ExpertGenQA: Open-ended QA generation in Specialized Domains

Haz Sameen Shahgir[1], Chansong Lim[1], Jia Chen[1], Evangelos E. Papalexakis[1], Yue Dong[1],

[1]University of California Riverside

Abstract

Generating high-quality question–answer (QA) pairs for specialized technical domains is essential for advancing knowledge comprehension, yet remains challenging. Existing methods often yield generic or shallow questions that fail to reflect the depth and structure of expert-written examples. We propose ExpertGenQA, a generation protocol that combines few-shot prompting with dual categorization by topic and question style to produce more diverse and cognitively meaningful QA pairs. ExpertGenQA achieves twice the efficiency of standard few-shot methods while maintaining 94.4% topic coverage. Unlike LLM-based judges, which often favor surface fluency, Bloom's Taxonomy analysis shows that ExpertGenQA better captures expert-level cognitive complexity. When used to train retrieval systems, our questions improve top-1 accuracy by 13.02%, demonstrating their practical value for domain-specific applications. [1]

1 Introduction

In high-stakes domains like law, transportation, and finance, expert-written questions capture how professionals reason, prioritize, and apply knowledge. They emphasize concepts essential for learning and grounded decision-making (Bai et al., 2023; Kale et al., 2024; Wang et al., 2023a; Lee et al., 2024). Different from generic questions that target factual recall, domain-specific questions support deeper understanding and better reflect real-world scenarios. They are also important for training AI systems in tasks such as information retrieval and question answering, where diverse, information-rich questions expose models to complex semantic structures and improve generalization to unseen human-written queries (Wang et al., 2023a).

However, generating such questions at scale requires substantial domain expertise and time, making automatic generation an attractive solution. While domain-adapted LLMs such as BloombergGPT (Wu et al., 2023), FinGPT (Wang et al., 2023b), EcomGPT (Li et al., 2024), BioGPT (Luo et al., 2022), and Med-PaLM (Singhal et al., 2023, 2025) are designed to answer domain-specific questions, they are not optimized for generating them. Prompt-based approaches like Med-Prompt (Nori et al., 2023) show that simply asking better questions can sometimes outperform fine-tuned models on domain benchmarks. Still, generating high-quality, domain-specific questions remains underexplored. Existing methods often default to generic, surface-level prompts (Liu et al., 2024b) that fail to capture the depth and structure of expert-authored questions. For instance, a legal professional gains little from a basic query like *"What is a subpoena?"* when their work requires complex, scenario-driven questions that synthesize statutes, precedents, and regulations.

Evaluating the usefulness of generated questions also remains a challenge. While LLMs have advanced in comparing generated answers, existing evaluation methods are less effective for questions. Reward models and LLM-as-judge approaches (Ouyang et al., 2022; Wang et al., 2024d,c; Liu et al., 2024a; Zheng et al., 2023) often prioritize fluency and syntactic form over semantic depth and task relevance. As a result, questions that score highly with LLM judges frequently perform poorly in downstream retrieval tasks, failing to meet the practical needs of domain experts.

We introduce a question generation pipeline that learns to produce domain-specific questions by imitating a small set of expert-written examples. Our approach focuses on generating question–answer pairs that are not only ***comprehensive in topic coverage*** but also ***capture the cognitive complexity and practical needs*** of domain experts. To achieve this, we ground generation in expert-written FAQs and apply a dual-categorization strategy based on

[1]The code and data are available at https://github.com/Patchwork53/ExpertGenQA

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2934–2955
November 4-9, 2025

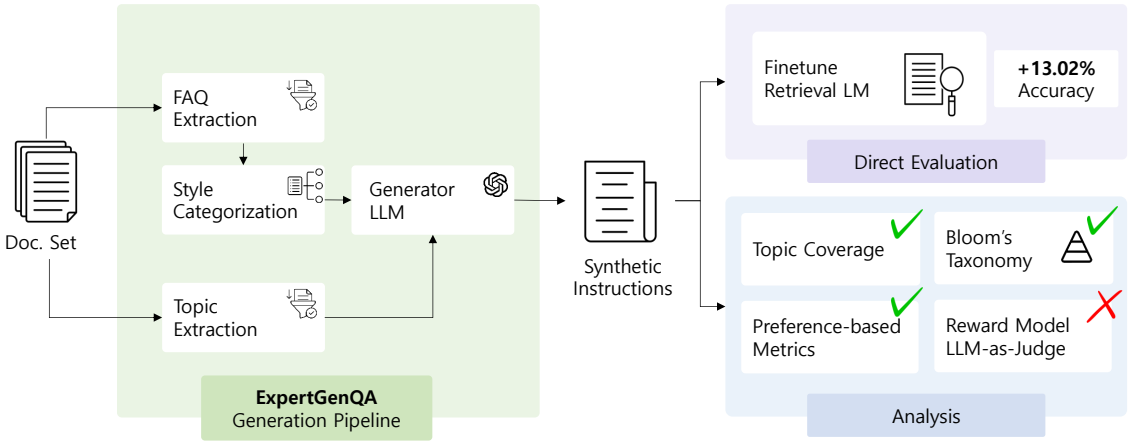

Figure 1: **Overview of the ExpertGenQA pipeline (left) and evaluation strategy (right).** Green checkmarks (✓) indicate interpretable metrics, including diversity, cognitive load, and topic coverage, that correlate with improved retrieval accuracy, our primary evaluation metric. The red cross (✗) highlights that Reward Models and LLM-as-Judge tend to favor surface-level fluency and do not align with retrieval-based measures of question quality.

question style and topic. Our proposed pipeline, ExpertGenQA (Figure 1), significantly outperforms standard few-shot prompting and template-based methods (e.g., MDCure (Liu et al., 2024b)), improving top-1 document retrieval accuracy on human-written test queries from 23.96% to 36.98%. Our analysis shows that these improvements stem from diversity, cognitive load (Bloom et al., 1956; Anderson and Krathwohl, 2001), and topic coverage, all of which strongly correlate with retrieval performance. ExpertGenQA also doubles the generation efficiency of baseline few-shot prompting while maintaining 94.4% topic coverage.

Why Retrieval as an Evaluation Metric? LLM-based evaluation metrics often reward surface-level fluency rather than task relevance. In contrast, retrieval accuracy provides a task-grounded measure of question quality by evaluating whether a question helps models retrieve conceptually relevant content. This better reflects expert reasoning, where questions encode meaningful structure, connect related concepts, and highlight contextually important information. Models trained on high-quality, expert-style questions learn to attend to these signals, resulting in stronger generalization to human-written queries. We use retrieval accuracy as the primary evaluation metric, supported by auxiliary analysis on diversity, cognitive load, and topic coverage (Soni and Roberts, 2021; Wang et al., 2023a).

Our Contributions are:

- We propose a question generation pipeline that uses a small set of expert-written examples to improve domain-specific QA and retrieval tasks.
- We propose a retrieval-based evaluation framework that leverages finetuned retrievers to measure the utility of synthetic questions in specialized domains, addressing the limitations of standard automatic metrics.
- We empirically analyze how question diversity, cognitive load, and topic coverage affect retrieval performance.

2 Related Work

QA Generation. Instruction-tuned large language models (LLMs) have made automated QA generation a scalable alternative to manual annotation. A straightforward strategy is to prompt pretrained LLMs directly (Wang et al., 2023c; Taori et al., 2023; Peng et al., 2023; Geng et al., 2023), enabling rapid generation with minimal setup. However, such outputs often lack diversity (Chen et al., 2024) and may include hallucinated content (Zhao et al., 2023). To improve quality and coverage, several prompt engineering techniques have been proposed. Template-based pipelines such as GenQA (Chen et al., 2024), MDCure (Liu et al., 2024b), and Persona Hub (Ge et al., 2024) guide generation using predefined styles or sampling strategies. For example, GenQA introduces topic-level prompts to encourage diversity, while MDCure improves

complexity by prompting over multi-document inputs. Other work uses augmentation techniques to increase variety and complexity in existing QA datasets (Xu et al., 2023; Mukherjee et al., 2023). Few-shot prompting (Brown et al., 2020) offers another approach, where high-quality exemplars guide generation without rigid structure. While few-shot examples often improve fluency and alignment with human-written QA, they tend to generate repetitive or narrowly focused outputs without explicit diversity controls (Xu et al., 2024b).

