and $\mathcal{D}_-$ are the set of annotated positive and negative document pairs for the given query q within batch; $s(q, d)$ is the dot-product similarity score between query q and document d embedding: $s(q, d) = E(q) \cdot E^T(d)$; and τ is a temperature hyperparameter. At inference, we compute the embeddings of all documents in the corpus offline and save in a vector database. For a given query, we compute its embedding and retrieve the top-k documents using similarity score search.

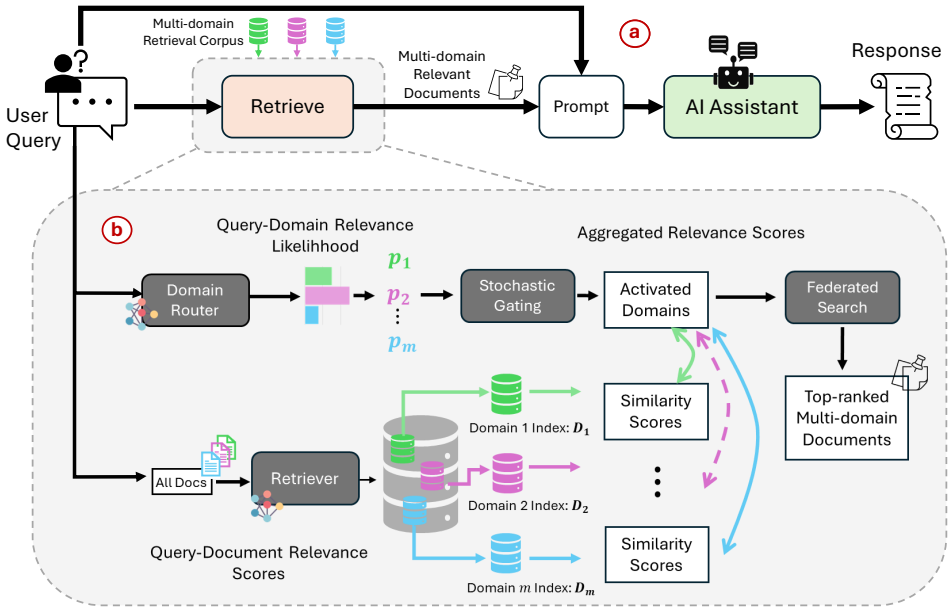

Figure 1: **Overview of the MKP-QA framework.** ⓐ The main RAG-QA pipeline: retrieval of multi-domain documents, prompt augmentation, and response generation. ⓑ Detailed view of the multi-domain knowledge augmentation: A domain router estimates query-domain relevance while a retriever finds relevant documents across product domains. A stochastic gating mechanism determines active domains, for which query-domain and query-document relevance scores are aggregated into a unified ranking of multi-domain documents. These top-ranked documents then augment the prompt, enabling effective cross-domain product QA.

3.4 Federated Search

For a given query q, we define the set of active domains $\mathcal{A}(q) = \{j \in M | \mathcal{G}(j,q) = 1\}$. For each active domain $j \in \mathcal{A}(q)$, we retrieve the top-k documents $D_j = \{d_j^1, \ldots, d_j^k\}$ and their respective query-document relevance scores $S_j = \{s_j^1, \ldots, s_j^k\}$ obtained from the bi-encoder retriever detailed above: $s_j^i = s(q, d_j^i) = E(q) \cdot E^T(d_j^i)$. Next, we aggregate these to a unified domain-aware retrieval scoring $U(j, q, d_j^i) = \mathcal{F}(q)[j] \cdot s(q, d_j^i) = p_j \cdot s_j^i$, where $U(\cdot)$ is the unified multi-domain ranking score function for query q, domain j and retrieved document d_j^i at position i in this domain. The final multi-domain ranked set of documents with federated search D^* is obtained by selecting the top-k documents across all the active domains: $D^* = \arg\max_k \{U(j, q, d_j^i) | j \in \mathcal{A}(q), d_j^i \in D_j\}$. These top-ranked documents from multiple domains are then augmented to the prompt and fed to LLM for the product QA.

