BetterBench: Dynamic Benchmark Generation Paradigm for Reliable and Task-Specific LLM Evaluation

Anonymous ACL submission

Abstract

Evaluating LLMs is challenging due to their extensive pre-training data and rapidly evolving capabilities, rendering traditional static benchmarks outdated or seen during training. Consequently, organizations are often forced to rely on general-purpose proxy benchmarks that fail to capture the nuanced complexities of specialized domains, leading to uncertainty in model selection and deployment. We introduce **BetterBench**, a dynamic benchmark generation framework that enables researchers to generate automated, task-specific benchmarks from any set of documents without manual annotation. To validate our approach, we release `betterbench b1`, a dataset comprising 40 documents from eight specialized domains collected after July 2024, along with over 330,000 generated questions and answers using models like GPT-4, LLaMA 70B, and Mistral Large. Our experiments demonstrate that BetterBench produces reliable, task-specific benchmarks that provide domain-specific insights while maintaining performance directionality with general-purpose benchmarks. We open-source BetterBench and the associated datasets to facilitate further research in automated LLM evaluation.

1 Introduction

Historically, machine learning model evaluation has been straightforward, epitomized by the use of pre-defined datasets like MNIST (Deng, 2012). However, the advent of large language models (LLMs) trained on massive web-scale datasets introduces significant challenges in constructing benchmarks that remain both challenging and unseen by the model. Even meticulously curated benchmarks can quickly become invalid as they integrate into the training data of future models, leading to inflated performance scores and misrepresenting advancements (Sainz et al., 2023; Zhang et al., 2024). Furthermore, static, human-written benchmarks quickly become outdated due to the rapid progress of LM capabilities (Ruder, 2024), necessitating repeated effortin developing newer benchmarks.

Human-written benchmarks often fail to capture nuanced complexities of specialized domains, resulting in uncertainty in model selection and deployment for specialized scenarios. For organizations or researchers with access to vast troves of task-specific or confidential data, the absence of suitable benchmarks forces reliance on general-purpose proxies with limited real-world guarantees. In this work, we ask: **Can we move beyond static, general-purpose benchmarks to develop dynamic approaches for evaluating LLMs?**

To explore the trajectory of LLM benchmark evolution, we look to the history of test automation in software engineering. Initially reliant on manually crafted tests, the 1970s saw the advent of automated test systems, culminating in the sophisticated, automated testing and benchmarking frameworks of today (Brooks, 2024). Despite the complexity inherent in LLM systems, we hypothesize that a similar trajectory toward automation and reliability in benchmark generation is both natural and necessary.

A primary challenge in dynamic benchmark generation by models is their susceptibility to unreliability and hallucination. However, akin to the BLEU metric in machine translation (Papineni et al., 2002), we propose focusing on the directionality of performance between models rather than definitive scores. For instance, *can we determine that one model outperforms another in specialized domains such as medicine or finance? Can we prove that a fine-tuning process is effective based on directional improvement over a base model? Or can we assess the superiority of certain Retrieval Augmented Generation (RAG) systems dynamically and task-specifically?*

We introduce **BetterBench**, a dynamic benchmark generation framework that enables researchers to zero-shot generate automated benchmarks for large language models from any set of documents. Unlike prior approaches that suffered from high rates of hallucination due to singular models, simplistic prompting and lack of structure (Krishna et al., 2024), BetterBench incorporates

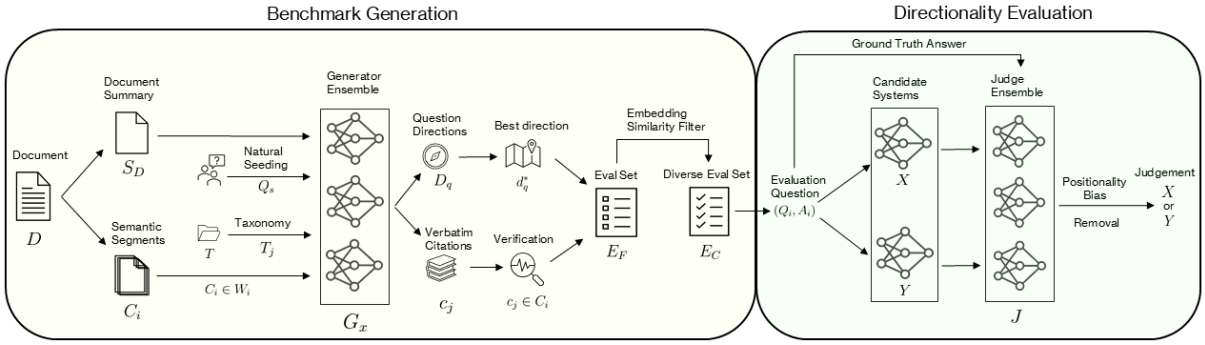


Figure 1: **Generation and Evaluation:** Overview of the BetterBench

multiple guardrails to enhance viability and reliability in benchmark generation.

Central to our methodology is the use of an ensemble of models for both question-answer generation and evaluation, thereby mitigating the directional biases inherent in any single language model. We develop a comprehensive taxonomy of ten distinct question types, graded based on verifiability and subjectivity, to assess various aspects of language model capabilities.

We conduct extensive experiments with state-of-the-art model families, including GPT, LLaMA, Mistral, and Qwen, performing preliminary scaling analyses to assess reliability in evaluation set generation. Our findings provide reliability guarantees and identify optimal trade-offs between computational cost and benchmark quality.

To facilitate these experiments, we curate a new dataset, `betterbench b1`, consisting of 40 documents from eight specialized domains collected post-July 2024 to ensure novelty. Using our taxonomy and a range of models—including GPT-4, Mistral Large, and LLaMA 70B—we generate over 330,000 questions and answers across ten question types. Additionally, we release a raw dataset containing over 1 million inference records encompassing question generation, answer generation, fidelity verification, and judging, utilizing state-of-the-art models. This dataset supports further research in automated benchmarking. This comprehensive dataset supports further research in automated benchmarking and is available both in full and as a condensed, semantically deduplicated version for public use.

Our contributions are summarized as follows:

- We propose and open-source **BetterBench**, a benchmark generation framework that enables zero-shot generation of task-specific benchmarks from document sets.

- We release `betterbench b1` and auxiliary datasets—including over 1 million inference records and over 330,000 generated questions and answers across ten question types—to support experiments in automated evaluation suite generation. [1]

- We provide comprehensive analyses demonstrating the effectiveness of BetterBench in producing reliable benchmarks, including scaling experiments, reliability guarantees, and optimality points regarding the trade-off between reliability and computational cost.

- We validate our approach through experiments with Retrieval Augmented Generation (RAG) systems, showing that performance directionality with general-purpose benchmarks is maintained.

2 Methods

The following section delineates our evaluation set generation process.

2.1 Preprocessing

Generating relevant and insightful questions from a document necessitates providing the question generator G with adequate context. To this end, we first generate a concise summary of the document, denoted as S_D, where D represents the input document. This summary serves as a high-level overview, enabling the LM to grasp the core themes and concepts within D. Subsequently, we segment D into distinct units of information, denoted as C_i, where $i \in \{1, ..., N\}$ represents the chunk index and N is the total number of chunks.

This chunking strategy is motivated by several factors. First, it allows for focused evaluation of the LM's question generation capabilities on smaller, manageable units, facilitating more reliable assessment of question quality and relevance. Second, it mitigates the risk of the LM's attention mechanism being dominated by a few salient parts of the document, ensuring broader coverage and representation of the information contained in D. This targeted approach not only enhances the quality of generated questions but also improves the

[1] All datasets are available anonymously at https://huggingface.co/anon-betterbench

overall efficiency of the question generation process. Thirdly, it enhances the ability of human reviewers to spot hallucinations, as lesser context is necessary to determine if a question is valid or invalid.

2.2 Question-Answer Pair Generation

Our methodology for generating question-answer pairs leverages context windows centered on document chunks. For each focus chunk C_i, we use surrounding chunks C_{i-1} and C_{i+1} to form a context window $W_i = \{C_{i-1}, C_i, C_{i+1}\}$, supplemented by a summary S_D of the document.

The generation process begins with a pre-stage reasoning phase $\mathcal{R}_i$, where the model G processes and understands W_i and S_D:

$$\mathcal{R}_i = \mathbb{E}_{C \in W_i \cup S_D}[G_{think}(C)]$$

Subsequently, G proposes potential question directions D_q and selects an optimal direction d_q^*:

$$d_q^* = \arg\max_{d \in D_q} P(d \mid J)$$

where J ensures alignment with seed questions Q_s.

Questions Q and answers A are generated strictly from the focus chunk C_i, with citations $\mathcal{C}$ extracted to maintain textual fidelity:

$$\mathcal{C} = \{c_1, c_2, \ldots, c_m\} \quad \text{where } c_j \in C_i$$

Lastly, the generated pairs are validated using a citation match rate ρ, ensuring the relevancy and accuracy of the content:

$$\rho = \frac{\sum_{j=1}^{m} \delta(c_j \in C_i)}{m} \times 100$$

Please refer to Appendix D for more comprehensive details on the QA generation steps and additional computational specifics.

2.3 Embedding Similarity Filter

We introduce an Embedding Similarity Filter to efficiently semantically deduplicate our generated evaluation set. Given a dataset $\mathcal{E}_{\mathcal{F}} = (q_i, a_i)_{i=1}^{N}$ where q_i represents a question and a_i its corresponding answer, we aim to identify and remove near-duplicate questions while retaining unique content.

Let $e : \mathcal{Q} \to \mathbb{R}^d$ be an embedding function that maps questions to d-dimensional vectors. We define the similarity between two questions q_i and q_j as the cosine similarity of their embeddings (Rahutomo et al., 2012):

$$\text{sim}(q_i, q_j) = \frac{e(q_i) \cdot e(q_j)}{|e(q_i)||e(q_j)|} \tag{1}$$

We consider two questions to be near-duplicates if their similarity exceeds a threshold τ. Formally, we define a cluster of near-duplicate questions as:

$$C_k = q_i \in \mathcal{D} \mid \exists q_j \in C_k : \text{sim}(q_i, q_j) \geq \tau \tag{2}$$

We develop an efficient, multi-GPU computation approach to cluster and randomly remove near-duplicates, details of which are provided in Appendix C. We then obtain our condensed evaluation set E_C.

2.4 Judging System

Using LLMs for evaluation presents unique challenges, particularly in their ability to provide reliable numerical scores for model outputs due to inconsistencies and biases in discrete scoring. To address this, we adopt a pairwise comparison approach inspired by the BLEU metric (Papineni et al., 2002), focusing on relative rather than absolute performance.

Scoring. Given a ground truth g and two candidate outputs x and y, the preference of each judge J_i is defined as:

$$J_i(x, y, g) = \begin{cases} 1 & \text{if } x \text{ is preferred over } y, \\ 0 & \text{if } y \text{ is preferred over } x. \end{cases}$$

The total preference for x over y is: $P_x = \sum_{i=1}^{N} J_i(x, y, g)$. The system generating x is considered superior if $P_x > \frac{N}{2}$.

For further details on the scoring model and additional equations, see Appendix E.

2.5 Question-Answer Pair Validation

To ensure the validity and source fidelity of the generated question-answer pairs, we introduce a randomized validation procedure. Let C_i denote the source text chunk, S_D the document summary, Q the generated question, and A the generated answer. We define a set of verifiers $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$, where each verifier $V_j : (C_i, S_D, Q, A) \to \{0, 1\}$ assesses the validity of the question-answer pair based on two criteria: (1) answer containment, i.e., whether the answer A contains information relevant to the question Q, and (2) source traceability, i.e., whether the question Q can be derived solely from the source text chunk C_i. Formally, $V_j(C_i, S_D, Q, A) = 1$ if both criteria are satisfied and 0 otherwise. We employ multiple verifiers to reduce the risk of individual verifier bias.

