Domain-Specific Data Generation Framework for RAG Adaptation

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) combines the language understanding and reasoning power of large language models (LLMs) with external retrieval to enable domain-grounded responses. Effectively adapting RAG systems to domain-specific settings requires specialized, context-rich training data beyond general-purpose question-answering. Here, we propose RAGen, a scalable and modular framework for generating domain-grounded question–answer–context (QAC) triples tailored to diverse RAG adaptation approaches. RAGen produces these QAC triples by identifying key concepts in documents, generating diverse questions guided by Bloom's Taxonomy-inspired principles, and pairing them with precise answers extracted from relevant contexts. RAGen supports multiple RAG adaptation strategies, including the optimization of key components such as the LLM, retriever, and embedding model, etc. Its modular pipeline features semantic chunking, hierarchical concept extraction, and multi-chunk retrieval, along with the introduction of curated distractor contexts to promote robust reasoning. Designed for scalability, RAGen efficiently handles large and evolving document corpora without redundant processing, making it especially suitable for dynamic evolving domains such as scientific research and enterprise knowledge bases.

1 Introduction

With the growing adoption of large language models (LLMs) in enterprise and organizational settings, there is increasing demand for integrating these models into domain-specific workflows (Chiarello et al., 2024; Qian et al., 2024). However, concerns over data privacy, regulatory compliance, and the high cost of commercial API usage often prevent organizations from deploying proprietary, cloud-hosted LLMs. As a result, many turn to open-source, locally deployed small- and medium-scale LLMs for internal use.

Despite their accessibility, smaller models inherently suffer from limited language understanding and reasoning capabilities compared to frontier LLMs (Chen et al., 2024c; Mallen et al., 2022). This performance gap motivates the use of Retrieval-Augmented Generation (RAG) (Lewis et al., 2020), which supplements an LLM with a retriever to provide external, context-specific information. RAG offers a practical and modular solution for grounding LLM outputs in proprietary knowledge bases without requiring massive model sizes.

However, simply applying off-the-shelf RAG pipelines to new domains often yields suboptimal results (Barnett et al., 2024). This is because general-purpose RAG systems are not tailored to domain-specific data distributions or terminology. RAG adaptation, therefore, becomes essential. We define RAG adaptation as the process of refining and optimizing individual components of the RAG pipeline, including the LLM, retriever, or embedding model, to better align with domain-specific requirements and improve end-to-end performance (Siriwardhana et al., 2023; Liu et al., 2025).

Recent methods such as RAFT (Zhang et al., 2024c) introduce distractor-aware fine-tuning to improve the robustness of LLMs in noisy RAG contexts. Meanwhile, inference-time techniques like Self-RAG and Open-RAG (Asai et al., 2023; Islam et al., 2024) aim to teach LLMs when and how to retrieve.

While these approaches provide valuable insights, they are often narrow in scope, each targeting only one component of the RAG pipeline. However, RAG is a multi-stage architecture, and optimizing a single module (e.g., just the retriever or just the LLM) is insufficient for robust, end-to-end performance. Many existing methods lack the flexibility to support multiple adaptation strategies and often rely on fixed, tightly coupled training procedures that assume the availability of high-quality,

domain-specific data for individual components which limits their generalizability across domains and architectures.

To address these limitations, we propose RAGen, a scalable and modular framework for generating high-quality, domain-specific training data to support diverse RAG adaptation strategies. RAGen constructs question–answer–context (QAC) triples by first identifying key document-level concepts, retrieving multi-chunk evidence, and generating diverse questions guided by Bloom's Taxonomy-inspired principles (Krathwohl, 2002). The resulting datasets support a wide range of training paradigms, including embedding model customization for improved retrieval, and LLM fine-tuning with curated distractors for robustness.

In contrast to prior methods, RAGen is explicitly designed for multi-component adaptation. It generates data with rich semantic structure and controlled difficulty, enabling flexible tuning across the entire RAG pipeline. Moreover, its modular design allows it to scale efficiently to large and evolving document corpora, making it suitable for real-world evolving use cases such as enterprise knowledge bases or scientific domains.

Empirical results across multiple domains demonstrate that RAGen-generated data significantly improve both retrieval quality and generation accuracy. Compared to baselines, our approach yields deeper, more holistic questions and enhances performance across a variety of adaptation tasks. These findings highlight RAGen as a practical and generalizable solution for building robust, domain-adapted RAG systems.