4 Dataset Creation and Statistics

4.1 Data Sources

The corpus is derived from the publicly available Adobe Experience League (ExL) documentation[1], focusing on three key products: Experience Platform (AEP), Target, and Customer Journey Analytics (CJA). These web-pages provide comprehensive information on product concepts, capabilities, troubleshooting guides, and usage instructions.

4.2 Data Pre-processing

The data preparation process involves several steps: ① *Web Crawling:* We employ a custom crawling script to extract content from the ExL web-pages. This script navigates through the documentation, capturing textual information while omitting images and converting clickable and in-section links to plain text for consistency. ② *Initial Segmentation:* The extracted content is initially segmented based on HTML header tags. This approach creates distinct sections that typically correspond to specific topics or tasks within each documentation. ③ *Document Chunking:* To optimize the corpus for efficient retrieval and context preservation, we implement the following chunking strategy. Each web-page is divided at every header level, creating initial chunks that align with the document's logical structure. If any section exceeds a pre-defined token limit (512 tokens), we utilize LangChain's hierarchical splitting approach based on a specified character list. This method prioritizes maintaining

[1] https://experienceleague.adobe.com/en/docs

the integrity of paragraphs, sentences, and words, ensuring that semantically related content remains together as much as possible.

4.3 Data Creation

Our dataset comprises query-document pairs from three Adobe product domains (AEP, Target, and CJA). We employed two complementary approaches for data creation: (i) *Subject Matter Expert (SME) Dataset:* Product experts manually created query-document pairs based on respective ExL web-pages for each domain. They wrote queries for documents extracted from web-pages and annotated the relevance of each pair based on their product expertise. (ii) *Synthetic LLM-Assisted (SLA) Dataset:* To ensure comprehensive coverage, we leveraged GPT-4 to generate queries for chunked documents extracted from ExL web-pages. Product experts from each domain subsequently reviewed these query-document pairs to guarantee accuracy and relevance. For cross-domain data creation, we followed a similar process where product experts first identified ExL web-pages with overlapping documentation across products; GPT-4 then generated queries for these cross-domain web-pages; and product experts reviewed the queries for relevance to the cross-domain documentation content. This approach ensured that positive document pairs per query were designed to span different domains, enhancing the dataset's utility for multi-domain product RAG-QA research.

To enhance the dataset with both positive and challenging negative examples, we employed a systematic approach for document pairing. For each question in the dataset, we utilized two strategies: (1) pairing the question with other document chunks from the same web-page as the golden document, and (2) when insufficient negative pairs were available from the original page, sampling document chunks from URLs closely related to the web-page containing the golden document. This method ensures a diverse and representative set of negative examples. Finally, we leveraged GPT-4 to annotate the relevance of each query-document pair. Using the prompt detailed in Appendix (Figure 6), GPT-4 assigns binary labels (Yes/No) to the relevance of all pairs.

4.4 Data Analysis and Statistics

Our dataset encompasses questions, documents, corresponding source web-page URLs and titles, and annotations across the Adobe AEP, Target, and CJA domains. The questions fall into two main categories: (1) *"What-is"* or *"Where-is"* questions about product concepts (e.g., "What is a union schema?", "What is an audience?"); and *"How-to"* questions about usage instructions (e.g., steps for "adding services to a datastream" or "looking up a sandbox"). Table 1 provides key data statistics for uni-domain and cross-domain datasets, including the count of question-document pairs, average lengths of questions and documents, and the ratio of positive pairs in the datasets.