3 Setup

3.1 Model Choice

In this study, we analyze a comprehensive range of state-of-the-art LLMs, encompassing the Qwen, Llama, Mistral, and GPT families. Model sizes range from 0.5B to 405B parameters. To provide a

more nuanced understanding of the evolution of capabilities with increasing scale, we include all available model sizes within each family. This approach allows for a direct comparison of how specific architectural choices and training methodologies impact performance across different scales within a consistent lineage. Further details regarding the specific models utilized in this study are provided in A.

3.2 Data Collection

To construct the `BetterBench` evaluation corpus, we tasked human data-labelers with collecting documents spanning eight diverse domains: Health, Government, Legal, Financial, Corporate, Research, News, and Miscellaneous. Documents were specifically selected from those created after August 2024 to minimize the risk of data contamination during testing, ensuring a clear separation between training data and evaluation data. All source documents were pre-processed into plain text format from their original source formats. We call this resulting dataset the `betterbench-b1` (referred to as `b1`). The summarization of each document to produce S_D is performed by a long-context LLM, Gemini 1.5 Pro Experimental (Reid et al., 2024). Further information regarding the specific categories within each domain and detailed data collection procedures can be found in Appendix B.

3.3 Segmentation

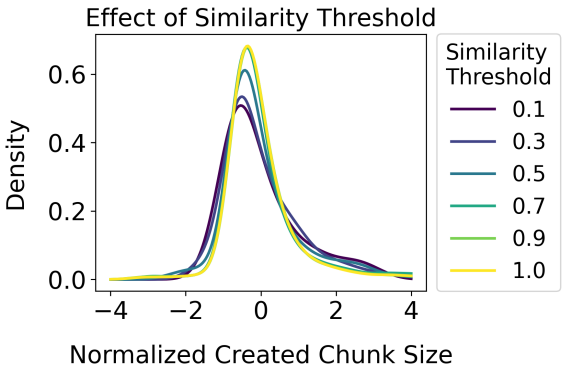


Figure 2: **Effect of Similarity Threshold on Generated Chunk Size**: Lower Thresholds → Larger chunk sizes. (1% outliers removed).

We employ a semantic document chunking strategy to decompose input documents into coherent and manageable units as described in §2.2. Initially, we tokenize the pre-processed document text into sentences utilizing the sentence tokenizer from the NLTK library (Bird, 2006). Subsequently, we compute sentence embeddings using Sentence-BERT (Reimers and Gurevych, 2019a). To identify potential chunk boundaries, we calculate pairwise cosine similarity scores between the embeddings of consecutive sentences. Boundaries are detected when similarity scores fall below a predefined threshold. We construct these chunks ensuring they adhere to specified size constraints and approximate a target chunk size. To optimize chunk creation, we conduct experiments with various similarity thresholds detailed in Figure 2. We observe that lower threshold values result in right-skewed distributions, and choose a value of `0.9` for consistency in token lengths.

3.4 Question Taxonomy

As described in §2.2, the question generation process is governed by a predefined taxonomy $\mathcal{T}$ encompassing 10 distinct question types, $\mathcal{T} = \{T_1, T_2, \ldots, T_{10}\}$, each meticulously designed to assess a wide range of capabilities of the generator G. We structure $\mathcal{T}$ into three levels of evaluation complexity, denoted as $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$, representing increasing degrees of verification difficulty and subjectivity.

Level $\mathcal{E}_1$ includes boolean (true-false) and factual questions characterized by their objective nature and ease of verification, often through standard information retrieval (IR) techniques or basic human evaluation. For all $T_i \in \mathcal{E}_1$, the verifiability $\mathcal{V}(T_i) \geq \theta_v$ and the subjectivity $\mathcal{S}(T_i) \leq \theta_s$, where θ_v and θ_s are predefined thresholds for high verifiability and low subjectivity. Examples in this level include boolean and factual questions.

Level $\mathcal{E}_2$ comprises question types that, while having a single correct answer, present greater challenges in evaluation, such as clarification and conceptual questions. For this level, $\theta'_v \leq \mathcal{V}(T_i) < \theta_v$ and $\theta'_s \geq \mathcal{S}(T_i) > \theta_s$, with θ'_v and θ'_s representing moderate verifiability and subjectivity.

Level $\mathcal{E}_3$ encompasses the most challenging and subjective question types, including edge case, open-ended, counterfactual, and false premise questions. These questions have low verifiability $\mathcal{V}(T_i) < \theta'_v$ and high subjectivity $\mathcal{S}(T_i) > \theta'_s$.

For detailed descriptions and the rationale behind the classification of each question type, please refer to Appendix F.

3.5 Question Generation

To assist with research reproducibility, we choose to use a single set of hyperparmeters and 0 temperature. [2]To ensure the quality and validity of the generated questions, we enforce a structured format during the generation process. We utilize a PyDantic model to define the structure of a valid question, encompassing its various components (e.g., question text, answer choices, difficulty level). The exact PyDantic model schema is detailed in Appendix I. Each LLM is instructed to generate a list of 3-5 questions conforming to this

[2]For increased diversity, we encourage different hyperparameter settings.

structure for each informational chunk, conditional on the chunk containing useful and valid information. This ensures that only relevant and well-formed questions are incorporated into our dataset.

3.6 Embedding Similarity Filter

To apply the filtering described in §2.3 we use the SentenceTransformers (Reimers and Gurevych, 2019b) model `msmarco-MiniLM-L-6-v3` as our embedding function e. We set the similarity threshold $\tau = 0.9$ based on empirical observations, finding it effectively identifies near-duplicates while preserving semantically distinct questions. Post filtering, we retain both a proportional diversity of QA-Pair generating models (Figure 10) and diversity of question types (Figure 9), without any necessity for explicit control.

3.7 Citation Verification during Q&A Generation

To assess the model's capacity for grounding its responses within the provided text, we incorporated a citation verification procedure during the question-answer pair generation process, as detailed in Section 2.2. Prior to the generation of any question-answer pair, the model was tasked with selecting, verbatim, the specific lines from the input text segment that it intended to utilize for formulating a question and providing the corresponding answer. This selection process served as the model's citation for its generated content. Following the model's selection, an algorithmic fuzzy string matching technique was employed to verify the presence of the cited text within the original text segment. We utilized the `partial_ratio` implementation from (Bachmann, 2024), which leverages the Levenshtein Edit Distance Algorithm (Levenshtein, 1965). The match threshold was set to 0.8, a value determined to provide a favorable balance between sensitivity and specificity. This verification step enabled us to evaluate the validity of the model's citations and identify potential instances of hallucination, where the model generates text that is not present in the source material, serving as a leading indicator for invalid questions.

3.8 Question Validity

As detailed in §2.5, the generated questions undergo a thorough validation process to ensure their quality and relevance. To maintain representativeness across the dataset, we employ stratified sampling when constructing the validation set (Singh and Mangat, 1996). The validation process leverages two distinct sets of verifiers: a group of strong LM models and a group of human annotators. This dual verification approach allows for a comprehensive assessment of question validity. To quantify the agreement between the LM models, we utilize Fleiss Kappa (Fleiss, 1981). Agreement between the LM models and the human annotators is measured using Cohen's Kappa (Cohen, 1960). This analysis provides insights into the reliability and consistency of the question validation process. We use GPT-4o (OpenAI et al., 2024) as the top performing closed source model, and Llama-405B (Dubey et al., 2024) as our top performing closed source model, to determine question fidelity.

3.9 Validation of Retrieval Augmented Generation Directionality

To assess the retention of directionality in retrieval-augmented generation (RAG) setups, we conducted experiments using two primary configurations: (a) BM25 (Robertson and Jones, 1976) and (b) BM25 integrated with a reranker (Clavié, 2024).

In the BM25 setup, given a question q, we retrieve the top 3 documents $\{d_1, d_2, d_3\}$ that are most relevant to q based on the BM25 scoring function. Formally, let $\mathcal{D}$ represent the document corpus. The retrieval process is represented as:

$$\{d_1, d_2, d_3\} = \arg\max_{d \in \mathcal{D}} \text{BM25}(d, q)$$

For the BM25 with reranker setup, we first retrieve the top 10 documents $\{d_i\}_{i=1}^{10}$ using BM25. These documents are subsequently re-ranked using the `metarank/ce-esci-MiniLM-L12-v2` model. The reranker assigns a new score to each document, and the top 3 documents are selected based on these scores:

$$\{d'_1, d'_2, d'_3\} = \arg\max_{d \in \{d_i\}_{i=1}^{10}} \text{Reranker}(d, q)$$

To evaluate the efficacy of these retrieval setups, we measure the overlap between the retrieved chunks, denoted C_{r_i}, and the gold chunk C_g, which is considered the most pertinent to answering the question. The evaluation metric $\mathcal{O}$, which quantifies the accuracy of the retrieval, is defined as:

$$\mathcal{O} = \frac{\sum_{i=1}^{3} |C_{r_i} \cap C_g|}{|C_g|}$$

where $|C_{r_i} \cap C_g|$ represents the number of elements common to both C_{r_i} and C_g, and $|C_g|$ represents the total number of elements in the gold chunk.

4 Results

4.1 Question Validity

To ensure the quality and groundedness of our generated question-answer (QA) pairs, we implemented a verification process to determine their validity with respect to the source text. As described in §3.8, this process utilizes both human and automated verifiers to assess whether each QA pair is (1) demonstrably sourced from the corresponding

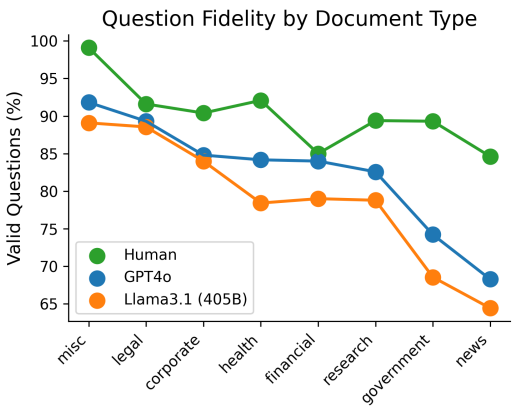


Figure 3: **Domain Distribution of Valid Questions.** Models are more likely to generate valid questions for objective domains

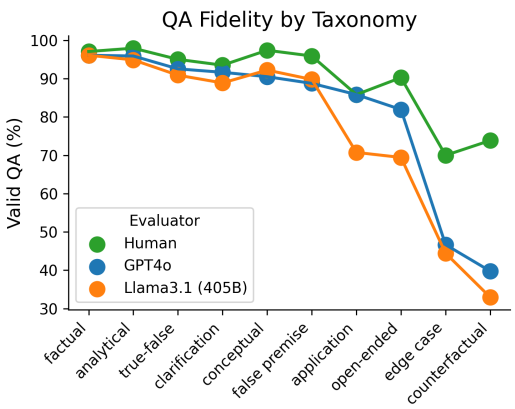


Figure 4: **Taxonomy Type Distribution.** Models tend towards generating "valid" QA pairs when there are objective answers

text chunk and (2) accurately answerable based on the information contained within that chunk. This verification step serves as a crucial control against hallucination, enabling us to quantify the error rate of our QA generation process. By calculating the overall verification percentage and further analyzing the breakdown of valid questions across various subcategories, we gain insights into the factors influencing question quality. Specifically, we examine the relationship between question validity and both document domain and question taxonomy. Our analysis reveals a strong correlation between these variables, indicating that the likelihood of generating a valid question is significantly influenced by the characteristics of the source text and the nature of the question itself. These findings are visually depicted in Figures 3 and 4, which provide detailed insights into the observed patterns.

