Automated Evaluation of Retrieval-Augmented Language Models with Task-Specific Exam Generation

Gauthier Guinet [1] Behrooz Omidvar-Tehrani [1] Anoop Deoras [1] Laurent Callot [1]

Abstract

We propose a new method to measure the task-specific accuracy of Retrieval-Augmented Large Language Models (RAG). Evaluation is performed by scoring the RAG on an automatically-generated synthetic exam composed of multiple choice questions based on the corpus of documents associated with the task. Our method is an automated, cost-efficient, interpretable, and robust strategy to select the optimal components for a RAG system. We leverage Item Response Theory (IRT) to estimate the quality of an exam and its informativeness on task-specific accuracy. IRT also provides a natural way to iteratively improve the exam by eliminating the exam questions that are not sufficiently informative about a model's ability. We demonstrate our approach on four new open-ended Question-Answering tasks based on Arxiv abstracts, StackExchange questions, AWS DevOps troubleshooting guides, and SEC filings. In addition, our experiments reveal more general insights into factors impacting RAG performance like size, retrieval mechanism, prompting and fine-tuning. Most notably, our findings show that choosing the right retrieval algorithms often leads to bigger performance gains than simply using a larger language model.

1. Introduction

Evaluating Large Language Models (LLM) beyond a set of limited tasks is notoriously challenging. General capabilities of LLMs conveyed through public benchmarks are not necessarily related to performance on narrow and highly specific customer tasks, even more so when such tasks involve specific domain knowledge corpus. Evaluation metrics aim at capturing different aspects of the performance of a LLM. No single metric can adequately capture every aspects of it.

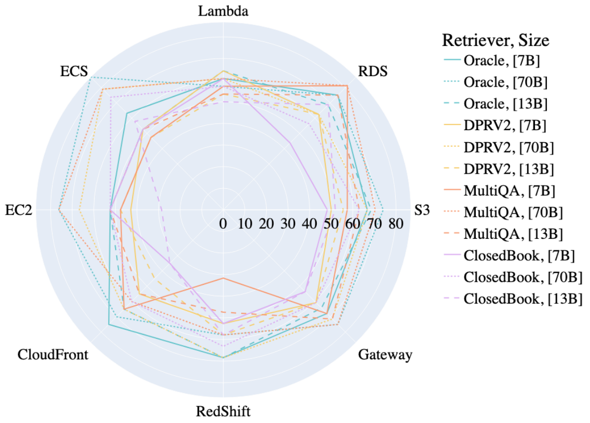

Figure 1. Granular results of our exam evaluation for the task of AWS DevOps troubleshooting. Accuracy is reported for different retrieval approaches and retriever sizes, on a % scale. Labels on the diameter shows the troubleshooting categories, i.e., AWS resources. Colors correspond to different retrieval approaches (`Oracle`, `DPRV2`, `MultiQA`, `ClosedBook`, as discussed in Section 4.2) and patterns correspond to the base LLM size (7B, 13B, and 70B). For instance, we observe that a small model such as Mistral-7B with MultiQA embeddings has an accuracy around 80% for the AWS resource "Relational Database Service" (RDS).

In this work, we propose an LLM-driven exam-based evaluation methodology to measure the accuracy of Retrieval-Augmented LLMs (RAG) on a given task. Our method is fully automated and does not require annotated ground-truth dataset. Our metrics focus on factual accuracy, i.e., the ability to retrieve and leverage the right information to correctly answer a user query. In addition to enabling users to select the optimal combination of components of a RAG system for their retrieval tasks, our methodology reveals more general insights into RAG performance factors like size, retrieval mechanism, prompting, and fine-tuning.

Our first contribution is a quantitative exam-based evaluation strategy which is fully automated, guaranteeing scalability in contrast to conventional human-in-the-loop evaluations, while concurrently mitigating expenses linked with expert or annotator engagement. Exams are generated by a

[1] AWS AI Labs. Correspondence to: Gauthier Guinet <guinetgg@amazon.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

LLM using the corpus of the data associated with the task at hand. Candidate RAG systems are evaluated on their ability to successfully answer the multiple-choice questions in the exam. Evaluation is always a trade-off between ease of scoring and representativeness. For factual evaluation, the ease of scoring multiple-choice exams does not compromise the objective of assessing factual knowledge efficiently and reliably. Moreover, comparative exam result analysis reveals areas for performance improvement, enabling continuous feedback-driven enhancements to the exam corpus. Figure 1 shows an example in the DevOps domain.

Our second contribution is a methodological improvement strategy within the automated exam generation process. Notably, we draw on Item Response Theory (IRT) to optimize the generated exam and maximize its informativeness on task-specific model performance.

We illustrate and evaluate our approach on open-ended question-answering tasks using 4 different knowledge corpora: AWS DevOps troubleshooting guides, Arxiv abstracts, StackExchange questions, and SEC Filings. In summary, here is the list of our contributions.

- We contribute a comprehensive methodology for *automatic evaluation* of Retrieval-Augmented Generation LLM pipelines based on task-specific synthetic exams.
- Leveraging Item Response Theory (IRT), we develop *robustness* and *interpretable* evaluation metrics to quantify and elucidate factors influencing model efficacy.
- We design a principled, fully automated technique to *construct* and *iteratively refine* the exams to maximize informativeness.
- We provide benchmark datasets for RAG systems evaluation, by creating four new tasks based on public datasets from diverse domains.
- We provide an open-source implementation of our proposed exam generation, evaluation and optimization framework allowing it to be executed on any RAG task. The source code is available at https://github.com/amazon-science/auto-rag-eval.

The paper is organized as follows: we review related work in Section 2. In Section 3, we discuss the problem of evaluating RAG pipelines, and propose two evaluation modalities. We introduce an extensive benchmark in Section 4 and present our experiments in Sections 5 (model evaluation) and 6 (exam evaluation). We conclude in Section 7.

2. Related Literature

We propose an automated exam generation method that enables standardized evaluation of RAG on specific tasks by tailoring multiple choice questions to each task's documents. To the best of our knowledge, this is the first work addressing RAG assessment with this particular contribution focus. However, our contribution builds upon existing literature in related domains like RAG systems, evaluation frameworks for language models, and item response theory.