For domain-specific tasks, several LLMs have been fine-tuned on medical, financial, and scientific corpora, including BloombergGPT (Wu et al., 2023), FinGPT (Wang et al., 2023b), EcomGPT (Li et al., 2024), BioGPT (Luo et al., 2022), and Med-PaLM (Singhal et al., 2023, 2025). These models primarily focus on answering questions, while question generation in specialized domains often relies on manually curated datasets such as MAmmoTH (Yue et al., 2023) and PubMedQA (Jin et al., 2019). In these settings, the quality and structure of generated QA data become especially important for supporting downstream tasks such as retrieval or tutoring. Recent studies suggest that high-quality, instruction-tuned QA data can substantially improve retrieval performance. For instance, Zhu et al. (2024) find that generic prompt-based generation fails to reflect the specialized query intent and document relevance often needed in real-world retrieval. Complementary work shows that LLMs' instruction-following behavior and information retrieval capabilities can reinforce each other (Weller et al., 2024; Wang et al., 2024a; Tran et al., 2024).

Together, these observations highlight the need for generation methods that support both domain specificity and structural diversity—while maintaining the reasoning patterns and coverage found in expert-authored questions.

Instruction Evaluation. Evaluating instruction-following behavior in LLMs is a central challenge in aligning generated outputs with human expectations. Reward models (RMs) predict user preferences based on supervised comparisons and are widely used during training and evaluation (Ouyang et al., 2022). Examples such as `Nemotron-70B` (Wang et al., 2024d,c) and `Skywork-2-27B` (Liu et al., 2024a) support a range of generation tasks across general domains.

An alternative approach, LLM-as-a-Judge (Zheng et al., 2023; Wang et al., 2024b), uses language models to assess outputs directly through structured ratings or rationale-based feedback. This method is model-agnostic, flexible across domains, and compatible with frontier models like GPT-4o.

While both strategies are widely adopted, their effectiveness in specialized domains, particularly for generated questions, remains underexplored. In tasks like question generation, where outputs must demonstrate semantic depth, domain relevance, and downstream utility (e.g., in retrieval or instruction tuning), standard preference models may fall short. This underscores the need for targeted evaluation frameworks that more accurately reflect task-specific objectives and failure modes.

3 Methodology

This work aims to generate synthetic questions in specialized domains that have practical utility for domain experts. To achieve this, we collect a small set of expert-written questions to serve as exemplars. Our ExpertGenQA pipeline learns domain-specific patterns from these examples and generates new questions that closely align with expert-written questions while maintaining high diversity and comprehensive coverage of source documents.

3.1 Domain-Specific Exemplar Construction

Railway safety is critical to U.S. infrastructure, with 28% of freight transported by rail[2]. As a specialized and highly technical domain that has seen limited applications of AI, it serves as an ideal test case for our approach. We select regulatory documents published by the U.S. Federal Railroad Administration (FRA), the primary federal agency that enforces safety standards and regulations across the decentralized and privatized U.S. rail industry.

We build our corpus by collecting 43 documents from the FRA's digital library containing nationwide railroad regulations and guidelines (totaling 1,158 pages)[3]. These PDFs are converted to text using the `pymupdf4llm`[4] Python package, and pages primarily consisting of tables, images, or diagrams are removed. We extract 147 expert-written QAs from FAQ sections within these documents to serve as exemplars of expert reasoning patterns. Since the original FAQs lack citations to specific source sections, we manually identify and extract the relevant supporting passages. We refer to this dataset as the **FRA Domain**.

[2] https://www.aar.org/industries-we-support/
[3] https://railroads.dot.gov/elibrary-search
[4] https://pypi.org/project/pymupdf4llm/

To validate our findings across diverse domains, we also collect 50 FAQs[5] from the Federal Aviation Administration (FAA) related to Unmanned Aerial Systems (UAS), and use Title 14 CFR Part 107 (Small Unmanned Aircraft Systems), Public Law 115–254, 49 U.S.C. §44809, and 49 U.S.C. §44807 as our source documents, totaling 601 pages. We refer to this domain as the **FAA-UAS Domain**. Further details on domain selection and criteria are provided in Appendix A.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ be the set of preprocessed documents. From these, we extract a set of expert-written QA pairs:

$$\mathcal{H} = \{(q^{(i)}, a^{(i)})\}_{i=1}^{|\mathcal{H}|}.$$

Each QA pair $(q^{(i)}, a^{(i)})$ is manually aligned to a supporting passage $p^{(i)} \subseteq d_j$ for some $d_j \in \mathcal{D}$.

3.2 ExpertGenQA: A Protocol for Diverse Question Generation

Most prior QA generation methods rely on rigid templates to promote diversity, but these often fail to align with expert intent in technical domains. Few-shot prompting improves quality by imitating expert-written questions but tends to lack topical diversity. To balance both, ExpertGenQA introduces a dual-axis prompting protocol: categorizing questions by **style** and **topic**. This separation allows targeted sampling of exemplars while ensuring document-wide coverage.

Style Categorization We manually classify the 147 expert-written questions from the FRA domain into three broad categories: Policy application, which addresses how specific regulations should be interpreted; Scenario-based, which presents specific situations requiring regulatory guidance; and Terminology clarification, which focuses on defining and explaining technical terms. These broad categories generalize well across documents and help steer the LLM toward consistent and realistic question framing.

For the FAA-UAS domain, we adopt four categories: How-To (procedures like filing reports), Certification-related (requirements for operating UAVs), Jurisdiction-related (which agencies control which airspaces), and Scenario-based. Examples are provided in Appendix H.

Topic Extraction To ensure broad coverage, we use LLM-based topic extraction to identify main topics from each document. Let $\mathcal{T}_d = \{t_1, t_2, \ldots, t_m\}$ denote the set of extracted topics for document d, where

$$f_{\text{topic}} : d \mapsto \mathcal{T}_d.$$

This prevents the model from focusing on only topics that it thinks are most salient which would lead to generating repetitive questions.

Question Generation For each document $d \in \mathcal{D}$ and each topic $t \in \mathcal{T}_d$, we aim to generate questions across different styles. For a given style $s \in \mathcal{S}$, we have a set of expert written question-answer pairs $\mathcal{H}_f$ which we use as few-shot examples. Since using different subsets of few-shot examples may boost generation diversity, we randomly sample K subsets of n few-shot examples from the expert QA pool $\mathcal{H}_f$, denoted as:

$$\mathcal{F}_{k,s} \subset \mathcal{H}_f, \quad |\mathcal{F}_{k,s}| = n.$$

Each question is then generated as:

$$q_{d,t,s,k} = \textsc{Generate}(d, t, \mathcal{F}_{k,s}),$$

where GENERATE is an LLM-based function that takes a document chunk, a topic, and a few-shot set as input. The full generated set is:

$$\mathcal{G} = \bigcup_{d \in \mathcal{D}} \bigcup_{s \in \mathcal{S}} \bigcup_{k=1}^{K} \bigcup_{t \in \mathcal{T}_d} \{(d, q_{d,t,s,k})\}.$$

This results in up to $|\mathcal{T}_d| \cdot |\mathcal{S}| \cdot K$ questions per document.

Efficiency Optimization Since each few-shot example contains a long section of a document and a question, the token cost of few-shot prompting is high. We circumvent this issue by prefix-caching the few-shot set $\mathcal{F}_{k,s}$ and reusing it when generating questions from each topic t in a document section d. This allows the LLM to avoid reprocessing repeated context and reduces the cost of ExpertGenQA substantially.

Deduplication After generation, we apply a fuzzy deduplication function to remove near-duplicate and paraphrased questions. Following Mu et al. (2024), we define the similarity between two questions q_i and q_j using normalized bigram overlap:

$$\text{Overlap}(q_i, q_j) = \frac{|B(q_i) \cap B(q_j)|}{\min(|B(q_i)|, |B(q_j)|)} > \theta,$$

where $B(q)$ is the set of bigrams in q, and θ is a fixed threshold. We keep only one question if the overlap exceeds θ.