The reasoning structures, used to determine question fidelity, as well as examples of determination, are listed in Appendix K.2

4.2 Citation Validity

Despite pydantic parsing and validation procedures, the citations generated by the model may be syntactically accurate, but factually inaccurate, i.e. a fabricated reference, indicating a tendency for hallucinated question answer pair generation. To address this concern, as discussed in §3.7, we implemented a citation validation process. Our findings, as detailed in Figure 5, indicate that Qwen 32B and Mistral Small offer a favorable price-to-performance ratio, while GPT4o and Mistral Large 132B emerge as the best performing models overall, with the latter being the top-performing open-source option. The percentage of questions, that contain invalid citations also does not vary with the question taxonomy, implying that model capability is the primaryu contributing factor in this case. Examples of valid and invalid citations are provided in Appendix L.

4.3 Preservation of Directionality

To evaluate the performance of various question-answering techniques, we conducted a comprehensive analysis using four models: Llama3.1 8B and 70B, and Qwen2.5 7B and 72B. Our evaluation set was derived through stratified sampling from a diverse question set generated post-embedding cluster deduplication. A robust benchmark must demonstrate that superior question-answering methods outperform their weaker counterparts consistently. To examine this, we employed multiple retrieval-augmented generation (RAG) techniques (Lewis et al., 2020) across five distinct scenarios, each varying in similarity to its predecessor: Zero Shot, Summary, BM25, Reranker, and Oracle.

In the Zero Shot scenario, the models were presented with only the question, devoid of any contextual information. The Summary scenario provided a summary of the document, excluding the document itself. For BM25, we employed the BM25Okapi method to select the top three relevant chunks, as detailed in 3.9. The Reranker scenario utilized a reranking technique as introduced in (Gao and Callan, 2022), and finally, the Oracle scenario supplied the model with both the document summary and the ground truth chunks. Based on established literature, we anticipated the performance ordering to align as Zero Shot $<$ Summary $<$ BM25 $<$ Reranker $<$ Oracle.

Our findings, as illustrated in Figure 6, validate this hierarchy. We observed that the ensemble judge could easily differentiate methods at either end of this spectrum, though this distinction narrows for intermediate methods. Notably, the win rate of a judge ensemble emerges as a reliable indicator of the technique's superiority. Specifically, BM25 and Reranker techniques exhibited

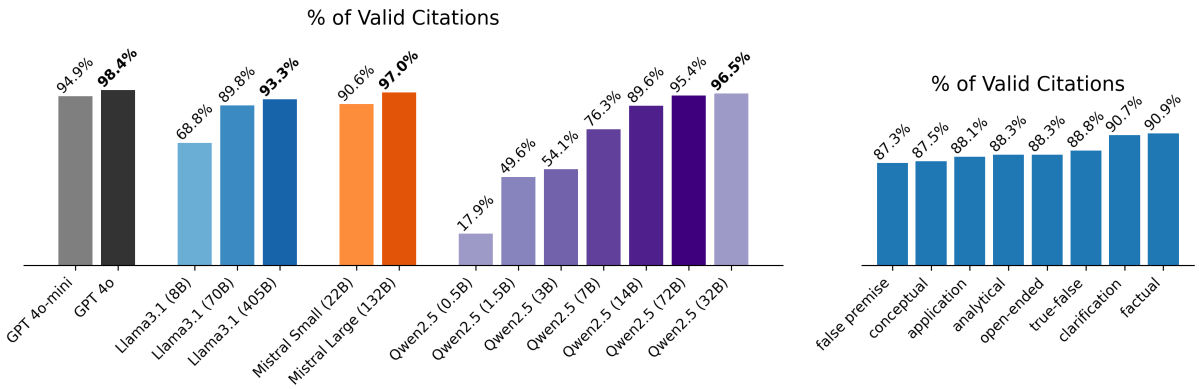


Figure 5: **Correlation of Valid Citation %** by QGen Model and Taxonomy

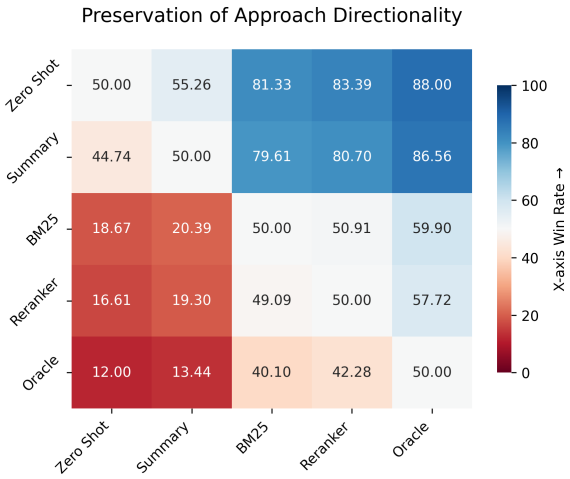


Figure 6: **Stronger Answers win More.** As judged by GPT4o, Mistral Large and Llama3.1 405B, higher quality answers are separable with automated judges.

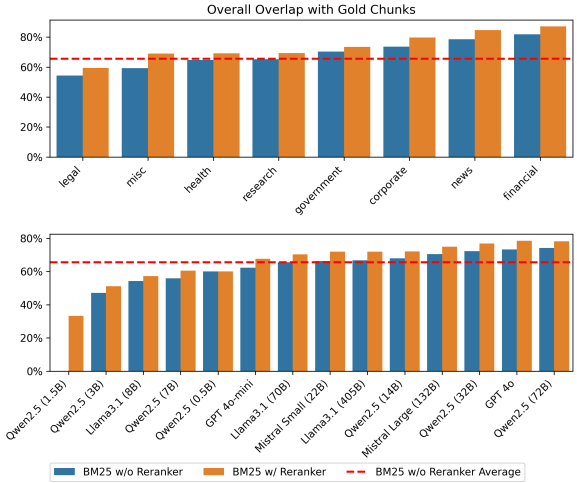


Figure 7: **Mean Retrieval Rates.** Passage Retrieval rate is highly dependent on QA generating model.

closely aligned performances, attributed to their similar zero-shot generation capabilities and information retrieval efficacy (Figure 7). This resemblance underscores the comparable retrieval difficulties presented by our document corpus, with minor improvements in retrievals slightly boosting win rates.

The heatmap in Figure 6 underscores our method's capability to position methods with similar proficiencies near each other while distinctly separating those with significant capability differences. Additionally, an emerging trend indicates that models generating questions easily retrievable by RAGs exhibit lower hallucination rates, maintaining a near-consistent ordering. This insight suggests that these models possess inherent strengths in question generation, facilitating effective retrieval. This phenomena is further investigated in Appendix H

In summary, our results substantiate the premise that stronger question-answering methodologies not only achieve better outcomes but are also discernible via automated judging systems.

5 Related Work

Benchmark Contamination and Evaluation Challenges The issue of benchmark contamination has received increasing attention in recent years as researchers have uncovered significant discrepancies between reported model performance and real-world capabilities. Gupta et al. (2024) demonstrated that simply altering the order of answer options in the widely-used MMLU benchmark (Hendrycks et al., 2021) can dramatically affect model accuracy. This finding raises serious concerns about the robustness of current evaluation methodologies and the extent to which models may be exploiting subtle patterns in benchmark data rather than demonstrating true understanding. Further evidence of benchmark contamina-

tion comes from the work of Zhang et al. (2024), who introduced GSM1K, a manually rewritten version of the popular GSM8K mathematical reasoning benchmark (Cobbe et al., 2021). Their study revealed substantial performance differentials between models that were purported to have similar capabilities based on the original benchmark. These results underscore the pressing need for more reliable and dynamic evaluation appro

Synthetic QA Generation. Recent works, such as Krishna et al. (2024), have attempted to synthetically generate evaluation benchmarks directly with large language models (LLMs) for assessing LLMs. However, these efforts suffer from significant issues of hallucination, with over 30% of their generated questions being hallucinated, thus necessitating extensive manual cleaning. A notable limitation of (Krishna et al., 2024) lies in their use of a rudimentary prompt and reliance on a single unspecified state-of-the-art LLM, thereby restricting the robustness and breadth of the generated questions. Additionally, the absence of a well-defined taxonomy of question types in their methodology limits the creative potential of the LLM, constraining its ability to produce diverse and domain-specific questions. Furthermore, unlike BetterBench, their approach does not generate citations or incorporate additional verification steps for the generated questions, which further undermines the reliability of the synthetic data. Consequently, while they illustrate some promise, these studies underscore the necessity for more sophisticated frameworks, such as BetterBench, which introduces multiple safeguards, an ensemble of models, and a comprehensive question taxonomy to improve the reliability and focus of synthetic benchmark generation. aches that can resist overfitting and provide a more accurate assessment of model performance.

Neural Question Generation While recent work has demonstrated the potential of large language models (LLMs) for generating educational questions, existing approaches often focus on creating factual questions from single documents or short contexts, relying on relatively older language models like T5 (Raffel et al., 2023). For instance, Bulathwela et al. (2023) propose EduQG, a model that leverages pre-training on scientific text and fine-tuning on science question datasets to generate educational questions. However, this approach primarily targets factual questions within a single document and does not address the need for more complex question types, such as multi-hop or open-ended questions, that require higher-level reasoning and synthesis of information across multiple documents. Furthermore, the lack of a dynamic benchmark generation component in EduQG limits its ability to adapt to evolving LLM capabilities and mitigate the risks of overfitting, which is a central concern in evaluating increasingly powerful LLMs.

Knowledge Base Question Generation with Large Language Models Recent work has explored the use of large language models (LLMs) for generating questions from knowledge bases (KBs), particularly in few-shot settings. Liang et al. (2023) introduce KQG-CoT, a framework that leverages CoT prompting to guide LLMs in generating questions from KB logical forms. Their approach demonstrates impressive performance in few-shot scenarios, outperforming several existing methods. However, KQG-CoT focuses specifically on generating questions from structured KB logical forms, which differs from BetterBench's objective of generating benchmarks from arbitrary text documents. Moreover, KQG-CoT primarily generates factual questions, while BetterBench aims to generate a wider range of question types to assess higher-level reasoning capabilities. Furthermore, KQG-CoT does not address the evaluation of answers, which is a crucial aspect of benchmark creation and a central component of BetterBench. Finally, BetterBench explicitly addresses the challenge of benchmark contamination through its relative comparison approach, a concern not explicitly tackled by KQG-CoT.

6 Conclusions and Future Work

In this work, we introduced **BetterBench**, a novel framework for dynamic benchmark generation, designed to enhance the evaluation of large language models (LLMs) by creating automated, task-specific benchmarks from arbitrary documents. Through extensive experiments, we validated the reliability of BetterBench in maintaining performance directionality consistent with general-purpose benchmarks, further solidifying its utility in domain-specific scenarios. We released the `betterbench b1` dataset, along with over 1 million inference records. By open-sourcing BetterBench and the associated datasets, we aim to stimulate further research into automated LLM evaluation and establish new standards for model assessment, ultimately paving the way for more robust and adaptive evaluation methodologies.

Alternative Uses. The benchmark generation stage, which involves the creation of question-answer pairs from given source documents, can be repurposed effectively. This generated data can serve as a training corpus for interactive chatbots designed with question-answering capabilities. While this dual application of BetterBench was not explored within the confines of our current study, future work should investigate this promising avenue.