Retrieval-Augmented Generation. Retrieval-Augmented Generation (RAG) integrates pre-trained language models with information retrieval techniques, enriching natural language processing tasks through external knowledge sources (Lewis et al., 2020). This methodology was further developed by (Khandelwal et al., 2019) who highlighted the effectiveness of nearest-neighbor search within language models. Subsequent advancements include the introduction of a self-supervised learning objective, synergizing language models with retrieval systems (Guu et al., 2020), and the expansion of retrieval-augmented methods for handling larger data scales (Borgeaud et al., 2021). The field is comprehensively surveyed by (Gao et al., 2023b).

Evaluation of NLP, LLM, and RAG. The evolution of evaluation in Natural Language Processing (NLP) has transitioned from classical, task-specific benchmarks, like BLEU scores for machine translation (Papineni, 2002), to more nuanced metrics (Es et al., 2023; Zheng et al., 2023; Hoshi et al., 2023; Saad-Falcon et al., 2023), like Answer Equivalence (Bulian et al., 2022), as the complexity of outputs has increased. This shift is exemplified by the work on GPT-3 which challenges traditional evaluation methods with its task-agnostic capabilities (Brown et al., 2020; Bender & Koller, 2020). Notwithstanding these advances, the field continues to confront well documented challenges (Deutsch et al., 2021; Bowman & Dahl, 2021; Bulian et al., 2022; Novikova et al., 2017; Fabbri et al., 2021), in particular to accurately measure models' understanding of nuanced human concepts.

When evaluating retrieval-augmented generation models, the difficulties are compounded by the multiplicity of components involved. (Gao et al., 2023b) offers a survey of the field, revealing *"a notable paucity of research dedicated to evaluating the distinct characteristics of RAG models, with only a handful of related studies."* Most recent work in this domain emphasize the integration of retrieved information with generated content (Lewis et al., 2020; Kamalloo et al., 2023; Chen et al., 2023). Some solutions like RAGAs (Es et al., 2023), RaLLe (Hoshi et al., 2023), and ARES (Saad-Falcon et al., 2023) are being increasingly used in industrial and research applications but offer limited interpretability. Furthermore, although certain benchmarks have been designed to assess specific aspects of LLMs (e.g, truthfulness (Lin et al., 2021), faithfulness (Adlakha et al., 2023), and factuality (Lee et al., 2022; Min et al., 2023; Yu et al., 2023; Muhlgay et al., 2023)), a comprehensive task-specific evaluation of these aspects remains a significant challenge. The absence of a canonical evaluation method for RAG pipelines is the main driver of our contribution in this work.

Item Response Theory. Item Response Theory (IRT) is a framework to study responses to questions (items) in evaluations conducted through examination. IRT offers a model-based approach for estimating both item characteristics and individual examinee abilities. Its formalization appears to originate with the works of (Rasch, 1960; Lord et al., 1968). This theory has been developed and extended to be applied to a wide range of problems which are reviewed in (Embretson & Reise, 2013; Cai et al., 2016) among others. In machine learning, IRT has found applications for providing interpretability (Yeung, 2019; Martínez-Plumed et al., 2019; 2016), improving recommender systems (Liu et al., 2023), or guiding human evaluation of chatbots (Sedoc & Ungar, 2020). To the best of our knowledge, this paper is the first one to leverage item response theory in order to develop an automated evaluation procedure for generative models.

3. Methodology

In this section, we define the key concepts upon which we build our contributions, discuss the problem of evaluation of RAG pipelines, and propose two evaluation modalities.

3.1. Preliminaries

RAG pipelines. We consider a RAG pipeline to be constituted by three components: the LLM, the retrieval mechanism, and the in-context learning part. First is the LLM which is used to generate an answer given some retrieved context and a prompting strategy. We rely on broadly available, pre-trained large language models. The second component is a retrieval mechanism which is used to identify documents in the corpus relevant to the user's question. These documents are then included in the LLM's prompt to provide helpful context for answering. Finally, the third component is the in-context learning part of the prompt given to the LLM. In this paper, the in-context learning mechanism is the number of examples of the task we provide in the prompt. Note that we could incorporate more complex RAG design choices: data processing, query refactoring, more elaborated prompting, fine-tuning, and post-generation processing. However, in the sake of generality, we focus on the three aforementioned choices and remark that our approach easily extends to other settings.

Tasks. The generic task we consider here is that of open-ended question-answering supported by a corpus of documents in which the answer is expected to be found. A task $t \in \mathcal{T}$ is characterized by a knowledge corpus which is composed of set of documents from a specific domain. The retrieval mechanism extracts from the corpus the documents that are most relevant to answering the user's question.

Evaluation. Evaluation should be seen through two lenses: *predictive* and *prescriptive*. The goal of a predictive evaluation is to design an estimator of the accuracy on a downstream task of interest. Prescriptive evaluation guides design decision by providing insights on the choice of model to make, as well as the impact of the different components. Our main contribution in this work, the exam-based evaluation methodology, is used for both predictive and prescriptive evaluation. For predictive evaluation, each RAG pipeline is evaluated independent from other pipelines by answering an exam composed of multiple-choice questions. This evaluation metric does not quantify all possible dimensions of interest, no single metric does. Our method is predictive in what is arguably the most important performance dimension for a RAG pipeline: the ability to retrieve and leverage external information. Prescriptive evaluation involves jointly examining multiple pipelines to understand broader patterns. This allows for model ranking and selection and reveals general insights on the drivers of RAG pipeline performance to guide design decisions.