Data Type	Metric	Uni-Domain		
		AEP	CJA	Target
SME	# of query-doc pairs	2,970	1,035	521
	Avg. length of queries	9.31	9.75	9.12
	Avg. length of docs	87.59	207.43	101.15
	% of positive pairs	8.95%	11.27%	10.23%
SLA	# of query-doc pairs	28,860	27,820	29,610
	Avg. length of queries	10.75	11.80	11.69
	Avg. length of docs	143.78	146.89	107.15
	% of positive pairs	17.53%	18.28%	20.26%

Data Type	Metric	Cross-Domain		
		AEP + CJA	AEP + Target	CJA + Target
SLA	# of query-doc pairs	880	1,370	480
	Avg. length of queries	14.70	14.92	13.68
	Avg. length of docs	141.15	97.72	95.49
	% of positive pairs	19.21%	19.56%	18.37%

Table 1: Statistics for the Adobe multi-product uni-domain (**top**), and cross-domain (**bottom**) RAG datasets

5 Experiments

Our experimental study aims to evaluate the effectiveness of MKP-QA in comparison with various baselines for multi-domain RAG-QA on Adobe datasets. We conducted a series of experiments on both uni-domain and cross-domain datasets. Throughout our experiments, we utilized `GPT-3.5-turbo-1106` and `GPT-4-0314` models from Azure OpenAI.

5.1 Baselines

Unified Index and Search (UIS): This baseline uses a single multi-domain index with a retriever fine-tuned on all three product domains. Search is performed across the entire index without considering domain relevance to the query.

Router Filter and Search (RFS): A domain router selects the most likely domain for each query, limiting the search to documents tagged for that domain within the unified index.

LLM Filter and Search (LFS): Using the ReSLLM method (Wang et al., 2024b), this baseline leverages GPT-4 in a zero-shot manner for domain selection, then searches within that domain's subset of the unified index (see Appendix Figure 9 for prompt details).

In all baselines, vector similarity is used to retrieve the top-5 most relevant documents by comparing the query's vector to document vectors. These documents are then augmented into the assistant's prompt for response generation.

5.2 Evaluation Methods

To comprehensively assess the performance of our multi-domain RAG-QA pipeline, we employ a variety of evaluation metrics targeting both retrieval accuracy and response quality: (*i*) *Retrieval Accuracy:* For evaluating retrieval performance across multiple domains, we use the Acc@Top1 metric. This metric represents the percentage of queries for which the golden (most relevant) multi-domain documents are correctly retrieved as the top-ranked candidate. Our focus on the first document is motivated by recent RAG studies (Liu et al., 2024; Xu et al., 2024) showing that the top-ranked document, when added to the prompt, most significantly influences the LLM's response. (*ii*) *Response Quality:* To assess the quality of generated answers for product QA, we employ Relevancy and Faithfulness analysis. In the former, we incorporate GPT-4, following the prompting strategy in (Zheng et al., 2024), to evaluate the relevancy and helpfulness of the generated responses to queries. Given the domain-specific nature of product QA, we also utilize the RAGAS[2] framework (Es et al., 2023) to assess the faithfulness of generated responses. In this metric, we decompose each response into individual statements and cross-check them against the ground truth documentation for each query with the help of GPT-4. The faithfulness score is computed as the percentage of statements that GPT-4 recognizes can be directly inferred from the provided context. Detailed prompts for these metrics are available in the Appendix.

5.3 Results and Analysis

Retrieval Performance Fig. 2 illustrates the retrieval accuracy (Acc@Top1) of our method and baselines across uni-domain and cross-domain datasets. Our approach consistently outperforms

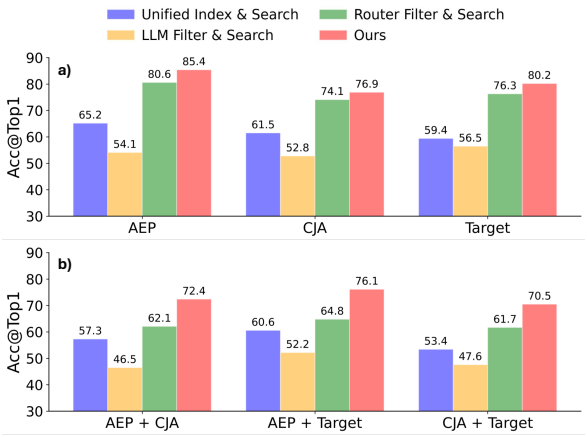

Figure 2: Performance comparison of retrieval accuracy (Top-1) across methods on **(a)** uni-domain, and **(b)** cross-domain datasets.