Limitations

A significant consideration for the practical deployment of BetterBench is the computational complexity associated with utilizing highly capable LLMs for both benchmark generation and evaluation. Our findings indicate that reliable generation necessitates the use of large models, which inherently entail substantial computational overhead. This presents a challenge for researchers and practitioners with limited computational resources. However, recent advancements in techniques like Direct Preference Optimization (DPO) (Rafailov et al., 2023), Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), and knowledge distillation (Hinton et al., 2015) offer promising avenues for mitigating this challenge. These approaches have demonstrated the feasibility of transferring the zero-shot capabilities of larger models into smaller, more efficient counterparts. Similarly, for judge training, techniques such as preference optimization and unsupervised learning of judgment rules, as proposed by Shashidhar et al. (2024), can potentially reduce the reliance on computationally expensive judge models. Future research should explore the effectiveness of these techniques in reducing the computational footprint of BetterBench while maintaining the reliability and quality of the generated benchmarks.

Ethical Considerations

While this work primarily focuses on the technical aspects of benchmark generation, it is important to acknowledge its potential ethical implications. Traditional benchmark creation often relies on extensive human annotation, which can raise concerns about fair labor practices and the potential exploitation of annotators, particularly in contexts with significant economic disparitie. BetterBench, by automating the benchmark generation process, offers a potential solution to mitigate these ethical concerns. By reducing the reliance on manual annotation, the framework minimizes the demand for potentially exploitative labor practices, thereby promoting fairness and ethical data practices within the AI community. Moreover, the use of automated benchmark generation can potentially lead to more efficient and cost-effective evaluation processes, further reducing the pressure to outsource annotation tasks to potentially vulnerable populations. While the framework itself does not directly address all ethical concerns related to AI development, its potential to promote fairer labor practices within the evaluation pipeline represents a positive step towards a more ethical and responsible approach to LLM research and deployment.

In our study, all human participants involved in cross-validation tasks were volunteers, ensuring ethical data collection practices.

References

Max Bachmann. 2024. RapidFuzz: Rapid fuzzy string matching in Python using various string metrics.

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics.

Heather Brooks. 2024. The history of test automation - testrigor ai-based automated testing tool.

Sahan Bulathwela, Hamze Muse, and Emine Yilmaz. 2023. Scalable educational question generation with pre-trained language models. *Preprint*, arXiv:2305.07871.

Benjamin Clavié. 2024. rerankers: A lightweight python library to unify ranking methods. *Preprint*, arXiv:2408.17344.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang,

Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratan-

chandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Joseph L. Fleiss. 1981. *Statistical Methods for Rates and Proportions*, 2nd edition. John Wiley & Sons, New York.

Luyu Gao and Jamie Callan. 2022. Long document re-ranking with modular re-ranker. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22. ACM.

Vipul Gupta, David Pantoja, Candace Ross, Adina Williams, and Megan Ung. 2024. Changing answer order can decrease mmlu accuracy. *Preprint*, arXiv:2406.19470.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2024. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *Preprint*, arXiv:2409.12941.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. *Preprint*, arXiv:2310.08395.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal

Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen E. Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *Journal of the Association for Information Science and Technology*.

Sebastian Ruder. 2024. The evolving landscape of llm evaluation.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Sumuk Shashidhar, Abhinav Chinta, Vaibhav Sahai, and Dilek Hakkani-Tür. 2024. Unsupervised human preference learning. *Preprint*, arXiv:2410.03731.

R. Singh and N.S. Mangat. 1996. Stratified sampling. In *Elements of Survey Sampling*, volume 15 of *Kluwer Texts in the Mathematical Sciences*. Springer, Dordrecht.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. 2024. A careful examination of large language model performance on grade school arithmetic. *Preprint*, arXiv:2405.00332.

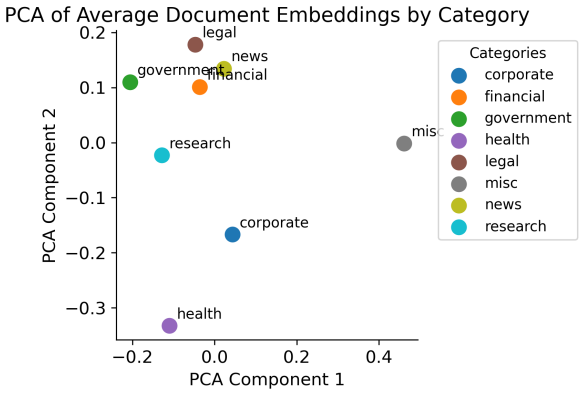
A Models Studied

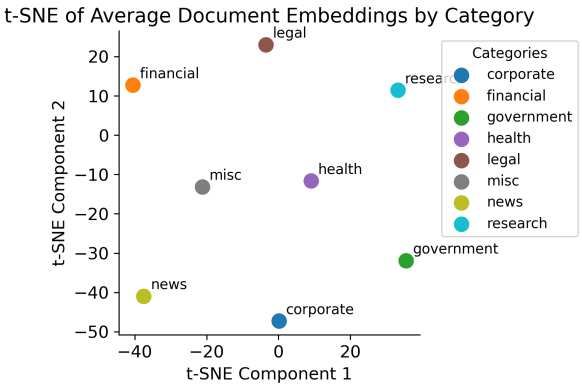
In this study, we evaluate a range of models across different families, ordered by the total number of parameters. For clarity, we categorize the models by family and use color coding to distinguish between them.

- **Meta-LLaMA Family**
  - **Meta-LLaMA-3.1-405B-Instruct-FP8** (405B parameters)[3]
  - **Meta-LLaMA-3.1-70B-Instruct** (70B parameters)
  - **Meta-LLaMA-3.1-8B-Instruct** (8B parameters)
- **Mistral Family**
  - **Mistral-Large-Instruct-2407** (132B parameters)
  - **Mistral-Small-Instruct-2409** (22B parameters)
- **Qwen Family**
  - **Qwen2.5-72B-Instruct** (72B parameters)
  - **Qwen2.5-32B-Instruct** (32B parameters)
  - **Qwen2.5-14B-Instruct** (14B parameters)
  - **Qwen2.5-3B-Instruct** (3B parameters)
  - **Qwen2.5-1.5B-Instruct** (1.5B parameters)
  - **Qwen2.5-0.5B-Instruct** (0.5B parameters)
- **GPT Family**
  - **GPT4o** (Unknown number of parameters)
  - **GPT4o-mini** (Unknown number of parameters)

As seen in the above list, models range from as small as 0.5 billion parameters to as large as 405 billion parameters. Additionally, the GPT family includes models where the number of parameters is currently unknown. This diverse set of models allows for a comprehensive analysis of performance across varying model sizes and architectures.

[3]Due to the scale of experimentation, FP8 was used reduce the computational cost of full FP16 inference. Chatbot Arena ranks the FP8 variant similarly to the FP16 variant.





B Dataset Preparation

The dataset was meticulously curated to ensure content diversity, fidelity, and relevance. Human annotators conducted manual web searches to collect diverse documents, prioritizing publicly available sources and tracking both content and source links. To ensure content fidelity and avoid duplication, all source links were de-duplicated. Annotators were instructed to select recent content (published within the past three months). In certain cases, Jina[4] was employed for converting HTML documents to plain text. Following data collection, humans manually cleaned the documents, removing HTML bloat and unnecessary content. The title of each document was designated as the first line of the corresponding markdown file.

C Efficient Similarity

To efficiently compute these clusters, we leverage GPU acceleration and parallel processing. Our implementation uses a batched computation approach, processing B questions simultaneously:

$$S_{ij} = \frac{E_i \cdot E_j^T}{|E_i||E_j|} \tag{3}$$

where $E_i \in \mathbb{R}^{B \times d}$ is a batch of normalized question embeddings, and $S_{ij} \in \mathbb{R}^{B \times N}$ is the resulting

[4]https://jina.ai/

similarity matrix. To address the computational challenges of processing large datasets, we employ a multi-GPU strategy. Let G be the number of available GPUs. We partition the dataset $\mathcal{E}_{\mathcal{F}}$ into G subsets $\mathcal{E}_{\mathcal{F}}gg = 1^G$ of approximately equal size. For each GPU g, we compute:

$$C_g = C_k \mid \exists q_i \in C_k : q_i \in \mathcal{E}_{\mathcal{F}g} \tag{4}$$

This parallel computation significantly reduces the time complexity from $O(N^2)$ to $O(N^2/G)$, where N is the total number of questions. To further optimize memory usage and computation time, we implement an iterative clustering algorithm. Instead of storing the full $N \times N$ similarity matrix, we process the data in chunks of size $M \ll N$. For each chunk, we compute:

$$C_m = C_k \mid \exists q_i \in C_k : q_i \in \mathcal{D}_m \tag{5}$$

where $\mathcal{E}_{\mathcal{F}m}$ is the m-th chunk of the dataset. We then merge the resulting clusters:

$$C = \bigcup_{m=1}^{\lceil N/M \rceil} C_m \tag{6}$$

This approach reduces the peak memory usage from $O(N^2)$ to $O(NM)$, allowing us to process datasets that would otherwise exceed available GPU memory.

D Detailed Question-Answer Generation Process

D.1 Context Window and Pre-stage Reasoning

The context window W_i is defined by the immediate chunks C_{i-1}, C_i, and C_{i+1}. The summary S_D provides thematic context. The pre-stage reasoning $\mathcal{R}_i$ involves parsing W_i and S_D using:

$$\mathcal{R}_i = \mathbb{E}_{C \in W_i \cup S_D}[G_{think}(C)]$$

where G_{think} distills essential information for generating coherent QA pairs.

D.2 Question Generation and Justification

G proposes directions D_q based on seed questions Q_s:

$$D_q = G_{propose}(Q_s, \mathcal{R}_i)$$

An optimal direction d_q^* is selected using a justification J:

$$d_q^* = \arg\max_{d \in D_q} P(d \mid J)$$

D.3 Citation Extraction and Validation

Citations $\mathcal{C}$ are extracted from C_i and validated using the citation match rate ρ:

$$\rho = \frac{\sum_{j=1}^{m} \delta(c_j \in C_i)}{m} \times 100$$

where δ checks string matches with C_i, ensuring reliability.

E Additional Scoring Model Details

The scoring system involves further nuances such as weighting votes from different judges based on their past performance and establishing inter-rater reliability metrics. For a more comprehensive description of these processes, please refer to the extended equations and contextual explanations below.

$$J_i(x, y, g) = \begin{cases} 1 & \text{if } x \text{ is preferred over } y, \\ 0 & \text{if } y \text{ is preferred over } x. \end{cases} \tag{7}$$

The combined judge preference score is then averaged to provide a stability metric for model comparisons:

$$P_x = \frac{1}{N} \sum_{i=1}^{N} J_i(x, y, g).$$

Further reliability metrics such as Cohen's kappa and Fleiss' Kappa are employed to measure the consistency of judgments across different judges. The formula for Cohen's kappa is:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{8}$$

where P_o is the observed agreement and P_e is the expected agreement by chance.

F Detailed Question Taxonomy

As outlined in Section 2.2, the question generation process is guided by a comprehensive taxonomy $\mathcal{T}$ that categorizes questions into distinct types based on their characteristics and the cognitive skills they assess. This appendix provides a detailed explanation of each question type, the rationale for their categorization, and their placement within the three levels of evaluation complexity.