2 Related Work

Question Generation Recent work has explored automatic QA pair generation to support domain-specific tasks and reduce the cost of manual annotation. CliniQG4QA (Yue et al., 2021) focuses on generating controlled and diverse QA pairs for the clinical domain using a two-step approach: question phrase prediction (QPP) followed by answer-aware question generation. While this improves control over question types, the method is template-driven and assumes access to clean, short text passages, making it less suitable for noisy, unstructured, or long-form enterprise documents. E2EQR (Hwang et al., 2024) proposes multi-hop QA generation by iteratively rewriting simple sub-questions into complex, compositional queries. However, it lacks explicit mechanisms for semantic evidence selection or grounding, which are crucial for ensuring answer faithfulness and supporting RAG-specific training. RAGEval (Zhu et al., 2024) addresses the evaluation of QA datasets in RAG contexts, introducing metrics for retrieval accuracy, answer grounding, and semantic coherence. While valuable as an evaluation tool, RAGEval relies on scenario-specific schema and configuration extraction, limiting its flexibility in diverse or evolving domains. Other recent efforts leverage large language models for synthetic QA generation. FinTextQA (Chen et al., 2024a), for example, targets financial text by combining semantic retrieval with sentence-windowing and LLM-based generation. However, it assumes the availability of an external question bank (e.g., from textbooks), restricting its applicability in low-resource or unseen domains. Similarly, QAG (Ushio et al., 2023) adopts an answer-first pipeline that identifies answer spans and generates corresponding multi-hop questions, but it is not designed to guide retrieval or align with RAG-specific reasoning processes.

While these efforts advance scalable QA generation, most fall short in capturing long-range semantic dependencies, integrating multi-source evidence, or producing training data aligned with the dual needs of retriever–generator interaction in RAG systems. In contrast, our approach emphasizes dynamic concept fusion, multi-chunk alignment, and semantic grounding, enabling the creation of high-quality QA data that is robust, diverse, and directly applicable to complex RAG adaptation paradigms.

Retrieval-Augmented Generation (RAG) RAG systems (Lewis et al., 2020) enhance language models by integrating them with neural retrievers that fetch relevant external documents to ground generated responses. A typical RAG pipeline consists of three core components: a retriever, which selects top-k relevant passages (often using dense or hybrid embeddings); an embedding model, which maps both queries and documents into a shared representation space for effective retrieval; and a language model, which synthesizes answers from the retrieved content.

Early work such as DPR (Karpukhin et al., 2020) focuses on improving retrieval through dense representation learning, while generation modules typically leverage pre-trained encoder–decoder models like BART (Lewis et al., 2019) or T5 (Raf-

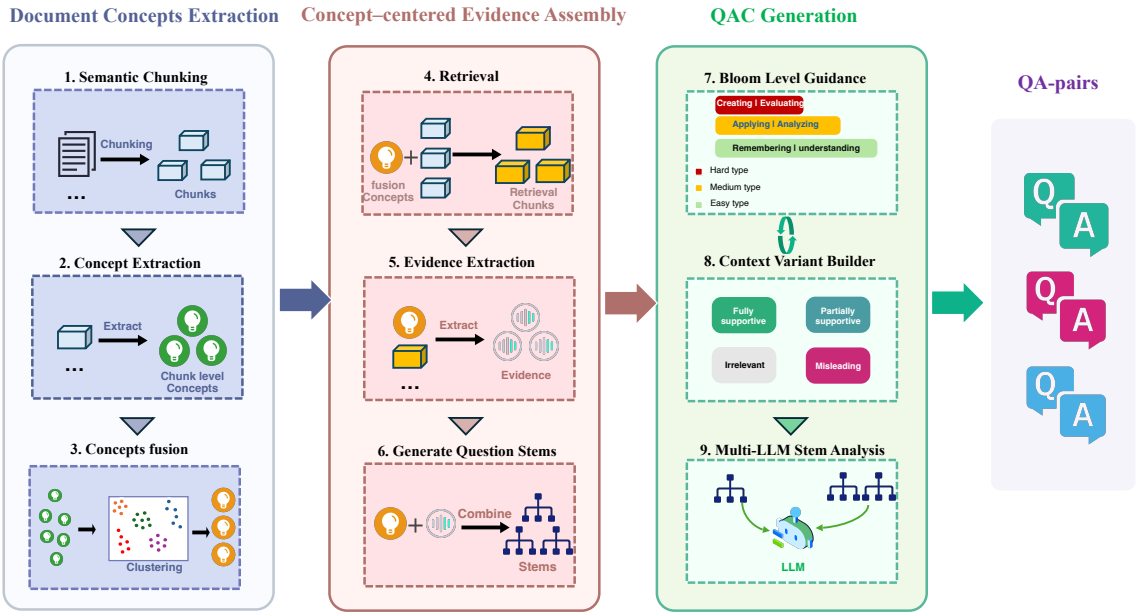

Figure 1: Overview of RAGen framework, a three-stage process that first extract document concepts and then construct question stems, and finally create Question-Answer-Context datasets.

fel et al., 2020). For improving the embedding space, MAFIN (Zhang et al., 2024a) proposes a method to fine-tune black-box embedding models by augmenting them with trainable, open-sourced embeddings on domain-specific tasks. To capture inter-passage relationships, GraphRAG (Edge et al., 2024) models retrieved content as a graph and performs graph-based traversal during retrieval and decoding. Although effective for structured data, GraphRAG relies on predefined graph schemas and lacks flexibility in adapting to new domains or dynamically constructing training data.