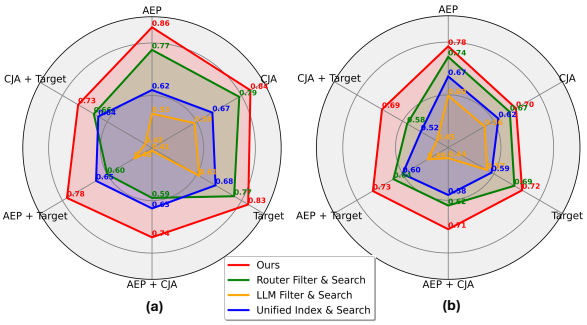

Figure 3: Performance comparison of response quality across methods on different datasets for LLM-based **(a)** Relevancy, and **(b)** Faithfulness metrics.

all baselines, with the performance gap widening in cross-domain settings. The RFS baseline shows the strongest performance among alternatives, particularly in uni-domain scenarios. This can be attributed to the simpler query-domain relevance in uni-domain settings. Conversely, the LFS baseline underperforms in retrieval accuracy, likely due to challenges general-purpose LLMs (e.g. GPT-4) face in domain selection and learning query-domain relevance for our specific product domains.

Response Performance Figure 3 presents the assistant's response quality evaluations across all methods and datasets, focusing on Relevancy and Faithfulness metrics. Our method outperforms baselines on both metrics, with its advantage becoming more pronounced in cross-domain settings. The RFS baseline consistently ranks second in response quality, while the LFS baseline performs poorest. Interestingly, the UIS baseline occasionally outperforms RFS on Relevancy in cross-domain datasets, but underperforms on Faithful-

[2] https://docs.ragas.io

ness. This can be attributed to the fact that faithfulness correlates more strongly with retrieval accuracy, while relevancy assessment is usually influenced more by other factors like response length and context diversity which is prevalent in the UIS baseline setting.

6 Path to Deployment

Deploying MKP-QA into production requires careful planning, extensive testing, and significant efforts across multiple dimensions. We highlight the following key aspects that are essential for ensuring a successful and robust deployment and discuss our plans:

Knowledge Precision Accurate multi-domain federated search is crucial for retrieving relevant content across product domains, as inaccurate retrievals in RAG can lead to irrelevant or misleading AI assistant responses. Our goal is to achieve a retrieval accuracy (Acc@Top1) of 90% or higher across both uni-domain and cross-domain queries. To improve this, we plan to implement regular retraining cycles for both the domain router and retriever models to adapt to evolving product documentation and user query patterns.

Latency Given the multi-step nature of our framework, managing response time is critical for user experience. Our target is to keep the end-to-end response time under 10 seconds for 95% of queries. To do so, we plan to implement parallel processing for domain routing and document retrieval; utilize caching for frequently accessed documents; and explore quantization techniques for the retriever model to reduce inference time without significant accuracy loss.

User Study A comprehensive user study is essential to evaluate performance in improving actual product QA experience. Once the system meets our accuracy and latency criteria, we plan to conduct A/B testing with a representative sample of users across different Adobe products, then, gather and analyze explicit and implicit user feedback through user surveys and interaction metrics (e.g., follow-up questions, task completion rates).

Continuous Monitoring and Iteration Following the deployment of this work, continuous performance monitoring and iterative improvements are essential. We intend to implement monitoring dashboards tracking key performance metrics across different product domains; and establish a feedback loop where user interactions and support team insights are regularly incorporated into model retraining and system refinement.