3.2. Exam Generation

The exam generator algorithm leverages a pre-trained LLM which generates a multi-choice exam with n questions for a given task t. The output $Q = \{q_1, q_2 \dots q_n\}$ is a set of questions. Each question is composed of a question description and a set of possible answers. There is one and only one correct answer among the possible answers. We leverage here a two-step approach: for each document in the knowledge corpus, we use the LLM and several prompt strategies to create candidates questions. This raw generation is insufficient to generate a high quality exam and thus we combined it with several NLP-based filters to remove low-quality questions along several axis such as length, incorrectness, and self-containment. We refer to this improvement steps as *a-priori verification* as the filters do not require candidate model answers. In particular, we note an interesting asymmetry: granted a document corpus, it is relatively easy for a LLM to generate a question and the correct answer, as this task is self-contained in the prompt in terms of knowledge. However, it is considerably more difficult to create high quality incorrect answers, commonly referred as discriminators. We leverage Jaccard and embedding based similarity metrics to filter out degenerated questions following this pattern. This methodology and the exam generation process is further detailed in Appendix A.

Throughout the work, we aimed at balancing each contributions on the automated generation of the exam, with equal new methodological contributions to both assess the quality and the evaluation impact of the exam. Among other, the next section introduces a novel *a-posteriori verification*, using Item Response Theory (IRT) to weight the contribution of each question towards the final model score by the inferred quality of the question. This ensures that our evaluation methodology is more robust to outliers and low quality questions.

3.3. Exam Evaluation

Pointwise Evaluation. To evaluate the performance of a RAG pipeline, we first treat it as a student who participates in an exam generated as described above, where we select for each question the answer with maximal length-penalized log-likelihood (Gao et al., 2023a). The score obtained by the RAG is simply the share of questions answered correctly. This examination modality allows us to order RAG pipelines as a function of their performance on a given exam generated using the corpus of documents associated with task t.

This simple exam-based evaluation modality is an automated, scalable, and computationally efficient way to obtain a performance ranking of RAG pipelines designed for a specific task. The performance metric, the grade of a given RAG, is trivially interpretable. Our approach to evaluation is capable of delivering deeper insights, which we discuss next.

Aggregate Evaluation and Item Response Theory. The aggregate evaluation method *jointly and simultaneously* evaluates multiple RAG pipelines together with the *quality* of the exam Q generated for task t on which the RAGs are graded. This allows to (i) increase robustness by providing weighted RAG ability scores that account for noisy or uninformative questions, (ii) reliably quantify the contribution of each individual RAG components on the final performance and (iii) quantify the exam informativeness over the task of interest. This last point is core in providing a set of quantitative exam analytics with high interpretability (Section 6.1 and 6.2) and iteratively improve the exam to maximize informativeness (Section 6.3).

To do so, we rely on Item Response Theory (IRT), a modern framework used to understand how exam-takers interact with individual items (i.e., questions) in an exam. Item Response Theory models the probability of a correct answer to an exam item $q_i \in Q$ as a function of the exam-taker's ability θ and of three parameters characterizing a specific question q_i: difficulty b_i, discrimination d_i, and guessing factor g_i, thanks to the logistic model:

$$\mathbb{P}(X = 1|\theta, g_i, d_i, b_i) = g_i + \frac{(1 - g_i)}{1 + exp(-d_i(\theta - b_i)))}, \quad (1)$$

where $X = \{1, 0\}$ indicates a correct or incorrect answer. On what follows, we use the abbreviation $p_i(\theta)$ for this quantity, omitting the dependency on g_i, d_i, b_i.

The capability of a question to distinguish between student of a given ability θ is captured by the difficulty parameter. Intuitively, an easy question (low d_i) will be answered correctly by all high-ability (high θ) students so it does not help distinguishing the best among those. A question with a high discrimination value d_i amplifies the difference in ability, meaning that the question is better at distinguishing between students that have close but different ability. In all multiple choice questions, there exists a probability to answer the question correctly by chance which is captured by g_i.

In this paper, we propose a variation of the standard IRT model of Equation 1 tailored to the task of evaluating RAG systems, which we call the *hierarchical* IRT model. The hierarchical model provides a higher resolution estimate of the ability of the RAG by breaking it down into its three components using the additive model $\theta_m = \theta_{llm(m)} + \theta_{ret(m)} + \theta_{icl(m)}$. The three parameters quantify the ability of the LLM, retrieval method, and in-context learning method, respectively. Extending this model to more complex RAG design choices only requires adding suited latent variables.

The hierarchical IRT model is one of the key contributions of this paper. It allows us to evaluate the performance of the components of a RAG pipeline independently, which simplifies the problem of model selection substantially. In addition it allows us to derive some general insights on the main drivers of the performance of RAG pipelines, discussed in details in Section 5.

3.4. Item Response Model Estimation

To fit the IRT model, we employ a log-likelihood optimization model to estimate the ability θ_m of the candidate models $m \in \mathcal{M}$, and to jointly estimate the three parameters $\{g_i, d_i, b_i\}$ characterizing each question $q_i \in \mathcal{Q}$ in the exam corpus. We maximize the log-likelihood function $\mathcal{L}$ over the parameters $\{\theta_m\}_{m\in\mathcal{M}}$ and $\{g_i, d_i, b_i\}_{q_i\in\mathcal{Q}}$ using the probability function $p_i(\theta)$ defined in Equation 1.

$$\mathcal{L} = \sum_{\substack{m\in\mathcal{M} \\ q_i\in\mathcal{Q}}} r_{i,m} \log p_i(\theta) + (1 - r_{i,m}) \log(1 - p_i(\theta)), \quad (2)$$

In Equation 2, $r_{i,m}$ is a binary function indicating whether model m provided the correct response to question i ($r_{i,m} = 1$) or incorrect ($r_{i,m} = 0$). For the hierarchical IRT model, we decompose θ_m as $\theta_{llm(m)} + \theta_{ret(m)} + \theta_{icl(m)}$ and maximize over this new space of latent variables. We further detail the estimation procedure and results in Appendix B. A model is considered to possess high ability if it can accurately respond to challenging questions. Conversely, difficulty questions are deemed so if only students with a high level of competence can answer them. This interdependent problem is what is addressed when maximising Equation 2.

4. Experiment Benchmark

In this section, we introduce an extensive benchmark instantiated based on the model defined in Section 3. We report the experiment results over this benchmark in Section 5.