In terms of LLM optimization, RAFT (Zhang et al., 2024c) introduces distractor-aware supervision to improve the model's robustness against noisy or irrelevant contexts. More recent work has focused on inference-time retrieval control, where the LLM actively guides what and when to retrieve. Representative approaches include Self-RAG (Asai et al., 2023), OpenRAG (Islam et al., 2024), and R1Searcher (Song et al., 2025) which adopt end-to-end training paradigms to align retrieval behavior with generative intent.

While these methods optimize specific components of RAG—such as retrievers, embedding models, or LLMs, they generally assume the availability of high-quality, domain-relevant training data targeting these specific components. By contrast, our work addresses this upstream challenge by proposing RAGen: a scalable framework for generating high-quality, semantically grounded question–answer–context (QAC) datasets. RAGen supports the training and optimization of various RAG components, thereby enabling end-to-end system enhancement across diverse RAG architectures and training paradigms.

3 Methodology

The RAGen pipeline is designed to automatically generate rich, high-quality question–answer–context (QAC) training data to support diverse RAG adaptation strategies. RAG adaptation refers to the process of systematically refining individual components of a Retrieval-Augmented Generation (RAG) system—such as the large language model (LLM), retriever, and embedding model—to enhance accuracy and robustness of the RAG system under dynamic domain-specific settings.

In the following, we will present the RAGen workflow, which comprises three main modules: *(i) Document concepts extraction*, *(ii) Question stems construction*, and *(iii) QA and context generation*. The overall workflow is illustrated in Fig.1.

3.1 Document Concepts Extraction

Semantic chunking. Given the domain documents D, we employ the standard *llamaindex chunker* to partition the text into a set of coherent chunks $\{d_1, d_2, \ldots\}$.

Chunk–level concept extraction. For each chunk d_i, ChatGPT-4o (OpenAI, 2025a) is

prompted to extract a set of concise, non-generic descriptors referred to as chunk-level concepts: $\mathcal{C}_i = \{c_1^i, c_2^i, \dots\}$, which capture the central themes of d_i.

Concept Fusion. To capture high-level semantics across a document, all chunk-level concepts are further fused based on semantic similarity, resulting in a de-duplicated set of representative *document-level concepts*: $O = \{o_1, o_2, \dots, o_K\}$.

The fusion process begins by eliminating redundant terms and synonyms from the chunk-level concepts. Each remaining concept is then embedded into a vector space using the OpenAI Ada embedding model (OpenAI, 2025b). Finally, the K-means clustering algorithm is applied to group these embeddings into K semantically coherent clusters, where K serves as a tunable hyperparameter. For each cluster, the concept closest to the centroid is selected as its representative, serving as a concept at the document level. Alternatively, an LLM-based summarization can be employed to abstract each cluster into a concise descriptor as the document-level concept.

This fusion step significantly reduces the dimensionality of chunk-level concept space, enabling the identification of core thematic ideas across the document. These document-level concepts guide subsequent cross-chunk retrieval and facilitate the generation of holistic, globally grounded questions—rather than shallow, localized ones.

3.2 Concept–centered Evidence Assembly

Cross-chunk Retrieval. Given the document-level concepts derived in the previous stage, we perform cross-chunk retrieval to collect semantically relevant contexts. For each concept, we use a retriever–reranker pipeline consisting of the dense retriever and *BGE-Reranker-Base* (Zhang et al., 2024b) to retrieve the top-N most relevant chunks from the document corpus. Due to the abstract and high-level nature of document-level concepts, this process often surfaces non-sequential chunks scattered across the document. This enables a departure from traditional single-chunk-based generation strategies, which tend to produce overly localized contexts and shallow questions. Instead, our approach supports the synthesis of holistic, multi-faceted questions grounded in distributed evidence.

Evidence Extraction. Although the retrieved chunks are semantically related, they are often coarse-grained and may contain information unrelated to the target concept. To isolate relevant content, we perform sentence-level filtering within each chunk to extract a concept-focused subset of text, referred to as the evidences e, via sentence window retriever, denoted as $d \xrightarrow{o_i} \{e_0^{o^i}, e_1^{o^i}, \dots, e_N^{o^i}\}$. This step simulates the human annotation process, where a reader selects specific spans of interest before crafting a question. By narrowing the scope to concept-relevant sentences, we ensure that the subsequent question generation process remains focused, interpretable, and controllable.