7 Conclusion

In this paper, we introduced MKP-QA , a novel multi-domain knowledge-augmented question-answering framework for complex enterprise software ecosystems. Leveraging federated search with stochastic gating, MKP-QA outperforms baselines in retrieval accuracy and response quality across uni-domain and cross-domain settings for Adobe's Experience Platform (AEP), Target, and Customer Journey Analytics (CJA) applications. We also introduced new datasets for multi-product QA, addressing the lack of suitable benchmarks in this domain. Our findings highlight the importance of multi-domain knowledge integration and specialized approaches for domain-specific nuances in enterprise product QA, while also revealing limitations of LLM-based domain selection techniques. Looking ahead, there are several avenues for future work and deployment optimization. These include implementing retraining cycles for router and retriever, exploring advanced caching and quantization techniques to reduce latency, and conducting comprehensive user studies to ensure alignment with real-world usage patterns.

References

Matthew Ashman, Thang D Bui, Cuong V Nguyen, Stratis Markou, Adrian Weller, Siddharth Swaroop, and Richard E Turner. 2022. Partitioned variational inference: A framework for probabilistic federated learning. *arXiv preprint arXiv:2202.12275*.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901, Singapore. Association for Computational Linguistics.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. 'chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. *arXiv preprint arXiv:2306.16092*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S Weld, and Marti A Hearst. 2021. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Tiansheng Huang, Weiwei Lin, Li Shen, Keqin Li, and Albert Y Zomaya. 2022. Stochastic client selection for federated learning with volatile clients. *IEEE Internet of Things Journal*, 9(20):20055–20070.

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)*, Online. Association for Computational Linguistics.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

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.

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

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Sanat Sharma, David Seunghyun Yoon, Franck Dernoncourt, Dewang Sultania, Karishma Bagga, Mengjiao Zhang, Trung Bui, and Varun Kotte. 2024. Retrieval augmented generation for domain-specific question answering. *arXiv preprint arXiv:2404.14760*.

Peng Shi, He Bai, and Jimmy Lin. 2020. Cross-lingual training of neural models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2768–2773, Online. Association for Computational Linguistics.

Milad Shokouhi and Luo Si. 2011. Federated search. *Found. Trends Inf. Retr.*, 5(1):1–102.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. 2024a. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 763–773.

Shuai Wang, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2024b. Resllm: Large language models are strong resource selectors for federated search. *arXiv preprint arXiv:2401.17645*.

Ridong Wu, Shuhong Chen, Xiangbiao Su, Yuankai Zhu, Yifei Liao, and Jianming Wu. 2024. A multi-source retrieval question answering framework based on rag. *arXiv preprint arXiv:2405.19207*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Appendix

A Query-Document Pair Examples

The following AEP query-document examples highlight the necessity for product domain-specific knowledge in providing accurate and detailed responses to the user's questions:

Question

How to create a segment?

Document

In the Experience Platform UI, click Segments in the left navigation, and then click Create segment. The Segment Builder appears. From the left Fields column and under the Attributes tab, click the folder named XDM Individual Profile and then click the folder with the namespace of your organization. The folder named Customer AI contains the results of prediction runs and are named after the instance the scores belong to. Click an instance folder to access its results of the desired instance. Located in the center of Segment Builder, drag and drop the Score attribute onto the rule builder canvas to define a rule. Under the right-hand Segment properties column, provide a name for the segment. Above the left-hand Fields column, click the gear icon and select a Merge policy from the drop-down. Finally, click Save to create the segment.

Question

How to delete existing fields from a schema?

Document

After you have added a field group to a schema in Experience Platform, you can remove any fields that you do not need. To remove a single field, select the field in the canvas and then select Remove in the right rail. If there are multiple fields you wish to remove, you can manage the field group as a whole. Select a field belonging to the group in the canvas, then select Manage related fields in the right rail. A dialog appears showing the structure of the field group in question. From here you can use the provided checkboxes to select or deselect the fields that you require. When you are satisfied, select Confirm to remove the selected fields.

Question

What are known and anonymous identities?