[5] https://www.faa.gov/faq

Algorithm 1: ExpertGenQA Framework

Input : Document chunks $\mathcal{D}$, question styles $\mathcal{S}$, human QA pairs $\mathcal{H}$, no. few-shot combinations per style K, no. few-shot examples n
Output : Generated question set $\mathcal{G}$

$\mathcal{G} \leftarrow \emptyset$
for $d \in \mathcal{D}$ **do**
  $\mathcal{T} \leftarrow \text{EXTRACTTOPICS}(d)$; `// LLM-based topic extraction (H.2.1)`
  for $s \in \mathcal{S}$ **do**
    for $k = 1, \ldots, K$ **do**
      $\mathcal{F} \leftarrow \text{SAMPLEFEWSHOT}(\mathcal{H}, s, n)$; `// Sample` n `style-specific examples`
      for $t \in \mathcal{T}$ **do**
        $q \leftarrow \text{GENERATE}(d, t, \mathcal{F})$; `// LLM generation with examples (H.2.2)`
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{(d, q)\}$
return $\mathcal{G}$

4 Evaluation and Experimental Setup

While a plethora of methods exist for evaluating LLM generated answers (Min et al., 2023; Bai et al., 2024), research into evaluating LLM generated questions is limited. Furthermore, we uncover key shortcomings in existing methods (Liu et al., 2024b) that use LLMs or reward models in our pilot experiments. As such, we opt for a retrieval-based evaluation framework in which we assess question quality through its impact on downstream retrieval performance, using it as a proxy for practical utility. This setup directly measures how well generated questions facilitate document navigation and comprehension in highly technical domains.

4.1 Retrieval-based Evaluation

Standard retrieval models perform poorly in specialized technical domains like railway regulations, where documents have similar vocabulary, structure, and overlapping terminology (Xu et al., 2024a; Lewis et al., 2020). Technical domains use specialized vocabulary where subtle distinctions carry significant regulatory implications, and general-purpose retrievers struggle to disambiguate these nuanced differences. Additionally, regulatory documents often share similar section structures and phrasing patterns, making it difficult for retrievers to distinguish between relevant and irrelevant passages.

We leverage these inherent challenges as a robust evaluation framework. The reasoning is straightforward: synthetic questions that better capture domain expertise should lead to measurably improved retrieval performance when used as training data. This approach directly measures the downstream practical utility of synthetic data for information retrieval.

For each synthetic generation pipeline, we finetune a retrieval LM, `gte-modernbert-base` (Zhang et al., 2024; Warner et al., 2024) using the generated document-query pairs and evaluate performance using the human-authored QA pairs as a test set. This provides a highly practical signal for comparing different question-generation approaches based on their utility for downstream retrieval tasks. We use the InfoNCE loss (Oord et al., 2018) to fine-tune retrieval LMs (See Eqn. 3). InfoNCE loss compares a positive pair of samples (like a query and its corresponding document) against multiple negative pairs, encouraging the model to maximize agreement between positive pairs while pushing apart negative pairs in the representation space.

$$L_{infoNCE} = -\log \frac{e^{s(q,d^+)/\tau}}{e^{s(q,d^+)/\tau} + \sum_{i=1}^{n} e^{s(q,d_i^-)/\tau}} \tag{3}$$

where $s(q, d)$ is the similarity function between query embedding q and a document embedding d, d^+ is a document embedding relevant to answering q and $\mathcal{D} = \{d_1^-, \ldots, d_n^-\}$ is a set of irrelevant document embeddings. τ is a temperature hyperparameter that controls the sharpness of the probability distribution over similarities. We use only in-batch negatives instead of mining hard negatives for simplicity (Lee et al., 2024).

4.2 Experimental Configuration

We use `GPT-4o` (Achiam et al., 2023) for all tasks including topic extraction, question generation, Bloom's Taxonomy classification, and response evaluation. Following Chen et al. (2024), we set

the generation temperature to $T = 1$ and sample 5 generations per input. For near-duplicate filtering, we apply a strict bigram overlap threshold of 0.3.

For retrieval evaluation, we use `NVEmbed-70B-V2` (Lee et al., 2024) as a zero-shot baseline and fine-tune `gte-modernbert-base` (Zhang et al., 2024) (150M parameters) using InfoNCE loss (Oord et al., 2018) with batch size 64, learning rate 1×10^{-5}, and temperature $\tau = 0.1$, using cosine similarity as the scoring function $s(q, d)$. We use only in-batch negatives. Generated questions are used as training queries; human-authored QA pairs serve as the test set. We filter out any generated question with ≥ 0.3 bigram overlap with any test question.

5 Results

5.1 Diversity of Generated Questions and Pipeline Efficiency

We evaluate the efficiency of ExpertGenQA against two baselines: few-shot prompting and MDCure (Liu et al., 2024b), a prompt-template-based pipeline that does not use examples. Generating diverse synthetic questions is important not only for downstream applications but also for efficiency reasons, as redundant generations result in wasted LLM calls.

Figure 2 demonstrates that, on the FRA domain, ExpertGenQA with 10 examples produces twice as many unique questions as the few-shot prompting baseline for the same number of LLM calls. More examples generally increase the efficiency of both ExpertGenQA and few-shot prompting. With 10 examples, ExpertGenQA generates $7,140$ unique questions from $17,622$ LLM calls, while 10-shot prompting generates only $3,658$ unique questions. In contrast, MDCure, being a purely template-based approach without examples, maintains a static efficiency of 15.71%, generating $8,030$ instructions from $51,100$ LLM calls. On the FAA-UAS domain, 10-shot prompting has an efficiency of 26.03% while ExpertGenQA achieves 38.37% efficiency. The detailed statistics of the generated questions can be found in Appendix B. We also include qualitative examples of synthetic questions are included in Appendix H .

5.2 Retrieval LM

Table 1 shows that finetuning a retrieval LM `gte-modernbert-base` (Zhang et al., 2024; Warner et al., 2024) on ExpertGenQA generations significantly improves Top-1 retrieval accuracy from 23.96% to 36.98% on the FRA domain and from 38% to 48% on the FAA-UAS domain, outperforming even the much larger generalist retrieval LM `NVEmbed-V2` (Lee et al., 2024). In contrast, finetuning on synthetic instructions from 10-shot prompting and MDCure yields more modest improvements of $+7.15\%$ and $+3.64\%$ respectively. Notably, the retrieval LM fine-tuned on MDCure-generated data achieves lower retrieval accuracy than the 10-shot pipeline, despite having more than twice the training data (8,030 instructions vs. 3,658). This demonstrates the importance of synthetic data matching the complexity and utility of expert-written QA for practical applications like retrieval.

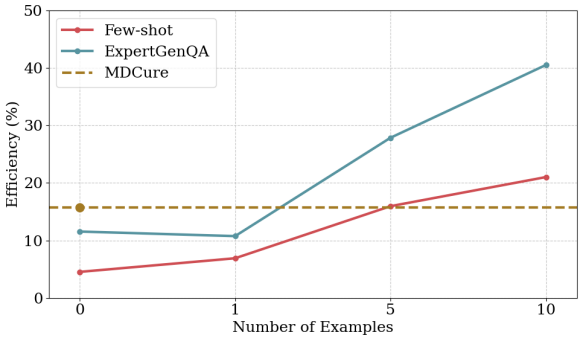

Figure 2: Comparison of efficiency across question-generation pipelines over the different number of few-shot examples on the FRA domain. We define efficiency as the fraction of unique generations over the total sampled generations.

6 Analysis

While directly finetuning a retrieval language model provides the most accurate measure of synthetic question effectiveness, it is impractical during pipeline development due to the compute required. We investigate alternative evaluation metrics using the larger **FRA domain** to determine which ones meaningfully correlate with improvements in retrieval language model performance. These metrics could serve as more practical proxies for assessing question quality during the development process.