F.1 Taxonomy Overview

We define the taxonomy $\mathcal{T}$ as follows:

- **True/False** (*Simple Boolean question answering*)
- **Factual** (*Test basic knowledge and recall of specific information*)
- **Clarification** (*Seek to resolve ambiguities or request additional information*)
- **Conceptual** (*Understanding broader ideas, theories, or principles*)
- **Analytical** (*Require interpretation and analysis of the information*)
- **Application-Based** (*How the information can be applied to real-world scenarios*)

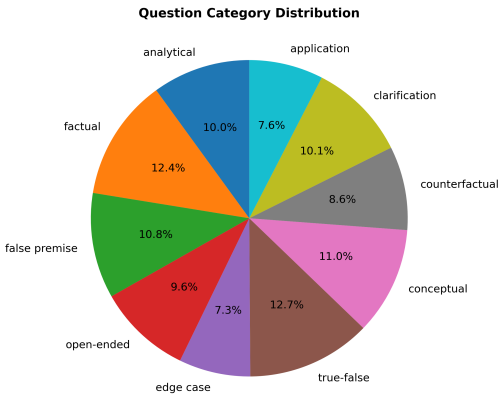


Figure 9: Resultant Question Distribution

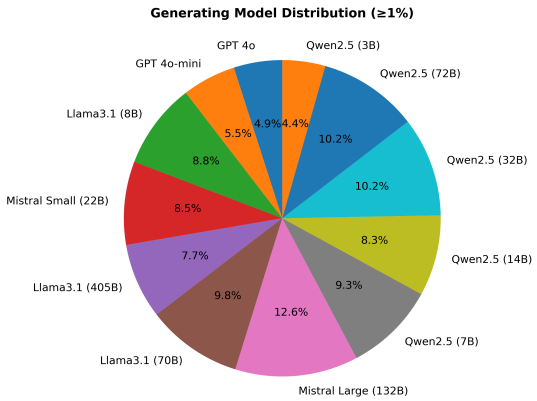


Figure 10: Resultant Model Distribution

- **Edge Case** (*Examine extreme or unusual scenarios to test the limits of a concept or rule*)
- **Counterfactual** (*Explore hypothetical scenarios that are contrary to the facts presented*)
- **Open-Ended** (*No single correct answer and encourage broader thinking and discussion*)
- **False Premise** (*Based on incorrect assumptions or misinterpretations*)

F.2 Levels of Evaluation Complexity

We categorize these question types into three levels of evaluation complexity, denoted by $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$, representing increasing degrees of difficulty in verification and subjectivity.

Level $\mathcal{E}_1$ (Low Complexity)

This level includes question types that are objective and straightforward to verify, often through direct retrieval of information. The questions have high verifiability $\mathcal{V}(T_i)$ and low subjectivity $\mathcal{S}(T_i)$. The following question types are included:

- **True/False**
- **Factual**

Level $\mathcal{E}_2$ (Moderate Complexity)

This level comprises question types that, while having definitive answers, require deeper understanding and reasoning. They have moderate verifiability and subjectivity. The included question types are:

- **Clarification**
- **Conceptual**
- **Analytical**
- **Application-Based**

Level $\mathcal{E}_3$ (High Complexity)

This level contains question types that are highly subjective or involve complex reasoning, making them challenging to evaluate. They have low verifiability and high subjectivity. The question types in this level are:

- **Edge Case**
- **Counterfactual**
- **Open-Ended**
- **False Premise**

F.3 Detailed Descriptions

True/False (Level $\mathcal{E}_1$)

Simple statements requiring a true or false answer, assessing the ability to recognize factual accuracy.

Example: *Is the capital of France Paris?*

Factual (Level $\mathcal{E}_1$)

Questions that ask for specific pieces of information or data explicitly stated in the text.

Example: *What is the boiling point of water at sea level?*

Clarification (Level $\mathcal{E}_2$)

Questions that address ambiguities or seek additional details to enhance understanding of the text.

Example: *Can you clarify what is meant by "sustainable development" in this context?*

Conceptual (Level $\mathcal{E}_2$)

Questions that focus on understanding broader ideas, theories, or principles underlying the content.

Example: *What is the theory behind photosynthesis in plants?*

Analytical (Level $\mathcal{E}_2$)

Questions that require interpretation, analysis, or critical thinking to infer information not directly stated.

Example: *How does the author support their argument about climate change?*

Application-Based (Level $\mathcal{E}_2$)

Questions that involve applying knowledge from the text to new or practical situations.

Example: *How would you apply Newton's laws to explain a car's acceleration?*

Edge Case (Level $\mathcal{E}_3$)

Questions that examine extreme or atypical scenarios to test the boundaries of concepts or rules.

Example: *What would happen to human circadian rhythms in a society without daylight?*

Counterfactual (Level $\mathcal{E}_3$)

Questions that explore hypothetical scenarios contrary to the facts presented, requiring speculative reasoning.

Example: *What might have happened if the internet was never invented?*

Open-Ended (Level $\mathcal{E}_3$)

Questions that encourage expansive thinking and discussion, with multiple possible valid responses.

Example: *What are the potential impacts of artificial intelligence on society?*

False Premise (Level $\mathcal{E}_3$)

Questions based on incorrect assumptions or misinterpretations, requiring identification and correction of errors.

Example: *Why is the Earth flat, and how does this affect global navigation?*

F.4 Rationale for Categorization

The categorization into levels $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ is based on the verifiability and subjectivity associated with each question type.

- **Level** $\mathcal{E}_1$: High verifiability ($\mathcal{V}(T_i) \geq \theta_v$) and low subjectivity ($\mathcal{S}(T_i) \leq \theta_s$). Answers can be easily verified against the text or external factual sources.

- **Level** $\mathcal{E}_2$: Moderate verifiability and subjectivity. Answers require more complex reasoning or understanding but remain within a realm where a single correct answer exists.

- **Level** $\mathcal{E}_3$: Low verifiability ($\mathcal{V}(T_i) < \theta'_v$) and high subjectivity ($\mathcal{S}(T_i) > \theta'_s$). Questions are challenging to evaluate objectively due to their speculative or interpretative nature.

G Reliability Per Parameter

In evaluating models for reliable citation generation, we propose a tradeoff function that balances performance and model size, called the Reliability Per Parameter, or RPP:

$$\text{RPP} = kP - cS^n \tag{9}$$

where P represents the percentage of valid citations produced, S denotes the model size in billions of parameters, k and c are scaling constants adjusting the influence of performance and size, respectively, and n is an exponent greater than 1, ensuring a super-linear penalty for increasing model size. This function exponentially rewards higher performance while penalizing larger models due to their increased computational demands. Specifically, we prioritized performance by setting k to 10^1 and the remaining parameters to 10^0.

Model	RPP
Qwen2.5 (32B)	932.98
Mistral Small (22B)	884.25
Qwen2.5 (72B)	882.14
Mistral Large (132B)	838.03
Llama3.1 (70B)	828.49
Qwen2.5 (7B)	756.42
Llama3.1 (8B)	680.46
Qwen2.5 (3B)	537.90
Llama3.1 (405B)	527.89
Qwen2.5 (1.5B)	494.33
Qwen2.5 (0.5B)	178.51

H Retreivability vs Valid Citation

In examining the relationship between retrievability and valid citations, we observe a notable correlation across model performances, as depicted in 7 and 5. Specifically, models that exhibit high scores in retrievability also tend to achieve high validity in citations. This phenomenon suggests that certain models are adept at using verbatim content from the given context, which in turn facilitates precise citations and the generation of questions embedding keywords optimally retrievable via BM25 mechanisms.

We hypothesize that this propensity to use verbatim text aids in both citation validity and the generation of questions with appropriate keywords, enhancing the retrievability metrics. To elucidate this hypothesis, we present a t-SNE visualization that highlights the diversity in questions generated by various models (11). This visualization showcases distinct clusters, reinforcing our observation of model-specific differences in generating retrievable and contextually accurate questions.

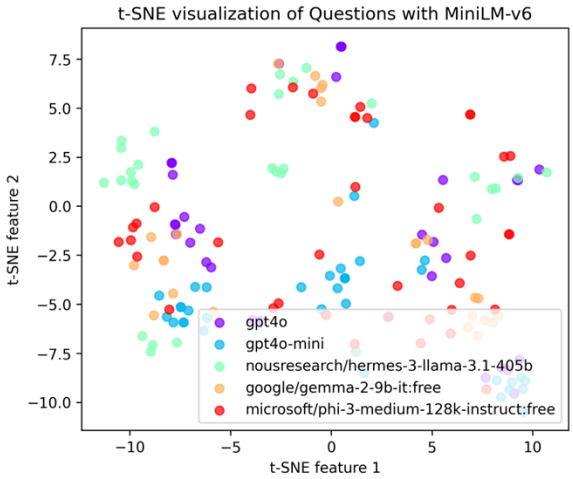


Figure 11: t-SNE Visualization of Questions For Various Models

I Pydantic Question Structure

Pydantic Structure

```python
class GeneratedQAPair(BaseModel):
    document_analysis: str = Field(..., description="The analysis of the document, given its summary.")
    chunk_analysis: str = Field(..., description="The analysis of the given text extract")
    potential_question_directions: List[str] = Field(..., description="The possible questions that a human would likely ask, based on the document summary, chunks, and cross-chunk analysis")
    best_direction: str = Field(..., description="The best question to ask, a decision made based on the question_directions. Why would it be a good question to ask, and what skills would it test?")
    direct_line_quotes: List[str] = Field(..., description="The precise verbatim excerpts from the current text chunk that serve as the basis for the question and answer. Each question must be directly answerable using only this chunk.")
    question: str = Field(..., description="The question to ask")
    answer: str = Field(..., description="The answer to the question")
    reasoning: str = Field(..., description="The reasoning for the answer, based on the question and the chunks")
    kind: str
    estimated_difficulty: int = Field(..., description="The estimated difficulty of the question, from 1 to 5")
```

J Prompts

Single Shot Analytical question generation

```
You will receive an extract from a document along with the preceding and succeeding chunks if they exist, and a summary of the document. Your task is to generate **analytical questions** based on these inputs.

First, read the summary to understand the document's context, then examine the text extract and its surrounding chunks.

# Guidelines
- All questions should be directly answerable from the current text chunk. DO NOT USE THE SUMMARY OR PREVIOUS OR NEXT CHUNKS TO GENERATE QUESTIONS.
- Focus on analytical questions that require critical thinking and deeper understanding, rather than simple recall of facts.
- Vary the difficulty level of the questions from 1 (very easy) to 5 (very challenging).
- Provide clear reasoning for the value or interest of each question.

# Task Steps
1. **Read the Summary:** Understand the general context and purpose of the document.
2. **Analyze the Chunks:** Examine the given chunk and its surrounding chunks for insights that can lead to analytical questions.
3. **Identify Potential Questions:** Consider several possible analytical questions based on the content of the chunk.
4. **Select the Best Questions:** Choose 3-5 questions that are valuable, testing critical thinking, analysis, or application of concepts.
5. **Identify Direct Quotes:** Gather exact, VERBATIM lines from the current chunk that will support the questions and answers. Remember, the question should be directly answerable from the current chunk.
6. **Create Question-Answer Pairs:** Formulate the questions and provide comprehensive answers, along with reasoning for each.
7. **Assign Difficulty Level:** Determine a difficulty level (1-5) for each question based on its complexity.

# Output Format

Output the questions as an array of JSON objects according to the specified Pydantic structure. This structure will be used to validate your JSON output.