4.1. Tasks

We introduce four different tasks in our benchmark $\mathcal{T} = \{t_{ops}, t_{arx}, t_{stk}, t_{sec}\}$. The task t_{ops} is defined over a knowledge corpus of 1249 webpages from AWS Knowledge Center[1] where each webpage troubleshoots one DevOps problem for AWS customers. The task t_{arx} is defined over 13000 ArXiv papers where each paper is represented by its abstract. The task t_{stk} is defined over 977 StackExchange[2] questions. Last, the task t_{sec} is defined on 188 documents submitted in yearly fashion to the U.S. Securities and Exchange Commission (SEC) by publicly traded companies, company insiders, and brokers[3]. Table 1 provides information about corpus associated with each task. Further details can be found in Appendix A.4.

We selected these four tasks to cover a broad spectrum of knowledge domains, ranging from technical operations and community-driven Q&A platforms to financial earnings and academic research, ensuring a diverse and comprehensive coverage of subjects.

4.2. RAG Pipelines

In our experiment benchmark, we consider 45 different RAG pipelines by combining 5 different retrieval mechanisms, 3 different LLMs, and 3 different ICL modes.

Retrieval Mechanism Variants. We consider the following 3 retrieval paradigms: Closed-Book, Classical Retrieval, and Oracle. Closed-Book and Oracle act as lower and upper bounds on the quality of the information that can be provided to the LLM from the corpus. We also introduce five different classical retrieval methods, totalling seven retrieval mechanisms.

Closed-Book Retrieval. No additional knowledge from the document corpus is provided to the LLM through retrieval. The exam-taker has only access to the question and the possible answers as well as the knowledge encoded in the weights of the LLM (i.e., parametric knowledge). We denote this method as `ClosedB`. A good evaluation score in this case relates to the LLM base knowledge of the question. Low `ClosedB` evaluation scores convey that the pretrained model knows little about the domain or that the question or its possible answers are poorly formulated.

Oracle. The exam-taker has access to the specific document used to generate the question and answer pair, in addition to the question itself and all possible candidate answers. In other words, the exam-taker has access to the ground truth knowledge. A good `Oracle` score relates not only to the LLM base knowledge of the question, but also the ability to extract the answer from the ground truth. High `Oracle` scores can be obtained if the questions are properly formulated and the exam-taker is competent enough to extract the information to correctly answer. The `Oracle` score is uniquely possible thanks to our exam-design strategy and is core to providing a calibrated evaluation metric.

Retrieval Models. The exam-taker is allowed to search over the knowledge corpus to combine the contextual knowledge with its parametric knowledge, using a given retrieval algorithm to better inform its answer. To give a representative perspective of the space of retrieval models, we compare a variety of methods.

- *Dense models:* We focus on two models: `MultiQA` embeddings (Talmor & Berant, 2019; Wang et al., 2020) and Siamese network embeddings (`SIAM`) (Koch et al., 2015).
- *Sparse models:* We focus on `BM25` (Robertson et al., 2009), a widely-used information retrieval technique which employs a probabilistic model to rank documents based on the frequency and distribution of query terms within them.
- *Hybrid models:* We consider ensembles of Dense and Sparse base retrievers where the output is re-ranked using a cross-encoder model (Yadav et al., 2022). We refer to the models as `DPR` (`SIAM` plus `BM25`) and `DPRV2` (`MultiQA` plus `BM25`) bellow.

Our analysis covers a spectrum of retrieval models, including contemporary models like `MultiQA` from Sentence Transformers and Cross-encoders in `DPR` and `DPRV2`, which are among the most used in the community (resp. 1.6M and 1.3M monthly downloads on HuggingFace, at the time of publication). BM25 is a standard bearer in Information Retrieval known for its robustness over modern methods. Our set of models is a combination of dense, sparse and hybrid models to ensure that our results are representative of all main classes.

LLM Variants. We employ Mistral-7B, LlamaV2-13B and LlamaV2-70B (Jiang et al., 2023; Touvron et al., 2023). We chose these three LLMs with the objective of investigating the spectrum of performance across different scales, aiming to gain insights into how the size of a model influences its language processing capabilities. These models offer a balance between advanced features, optimal performance at the

[1] *AWS Knowledge Center: https://repost.aws/knowledge-center.*

[2] *StackExchange network: https://stackexchange.com/.*

[3] *Company Filings: https://www.sec.gov/edgar*

Attribute	t_{ops} (DevOps)	t_{arx} (Arxiv)	t_{stk} (StackExchange)	t_{sec} (SEC Filings)
# Documents	1249 webpages	13 000 abstracts	977 Stack questions	493 sections
# Documents Chunks	4 536	13 000	977	11 658
Average Document Length (words)	254	189	144	187
Total # Words	1 153 149	2 459 804	140 859	2 175 250
Vocabulary Size	9 175	39 551	44 084	11 229
# Topics	18	13	20	10

Table 1. Description of all four tasks used in the experiment benchmark. Word count is computed using NLKT word tokenizer and punctuation remover (Bird et al., 2009).

time of the publication, community support, and practical considerations like resource availability and computational efficiency. Our original analysis also considered LlamaV2-7B, Falcon-40B and Alpaca-13B, which we discarded as they were consistently outperformed.

Finally, we consider the following 3 in-context demonstration modes: ICL@0, ICL@1, and ICL@2. In the former, no in-context example is added to the prompt. In the two others, respectively one and two examples are provided in the prompt (question, candidate answers and correct one). While we examine performance on these specific RAG settings, our broader goal is not to maximize metrics on any single RAG formulation, but rather to have an evaluation system that is adaptable and extensible. Acknowledging the high frequency of top models release, our exam-based framework is intentionally crafted to be independent of the choice of retrieval approach or LLM, as discussed in Section 3.3. On a per-application basis, our approach allows to flexibly incorporate additional RAG dimensions like data processing methods, query reformulation, fine-tuning, etc.

5. Experiment Results for Model Evaluation

In this section, we present the experimental results for model evaluation by following the methodology discussed in Section 3 and the benchmark introduced in Section 4. We first introduce point-wise evaluation results of RAG pipelines in Table 2 and then discuss IRT-based ability levels for individual RAG components in Table 3. Such results are used at task level to make optimal design decisions and across tasks to infer RAG system patterns.