Unlike existing QA generation methods that operate on isolated, single chunks, our approach assembles evidences from multiple, non-contiguous chunks scattered across the document. The resulting set of evidences for each concept forms a semantically grounded *Question Stem*, denoted as $\mathcal{S}$, which serves as the basic unit for downstream question generation.

While single-stem inputs enable the generation of concept-focused, context-aware questions, we further support multi-stem combinations—allowing the question generator to condition on multiple concepts simultaneously. This enables the creation of global, cross-concept questions that require deeper reasoning and more complex logical chaining. As such, our approach supports the generation of holistic, semantically rich questions that go beyond the limitations of single-chunk-based methods, better simulating human-level comprehension and reasoning over long-form content.

3.3 QAC Generation

Bloom's question-type. After constructing a list of K question stems, each consisting of concept-centered evidence, we sample them to form input to the question generator. We define the number of stems combined per input as the combination level, denoted by ℓ. When $\ell = 1$, we iterate through all individual stems. For $\ell \geq 2$, the number of possible combinations becomes C_K^ℓ, which can grow rapidly. To manage this combinatorial explosion, we impose an upper limit on the number of questions generated for each level ℓ; once this threshold is met, we stop enumerating further combinations at that level. For each input consisting of one or more Question Stems, we prompt ChatGPT-4o to generate diverse types of questions supported by the associated evidences. To guide this process, we adopt Revised Bloom's Taxonomy (Krathwohl,

2002), a widely used pedagogical framework that categorizes cognitive learning objectives in ascending order of complexity:

- *Remembering*: Recognizing or recalling information,
- *Understanding*: Constructing meaning from information.
- *Applying*: Using knowledge in new situations,
- *Analyzing*: Breaking down information into parts and finding evidence,
- *Evaluating*: Making judgments based on criteria,
- *Creating*: Putting elements together to form a coherent whole.

By aligning question types with Bloom's Taxonomy, we simulate the cognitive learning trajectory of humans and enable the generation of questions that span from factual recall to complex synthesis and reasoning. This approach allows us to explicitly control the difficulty distribution of the generated dataset, ensuring a balanced mix of lower-order and higher-order cognitive questions. In addition, the flexible combination of stems—especially at higher ℓ levels—naturally promotes diversity in both content and reasoning depth, enabling the dataset to cover a wider range of topics and inferential patterns.

Notably, for combinations where $\ell \geq 2$, it is possible that no meaningful question can be inferred—particularly when the concepts in the stems are semantically unrelated. In such cases, we discard the current combination and move on to the next.

By combining chunk-level concept fusion with multi-stem aggregation, our framework supports both cross-chunk and cross-concept reasoning. This layered design promotes the generation of high-quality, pedagogically diverse, and cognitively rich question–answer–context samples suitable for domain-specific RAG adaptation.

Question Generation. Conditioned on the selected stem combination and Bloom's Taxonomy levels, we prompt ChatGPT-4o to generate the question, its reference answer, a concise reasoning trace, and the supporting evidences.

To enhance retrieval sensitivity and robustness, we further associate each question–answer (QA) instance with four curated context variants (Below, we use the question "*what are the possible colors of apple?*" as the example):

- *Fully-supportive*: Sentences directly drawn from the evidence set that completely answer the question. Example: "*Apples have various colors: red, green, yellow depending on the variety.*"
- *Partially-supportive*: A subset of the evidence that contains incomplete information, requiring cross-evidence reasoning. Example: "*Fuji apples are famous for their red surface.*"
- *Irrelevant*: Content from the same domain but unrelated to the question. Example: "*Bananas turn from green to yellow when they ripen.*"
- *Misleading*: Topically related but semantically insufficient content that could plausibly mislead a reader. Inspired by human reading comprehension distractors, these passages share surface similarity but fail to answer the question. Example: "*Apple trees have flowers that are mainly white or light pink.*"

Unlike prior methods that rely solely on randomly sampled chunks as distractors, our well-curated distractors increases the semantic difficulty of the retrieval task while encourages higher-order reasoning and a deeper understanding of domain semantics during model adaptation.

Through the RAGen pipeline, we finaly generate high-quality, domain-specific datasets from seed documents to support a variety of RAG adaptation strategies. Each data sample includes a question, the associated concepts, a corresponding answer, and multiple curated contexts. These elements collectively enable fine-grained control over question difficulty and content diversity.

4 Experiments

We evaluate the proposed RAGen framework by constructing three domain-specific datasets: PPFS, TradePolicy, and AIBusiness. PPFS is derived from APEC Policy Partnership on Food Security meeting documents covering topics such as water management, rural development, and sustainable agriculture. TradePolicy includes import/export regulations (primarily for meat and seafood) collected from eight APEC economies. BusinessAI consists of technical reports on AI adoption across various business sectors. All data are collected from publicly available websites.