```python
class GeneratedQAPair(BaseModel):
    document_analysis: str = Field(..., description="The analysis of the document, given its summary.")
    chunk_analysis: str = Field(..., description="The analysis of the given text extract")
    potential_question_directions: List[str] = Field(..., description="The possible questions that a human would likely ask, based on the document summary, chunks, and cross-chunk analysis")
    best_direction: str = Field(..., description="The best question to ask, a decision made based on the question_directions. Why would it be a good question to ask, and what skills would it test?")
    direct_line_quotes: List[str] = Field(..., description="The precise verbatim excerpts from the current text chunk that serve as the basis for the question and answer. Each question must be directly answerable using only this chunk.")
    question: str = Field(..., description="The question to ask")
    answer: str = Field(..., description="The answer to the question")
    reasoning: str = Field(..., description="The reasoning for the answer, based on the question and the chunks")
    kind: str
    estimated_difficulty: int = Field(..., description="The estimated difficulty of the question, from 1 to 5")
```

Examples
The questions should be like:
- Example 1: "What are the potential impacts of the season finale of season 8 of The Walking Dead on the show's future plot development?"
- Example 2: "How does the myth of Persephone reflect ancient Greek views on life and death?"
- Example 3: "In what ways did the introduction of robotics in surgery revolutionize medical practices?"

Notes
- If the text does not contain useful data, return a blank array.
- Ensure the JSON output passes Pydantic validation.
```

```
- Use a mental scratchpad to take notes as you read through the chunks and the document. Enclose your notes within <scratchpad> tags.
- Use the previous, and next chunks to help you understand the document. In NO CASES should you generate questions from the summary, previous chunk, or next chunk to avoid information leakage.

Here is an example of how your output JSON should look like. Enclose your output json in <generated_questions> tags:

<generated_questions>
[
    {
        "document_analysis": "The document appears to be discussing the mythology around Greek gods and goddesses, with a particular focus on their attributes, stories, and the cultural significance of these deities.",
        "chunk_analysis": "This chunk describes Demeter as the goddess of harvest and how her sadness affects the crops, tying her emotions directly to the seasons. The following chunk mentions Persephone, her daughter, being taken to the underworld, which causes Demeter's grief. This mythology explains the changing of the seasons, especially winter when Persephone is away from Demeter.",
        "potential_question_directions": [
            "How does Demeter's emotional state influence the natural world in the myth?",
            "What does the relationship between Demeter and Persephone symbolize in Greek culture?",
            "How does the myth explain natural phenomena through divine narratives?",
            "In what ways does the myth of Demeter and Persephone address themes of loss and renewal?"
        ],
        "best_direction": "The best question to ask is: 'In what ways does the myth of Demeter and Persephone address themes of loss and renewal?' This question is particularly insightful as it requires the responder to analyze the emotional dynamics between Demeter and Persephone, exploring how these emotions manifest in the natural world. It tests the reader's understanding of the interconnectedness of human experiences and seasonal changes, encouraging a deeper interpretation of the myth's significance in reflecting cultural values around grief and regeneration.",
        "direct_line_quotes": [
            "'As Persephone was dragged into the dark realm of Hades, Demeter's heart shattered, and with it, the fertility of the land.'",
            "'The earth lay barren as Demeter roamed in her sorrow, her grief seeping into the soil, leaving winter in her wake.'",
            "'When Persephone returned, the flowers blossomed, and life surged through the fields once more, a sign of Demeter's restored hope.'"
        ],
        "question": "In what ways does the myth of Demeter and Persephone address themes of loss and renewal?",
        "answer": "The myth illustrates loss through Demeter's grief when Persephone is taken to the underworld, causing the earth to become barren (winter). Renewal is depicted when Persephone returns, and Demeter's joy restores fertility to the land (spring). This cyclical pattern reflects the natural cycle of seasons and symbolizes hope and regeneration after periods of hardship.",
        "reasoning": "The chunk connects Demeter's emotions with the fertility of the land, embodying the cycles of nature. Analyzing this relationship reveals deeper themes of loss and renewal inherent in the myth.",
        "kind": "analytical",
        "estimated_difficulty": 4
    }
]
</generated_questions>
```

Single Shot Factual question generation

```
You will receive an extract from a document along with the preceding and succeeding chunks if they exist, and a summary of the document. Your task is to generate factual questions based on these inputs.

First, read the summary to understand the document's context, then examine the text extract and its surrounding chunks.

# Guidelines
- All questions should be directly answerable from the current text chunk. DO NOT USE THE SUMMARY OR PREVIOUS OR NEXT CHUNKS TO GENERATE QUESTIONS.
- Focus on factual, verifiable information rather than opinions.
- Vary the difficulty level of the questions from 1 (very easy) to 5 (very challenging).
- Provide clear reasoning for the value or interest of each question.

# Task Steps
1. **Read the Summary:** Understand the general context and purpose of the document.
2. **Analyze the Chunks:** Examine the given chunk and its surrounding chunks for factual information.
3. **Identify Potential Questions:** Consider several possible questions based on the chunk's content.
4. **Select the Best Questions:** Choose 3-5 questions that are valuable, testing specific knowledge or skills.
5. **Identify Direct Quotes:** Gather exact, VERBATIM lines from the current chunk that will support the questions and answers. Remember, the question should be directly answerable from the current chunk.
6. **Create Question-Answer Pairs:** Formulate the questions and provide comprehensive answers, along with reasoning for each.
7. **Assign Difficulty Level:** Determine a difficulty level (1-5) for each question based on its complexity.

# Output Format

Output the questions as an array of JSON objects according to the specified Pydantic structure. This structure will be used to validate your JSON output.

```python
class GeneratedQAPair(BaseModel):
    document_analysis: str = Field(..., description="The analysis of the document, given it's summary.")
    chunk_analysis: str = Field(..., description="The analysis of the given text extract")
    potential_question_directions: List[str] = Field(..., description="The possible questions that a human would likely ask, based on the document summary, chunks, and cross-chunk analysis")
    best_direction: str = Field(..., description="The best question to ask, a decision made based on the question_directions. Why would it be a good question to ask, and why skills would it test?")
    direct_line_quotes: List[str] = Field(..., description="The precise verbatim excerpts from the current text chunk that serve as the basis for the question and answer. Each question must be directly answerable using only this chunk.")
    question: str = Field(..., description="The question to ask")
    answer: str = Field(..., description="The answer to the question")
    reasoning: str = Field(..., description="The reasoning for the answer, based on the question and the chunks")
    kind: str
    estimated_difficulty: int = Field(..., description="The estimated difficulty of the question, from 1 to 5")
```

Examples
The questions should be like:
```

```
- Example 1: "When is the season finale of season 8 of The
↪  Walking Dead?"
- Example 2: "In greek mythology who was the goddess of
↪  spring growth?"
- Example 3: "When was the first robot used in surgery?"

# Notes
- If the text does not contain useful data, return a blank
↪  array.
- Ensure the JSON output passes Pydantic validation.
- Use a mental scratchpad to take notes as you read
↪  through the chunks and the document. Enclose your
↪  notes within <scratchpad> tags.
- Use the previous, and next chunks to help you understand
↪  the document. In NO CASES should you generate
↪  questions from the summary, previous chunk, or next
↪  chunk to avoid information leakage.

Here is an example of how your output JSON should look
↪  like. Enclose your output json in
↪  <generated_questions> tags:

<generated_questions>
[
    {
        "document_analysis": "The document appears to be
        ↪  discussing the mythology around Greek gods and
        ↪  goddesses, with a particular focus on their
        ↪  attributes, stories, and the cultural
        ↪  significance of these deities.",
        "chunk_analysis": "This chunk describes Demeter as
        ↪  the goddess of harvest and how her sadness
        ↪  affects the crops, tying her emotions directly
        ↪  to the seasons. The following chunk mentions
        ↪  Persephone, her daughter, being taken to the
        ↪  underworld, which causes Demeter's grief. This
        ↪  mythology explains the changing of the
        ↪  seasons, especially winter when Persephone is
        ↪  away from Demeter.",
        "potential_question_directions": [
            "Who is Demeter in Greek mythology?",
            "What event causes Demeter's sadness?",
            "How does the myth of Demeter explain the
            ↪  seasons?",
            "Who is Persephone, and what happens to her in
            ↪  the myth?"
        ],
        "best_direction": "The best question to ask is:
        ↪  'How does the myth of Demeter and Persephone
        ↪  explain the seasons?' This question is
        ↪  particularly effective because it requires the
        ↪  responder to synthesize information about the
        ↪  emotional dynamics between Demeter and
        ↪  Persephone and their direct impact on
        ↪  agricultural cycles. It tests the reader's
        ↪  comprehension of how mythology serves to
        ↪  personify natural phenomena, thereby deepening
        ↪  their understanding of the cultural
        ↪  significance of seasonal changes in ancient
        ↪  Greek society.",
        "direct_line_quotes": [
            "Demeter, goddess of the harvest, weeps as her
            ↪  fields grow barren, her sorrow mirrored by
            ↪  the earth itself.",
            "Persephone, her beloved daughter, is taken to
            ↪  the underworld, leaving Demeter in
            ↪  profound grief that halts the growth of
            ↪  crops.",
            "The earth, barren and cold during
            ↪  Persephone's absence, blooms again upon
            ↪  her return, marking the changing seasons."
        ],
        "question": "How does the myth of Demeter and
        ↪  Persephone explain the seasons?",
        "answer": "The myth explains the changing of the
        ↪  seasons through Demeter's grief when her
        ↪  daughter Persephone is taken to the
        ↪  underworld. During the months Persephone is
        ↪  away, Demeter's sadness causes the earth to
        ↪  become barren, which corresponds to winter.
        ↪  When Persephone returns, Demeter's joy
        ↪  restores fertility to the land, which
        ↪  corresponds to spring.",
        "reasoning": "The chunk explains that Demeter's
        ↪  sadness, caused by the loss of Persephone,
        ↪  leads to barren crops, connecting the myth to
        ↪  the natural phenomenon of winter and spring.",
        "kind": "factual",
        "estimated_difficulty": 3
    }
]

</generated_questions>
```

Single Shot True False question generation

```
You will receive an extract from a document along with the
↪  preceding and succeeding chunks if they exist, and a
↪  summary of the document. Your task is to generate
↪  true-false questions based on these inputs.

First, read the summary to understand the document's
↪  context, then examine the text extract and its
↪  surrounding chunks.

# Guidelines
- All statements should be directly answerable as true or
↪  false from the current text chunk. DO NOT USE THE
↪  SUMMARY OR PREVIOUS OR NEXT CHUNKS TO GENERATE
↪  QUESTIONS.
- Focus on factual, verifiable information rather than
↪  opinions.
- Vary the difficulty level of the statements from 1 (very
↪  easy) to 5 (very challenging).
- Provide clear reasoning for the value or interest of
↪  each statement.

# Task Steps
1. **Read the Summary:** Understand the general context
↪  and purpose of the document.
2. **Analyze the Chunks:** Examine the given chunk and its
↪  surrounding chunks for factual information.
3. **Identify Potential Statements:** Consider several
↪  possible true-false statements based on the chunk's
↪  content.
4. **Select the Best Statements:** Choose 3-5 statements
↪  that are valuable, testing specific knowledge or
↪  skills.
5. **Identify Direct Quotes:** Gather exact, VERBATIM
↪  lines from the current chunk that will support the
↪  questions and answers. Remember, the question should
↪  be directly answerable from the current chunk.
6. **Create Question-Answer Pairs:** Formulate the
↪  questions and provide comprehensive answers, along
↪  with reasoning for each.
7. **Assign Difficulty Level:** Determine a difficulty
↪  level (1-5) for each question based on its complexity.

# Output Format

Output the statements as an array of JSON objects
↪  according to the specified Pydantic structure. This
↪  structure will be used to validate your JSON output.