6.1 Qualitative Analysis

We begin by examining the expert-written QAs and comparing them with 100 randomly selected QAs from each of the generation methods: MDCure, few-shot prompting, and ExpertGenQA.

As shown in the examples provided in Appendix H, we immediately notice that expert-

Model	# Params	Top-1	Top-5
FRA (147 QA)			
`gte-baseline`	150M	23.96	55.73
`NVEmbed-V2`	7B	29.17	60.94
`gte[MDCure]`	150M	27.60	53.65
`gte[10-Shot]`	150M	31.11	60.50
`gte[ExpertGenQA]`	150M	**36.98**	**77.08**
FAA-UAS (50 QA)			
`gte-baseline`	150M	38.00	76.00
`NVEmbed-V2`	7B	40.00	88.00
`gte[10-Shot]`	150M	42.00	84.00
`gte[ExpertGenQA]`	150M	**48.00**	**88.00**

Table 1: Retrieval performance (in terms of Top$-k$ accuracy) of retrieval LMs and finetuned variants. The second column contains the number of parameters. `gte[X]` means the `AlibabaNLP/gte-modernbert-base` was fine-tuned on synthetic instructions from the respective dataset generated using the `X` pipeline. Both `few-shot` and `ExpertGenQA` pipelines use 10 examples.

written QAs are *human-centric*, often expressed from a first-person perspective (e.g., *"How do I decide if a case is work-related when the employee is working from home?"*). In contrast, questions generated by MDCure tend to focus on specific factual details (e.g., *"Railroad injury and illness reporting conditions?"*). Few-shot prompting and ExpertGenQA, both of which use expert-written QAs as exemplars, produce more complex questions.

A common weakness shared by all three synthetic generation methods is their difficulty in appropriately using domain-specific terminology. For instance, Form 6180.57 ('Highway Incident Report') is well-known among FRA domain experts, and referencing it simply as "Form 57" without further elaboration is common. Conversely, Form 6180.99x ('31 & 92 Service Day Report') is a more niche document, only occasionally used. LLMs lack the domain expertise required to recognize this distinction and generate questions involving niche terminology without necessary clarifications. We speculate that expert-crafted prompts or domain-specific pretraining would help in this regard.

6.2 Reward Models and LLM-as-Judge

We test the ability of state-of-the-art Reward Models (RM) to judge question quality, based on which we compare the quality of expert-written questions with synthetically generated questions using the

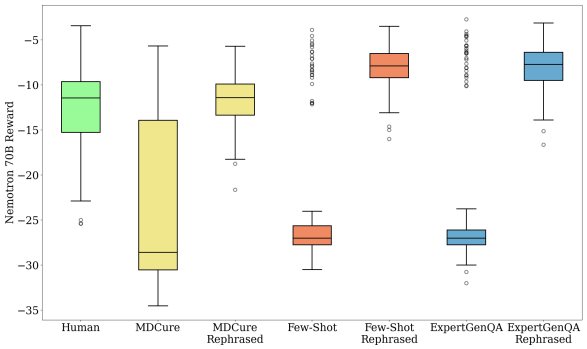

Figure 3: Box plot of reward assigned to FRA synthetic questions by `Llama-3.1-Nemotron-70B-Instruct` Reward Model. Notably, merely rephrasing synthetic questions to sound *human-like* (see App. H.2.3) drastically increases the assigned reward score although the semantic content hasn't changed.

template shown in Appendix H.2.4. We leveraged GPT4o to automatically rephrase synthetic questions that sound human-like using the prompt in Appendix H.2.3. In Fig. 3, we demonstrate the reward scores of human-written questions (see "Human"), synthetic questions using the aforementioned three pipelines, and synthetic questions after rephrasing (see "X Rephrased" where "X" is the corresponding question generation pipeline). Clearly, 1) merely rephrasing LLM generations drastically increases the score awarded by the RM; and 2) synthetic generations with rephrasing achieve higher rewards than expert-written questions. Thus, the results in Fig. 3 imply that `Nemotron-70B-Instruct` RM (Wang et al., 2024d,c) exhibits a strong bias based on writing style rather than content quality. We also observe such bias in another state-of-the-art RM `Skywork-Reward-Gemma-2-27B-v0.2` (Liu et al., 2024a) while using *GPT4o-as-Judge* (see Appendix E). These findings indicate that RMs are highly sensitive to superficial stylistic changes and **do not correlate** with the clear differences between different pipelines shown in Table 1.

6.3 Cognitive Complexity Distribution

To evaluate the cognitive complexity and educational value of generated questions, we leverage Bloom's Revised Taxonomy (Bloom et al., 1956; Anderson and Krathwohl, 2001), a well-established framework from cognitive science that categorizes learning objectives into six hierarchical levels: Remember, Understand, Apply, Analyze, Evaluate, and Create. Each level represents increasingly complex cognitive processes, from basic recall to sophisticated synthesis. We use GPT4o to classify

both human-written and synthetic questions according to these taxonomic levels, allowing us to assess and compare the distribution of cognitive demands across different instruction sets.

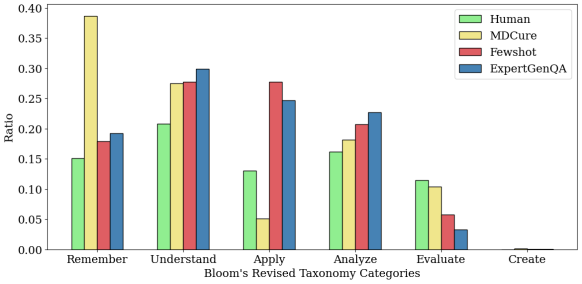

Figure 4: Distribution of cognitive complexity levels in human-written and synthetic instructions according to Bloom's Revised Taxonomy. MDCure shows higher concentration in lower cognitive levels.

Fig. 4 shows the distribution of instructions across Bloom's Taxonomy levels for human-written and synthetic data. MDCure shows a notable skew toward lower-level cognitive tasks, with approximately 39% of instructions falling into the *Remember* category. The distribution of human-written questions demonstrates greater uniformity across cognitive levels, reflecting their origin from domain experts crafting questions for other domain experts. Few-shot prompting and ExpertGenQA produce distributions more closely aligned with human-written questions, emphasizing the value of using expert examples in specialized domains.

6.4 Topic Coverage and Preference Metrics

A key challenge in question generation is ensuring comprehensive coverage of source materials, as missing critical topics could lead to gaps in downstream capabilities. To ensure that generated questions span the full scope of document content, we measure topic coverage: $TC = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}}\frac{|Q(d)|}{|T(d)|}$ where $Q(d)$ represents the topics covered by generated questions for document d, $T(d)$ represents the topics in document d, $\mathcal{D}$ is the document set, and $|\cdot|$ is the set cardinality operator.

Reward models are generally trained to evaluate responses to questions rather than the questions themselves. Yu et al. (2025) has shown that the rewards assigned to responses can be used as a proxy metric for instruction quality. Following their methodology, we sample $N = 10$ responses for each context-question x and evaluate them using a RM. From these responses, we identify the *chosen* response y_w with the highest reward and the *rejected* response y_l with the lowest reward. We analyze three key metrics:

- Rejected response reward $\texttt{RM}(y_l|x)$; higher is better
- Rejected response length ratio $\frac{\texttt{len}(y_l)}{\texttt{len}(x)}$; higher is better
- Reward gap $\Delta\texttt{RM}(\cdot) = \texttt{RM}(y_w|x) - \texttt{RM}(y_l|x)$; lower is better

The intuition behind these metrics, as demonstrated by Yu et al. (2025), is that high-quality instructions should produce longer and more coherent responses even when they are "rejected" and should lead to more consistent response quality.