5.1. Accuracy and Ability Analysis

In summary, our experiments result in the following four findings: Firstly, there's no one size that fits all, i.e., the optimal choice of retrieval method, and to a lesser extent LLM, is typically task-dependent. Depending on the task and retrieval, Mistral-7B and LlamaV2-13B ranking varies. LlamaV2-70B is even outperformed in no-retrieval settings. Similarly, for some tasks such as t_{sec} and t_{arx}, `BM25` outperforms `MultiQA` and `SIAM`, which indicates that sparse retrieval is typically better than dense retrieval for these tasks. A conjecture is that such tasks often contain easily identifiable terms (e.g., AWS service names in t_{ops}) which can be retrieved with keyword search, while other tasks like t_{stk} mostly contains common words. However, our findings do indicate that hybrid ensemble models, which integrate both dense and sparse retrieval techniques (e.g., `DPRV2`), generally offer greater robustness and adaptability across a variety of tasks compared to exclusively dense or sparse models.

	Best Absolute Accuracy in %					
	Retrieval	t_{ops}	t_{stk}	t_{arx}	t_{sec}	Avg.
Mistral-7B	`ClosedB`	52.2	48.6	54.5	49.5	51.2
	`SIAM`	45.5	50.0	57.0	47.6	50.0
	`DPR`	52.2	58.3	60.3	60.5	57.8
	`BM25`	**58.0**	60.4	69.5	55.3	60.8
	`MultiQA`	57.7	**72.2**	**69.5**	53.6	63.2
	`DPRV2`	55.1	70.1	69.4	**63.9**	64.6
	`Oracle`	63.8	74.3	68.6	70.9	69.4
LlamaV2-13B	`ClosedB`	50.4	42.4	45.5	45.8	46.3
	`SIAM`	44.6	48.6	56.2	42.7	48.0
	`DPR`	51.3	54.9	55.4	61.0	55.7
	`BM25`	54.5	63.2	66.9	55.9	60.1
	`MultiQA`	**58.0**	67.4	66.1	53.8	61.3
	`DPRV2`	55.7	**66.7**	**69.4**	**63.9**	63.9
	`Oracle`	63.0	68.8	68.6	69.9	67.6
LlamaV2-70B	`ClosedB`	63.0	38.2	47.1	48.7	49.3
	`SIAM`	55.4	53.5	57.9	49.5	54.1
	`DPR`	61.5	59.7	60.3	**68.3**	62.5
	`BM25`	71.4	65.3	76.9	64.3	69.5
	`MultiQA`	**72.6**	**75.7**	74.4	58.6	70.3
	`DPRV2`	69.7	72.2	**77.7**	67.6	71.8
	`Oracle`	72.6	73.6	76.0	77.1	74.8

Table 2. Point-wise evaluation results. The score is the percentage of correctly answered questions by the RAG. More precisely, we denote the maximum score among the three ICL passes as *best absolute accuracy* for a RAG. For each LLM, we indicate the top performing retriever in bold.

Secondly, the right choice of the retrieval method can often lead to performance improvements surpassing those from simply choosing larger LLMs, as seen when comparing

		Ability Level			
	Retrieval	t_{ops}	t_{stk}	t_{arx}	t_{sec}
LLM	Mistral-7B	-0.38	-0.59	-1.03	-0.48
	LlamaV2-13B	-0.04	-0.51	-0.78	-0.36
	LlamaV2-70B	1.00	0.40	-0.05	0.18
Retrieval	`ClosedB`	-0.86	-1.39	-0.62	-1.29
	`SIAM`	-2.74	-0.06	-0.21	-1.39
	`DPR`	-0.87	0.36	-0.01	0.10
	`BM25`	-0.56	0.62	0.60	-0.22
	`MultiQA`	-0.43	1.06	0.62	-0.42
	`DPRV2`	-0.54	0.99	0.72	0.22
	`Oracle`	-0.20	1.14	0.75	0.59
ICL	ICL@0	-0.54	-0.77	-0.11	-0.83
	ICL@1	0.66	0.02	0.90	0.05
	ICL@2	0.46	0.04	1.06	0.11

Table 3. IRT evaluation results for each RAG component $(\theta_{llm(m)}, \theta_{ret(m)}, \theta_{icl(m)})_{m\in\mathcal{M}}$. A higher level of model ability level corresponds to a higher value of θ, and values are relative: for instance, to assess the ability gain of a given retrieval model MultiQA, we consider $\theta_{\texttt{MultiQA}} - \theta_{\texttt{ClosedB}}$. Note that results are not normalized across tasks and thus not directly comparable. See Table 5 for question-based parameters $(g_i, b_i, d_i,)_{i\in\mathcal{M}}$.

marginal gains in Table 3: in t_{sec}, we gain more ability gain by switching from `SIAM` to `DPRV2` compared to switching to larger LLMs. Thirdly, for tasks involving *closed source knowledge*, the accuracy bottleneck is typically the LLM and not the retrieval method.By *closed source*, we refer to confidential data, proprietary to companies, such as internal financial statements, proprietary codebase, internal FAQs or documents. This type of corpus is particularly relevant given that the LLM wasn't exposed to it during the pre-training: all the information flows through the retrieval. Fourthly, poorly aligned retriever component can lead to a worse accuracy than having no retrieval at all, as seen for `SIAM` performance compared to `ClosedB` in Tables 2 and 3.

Finally, a noteworthy phenomena in RAG systems is when there is strong information overlap between documents. Notably, this explains why the Oracle might be outperformed by some retriever, as seen in Table 2 for t_{arx} and in Figure 7: certain document chunks are more helpful to answer question than the one used to actually generate the question.

5.2. Evaluating the Evaluation: Meta-Evaluation

In Section 5.1, we presented how our evaluation framework assesses various RAG pipelines by utilizing point-wise evaluation results and IRT-based ability levels for individual RAG components. Another critical question is how to evaluate our evaluation framework itself.