We generate QAC datasets from these seed documents using RAGen and compare them against two baselines: 1. AutoRAG – an automated framework that searches for optimal RAG pipeline configurations on user-provided data, including a built-in

Domain	Corpus No.	Questions No.
PPFS	15 /3	2726 /2502 /2084
TradePolicy	20 /5	1977 /1820 /1500
BusinessAI	17 /3	2228 /2118 /2072

Table 1: Corpus size (training/evaluation) and number of generated questions (RAGen / LlamaIndex / AutoRAG) for each domain.

dataset generation module. 2. LlamaIndex Dataset Generator(LlamaIndex, 2025) – an open-source QA data generator for RAG evaluation. We refer to it as LlamaIndex in this paper.

Both baselines follow a single-chunk question generation paradigm: AutoRAG uses a simplified Bloom-style taxonomy (factual/conceptual), while LlamaIndex applies intra-chunk retrieval similar to our evidence extraction step. We exclude RAGEval due to its reliance on structured schemas, which are incompatible with our unstructured corpora.

Each dataset is constructed from self-contained documents, enabling standalone QA generation without cross-document reasoning. Evaluation splits are shown in Table 1. We apply the same document partitions and maintain comparable question volumes across RAGen, AutoRAG, and LlamaIndex to ensure fairness.

To assess the impact of RAGen data, we conduct experiments on both embedding model customization and LLMs fine-tuning using 4×NVIDIA RTX 3090 GPUs. Results consistently show that RAGen-generated datasets lead to improved performance across multiple adaptation settings.

Hyperparameter discussion During question generation, all methods segment documents into 1024-token chunks with a 200-token overlap. For single-chunk baselines (AutoRAG, LlamaIndex), question generation is controlled by a single hyperparameter: the number of questions per chunk. However, this approach is inherently constrained by the limited semantic scope of each chunk, and increasing the value often leads to redundant or low-quality questions. To balance question quantity and quality, we carefully tune this hyperparameter for both baselines. As shown in Table 1, AutoRAG consistently produces the fewest questions across all domains.

In contrast, RAGen generates questions from document-level concept stems, which reflect higher-level semantics across chunks. The number of stems scales with content richness, and RAGen further supports multi-stem combinations, enabling

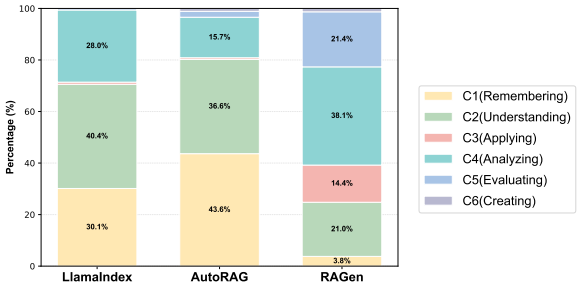

Figure 2: Cognitive level distribution in PPFS domain.

cross-concept, cross-chunk reasoning. To ensure fairness, we restrict generation to combination levels $\ell \leq 2$ with a cap of 50 questions for $\ell = 2$. Even under this constraint, RAGen consistently yields more diverse and semantically rich questions than single-chunk methods.

4.1 Dataset Analysis

Cognitive Level Coverage. Fig.2 shows the distribution of Bloom's cognitive levels for questions generated by LlamaIndex, AutoRAG, and RAGen. Compared to the other two, RAGen produces a markedly richer mix of higher-order question types (Analyzing, Evaluating, Creating) while drastically reducing low-level (Remembering and Understanding) questions. This indicates that RAGen-generated data are more holistic and conceptually comprehensive, moving beyond surface-level recall to support deeper reasoning and complex learning objectives—essential for building robust, domain-adapted RAG systems.

Cross-concept and Cross-chunk Questions. RAGen supports multi-stem conditioning, where multiple document-level concepts—each associated with evidence from distinct chunks—are jointly used to generate a single question. This design naturally enables the generation of cross-concept questions, which often span multiple chunks, resulting in more holistic and semantically rich QA pairs. As illustrated in Fig. 3, such questions require deeper reasoning and capture relationships across disparate parts of a document. In contrast, single-chunk methods like LlamaIndex are limited to localized questions, reducing both answer completeness and dataset diversity. RAGen's ability to support multi-faceted, cross-concept reasoning reflects a key advantage for developing realistic RAG systems.

4.2 Embedding model customization

In domain-specific RAG systems, the embedding model plays a pivotal role in retrieval accuracy, which directly influences generation quality. While

Question: How can the integration of document drafting agents impact the incremental profit and loss in life sciences companies? (Concept: Document Drafting Agent & Profit and Loss)

Evidence(from Chunk4): [...agents could free up 25 to 40 percent of employees' workloads... allowing employees to focus on more strategic, value-adding, and productive work...]