Pipeline	#Questions	$TC\uparrow$	$\texttt{RM}(y_l\|x)\uparrow$	$\frac{\texttt{len}(y_l)}{\texttt{len}(x)}\uparrow$	$\Delta\texttt{RM}(\cdot)\downarrow$
MDCure	8,030	0.626	-8.67	0.27	**4.38**
Few-Shot	3,658	0.726	-7.87	0.59	5.15
EGenQA	7,140	**0.944**	**-7.75**	**0.61**	5.05

Table 2: Comparison of pipelines across topic coverage (TC) and response preference metrics. ExpertGenQA (EGenQA) achieves the best scores in TC, rejected response quality $\texttt{RM}(y_l|x)\uparrow$, and rejected response length ratio $\frac{\texttt{len}(y_l)}{\texttt{len}(x)}\uparrow$.

Table 2 reveals interesting trade-offs between the three approaches. ExpertGenQA has the highest rejected response reward, rejected response length ratio, and topic coverage even after filtering, highlighting the effectiveness of the ExpertGenQA generation protocol. Further investigation into the reward gap $\Delta\texttt{RM}(\cdot)$ reveals an interesting pattern when analyzed alongside Bloom's Taxonomy levels, as shown in appendix F table 6. While MDCure achieves the lowest reward gap despite generating simpler questions, this appears to be a natural consequence of its approach - simpler questions tend to elicit more consistent responses from LLMs, resulting in smaller reward gaps. From the four metrics, topic coverage strongly correlates with the retrieval performance in Table 1.

7 Conclusion

This work introduces ExpertGenQA, a protocol for combining structured categorization with few-shot learning to generate high quality domain-specific questions. Our evaluation reveals limitations in current automated assessment methods: both reward models and LLM-as-judge approaches struggle to meaningfully evaluate technical content quality. The cognitive complexity

analysis shows ExpertGenQA better preserves the distribution of expert-level thinking demands compared to template-based methods. Retrieval LMs using ExpertGenGA achieves improved retrieval performance even compared to the much larger `NVEmbed-2-70B`, although the modest absolute performance highlights ongoing challenges in technical domain retrieval. In the future, we will extend this approach to other specialized fields where expert knowledge is crucial but limited.

8 Limitations

Our study has several limitations. Firstly, we evaluate only on the Federal Railway Administration and Federal Aviation Administration domains because they offered well-structured corpora of regulatory documents with expert-written FAQs, making them ideal testing grounds for our approach. However, Our proposed pipeline ExpertGenQA and evaluation metrics should be effective in other specialized domains as well. We leave this as future work.

Secondly, while ExpertGenQA significantly improves retrieval performance compared to baselines, the best top-1 accuracy remains below 40%. Scaling up synthetic data generation is a promising direction for achieving practically viable performance levels.

Finally, the few-shot prompting component of ExpertGenQA, while effective for quality, incurs substantial compute costs in terms of token usage during generation. Future research could explore optimizing the efficiency-quality tradeoff.

9 Ethical Considerations

This work focuses on generating high-quality question-answer pairs for specialized technical domains. We acknowledge the following ethical considerations:

- **Data Source and Copyright:** We used publicly available U.S. Federal Railroad Administration (FRA) documents as a case study. While these documents are in the public domain, it's important to recognize that not all information on the internet is free for unrestricted use. In this work, we processed the PDF documents using the `pymupdf4llm` library, adhering to its intended use and licensing terms.

- **Risk of Data Poisoning:** While our current work uses a curated set of official FRA documents, extending this approach to less controlled environments introduces the risk of data poisoning. Malicious actors could intentionally introduce incorrect or misleading information into the source documents used for question generation. This could lead to the generation of inaccurate or biased question-answer pairs, ultimately impacting the reliability of downstream applications like retrieval systems.

- **Ensuring Trustworthy Information:** The primary goal of this work is to improve information access and knowledge assessment for domain experts. However, there is a risk that errors in the generated questions or retrieved information could lead to incorrect conclusions or decisions by these experts. Ensuring the accuracy and reliability of the generated content is crucial for building trustworthy AI systems.

We believe that the benefits of this research, particularly in providing more efficient access to critical information in specialized domains, are substantial. However, we emphasize the importance of responsible development and deployment, with careful consideration of data quality, potential risks, and the need for ongoing validation to ensure trustworthy and reliable results.

10 Acknowledgements

This work was supported by the University Transportation Center for Railway Safety (UTCRS) at UTRGV through the USDOT UTC Program under Grant No. $69A3552348340$ and NSF CREST Center for Multidisciplinary Research Excellence in Cyber-Physical Infrastructure Systems (MECIS) grant no. 2112650. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding parties.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Flora Amato, Egidia Cirillo, Mattia Fonisto, and Alberto Moccardi. 2024. Optimizing legal information access: Federated search and rag for secure ai-powered

legal solutions. In *2024 IEEE International Conference on Big Data (BigData)*, pages 7632–7639.

Bang An, Shiyue Zhang, and Mark Dredze. 2025. Rag llms are not safer: A safety analysis of retrieval-augmented generation for large language models. *Preprint*, arXiv:2504.18041.

Lorin W Anderson and David R Krathwohl. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, Jiayin Zhang, Juanzi Li, and Lei Hou. 2023. Benchmarking foundation models with language-model-as-an-examiner. *ArXiv*, abs/2306.04181.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv preprint arXiv:2408.07055*.

Benjamin S Bloom et al. 1956. Taxonomy of educational objectives, handbook i: Cognitive domain. new york: David mckay company. *Inc., 1956. 207 pp*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiuhai Chen, Rifaa Qadri, Yuxin Wen, Neel Jain, John Kirchenbauer, Tianyi Zhou, and Tom Goldstein. 2024. Genqa: Generating millions of instructions from a handful of prompts. *ArXiv*, abs/2406.10323.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Xiaotong Ji, Shyam Sundhar Ramesh, Matthieu Zimmer, Ilija Bogunovic, Jun Wang, and Haitham Bou Ammar. 2025. Almost surely safe alignment of large language models at inference-time. *Preprint*, arXiv:2502.01208.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Preprint*, arXiv:1909.06146.

Sahil Kale, Gautam Khaire, and Jay Patankar. 2024. Faq-gen: An automated system to generate domain-specific faqs to aid content comprehension. *ArXiv*, abs/2402.05812.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

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.

Yangning Li, Shirong Ma, Xiaobin Wang, Shen Huang, Chengyue Jiang, Hai-Tao Zheng, Pengjun Xie, Fei Huang, and Yong Jiang. 2024. Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18582–18590.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024a. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Gabrielle Kaili-May Liu, Bowen Shi, Avi Caciularu, Idan Szpektor, and Arman Cohan. 2024b. Mdcure: A scalable pipeline for multi-document instruction-following. *ArXiv*, abs/2410.23463.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Yida Mu, Mali Jin, Xingyi Song, and Nikolaos Aletras. 2024. Enhancing data quality through simple de-duplication: Navigating responsible computational social science research. *ArXiv*, abs/2410.03545.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, pages 1–8.

Sarvesh Soni and Kirk Roberts. 2021. An evaluation of two commercial deep learning-based information retrieval systems for covid-19 literature. *Journal of the American Medical Informatics Association*, 28(1):132–137.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hieu Tran, Zhichao Yang, Zonghai Yao, and Hong Yu. 2024. Bioinstruct: Instruction tuning of large language models for biomedical natural language processing. *Preprint*, arXiv:2310.19975.

Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2024a. Instructretro: Instruction tuning post retrieval-augmented pretraining. *Preprint*, arXiv:2310.07713.

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

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023b. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *arXiv preprint arXiv:2310.04793*.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024b. Direct judgement preference optimization. *Preprint*, arXiv:2409.14664.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. *Preprint*, arXiv:2212.10560.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2024c. Helpsteer2-preference: Complementing ratings with preferences. *Preprint*, arXiv:2410.01257.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024d. Helpsteer2: Open-source dataset for training top-performing reward models. *ArXiv*, abs/2406.08673.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. 2024. Followir: Evaluating and teaching information retrieval models to follow instructions. *Preprint*, arXiv:2403.15246.

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*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *Preprint*, arXiv:2304.12244.

Ran Xu, Hui Liu, Sreyashi Nag, Zhenwei Dai, Yaochen Xie, Xianfeng Tang, Chen Luo, Yang Li, Joyce C Ho, Carl Yang, et al. 2024a. Simrag: Self-improving retrieval-augmented generation for adapting large language models to specialized domains. *arXiv preprint arXiv:2410.17952*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024b. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *Preprint*, arXiv:2406.08464.