Comparing and evaluating the evaluation methods of LLMs, including our exam-based evaluation model, is a complex meta-evaluation task. Granted current challenges for direct evaluation of LLMs, we highlight that performing meta-evaluation is a step above in terms of difficulty. Beyond that, meta-evaluation of LLM assessment is a multi-objective problem due to the multidimensional nature of LLM performance: LLMs are assessed on varied capabilities like factuality, linguistic understanding, coherence, and ethical considerations, each requiring specific evaluation criteria. The rapid evolution of LLM technology adds to this complexity, as new models may exhibit behaviors not previously considered, necessitating continuous updates to meta-evaluation methodologies. Furthermore, the subjective nature of language processing and the diversity of LLM applications demand different performance metrics, further complicating the meta-evaluation process. The reliance on human judgment as a benchmark introduces variability, making it challenging to establish a universal evaluation framework that balances technical accuracy with diverse human perspectives and real-world applicability (Howcroft et al., 2020).

Typical NLP evaluation methods like ROUGE, BLEU, and BERTScore, commonly used for evaluating specific aspects of language models, are too narrow for effectively meta-evaluating LLMs, lacking breadth, interpretability, and feedback to assess capabilities and guide improvements. Recent LLM-based evaluation methods (Es et al., 2023; Fu et al., 2023; Zheng et al., 2023; Xu et al., 2023) are promising but still have limitations in scope, adaptability, interpretability, bias reduction, or actionable feedback required for comprehensive LLM evaluation. A key distinction of our exam-based evaluation approach compared to other methods is that it is interpretable and provides predictive and prescriptive guidance on areas where the RAG needs improvement.

6. Experiment Results for Exam Evaluation

Properly defining what is a good exam is a difficult question: although perfectly correct from a content perspective, an exam can still be of lower quality by not being discriminative enough across models nor informative enough on the task of interest. To quantitatively measure and improve upon this, we present in this section an analysis of the exam questions generated by our framework across different categorization schemes. Specifically, we leverage Bloom's taxonomy to categorize questions by cognitive complexity and introduce an *item information function* to quantify the informativeness of questions for evaluating model performance. Figures 3 and 4 illustrate this process in the context of StackExchange task. We conclude by presenting a methodology to iteratively maximize the informativeness of the exam, a key contribution of our work.

6.1. Exam Informativeness

To measure the informativeness of the exam with respect to the task and models, we introduce the *item information function*, aka, Fischer information (Hambleton et al., 1991).

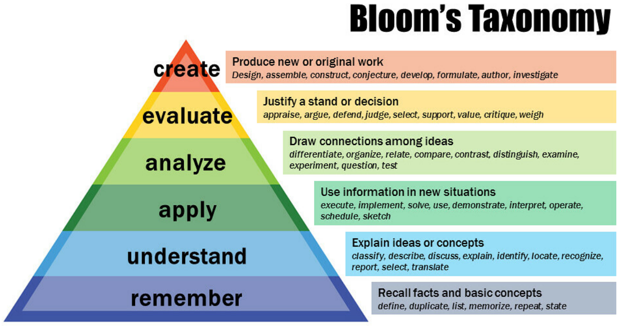
Bloom's Taxonomy
create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
evaluate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh
analyze
Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
apply
Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
understand
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
remember
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Figure 2. Representation of Bloom's revised taxonomy. The cognitive complexity of skills increase from the bottom to the top of the pyramid. Source: Vanderbilt University Center for Teaching

This function quantifies the amount of information that an observable random variable X provides about the unknown ability parameter θ, through a measure of the curvature of the log-likelihood function $\mathcal{L}$. Thereby, it offers a pivotal metric for assessing the precision of statistical estimators in parameter estimation theory and more precisely the discriminating power of the exam question over the space of candidate models at different levels of ability. It is defined for an individual question as:

$$I(\theta|g_i, d_i, b_i) = d_i^2 \frac{(p_i(\theta) - g_i)^2}{(1 - g_i)^2} \frac{1 - p_i(\theta)}{p_i(\theta)}, \tag{3}$$

In Figure 8, we highlight individual item information functions for this task. Note that the item information function reaches its maximum value at the question's difficulty parameter. Thus, questions provide the most information for estimating θ at an ability level close to their difficulty, and provide less information at ability levels further away from their difficulty. In this way, the item information function formally characterizes a question's capacity for discriminating between individuals and around a particular ability level. To asses the overall effect of a given subset of questions $\mathcal{R} \subset \mathcal{Q}$, we introduce the aggregated Information function:

$$\bar{I}_{\mathcal{R}}(\theta) = \frac{1}{|\mathcal{R}|} \sum_{i \in \mathcal{R}} I(\theta|g_i, d_i, b_i), \tag{4}$$

6.2. Categorization of Exams and Questions

Once an exam is generated, we perform an automated question categorization to determine the relevant dimensions for a given question. Question categories enable a more granular understanding of types of questions that RAG pipelines are better or worse at as well as the ones that helps to better discriminate across models, through the usage of the item information function introduced above. For this aim, we leverage Bloom's revised taxonomy (Bloom et al., 1956; Krathwohl, 2002) illustrated in Figure 2. Bloom's taxonomy is a hierarchical model that classifies learning objectives into different levels of cognitive complexity. Table 6 in Appendix C illustrates the levels of the revised Bloom's Taxonomy, from the lowest to the highest, along with a brief description and examples of how they might translate into multiple-choice questions. They differentiate between the knowledge dimension (factual, conceptual, procedural, and meta-cognitive) and the cognitive process dimension (remember, understand, apply, analyze, evaluate, and create).

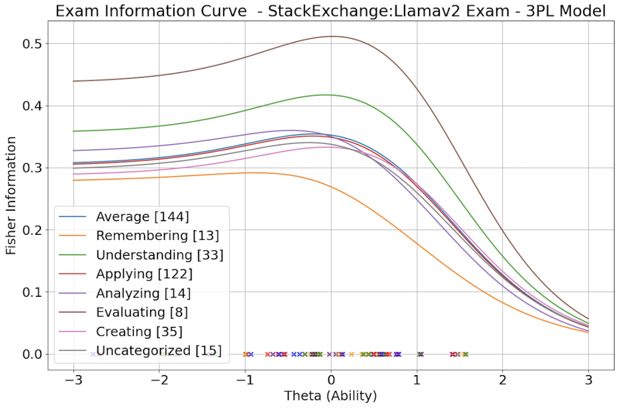

Figure 3. Aggregated Information function $\bar{I}_{cat}(\theta)$ for t_{stk}, averaged according to Bloom taxonomy. Each cross on the x-axis correspond to a given model ability θ_m, with no particular signification granted to their colors.