Evidence(from Chunk6): [...the full potential of enterprise-wide agentic transformation could boost top medtechs' EBITDA by 2.2 to 4.7 percentage points...]

Evidence(from Chunk11): [...documentation agents can achieve 75 to 80 percent productivity gains for initial document generation...]

Answer: ① ...can automate the generation of manufacturing practice documents, achieving productivity gain. ② ...allow employees to focus on more strategic tasks, potentially freeing up their workload. ③ ...the full potential of enterprise-wide agentic transformation could boost top medtechs' EBITDA...

Figure 3: Cross-concept question sample: By drawing on 3 non-adjacent evidence sources, cross-concept questions promote deeper, more holistic reasoning, moving beyond localized facts to capture broader operational and financial implications.

pre-trained models provide general-purpose embeddings, they may underperform in specialized domains. For example, the word "pitch" carries very different meanings in sports domain ("The baseball pitch was perfect.") and business domain ("The startup delivered a great pitch to investors."), illustrating how domain context shapes semantic interpretation. To address this, we follow prior works in embedding fine-tuning (Wang et al., 2023; Xiao et al., 2023; Zhang et al., 2023) and adopt the open-source framework *FlagEmbedding* (BAAI, 2025) for both embedding model fine-tuning and evaluation to investigate how fine-tuning embedding models through contrastive training with synthetic domain-specific data can enhance retrieval performance in domain-adapted RAG settings.

Setup. We conducted embedding model customization experiments on all three domain-specific datasets. To demonstrate the effectiveness of the RAGen datasets, we select three different embedding models: BGE-large-v1.5 (BAAI, 2024) (hereafter referred to as BGE-large), BGE-m3 (Chen et al., 2024b), and E5-large-v2 (Wang et al., 2022). Under the InfoNCE objective (Oord et al., 2018), we set the learning rate to 1e-5 for 3 epochs, with the temperature parameter $\tau = 0.02$ and the number of negative samples set to 2 . All models are fine-tuned using a full-parameter training setup with consistent hyperparameters across all runs. For evaluation, we assess the fine-tuned model on the split-out evaluation datasets of the three domains. Specifically, for each domain, we randomly select 300 samples from the AutoRAG, LlamaIndex, and RAGen evaluation datasets respectively to form the final evaluation set. We adopt Recall@K (K=1, 5, 10) and Mean Reciprocal Rank (MRR@10) as the evaluation metrics, which are widely used in the evaluation of information retrieval system.

For all methods, we construct contrastive training triplets following the standard contrastive learning format. The positive sample is the original chunk used to generate a QA pair. For the AutoRAG and LlamaIndex datasets, the negative samples consist of two randomly selected chunks from the same corpus, which serve as 2 irrelevant context negatives. In contrast, for the RAGen dataset, two negative samples are used: one irrelevant context and one misleading context. retrieval systems.

Results. Table 2 presents the complete results. All customized models outperform the uncustomized baseline (denoted as Vanilla), confirming the necessity of domain-specific embedding customization. Datasets generated by RAGen consistently achieve superior performance across all domains and models, demonstrating the effectiveness of our data generation strategy.

4.3 LLMs Supervised Fine-tuning

Setup. We perform standard LoRA-based supervised fine-tuning (Hu et al., 2022) on the Qwen2.5-1.5B and Qwen2.5-3B models (Qwen et al., 2025) using the QAC datasets generated from the three domains. All experiments are conducted using the open-source LlamaFactory framework (Zheng et al., 2024), with a fixed learning rate of 1e-5, five training epochs, and a 10% validation split.

For input construction, we follow a consistent schema across all methods. In AutoRAG and LlamaIndex, the original chunk used to generate each question (the *golden context*) is concatenated with the question to form the model input. For RAGen, all supportive evidence chunks are concatenated as the golden context.

To ensure a fair evaluation, we randomly sample 300 questions from the evaluation sets of each methods across all domains. Given that the task involves long-form QA, we adopt ROUGE-L and BERT-F1 as metrics for assessing lexical overlap