Ping Yu, Weizhe Yuan, Olga Golovneva, Tianhao Wu, Sainbayar Sukhbaatar, Jason Weston, and Jing Xu. 2025. Rip: Better models by survival of the fittest prompts. *arXiv preprint arXiv:2501.18578*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *Preprint*, arXiv:2309.05653.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*.

Lingjun Zhao, Khanh Nguyen, and Hal Daum'e. 2023. Hallucination detection for grounded instruction generation. In *Conference on Empirical Methods in Natural Language Processing*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zheng Liu, Ji-Rong Wen, and Zhicheng Dou. 2024. Inters: Unlocking the power of large language models in search with instruction tuning. *Preprint*, arXiv:2401.06532.

A Target Domains

A.1 Federal Railway Administration (FRA)

The U.S. railway system operates primarily under private ownership, with freight railroads owned and operated by corporations such as Union Pacific, BNSF, and CSX. These companies handle a significant portion of the nation's freight transportation, moving over a third of goods by ton-miles. Passenger rail services, which are much more limited in scope, include Amtrak (a federally supported corporation) and various regional commuter systems such as Metrolink, BART, and SEPTA. Most passenger rail services operate on infrastructure owned and maintained by private freight railroads, creating a complex system of shared use that requires extensive oversight and coordination.

The Federal Railroad Administration (FRA), part of the U.S. Department of Transportation (USDOT), serves as the primary federal agency responsible for regulating and supporting this privately managed rail system. The FRA's role includes developing and enforcing safety standards for infrastructure, rail equipment, operations, and employee working conditions. Its inspectors ensure compliance across the industry, enforce safety mandates, and investigate accidents to improve future practices.

In addition to enforcing safety standards, the FRA administers funding programs, such as the Consolidated Rail Infrastructure and Safety Improvements (CRISI) grant initiative, which supports infrastructure modernization, capacity improvements, and the implementation of new technologies. The agency also collects and distributes data on accident trends, track performance, and operator compliance, providing essential insights for railroads, policymakers, and the public to guide decision-making and planning. The FRA maintains an extensive online repository of regulatory and informational documents through its eLibrary to support industry stakeholders, researchers, and the public. The eLibrary [6] contains more than 9000 documents spanning from 1966 to the present.

In this work, We have curated 43 up-to-date documents from the FRA eLibrary based on the following criteria: sufficient textual content and expert-written QA pairs that are not tied to specific events or overly focused on temporary or local programs. A significant portion of the qualified QA pairs comes from the *Federal Railroad Administration Guide for Preparing Accident/Incident Reports*, which provides comprehensive regulatory explanations and practical QA pairs for each section. Additional sources include FAQs and QA-focused documents covering topics such as workers, programs, operations, and services. Examples include *Questions and Answers Concerning Wheelchairs and Bus and Rail Service* and *RCL Operations Q&As*.

A.2 Federal Aviation Administration (FAA)

The Federal Aviation Administration (FAA) regulates nearly all aspects of civil aviation in United States airspace, encompassing commercial airlines, private and recreational flights, general aviation, and unmanned aircraft systems (UAS). It administers aircraft certification and pilot licensing, oversees air traffic control, and enforces operational safety standards. Certain aviation operations—such as military flights—fall under the jurisdiction of other agencies, including the Department of Defense, the Department of Homeland Security, and the National Park Service. The FAA's statutory authority derives from Title 49 of the United States Code, while its operational rules are codified in Title 14 of the Code of Federal Regulations. Although the FAA establishes overarching airspace regulations, enforcement and local restrictions often involve inter-agency collaboration depending on the context.

This work focuses on FAA regulations governing UAS, commonly known as drones. UAS operations in U.S. airspace are governed by Title 14 CFR Part 91 (General Operating and Flight Rules) and Part 107 (Small Unmanned Aircraft Systems), as well as Public Law 115–254 (FAA Reauthorization Act of 2018). Exemptions are specified in 49 U.S.C. § 44809 (Exception for Limited Recreational Operations of Unmanned Aircraft) and 49 U.S.C. § 44807 (Special Authority for Certain UAS, replacing Section 333 exemptions under Public Law 112–95).

To build our QA dataset, we initially collected 72 FAQs[7] from the FAA's online portal related to UAS regulations. For each item, we used the official answers to locate the corresponding regulatory passages

[6] https://railroads.dot.gov/elibrary-search

[7] https://www.faa.gov/faq

that substantiate them. Twenty-two FAQs lacked clear supporting text and were therefore excluded, leaving 50 human-authored questions paired with their regulatory citations. Our seed corpus for QA generation totals 601 pages, drawn from the regulations listed above, with the exception of Part 91, which does not address UAS-specific requirements.

B Diversity and Efficiency

Strategy	#Shots	#LLM()	#Unique	Efficiency
FRA Domain				
MDCure	0	51,100	8,030	15.71%
Few-shot	0	17,400	788	4.53%
	1	17,400	1,220	6.90%
	5	17,400	2,778	15.96%
	10	17,400	3,658	21.02%
ExpertGenQA	0	17,622	2,035	11.55%
	1	24,030	2,584	10.75%
	5	19,224	5,355	27.86%
	10	17,622	7,140	40.52%
FAA-UAS Domain				
Few-shot	1	11,400	673	5.90%
	10	11,400	2,967	26.03%
ExpertGenQA	1	12,338	1356	10.99%
	10	11,331	4,348	38.37%

Table 3: Efficiency of different generation pipelines. #Shots denotes the number of few-shot examples used, #LLM() is the number of LLM calls, #Unique is the number of questions left after deduplication, and Efficiency denotes the ratio of unique questions over total LLM calls used.

C Data Generation with MDCure

MDCure (Liu et al., 2024b) is a pipeline for generating question-answers from single or multiple documents in a zero-shot setting. After generation, it uses the MDCure Reward Model (MDCureRM) to filter the generations. MDCure uses three categories of prompts to encourage generation diversity: generic, template-based, and snippet-based. MDCure first clusters documents by their embeddings. For each cluster, generic prompts ask the model to generate questions requiring all the cluster documents to answer. Template-based prompts are constructed by randomly combining restrictions on the question and answer such as question type (summarization, paraphrasing, inference, etc.), answer length, and question style (declarative, imperative, etc.). Finally, snippet-based prompts work on similar pairs of documents instead of clusters. MDCure first extracts random snippets from each document and prompts a model to generate a question and answer based on the two snippets.

We generate 5170 QAs using generic prompts, 14300 with template-based prompts, and 31080 with snippet-based prompts from our FRA documents. For a fair comparison with ExpertGenQA, We sample 5 completions per prompt. We use MDCureRM to score the generations and keep the top 50% generations by score. Similar to ExpertGenQA, we further filter near-duplicates by word overlap. The complete pipeline yields 8030 QA pairs from 51100 sampled generations.

D ExpertGenQA Ablation

In this section, we study the effect of topic categorization and style categorization in isolation. Table 4 shows that Topic Categorization boosts efficiency by 4% over random few-shot prompting. In contrast, Style Categorization has minimal benefits when the number of examples is low since the LLM cannot grasp the correct style with fewer examples. As the number of examples is increased, the LLM can better understand the style of the examples, the gain afforded by Style Categorization gets more pronounced. Finally, combining the two forms of categorization into the complete ExpertGenQA pipeline has compounding effects.

#Shots	Strategy	#LLM()	#Unique	Efficiency
1	Random	17,400	1,220	6.90%
	Topic	15,905	1645	10.34%
	Style	17,400	1,300	7.47%
	Topic+Style	24,030	2,584	10.74%
5	Random	17,400	2,778	15.96%
	Topic	15,905	2,897	18.84%
	Style	17,400	3,172	18.22%
	Topic+Style	19,224	5,355	27.86%
10	Random	17,400	3,658	21.02%
	Topic	15,905	4,101	25.78%
	Style	17,400	4,721	27.13%
	Topic+Style	17,622	7,140	40.52%

Table 4: Ablation study of the two major components of ExpertGenQA: topic categorization and style categorization. #Shots denotes the number of few-shot examples used, #LLM() is the number of LLM calls, #Unique is the number of questions left after deduplication, and Efficiency denotes the ratio of unique questions over total LLM calls used.