In Figure 3, we present the average item information function $\bar{I}_{cat}(\theta)$ for each Bloom category for t_{stk}. Lines are fitted by maximizing the log-likelihood $\mathcal{L}$ defined in Equation 2, using the optimization process described in Appendix B.1. Informativeness is an increasing quantity, meaning that higher values are better. As discussed in Section 3.3, it is also a function of the ability level. Therefore, some questions might be more informative at certain level of ability, for instance to discriminate among medium ability students and less at others, such as for high ability students. For this specific task t_{stk}, we observe that *evaluating* and *understanding* are the most discriminate dimensions in Bloom's taxonomy across different ability levels, where *remembering* is the least discriminatory. Such task-specific insights empower the decision-makers to better evaluate and understand the task, and highlight model strengths and limitations.

Similarly, Figure 4 shows a clustering of questions in the task t_{stk} based on semantic type (e.g., where, what). We observe that *What* and *Which* were the most discriminatory for lower ability levels, and *When* discriminated more at higher ability levels. One interpretation is that *What* and *How* questions tend to be more factual and syntax-based in the t_{stk} domain, and hence RAG with lower ability level struggle

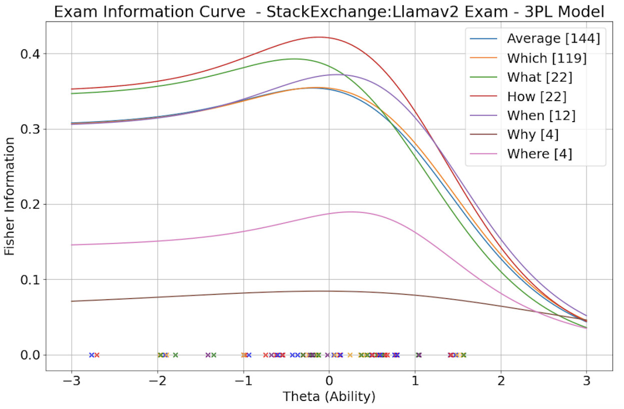

Figure 4. Aggregated Information function $\bar{I}_{cat}(\theta)$ for t_{stk}, averaged according to semantic taxonomy. Each cross on the x-axis correspond to a given model ability θ_m.

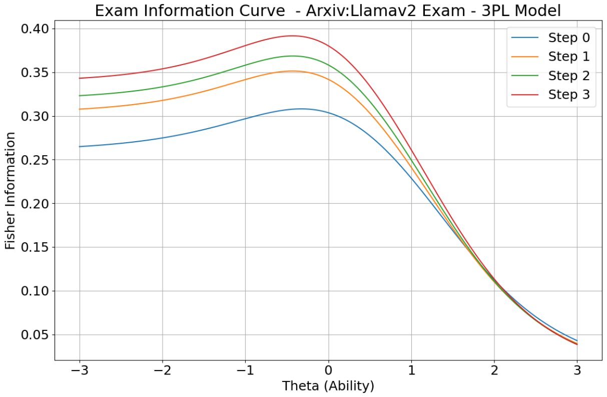

Figure 5. Evolution of Exam Informativeness during the iterative process, for ArXiv task. Each curve represents the exam aggregated Information function $\bar{I}_{\mathcal{Q}_i}(\theta)$, at step i.

more with these genres of questions. *When* question may also involve more situational logic where RAG with higher ability level are better equipped to answer. We refer the reader to Appendix B for further discussions and extended analysis of the Hierarchical IRT Model on the other tasks.

Moreover, we argue that our novel programmatic application of Bloom's Taxonomy and Item Response Theory provides a more comprehensive understanding of the exam's framework, thereby aiding practitioners in identifying potential biases. Specifically, Figure 3 showcases the distribution of question types (e.g., where, what, who), Figure 4 details the taxonomy of questions (e.g., creating, evaluating, remembering...), and Table 4 offers key statistics on questions and answers. Together, these elements offer fresh insights into the exam structure and contribute significantly to the identification and mitigation of biases.

6.3. Iterative Exam Improvement

Lastly, in order to increase the quality of the exam and thus better distinguish among the highest performing RAG pipelines, we introduce an iterative method to generate new exams $\mathcal{Q}_1 \mapsto \mathcal{Q}_2 \cdots \mapsto \mathcal{Q}_n$ by adaptively selecting questions to maximize the informativeness: $\bar{I}_{\mathcal{Q}_1}(\theta) \preceq \bar{I}_{\mathcal{Q}_2}(\theta) \cdots \preceq \bar{I}_{\mathcal{Q}_n}(\theta)$. More precisely, we apply an alternate process of IRT model fitting and question discarding based on the inferred discrimination parameter $(d_i)_{i \in \mathcal{Q}}$. The methodology is discussed in details in Appendix B.3. Figure. 5 illustrate the maximization process for t_{arx} as the exam and IRT estimation evolves; other tasks are presented in Appendix B.3. For t_{arx} or t_{ops}, we witness a continuous Pareto-dominating improvement, although mostly in the low to medium ability levels: the exam becomes more and more informative with the iterations. Such improvement is also witnessed for t_{stk}, although convergence happens faster. Finally, for t_{sec}, the evolution is non-monotonic and interestingly mostly happens in high ability regions. To conclude, this process is the first step towards a data-driven continuous optimization of the exam and we believe it is one of the most promising follow-up direction for the field of automated evaluation.

7. Conclusion

In this paper, we proposed and demonstrated a robust method to evaluate the performance of Retrieval-Augmented Large Language Models on specific tasks. By automatically generating multiple choice exams tailored to the document corpus associated with each task, our approach enables standardized, scalable, and interpretable scoring of different RAG systems. Through iterative optimization guided by Item Response Theory, we create highly informative exams that surface the strengths and weaknesses of different model configurations. Our experiments on question answering across four distinct domains reveal key insights into the factors driving RAG performance. Notably, we find that optimization of the retrieval mechanism can unlock bigger gains than simply scaling model size, highlighting the importance of a co-design approach. Overall, our work provides an efficient, reproducible paradigm for benchmarking and improving RAG for real-world applications.