Vanilla Model	Finetune Strategy	PPFS				TradePolicy				BusinessAI			
		R@1	R@5	R@10	MRR@10	R@1	R@5	R@10	MRR@10	R@1	R@5	R@10	MRR@10
BGE-large(BAAI, 2024)	Vanilla	0.1548	0.4368	0.5549	0.2722	0.1961	0.4691	0.6214	0.3154	0.1068	0.3291	0.4263	0.2019
	AutoRAG	0.1877	0.5183	0.6712	0.3342	0.2247	0.5505	0.6606	0.3573	0.1560	0.4818	0.6325	0.2972
	LlamaIndex	0.2024	0.5604	0.6987	0.3548	0.2474	0.5686	0.6893	0.3789	0.1624	0.4893	0.6261	0.3036
	RAGen	**0.3095**	**0.6584**	**0.7821**	**0.4626**	**0.3891**	**0.8069**	**0.8899**	**0.5586**	**0.3002**	**0.6827**	**0.8120**	**0.4693**
BGE-m3(Chen et al., 2024b)	Vanilla	0.2115	0.5018	0.6136	0.3359	0.2368	0.5309	0.6516	0.3584	0.1368	0.4241	0.5417	0.2602
	AutoRAG	0.2015	0.5055	0.6383	0.3377	0.2594	0.5807	0.6953	0.3909	0.1603	0.5043	0.6271	0.3066
	LlamaIndex	0.2125	0.5687	0.7042	0.3664	0.2881	0.5792	0.7074	0.4114	0.1538	0.4947	0.6282	0.3000
	RAGen	**0.2692**	**0.6255**	**0.7647**	**0.4261**	**0.3665**	**0.7888**	**0.8944**	**0.5355**	**0.2318**	**0.6677**	**0.7906**	**0.4232**
E5-large-v2(Wang et al., 2022)	Vanilla	0.1749	0.4844	0.6273	0.3052	0.1131	0.4449	0.5913	0.2472	0.1015	0.3205	0.4573	0.1977
	AutoRAG	0.1905	0.5201	0.6465	0.3274	0.1388	0.4449	0.6199	0.2685	0.1047	0.3226	0.4679	0.2049
	LlamaIndex	0.1996	0.5348	0.6767	0.3451	0.1976	0.5158	0.6440	0.3259	0.1026	0.3568	0.4979	0.2123
	RAGen	**0.2665**	**0.6511**	**0.7848**	**0.4345**	**0.3469**	**0.7677**	**0.8778**	**0.5074**	**0.2767**	**0.6912**	**0.8066**	**0.4554**

Table 2: Retrieval performance on 3 domains. the best results are in bold. All results are averaged over 3 runs.

and semantic similarity to evaluate model performance against reference answers.

Results. Table 3 presents the results across all domains. Models fine-tuned on RAGen-generated data consistently outperform those trained on the AutoRAG and LlamaIndex datasets across both evaluation metrics—ROUGE-L and BERT-F1—thereby demonstrating superior factual consistency and semantic relevance.

These improvements validate the effectiveness of RAGen datasets. Notably, RAGen maintains its advantage across all three domains, indicating strong generalization ability beyond a single knowledge area. Furthermore, the consistent gains observed on both Qwen2.5-1.5B and Qwen2.5-3B confirm the scalability of our approach across model sizes.

Distractor Supervision Setup. Motivated by RAFT (Zhang et al., 2024c), which demonstrates the benefits of distractor exposure during training,

Domain	Method	ROUGE-L	BERT-F1
	Qwen2.5–1.5B Instruct		
PPFS	AutoRAG	0.2876	0.8847
	LlamaIndex	0.3293	0.8903
	RAGen	**0.3955**	**0.9094**
TradePolicy	AutoRAG	0.2775	0.8726
	LlamaIndex	0.2698	0.8696
	RAGen	**0.3911**	**0.9033**
BusinessAI	AutoRAG	0.2701	0.8852
	LlamaIndex	0.3223	0.8925
	RAGen	**0.3392**	**0.9038**
	Qwen2.5–3B Instruct		
PPFS	AutoRAG	0.3436	0.8979
	LlamaIndex	0.3253	0.8952
	RAGen	**0.3815**	**0.9079**
TradePolicy	AutoRAG	0.3388	0.8875
	LlamaIndex	0.3346	0.8861
	RAGen	**0.3747**	**0.9004**
BusinessAI	AutoRAG	0.3284	0.8985
	LlamaIndex	0.3597	0.9036
	RAGen	**0.3682**	**0.9091**

Table 3: Performance comparison of Qwen2.5–1.5B and –3B models on 3 domains. All results are averaged over 3 runs. The best result is in bold.

Method	ROUGE-L	BERT-F1
$\text{RAGen}_{\text{w/o dis}}$	0.3143	0.8957
$\text{RAGen}_{\text{dis}}$	**0.4074**	**0.9121**

Table 4: Evaluation of Qwen2.5-3B on the PPFS domain under real-world RAG inference (k=3) settings. $\text{RAGen}_{\text{w/o dis}}$ is trained with golden contexts only, whereas $\text{RAGen}_{\text{dis}}$ incorporates distractor supervision. All results are averaged over 3 runs.

we conduct additional experiments to evaluate how distractor-based supervision impacts LLM robustness in real-world RAG settings. We fine-tune models using both golden contexts and 2 distractors (irrelevant and misleading), and evaluate them using a fixed retriever with top-k=3 retrieved chunks on the customized embedding model trained in Sec.4.2.