E Reward Models and LLM-as-Judge

	Human	MDCure	FewShot	ExpertGenQA
Relevance	4.44	4.18	4.49	**4.48**
Coherence/Factuality	4.23	4.19	**4.33**	4.31
Creativity	2.99	3.13	3.11	**3.16**
Context Integration	3.32	3.47	**3.48**	3.43
Intra-doc Relations	3.62	3.58	**3.81**	3.66
Complexity	3.32	3.46	3.44	**3.52**

Table 5: Fine-grained scores assigned by `GPT4o-as-Judge` using the MDCure prompt (Liu et al., 2024b) on the FRA domain. The best score for each metric is in bold. The weighted-average score is shown in Fig. 5. We use the weights proposed by MDCure.

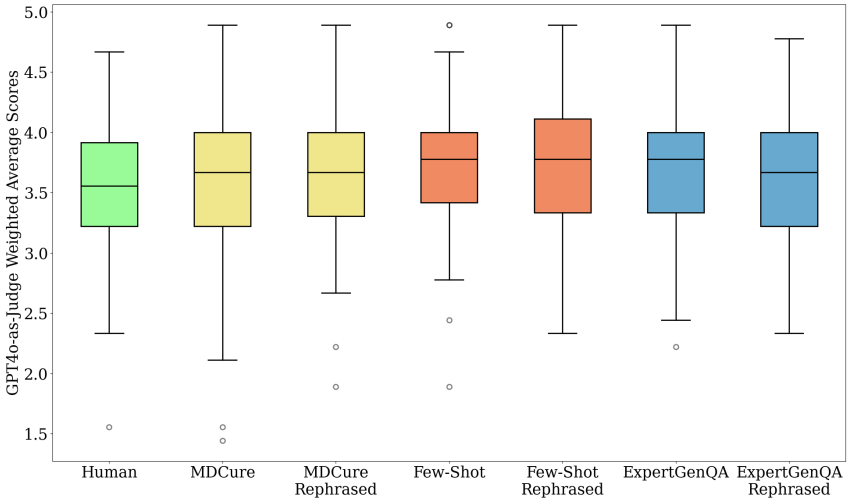

Figure 5: Box plot of scores assigned by `GPT4o-as-Judge` using the MDCure prompt (Liu et al., 2024b) on the FRA domain. `GPT4o-as-Judge` assigned similar scores for all generation methods and hence does not correlate with the clear differences in downstream task improvements shown in Table 1.

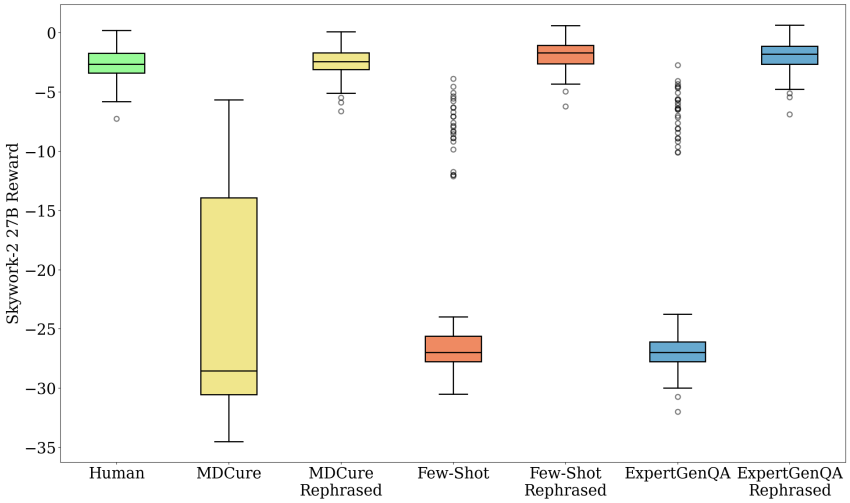

Figure 6: Box plot of reward assigned by `Skywork-Reward-27B` Reward Model on the FRA domain. Merely rephrasing synthetic instructions to sound *human-like* drastically increases the assigned reward showing that RMs are not suitable for judging synthetic instruction quality.

F Evaluation via Response Generation

Level	$\mathbf{RM}(\mathbf{y_l}\|\mathbf{x}) \uparrow$			$\frac{\mathbf{len}(\mathbf{y_l})}{\mathbf{len}(\mathbf{x})} \uparrow$			$\mathbf{\Delta RM}(\cdot) \downarrow$		
	MD	FS	EX	MD	FS	EX	MD	FS	EX
Remember	**-7.77**	-8.29	-7.84	0.20	**0.56**	0.40	**3.79**	5.23	4.62
Understand	-9.66	**-6.89**	-7.52	0.31	0.55	**0.59**	5.00	**4.03**	4.84
Apply	-9.52	-8.21	**-7.77**	0.35	0.66	**0.80**	6.08	5.56	**5.40**
Analyze	**-7.37**	-8.26	-8.23	0.35	0.59	**0.69**	**3.23**	5.83	5.76
Evaluate	-11.55	-8.50	**-5.91**	0.17	**0.56**	0.44	6.03	6.23	**3.16**
Average	-8.67	-7.87	**-7.75**	0.27	0.59	**0.61**	**4.38**	5.15	5.05

Table 6: Comparison of response preference metrics against Bloom's Taxonomy on the FRA domain. MD: MDCure, FS: FewShot, EX: ExpertGenQA. The best performance is in bold.

G Safety Risks in Expert Domains

Safety and trustworthiness of LLM generations is an expansive topic with several tentative approaches, such as reinforcement learning (RL) safety training (Bai et al., 2022), training a critic on internal states (Ji et al., 2025), and employing retrieval-augmented generation (RAG) (Amato et al., 2024). However, this remains an active field of research, as An et al. (2025) demonstrate that RAG LLMs are not necessarily safer. Given these challenges, a comprehensive exploration of LLM safety across specialized domains lies beyond the scope of our work; instead, we outline key risks that arise when applying LLMs to expert contexts.

H Qualitative Examples

FRA Domain - Expert-written Questions (Randomly Sampled)

Policy Application

1. Our employees are frequently tested for drug or alcohol use after an accident/incident. Company policy prohibits an employee from returning to work until the results of the tests are known and it is established that there is no risk factor due to impairment. Must we make a report because of the days the employee was held out of service while awaiting test results?
2. Our employees are frequently tested for drug or alcohol use after an accident/incident. Company policy prohibits an employee from returning to work until the results of the tests are known and it is established that there is no risk factor due to impairment. Must we make a report because of the days the employee was held out of service while awaiting test results?
3. How do I decide if a case is work-related when the employee is working at home or telecommuting from another location?

Scenario-based

4. If the injured or ill worker produces fewer goods or services than he or she would have produced prior to the injury or illness, but otherwise performs all of the routine functions of his or her work, is the case considered a restricted work case?
5. Say that a highway user struck a signal stand at a highway-rail grade crossing and was injured, but there was no on-track equipment present, nor were employees of the railroad in the vicinity. Is this reportable?
6. One of our employees experienced minor musculoskeletal discomfort. The health care professional who examined the employee only provided first aid treatment. In addition, it was determined that the employee is fully able to perform all of her routine job functions. When the employee returned to work, we decided to limit the duties of the employee for the purpose of preventing a more serious condition from developing. Is this a restricted work case?

Terminology Clarification

7. Is a physical therapist considered a "health care professional" under the definition of health care professional?
8. Removing splinters or foreign material from areas other than the eye by irrigation, tweezers, cotton swabs, or other simple means... What are "other simple means" of removing splinters that are considered first aid?
9. What does "other potentially infectious material" mean?