```python
class GeneratedQAPair(BaseModel):
    document_analysis: str = Field(..., description="The
    ↪  analysis of the document, given its summary.")
    chunk_analysis: str = Field(..., description="The
    ↪  analysis of the given text extract.")
    potential_question_directions: List[str] = Field(...,
    ↪  description="The possible true-false statements
    ↪  that a human would likely consider, based on the
    ↪  document summary, chunks, and cross-chunk
    ↪  analysis.")
    best_direction: str = Field(..., description="The best
    ↪  statement to test, a decision made based on the
    ↪  potential statements. Why would it be a good
    ↪  statement to test, and what skills would it
    ↪  assess?")
    direct_line_quotes: List[str] = Field(...,
    ↪  description="The precise verbatim excerpts from
    ↪  the current text chunk that serve as the basis for
    ↪  the question and answer. Each question must be
    ↪  directly answerable using only this chunk.")
    question: str = Field(..., description="The true-false
    ↪  statement to present.")
    answer: str = Field(..., description="Indicate 'True'
    ↪  or 'False'.")
    reasoning: str = Field(..., description="The reasoning
    ↪  for the answer, based on the statement and the
    ↪  chunks.")
    kind: str = Field(..., description="The type of
    ↪  question, e.g., 'true-false'.")
```
```

```
    estimated_difficulty: int = Field(...,
    ↪  description="The estimated difficulty of the
    ↪  statement, from 1 to 5.")
```

Examples
The true-false statements should be like:
- Example 1: "The season finale of season 8 of 'The
↪ Walking Dead' aired on April 15th."
- Example 2: "In Greek mythology, Demeter is the goddess
↪ of spring growth."
- Example 3: "The first robot used in surgery was
↪ introduced in 2000."

Notes
- If the text does not contain useful data, return a blank
↪ array.
- Ensure the JSON output passes Pydantic validation.
- Use a mental scratchpad to take notes as you read
↪ through the chunks and the document. Enclose your
↪ notes within <scratchpad> tags.
- Use the previous and next chunks to help you understand
↪ the document. In NO CASES should you generate
↪ statements from the summary, previous chunk, or next
↪ chunk to avoid information leakage.

Here is an example of how your output JSON should look
↪ like. Enclose your output json in
↪ <generated_questions> tags:

<generated_questions>
[
    {
        "document_analysis": "The document appears to be
        ↪ discussing Greek mythology, focusing on gods
        ↪ and goddesses, their stories, and their
        ↪ cultural significance.",
        "chunk_analysis": "This chunk describes Demeter as
        ↪ the goddess of the harvest and explains how
        ↪ her emotions affect the fertility of the
        ↪ earth. It mentions that her sadness leads to
        ↪ barren lands, linking her mood to the changing
        ↪ seasons. The following chunk discusses her
        ↪ daughter Persephone's abduction to the
        ↪ underworld, which causes Demeter's grief.",
        "potential_question_directions": [
            "Demeter is the goddess of the harvest.",
            "Persephone is Demeter's mother.",
            "Demeter's sadness causes the earth to
            ↪ flourish.",
            "The myth explains the origin of the seasons."
        ],
        "best_direction": "The best statement to test is:
        ↪ 'Demeter's sadness causes the earth to
        ↪ flourish.' This statement is particularly
        ↪ effective because it challenges the reader to
        ↪ critically analyze the cause-and-effect
        ↪ relationship presented in the myth. It
        ↪ requires an understanding of how Demeter's
        ↪ emotional state directly impacts the fertility
        ↪ of the earth, thus assessing the reader's
        ↪ comprehension of the narrative's underlying
        ↪ themes regarding grief and its consequences on
        ↪ nature.",
        "direct_line_quotes": [
            "Demeter's grief cast a shadow over the earth,
            ↪ and in her sorrow, the lands became barren
            ↪ and cold, leading to the onset of
            ↪ winter.",
            "As long as Persephone remained in the
            ↪ underworld, Demeters sadness made the
            ↪ earth infertile, depriving it of its
            ↪ bountiful harvests."
        ],
        "question": "Demeter's sadness causes the earth to
        ↪ flourish.",
        "answer": "False",
        "reasoning": "In the myth, Demeter's sadness
        ↪ causes the earth to become barren, not
        ↪ flourish. Her grief over Persephone's
        ↪ abduction leads to a lack of fertility,
        ↪ representing the winter season.",
        "kind": "true-false",
        "estimated_difficulty": 3,
    }
]
</generated_questions>

Fidelity Check

Analyze the summary, text chunk, question, and ground
↪ truth answer, and determine if the question and answer
↪ can be sourced from the text chunk.

Steps

1. **Document Understanding**: Analyze the document
↪ summary to understand its key themes.
   - Enclose this analysis in `<document_understanding>`
   ↪ XML tags.

2. **Chunk Understanding**: Examine the text chunk to
↪ identify key elements or concepts.
   - Enclose this analysis in `<chunk_understanding>` XML
   ↪ tags.

3. **Question Understanding**: Comprehend the question
↪ being asked for clarity and intent.
   - Enclose this analysis in `<question_understanding>`
   ↪ XML tags.

4. **Ground Truth Answer Understanding**: Identify key
↪ points from the provided ground truth answer.
   - Enclose this analysis in
   ↪ `<ground_truth_answer_understanding>` XML tags.

5. **Analysis**: Evaluate if the question and answer can
↪ be directly derived from the text chunk.
   - Enclose this evaluation in `<analysis>` XML tags.

6. **Final Answer**: Decide and state whether the
↪ information needed to answer the question is present
↪ in the text chunk.
   - Use `<final_answer>YES</final_answer>` or
   ↪ `<final_answer>NO</final_answer>` XML tags for this
   ↪ final determination.

Output Format

Output each analysis and decision-making step using the
↪ specified XML tags. The final output should begin with
↪ the understanding analyses and conclude with the
↪ analysis decision and final answer in XML format.

Examples

Example 1:
*Input:*
- <document_summary>"[Summary of the
↪ document]"</document_summary>
- <chunk_understanding>"[Specific piece of
↪ text]"</chunk_understanding>
- <question_understanding>"[Question being
↪ asked]"</question_understanding>
- <ground_truth_answer>"[Ground Truth
↪ answer]"</ground_truth_answer>

*Output:*

```xml
<document_understanding>
Understanding of the summary including key themes...
</document_understanding>

<chunk_understanding>
Analysis of the piece of text...
</chunk_understanding>

<question_understanding>
Comprehension of the question being asked...
</question_understanding>

<ground_truth_answer_understanding>
Key points from the gold answer...
</ground_truth_answer_understanding>

<analysis>
Evaluation on whether the question and answer can be
↪  sourced directly from the chunk of text...
</analysis>

<final_answer>
```

YES
</final_answer>
```

# Notes

- Ensure each section is complete and clear.
- Maintain the logical flow from understanding the components to concluding whether they align.
- Handle specific language or nuances in the question, answer, or text chunk that may affect clarity or conclusion.

---

## Judge Prompt

---

You will be provided with the summary of a document, a piece of text, a question generated from that text, and the correct or "gold" answer to the question. Additionally, you will receive two answers: Answer A and Answer B. Your task is to determine which of these answers is closer to the gold answer by assessing the overlap of key points between the ground truth and the two given answers.

# Steps

1. **Document Understanding**:
   - Analyze the provided document summary to grasp the context and main themes.

2. **Chunk Understanding**:
   - Examine the provided text (chunk) to understand its content.

3. **Question Understanding**:
   - Interpret the given question to fully comprehend what is being asked.

4. **Ground Truth Answer Understanding**:
   - Understand the provided ground truth answer, identifying its key points.

5. **Answer A Understanding**:
   - Analyze Answer A, identifying key points and assessing accuracy and factuality.

6. **Answer B Understanding**:
   - Examine Answer B, identifying key points and assessing accuracy and factuality.

7. **Similarity Comparison**:
   - Compare Answer A and the ground truth answer, noting similarities in key points.
   - Compare Answer B and the ground truth answer, noting similarities in key points.

8. **Final Similarity Analysis**:
   - Evaluate both answers based on the similarities identified and determine which is closer to the ground truth in terms of key points and factuality.

# Output Format

- Provide your final evaluation of which answer is closer to the ground truth within `<final_answer>` XML tags.
- Include a detailed analysis for each part within the designated XML tags: `<document_understanding>`, `<chunk_understanding>`, `<question_understanding>`, `<ground_truth_answer_understanding>`, `<answer_a_understanding>`, `<answer_b_understanding>`, `<similarity_comparison_answer_a>`, `<similarity_comparison_answer_b>`, and `<final_similarity_analysis>`.

# Examples

**Input**:
```xml
<document_summary>
[Summary]
</document_summary>

<piece_of_text>
[Text]
</piece_of_text>

<question>
[Question]
</question>

<gold_answer>
[Gold Answer]
</gold_answer>

<answer_a>
[Answer A]
</answer_a>

<answer_b>
[Answer B]
</answer_b>
```
Output:
```xml

<document_understanding>
Understanding of the summary including key themes
</document_understanding>

<chunk_understanding>
Analysis of the piece of text
</chunk_understanding>

<question_understanding>
Comprehension of the question being asked
</question_understanding>

<ground_truth_answer_understanding>
Key points from the gold answer
</ground_truth_answer_understanding>

<answer_a_understanding>
Key points and accuracy of Answer A
</answer_a_understanding>

<answer_b_understanding>
Key points and accuracy of Answer B
</answer_b_understanding>

<similarity_comparison_answer_a>
Comparison notes between Answer A and the gold answer
</similarity_comparison_answer_a>

<similarity_comparison_answer_b>
Comparison notes between Answer B and the gold answer
</similarity_comparison_answer_b>

<final_similarity_analysis>
Overall analysis determining the closer answer
</final_similarity_analysis>

<final_answer>
Answer X (where X is the option you pick)
</final_answer>
```

Notes

- Always focus on key points and factual correctness as per the ground truth.
- Avoid any biases and rely solely on the evidence presented.
- Enclose all evaluations and analyses in the specified XML tags for clarity and structure.

```

K Qualitative Examples

K.1 Similarity Clusters

```
# Example Cluster 1:
==================================================
Question ID: 6d19739f-9d9a-419a-bb36-00bfc9fe0c6d
Document ID: f3370f14-180d-4375-8255-7c41417a61c9
Question: How does the Mentu program aim to engage
↪  students in learning math?
Answer: The Mentu program aims to engage students by
↪  presenting math learning in a motivational and
↪  interesting format, likened to a video game. This
↪  approach is designed to make learning enjoyable while
↪  simultaneously improving math skills.
Kind: analytical
Document Title: Three ways McKinsey is partnering to
↪  transform education in Latin America
Generating Model: gpt-4o-mini
----------------------------------------
Question ID: 79cc3ab6-dd55-4e87-a399-c66ec46c3d53
Document ID: f3370f14-180d-4375-8255-7c41417a61c9
Question: How does the Mentu program leverage technology
↪  to engage students in learning math?
Answer: The Mentu program uses an online platform designed
↪  to be motivational and engaging, likening the learning
↪  experience to a video game. This gamification approach
↪  helps capture students' interest and encourages them
↪  to improve their math skills in a fun and interactive
↪  way.
Kind: open-ended
Document Title: Three ways McKinsey is partnering to
↪  transform education in Latin America
Generating Model: gpt-4o-mini
----------------------------------------
Question ID: 1ff7b3c8-a128-4f29-9bd4-25ae1775dcd7
Document ID: f3370f14-180d-4375-8255-7c41417a61c9
Question: How does the online math program Mentu aim to
↪  engage students and improve their math skills?
Answer: Mentu aims to engage students by being
↪  motivational and interesting, similar to a video game.
↪  This design helps to make learning math more enjoyable
↪  and effective, thereby improving students' math
↪  skills.
Kind: analytical
Document Title: Three ways McKinsey is partnering to
↪  transform education in Latin America
Generating Model: gpt-4o
----------------------------------------

# Example Cluster 2
==================================================
Question ID: a9c45d82-34fd-498f-a4d9-e8cfa0827be9
Document ID: f3370f14-180d-4375-8255-7c41417a61c9
Question: Did fewer than ten percent of schools in Ecuador
↪  achieve an 'excellent' ranking in standardized tests?
Answer: No, fewer than one percent of schools in Ecuador
↪  achieved an 'excellent' ranking in standardized tests,
↪  highlighting the significant educational challenges in
↪  the country.
Kind: false premise
Document Title: Three ways McKinsey is partnering to
↪  transform education in Latin America
Generating Model: gpt-4o-mini
----------------------------------------
Question ID: 4bc30b5c-853a-4746-8489-65100d9617c9
Document ID: f3370f14-180d-4375-8255-7c41417a61c9
Question: Fewer than one percent of schools in Ecuador
↪  achieve an 'excellent' ranking in standardized tests.
Answer: True
Kind: true-false
Document Title: Three ways McKinsey is partnering to
↪  transform education in Latin America
Generating Model: gpt-4o-mini
----------------------------------------
Question ID: ccf6c2e6-e2d8-4b77-91e6-bcf0e730a712
Document ID: f3370f14-180d-4375-8255-7c41417a61c9
Question: Fewer than one percent of schools in Ecuador
↪  achieve an 'excellent' ranking in standardized tests.
Answer: True
Kind: true-false
Document Title: Three ways McKinsey is partnering to
↪  transform education in Latin America
Generating Model: gpt-4o
----------------------------------------
```