Results. Table 4 presents the evaluation results on the PPFS domain using the Qwen-3B model and RAGen dataset. We observe a substantial performance drop when the model is fine-tuned without distractors and then exposed to noisy retrieved contexts during inference. In contrast, training with distractor-augmented supervision significantly improves robustness, yielding notable gains in both ROUGE-L and BERT-F1. These findings highlight the effectiveness of distractor-aware training in enhancing model resilience under realistic retrieval conditions.

5 Conclusion

We present RAGen, a scalable and modular framework for generating high-quality, domain-specific QAC datasets to support diverse RAG adaptation strategies. Extensive experiments across multiple domains demonstrate its effectiveness in enhancing retrieval accuracy and answer quality, leading to more effective domain-adapted RAG systems. RAGen offers a practical solution for building domain-adapted RAG systems in complex, evolving knowledge environments.

Limitations

While RAGen demonstrates strong performance in generating high-quality, domain-specific QAC datasets, several limitations remain:

First, our current pipeline operates exclusively on text-formatted documents. However, in many real-world enterprise scenarios, proprietary knowledge is stored in PDFs or other multimodal formats (e.g., scanned documents, tables, or images). Extending RAGen to robustly handle non-text and multimodal inputs remains an open challenge.

Second, the quality of seed documents significantly impact the effectiveness of the generated QAC samples. Inconsistent content, low-quality sources may limit the utility of the resulting data.

Third, RAGen requires manual specification of the number of document-level concept —a hyperparameter that depends on the semantic richness of each document. Automating this selection process remains a direction for future improvement.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

BAAI. 2024. Bge large en v1.5 model. https://huggingface.co/BAAI/bge-large-en-v1.5.

BAAI. 2025. Flag embedding. https://github.com/FlagOpen/FlagEmbedding.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 194–199.

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

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024c. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Filippo Chiarello, Vito Giordano, Irene Spada, Simone Barandoni, and Gualtiero Fantoni. 2024. Future applications of generative large language models: A data-driven case study on chatgpt. *Technovation*, 133:103002.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Seonjeong Hwang, Yunsu Kim, and Gary Geunbae Lee. 2024. Explainable multi-hop question generation: An end-to-end approach without intermediate question labeling. *arXiv preprint arXiv:2404.00571*.

Shayekh Bin Islam, Md Asib Rahman, KSM Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez. 2024. Open-rag: Enhanced retrieval-augmented reasoning with open-source large language models. *arXiv preprint arXiv:2410.01782*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

David R Krathwohl. 2002. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Hui Liu, Wenya Wang, Hao Sun, Chris Xing Tian, Chenqi Kong, Xin Dong, and Haoliang Li. 2025. Unraveling the mechanics of learning-based demonstration selection for in-context learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2623–2641, Vienna, Austria. Association for Computational Linguistics.

LlamaIndex. 2025. Llamaindex datasetgenerator. https://developers.llamaindex.ai/python/framework-api-reference/evaluation/dataset_generation/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

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

OpenAI. 2025a. Openai chatgpt4o. https://platform.openai.com/docs/models/chatgpt-4o-latest.

OpenAI. 2025b. Openai embedding api. https://platform.openai.com/docs/models/text-embedding-ada-002.

Crystal Qian, Michael Xieyang Liu, Emily Reif, Grady Simon, Nada Hussein, Nathan Clement, James Wexler, Carrie J Cai, Michael Terry, and Minsuk Kahng. 2024. The evolution of llm adoption in industry data curation practices. *arXiv preprint arXiv:2412.16089*.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Asahi Ushio, Fernando Alva-Manchego, and Jose Camacho-Collados. 2023. An empirical comparison of lm-based question and answer generation methods. *arXiv preprint arXiv:2305.17002*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Xiang Yue, Xinliang Frederick Zhang, Ziyu Yao, Simon Lin, and Huan Sun. 2021. Cliniqg4qa: Generating diverse questions for domain adaptation of clinical question answering. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 580–587. IEEE.

Mingtian Zhang, Shawn Lan, Peter Hayes, and David Barber. 2024a. Mafin: Enhancing black-box embeddings with model augmented fine-tuning. *arXiv preprint arXiv:2402.12177*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2024b. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*, 2(3):5.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *Preprint*, arXiv:2310.07554.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024c. Raft: Adapting language model to domain specific rag. In *First Conference on Language Modeling*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Kunlun Zhu, Yifan Luo, Dingling Xu, Yukun Yan, Zhenghao Liu, Shi Yu, Ruobing Wang, Shuo Wang, Yishan Li, Nan Zhang, and 1 others. 2024. Rageval: Scenario specific rag evaluation dataset generation framework. *arXiv preprint arXiv:2408.01262*.