FRA Domain - Synthetic Instructions from MDCure (Randomly Sampled)

1. How do the reporting requirements for railroad accidents and incidents ensure timely and accurate accountability while also protecting the rights of employees involved, particularly in cases where human factors are cited as a cause?
2. What are the requirements for a written request to treat subsidiary railroads as a single, integrated railroad system, and what does railroad transportation encompass according to the regulations?
3. What are the specific reporting criteria and procedures for railroads regarding suicide data, as well as exceptions related to injuries or illnesses incurred by employees, contractors, and volunteers?
4. Can a person who is not on railroad property be involved in railroad operations?
5. What are the reporting criteria for workplace injuries and the investigation procedures for rail accidents regarding substance use?
6. Railroad injury and illness reporting conditions?
7. What must be submitted for FRA review?
8. How does the categorization of accidents and the reporting thresholds relate to the documentation requirements for rail equipment incidents and worker injuries within the railroad industry?
9. Where to download FRA forms and guide?
10. What are the primary purposes of Part 225 regulations compared to the applicability restrictions outlined in § 225.3?

FRA Domain - Synthetic Instructions from 10-shot Prompting (Randomly Sampled)

1. If an accident involves hazardous materials but no evacuation was necessary, should the number of people evacuated still be reported as "0," or is it considered not applicable?
2. If a volunteer railroad worker is injured while performing safety-sensitive functions, does that injury require reporting under FRA regulations?
3. If a railroad operates another company's freight train and runs a total of 1,000 miles with its crew during the month, should those miles be reported in the total for the operating railroad or the railroad that owns the freight train?
4. In the situation where an employee broke their arm during a physical altercation with a coworker in the company parking lot before clocking in for work, is there a justification for classifying this injury as non-work-related, or must it be reported as a work-related incident?
5. Are incidents involving damage to idle railroad cars due to vandalism by nonrailroad employees subject to reporting if there is no involvement of railroad employees?
6. If a railroad employee suffers a reportable injury and the railroad receives information about it six days later, what is the latest date by which the railroad must enter that reportable case on the appropriate record?
7. What should a railroad do if they receive an Employee Statement Supplementing Railroad Accident Report after initially filing the Rail Equipment Accident/Incident Report?
8. If a railroad experiences a significant change in their reported damage costs for a rail equipment accident after initially filing a report, what is the percentage variance that would necessitate an amended report?
9. An employee was injured when a heavy object fell on them while they were chatting with a co-worker in the break room. How should we determine if this injury is considered work-related under the FRA guidelines?
10. Are railroads required to include suicide data in their periodic reports to FRA, and if not, how is such data handled?

FRA Domain - Synthetic Instructions from ExpertGenQA (Randomly Sampled)

1. What is the significance of the FRA Guide for Preparing Accident/Incident Reports in relation to Part 225, and how does it serve railroad companies in meeting their recordkeeping and reporting obligations?
2. If an employee tested positive for drug use following an accident and further investigation indicates

that drug use did not impair their ability to perform their job responsibilities, how should this be documented in the accident report narrative? What specific information should be included to clearly explain this determination?
3. In the context of reporting an incident involving a highway user and railroad on-track equipment, how should a railroad handle a situation where a highway user attempted to avoid the incident but was struck at a different location than the crossing?
4. What guidelines must be followed when determining whether a case falls under the exceptions for reporting injuries or illnesses?
5. What types of professionals are classified as "qualified health care professionals," and what does this classification entail regarding their scope of practice?
6. If the employee injured during a smoke break was on a designated break time and the employer has a policy allowing such breaks, would this change the work-relatedness assessment for the slip on ice, leading it to be reportable?
7. What criteria define a "significant injury" or "significant illness" in the context of reporting railroad accidents or incidents?
8. What information is required to be maintained in a railroad's injury and illness record, and can alternative recordkeeping formats be used?
9. What defines occupational tuberculosis in the context of railroad employees?
10. What are the three primary groups into which reportable railroad accidents and incidents are categorized, and what are the specific reporting requirements for each group?

FAA-UAV Domain - Expert-written Questions (Randomly Sampled)

Procedure-related
1. How would I report a drone operator potentially violating the FAA rules or regulations?
2. How does Beyond Visual Line of Sight (BVLOS) currently apply to public safety in terms of waivers and restrictions?
3. How will ATC facilities get in contact with a small UAS or drone operator if there is an issue or problem?
Certification-related
4. After a Part 107 pilot completes the online ALC training course to renew his/her remote pilot currency does the FAA issue a new remote pilot certificate?
5. Will the FAA recognize any previous UAS or drone training I've taken?
6. I don't see an expiration date on my Part 107 remote pilots certificate . Do I have to take a test annually?
Jurisdiction-related
7. Are local government bodies able to set and enforce their own drone regulations above and beyond the FAA?
8. Do the FAA rules and regulations apply to a commercial UAS or drone operations conducted indoors ONLY?
9. Is law enforcement able to fly UASs around airports if they have multiple airports in their jurisdiction and the towers are notified?
Scenario-based
10. I applied for a Section 333 exemption, an exemption under the Special Authority for Certain Unmanned Systems (U.S.C. 44807), or have a pending request for amendment. What do I do?
11. If my registered UAS or drone is destroyed or is sold, lost, or transferred, what do I need to do?
12. My blanket Certificate of Waiver of Authorization (COA) says I can fly a drone at night, but does it have to be in an emergency situation? How do you train for this if you can't fly at night?

H.1 FAA-UAS Domain - Synthetic Instructions from ExpertGenQA (Randomly Sampled)

1. Under what circumstances can communications related to unmanned aircraft systems be disclosed

outside the Department of Homeland Security or the Department of Justice? 2. Can institutions of higher education operate unmanned aircraft systems for recreational purposes, or are there specific guidelines they must follow for educational or research purposes?
3. Is it permissible for a remote pilot to operate more than one small unmanned aircraft at the same time?
4. If I change my name and need to update my remote pilot certificate, what documents do I need to provide with my application?
5. Can test range operators receive federal funding or in-kind contributions from participants to support their research and testing objectives?
6. How will the program for unmanned aircraft test ranges ensure coordination with the Next Generation Air Transportation System?
7. How long must a person who submits a declaration of compliance retain the supporting information used to demonstrate that their small unmanned aircraft meets regulatory requirements?
8. Under what circumstances can the FAA rescind its acceptance of a means of compliance for small unmanned aircraft systems?
9. How can unmanned aircraft systems (UAS) support tribal law enforcement and emergency response activities?
10. How does the Administrator of the Federal Aviation Administration plan to assist Federal civilian Government agencies that operate unmanned aircraft systems in relation to enhancing public health and safety?

H.2 Prompt Template

H.2.1 ExpertGenQA Topic Extraction Prompt

```
Passage: {{PASSAGE}}

-----

Please analyze the given passage and identify its main topics. Provide your response
in JSON format where the key is 'topics' and its value is an array of the main
topic names. For example:

{
'topics': ['topic1', 'topic2', 'topic3']
}
```

H.2.2 ExpertGenQA Generation Prompt

```
Passage: {{PASSAGE}}

-----

The passage above covers the following topics:
{{TOPICS_IN_PASSAGE}}

Generate a question from the passage related to '{{SELECTED_TOPIC}}'.
```

H.2.3 Paraphrasing with Examples - User Instruction

```
<target_question>
{{QUESTION}}
</target_question>

<examples>
```

```
{{EXAMPLES}}
</examples>

Please paraphrase the target question to match the style of the examples. Do not make
any changes that would alter the meaning and change its answer. Do not answer the
question. Respond with only the rephrased question (without any tags).
```

H.2.4 Reward Model Input for Instruction Quality

```
System
A chat between a curious user and an artificial intelligence assistant. The
assistant gives helpful, detailed, and polite answers to the user's questions.

User
Passage: {{PASSAGE}}
-----
Please generate a question from the passage above.

Assistant
{{INSTRUCTION}}
```

A reward model (RM) assigns a single scalar value, i.e. a reward depending on the quality of the assistant response. Ideally, the RM learns to distinguish implicitly desirable properties of the response such as quality, factuality, helpfulness, creativity, etc.