Document

A known identity in Experience Platform refers to an identity value that can be used on its own or with other information to identify, contact, or locate an individual person. Examples of known identities may include email addresses, phone numbers, and CRM IDs. An anonymous identity in Experience Platform refers to an identity value that cannot be used on its own or with other information to identify, contact, or locate an individual person (such as a cookie ID).

Figure 4: Examples of user query and relevant product documentation in the dataset.

B LLM Prompts

This section provides an overview of the high-level structures for prompts utilized in our study

Query Generation LLM Prompt

````
You are a smart assistant designed to act as a user
coming up with questions about a product.

Given a piece of document, you must come up with a
question that can be used to mimic user's behavior.

When coming up with this question, you must respond in
the following format:
    ```
    {{
        "question": "$YOUR_QUESTION_HERE",
    }}
    ```
Everything between the ``` must be valid json.

Please come up with a question, in the specified JSON
format, for the following document:
----------------
{{DOCUMENT}}
````

Figure 5: LLM Prompt for Query Generation: Simulating user behavior for document-based question synthesis

Pair Annotation LLM Prompt

````
You are an assistant tasked with determining if a given
document contains information relevant to answering a
user's question about a product

User Question: {{QUERY}}

Document Content: {{DOCUMENT}}

The last sentence in your response should include the
Final Answer, by choosing one from: 'Yes' or 'No'.
Let's think step by step.

You must respond in the following format:
    ```
    {{
     "reasoning": "$YOUR_REASONING_HERE",
     "final_answer": "$YOUR_ANSWER_HERE",
    }}
    ```
Everything between the ``` must be valid json.
````

Figure 6: LLM Prompt for Query-Document Relevance Annotation: Binary labeling with explanatory reasoning for the relevance annotation.

Relevancy Judge LLM Prompt

```
[Instruction]
Please act as an impartial judge and evaluate the
quality of the response provided by an AI assistant to
the user question displayed below. Your evaluation
should consider factors such as the helpfulness,
relevance, accuracy, depth, and level of detail of the
response. Begin your evaluation by providing a short
explanation. Be as objective as possible. After
providing your explanation, you must rate the response
on a scale of 1 to 10 by strictly following this
format: "[[rating]]",
for example: "Rating: [[5]]".

[Question]
{{QUERY}}

[The Start of Assistant's Answer]
{{RESPONSE}}
[The End of Assistant's Answer]
```

Figure 7: LLM Prompt for Query-Response Relevance Judge: Rating from 1 to 10 for the response to a given query, following the prompt in (Zheng et al., 2024).

Faithfulness Judge LLM Prompt

```
Your task is to judge the faithfulness of a series of
statements based on a given context. For each statement
you must return verdict as 1 if the statement can be
directly inferred based on the context or 0 if the
statement can not be directly inferred based on the
context. Let's think step by step.

Context: {{DOCUMENTS}}

Statements: {{RESPONSE STATEMENTS}}

You must respond in the following format:
    ```
    {{
    "statement_i": "$YOUR_STATEMENT_i_HERE",
    "reason_i": "$YOUR_REASONING_HERE",
    "verdict_i": "$YOUR_VERDICT_HERE",
     }}
    ```
Everything between the ``` must be valid json.
```

Figure 8: LLM Prompt for Response Faithfulness Judge: Binary rating for inferring each response statement from query's golden documents, following the prompt in (Es et al., 2023).

LFS Resource Selection LLM Prompt

```
[System]
Federated search retrieves information from a variety
of sources via a search application built on top of one
or more search domains. A user makes a single query
request. The federated search then selects only the
search domains that the query should be sent to from a
list of domains, and aggregates the result for
presentation of high-quality result to the user. The
task is called resource selection.

The following is a real user query:
Query: {{QUERY}}

The following are some context from this search domain,
providing an overview of the domain:
Domain Context: {{DOMAIN CONTEXT}}

Now, please reply only 'Yes' or 'No' to indicate if the
query should be sent to the search domain.
```

Figure 9: LLM Prompt for Resource Selection step in the LFS baseline, following the prompt in (Wang et al., 2024b).