K.2 Question Fidelity

```xml
<document_understanding>
    The document summarizes key points from Jim Jones'
    ↪  testimony before the House Subcommittee on Health
    ↪  about the new Human Foods Program (HFP). The
    ↪  summary highlights the program's focus on
    ↪  improving food safety and nutrition by addressing
    ↪  foodborne illnesses, food chemical safety, and
    ↪  diet-related chronic diseases. It mentions
    ↪  efforts and challenges in areas like infant
    ↪  formula safety and collaboration with
    ↪  stakeholders for public health.
</document_understanding>

<chunk_understanding>
    The provided text chunk discusses the FDA's actions
    ↪  related to lead contamination in certain food
    ↪  products, the role of state partnerships, and the
    ↪  need for enhanced information-sharing
    ↪  capabilities. It mentions statutory restrictions
    ↪  that limit FDA's ability to share nonpublic
    ↪  information with regulatory partners, which can
    ↪  hinder swift action to protect the supply chain.
    ↪  The chunk also touches upon other FDA concerns,
    ↪  such as dietary supplement safety and addressing
    ↪  diet-related chronic diseases.
</chunk_understanding>

<question_understanding>
    The question asks if the FDA is currently able to
    ↪  share all nonpublic information with regulatory
    ↪  partners, which pertains to the agency's capacity
    ↪  for information dissemination and regulatory
    ↪  compliance.
</question_understanding>

<ground_truth_answer_understanding>
    The ground truth answer, "False," indicates that the
    ↪  FDA is not able to share all nonpublic
    ↪  information with regulatory partners.
</ground_truth_answer_understanding>

<analysis>
    The text chunk clearly states that the FDA is
    ↪  restricted by statute from sharing certain
    ↪  nonpublic information with regulatory partners
    ↪  and that these limitations can prevent the FDA
    ↪  from taking swift action. Given this insight, it
    ↪  supports the answer that the FDA cannot currently
    ↪  share all nonpublic information with regulatory
    ↪  partners.
</analysis>

<final_answer>
    YES
</final_answer>
```

```xml
<document_understanding>
    This study focuses on evaluating the effects of solar
    ↪  cook stoves on rural households in Zambia,
    ↪  particularly looking at meal preparation and fuel
    ↪  consumption using a randomized controlled trial.
    ↪  Key findings include a significant reduction in
    ↪  firewood consumption, time spent collecting
    ↪  firewood, and overall fuel costs, though no
    ↪  substantial changes were observed in dietary
    ↪  diversity or cooking frequency. The study
    ↪  highlights the potential economic benefits of
    ↪  solar stoves and suggests further research on
    ↪  their long-term nutritional impacts and
    ↪  complementary interventions to promote dietary
    ↪  diversification.
</document_understanding>

<chunk_understanding>
```

```
    The text chunk consists of various summary statistics
    ↪  tables related to dietary outcomes, cooking
    ↪  frequency, and fuel collection, among others. It
    ↪  also discusses robustness checks, detailing
    ↪  approaches like calculating the local average
    ↪  treatment effect (LATE) to determine the solar
    ↪  stove's impact more accurately. The primary
    ↪  results rely on intent-to-treat (ITT) effects due
    ↪  to random assignment, with noted underestimates
    ↪  of average treatment effects (ATE).
</chunk_understanding>

<question_understanding>
    The question asks how the findings of the study
    ↪  regarding solar stoves could influence household
    ↪  energy efficiency in rural Zambia. This seeks to
    ↪  understand the broader implications of the
    ↪  reduction in fuel consumption and time saved
    ↪  collecting firewood noted in the study's results.
</question_understanding>

<ground_truth_answer_understanding>
    The ground truth answer states that solar stoves can
    ↪  significantly reduce household energy
    ↪  consumption, especially in firewood usage and
    ↪  time spent collecting firewood, leading to
    ↪  improved energy efficiency. It also implies
    ↪  financial relief for households and a move
    ↪  towards sustainable energy practices, while
    ↪  noting the lack of impact on dietary diversity or
    ↪  cooking frequency.
</ground_truth_answer_understanding>

<analysis>
    The text chunk provides detailed statistics and
    ↪  methodological robustness checks but does not
    ↪  explicitly discuss the impact of solar stoves on
    ↪  household energy efficiency in descriptive terms.
    ↪  It focuses more on presenting data and
    ↪  methodological approaches rather than directly
    ↪  linking these to broader outcomes like energy
    ↪  efficiency or household benefits. Therefore, the
    ↪  question's answer cannot be fully sourced from
    ↪  the chunk provided, as the chunk lacks explicit
    ↪  qualitative analysis on energy efficiency impacts.
</analysis>

<final_answer>
    NO
</final_answer>
```

L Qualitative Citation Examples

Metadata	**Chunk ID:** d5279abc-1a3f-480b-b516-9835333b7976 **Kind:** Conceptual **Category:** Government **Model:** Mistral 132B **Document Summary:** The U.S. Department of Energy (DOE) announced nearly $44 million in funding to support clean energy projects in Tribal communities. $18.8 million will be awarded to six American Indian and Alaska Native communities to install solar and wind energy infrastructure, while an additional $25 million will be available for future project planning and development. These projects aim to enhance Tribal energy sovereignty, improve energy resilience, lower energy costs, and create jobs, with the DOE highlighting specific projects and their anticipated benefits. The DOE also announced a new funding opportunity for Tribal communities to plan and develop clean energy projects, further demonstrating their commitment to Tribal energy independence.
Text Chunk	Tribal Nations are helping to lead the way as we transition to build more cost-effective, stronger, and resilient communities for the future." The six cost-shared energy projects selected today are the result of a competitive funding opportunity announcement released Feb. 27, 2024. Through these grants, the DOE Office of Indian Energy will continue its efforts to advance Tribal energy sovereignty and maximize the deployment of clean, reliable, affordable, and local energy solutions. Collectively, these projects are estimated to result in nearly 5.6-MW of clean energy generation and over 3.2-megawatt-hours (MWh) of battery storage, impacting over 100 Tribal buildings and saving those communities nearly $54 million over the life of those systems. These investments will yield tangible results year after year to improve the quality of life for these underserved communities. The projects competitively selected for negotiation of award, in alphabetical order, are: Agua Caliente Band of Cahuilla Indians in Palm Springs, CA, will install a cumulative of 551-kilowatts (kW) of solar photovoltaics (PV) and a 1,311-kilowatt-hour (kWh) battery energy storage system (BESS) on five essential buildings, significantly enhancing grid resiliency, reducing energy costs, and promoting environmental sustainability.
Generated Citations	1. These investments will yield tangible results year after year to improve the quality of life for these underserved communities. 2. Collectively, these projects are estimated to result in nearly 5.6-MW of clean energy generation and over 3.2-megawatt-hours (MWh) of battery storage, impacting over 100 Tribal buildings and saving those communities nearly $54 million over the life of those systems.
Generated Question	How does the DOE's funding support the transition to clean energy in Tribal communities?
Generated Answer	The DOE's funding supports the transition to clean energy in Tribal communities by providing financial resources for the installation of solar and wind energy infrastructure. This funding is expected to result in nearly 5.6-MW of clean energy generation and over 3.2-megawatt-hours (MWh) of battery storage, impacting over 100 Tribal buildings and saving those communities nearly $54 million over the life of the systems. This financial support enhances energy security, resilience, and affordability for the communities.

Table 1: Valid Citations Generated By Mistral 132B.

Metadata	**Chunk ID:** c6bf6797-6bf3-49c5-bf10-cb2d47ce01c1 **Kind:** clarification **Category:** Research **Model:** Llama3.1 8B **Document Summary:** This study investigated the impact of solar cook stoves on household meal preparation and fuel consumption in rural Zambia using a randomized controlled trial. While solar stove adoption was high (40% of dishes), the study found no significant effects on dietary diversity or cooking frequency. However, there were significant reductions in firewood consumption, time spent collecting firewood, and overall fuel costs. These findings suggest that while solar stoves may not change dietary patterns in the short-term, they can help reduce reliance on biomass fuels and offer economic benefits to households. Further research may be needed to explore the long-term impacts of solar stoves on nutrition and to identify complementary interventions that could promote dietary diversification alongside fuel-efficient cooking technologies.
Text Chunk	[....]n measure. First, we use the FAO's Household Dietary Diversity Score (HDDS), which categorizes ingredients into one of 12 FAO-designated food group categories. The HDDS is simply the count of the unique food group categories present in a given dish, meal, day, etc., where one indicates low dietary diversity, and 12 indicates high dietary diversity. Second, following lachat2018dietary, we include dietary species richness (DSR) as a measure of dietary diversity. The DSR is calculated by matching ingredients to species and tabulating the number of unique species consumed during a given period. [...]has no upper bound since it counts the number of unique species present in a given dish, meal, day, etc.
Next Chunk	[...] component of the AAS program was to encourage households to grow and consume more legumes. Legumes are viewed as an important component of sustainable or climate-smart agriculture. They fix nitrogen in the soil, reducing the need to add nitrogen to the soil through inorganic fertilizer. Legumes are also protein-rich, so adding them to one's diet can offset reliance on animal-based protein. One hurdle to adopting legumes is their long cooking time, which can make preparing them prohibitively expensive if one must purchase fuelwood for cooking. Our measure of legume consumption is simply counting the number of times legumes were present in a given dish, meal, day, etc. [...]).
Generated Citations	1. Our measure of legume consumption is simply counting the number of times legumes were present in a given dish, meal, day, etc. 2. Legumes are viewed as an important component of sustainable or climate-smart agriculture. 3. Legumes are also protein-rich, so adding them to one's diet can offset reliance on animal-based protein.
Generated Question	How does the measure of legume consumption work, and what are the benefits of including legumes in one's diet?
Generated Answer	The authors measure legume consumption by counting the number of times legumes are present in a given dish, meal, day, etc. Legumes are important in sustainable agriculture because they fix nitrogen in the soil, reducing the need to add nitrogen through inorganic fertilizer. They are also protein-rich, making them a valuable addition to one's diet, which can offset reliance on animal-based protein.
Failure Reason	Citations were instead generated from the next chunk (which is also given to the model during question generation to help with contextual understanding). **This highlights a weakness in instruction following.**

Table 2: Invalid generated citations by Llama3.1 8B.