Natural extensions of our work include investigating multi-language applications, incorporating agent-based systems for sequential decision-making tasks to extend beyond RAG systems, and utilizing the exam-based approach in traditional NLP problems like summarization and translation, thereby fostering the creation of more nuanced benchmarking datasets.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here. While enhanced factual accuracy for language models could have broad positive applications, we acknowledge there may also be risks if used improperly.

References

Adlakha, V., BehnamGhader, P., Lu, X. H., Meade, N., and Reddy, S. Evaluating correctness and faithfulness of instruction-following models for question answering. *arXiv preprint arXiv:2307.16877*, 2023.

Bender, E. M. and Koller, A. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 5185–5198, 2020.

Bird, S., Klein, E., and Loper, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., and Krathwohl, D. R. *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain*. McKay New York, 1956.

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., van den Driessche, G., Lespiau, J., Damoc, B., Clark, A., de Las Casas, D., Guy, A., Menick, J., Ring, R., Hennigan, T., Huang, S., Maggiore, L., Jones, C., Cassirer, A., Brock, A., Paganini, M., Irving, G., Vinyals, O., Osindero, S., Simonyan, K., Rae, J. W., Elsen, E., and Sifre, L. Improving language models by retrieving from trillions of tokens. *CoRR*, abs/2112.04426, 2021. URL https://arxiv.org/abs/2112.04426.

Bowman, S. R. and Dahl, G. E. What will it take to fix benchmarking in natural language understanding? In *2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pp. 4843–4855. Association for Computational Linguistics (ACL), 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bulian, J., Buck, C., Gajewski, W., Boerschinger, B., and Schuster, T. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. *arXiv preprint arXiv:2202.07654*, 2022.

Cai, L., Choi, K., Hansen, M., and Harrell, L. Item response theory. *Annual Review of Statistics and Its Application*, 3:297–321, 2016.

Chen, J., Lin, H., Han, X., and Sun, L. Benchmarking large language models in retrieval-augmented generation. *arXiv preprint arXiv:2309.01431*, 2023.

Deutsch, D., Dror, R., and Roth, D. A statistical analysis of summarization evaluation metrics using resampling methods. *Transactions of the Association for Computational Linguistics*, 9:1132–1146, 2021.

Embretson, S. E. and Reise, S. P. *Item response theory*. Psychology Press, 2013.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*, 2023.

Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., and Radev, D. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

Fu, J., Ng, S.-K., Jiang, Z., and Liu, P. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 12 2023a. URL https://zenodo.org/records/10256836.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023b.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. REALM: retrieval-augmented language model pretraining. *CoRR*, abs/2002.08909, 2020. URL https://arxiv.org/abs/2002.08909.

Hambleton, R. K., Swaminathan, H., and Rogers, H. J. *Fundamentals of item response theory*, volume 2. Sage, 1991.

Hoshi, Y., Miyashita, D., Ng, Y., Tatsuno, K., Morioka, Y., Torii, O., and Deguchi, J. Ralle: A framework for developing and evaluating retrieval-augmented large language models. *arXiv preprint arXiv:2308.10633*, 2023.

Howcroft, D. M., Belz, A., Clinciu, M., Gkatzia, D., Hasan, S. A., Mahamood, S., Mille, S., Van Miltenburg, E., Santhanam, S., and Rieser, V. Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *13th International Conference on Natural Language Generation 2020*, pp. 169–182. Association for Computational Linguistics, 2020.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.

Kamalloo, E., Jafari, A., Zhang, X., Thakur, N., and Lin, J. Hagrid: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv e-prints*, pp. arXiv–2307, 2023.

Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., and Lewis, M. Generalization through memorization: Nearest neighbor language models. *CoRR*, abs/1911.00172, 2019. URL http://arxiv.org/abs/1911.00172.

Koch, G., Zemel, R., Salakhutdinov, R., et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

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

Lee, N., Ping, W., Xu, P., Patwary, M., Fung, P. N., Shoeybi, M., and Catanzaro, B. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599, 2022.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401, 2020. URL https://arxiv.org/abs/2005.11401.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Liu, Y., Medlar, A., and Glowacka, D. What we evaluate when we evaluate recommender systems: Understanding recommender systems' performance using item response theory. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 658–670, 2023.

Lord, F., Novick, M., and Birnbaum, A. Statistical theories of mental test scores. 1968.

Martínez-Plumed, F., Prudêncio, R. B., Martínez-Usó, A., and Hernández-Orallo, J. Making sense of item response theory in machine learning. In *ECAI 2016*, pp. 1140–1148. IOS Press, 2016.

Martínez-Plumed, F., Prudêncio, R. B., Martínez-Usó, A., and Hernández-Orallo, J. Item response theory in ai: Analysing machine learning classifiers at the instance level. *Artificial intelligence*, 271:18–42, 2019.

Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P. W., Iyyer, M., Zettlemoyer, L., and Hajishirzi, H. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*, 2023.

Muhlgay, D., Ram, O., Magar, I., Levine, Y., Ratner, N., Belinkov, Y., Abend, O., Leyton-Brown, K., Shashua, A., and Shoham, Y. Generating benchmarks for factuality evaluation of language models. *arXiv preprint arXiv:2307.06908*, 2023.

Novikova, J., Dušek, O., Curry, A. C., and Rieser, V. Why we need new evaluation metrics for nlg. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2241–2252, 2017.

Papineni, K. Machine translation evaluation: N-grams to the rescue. In *LREC*, 2002.

Rasch, G. Studies in mathematical psychology: I. probabilistic models for some intelligence and attainment tests. 1960.

Robertson, S., Zaragoza, H., et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Saad-Falcon, J., Khattab, O., Potts, C., and Zaharia, M. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*, 2023.

Sedoc, J. and Ungar, L. Item response theory for efficient human evaluation of chatbots. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pp. 21–33, 2020.

Talmor, A. and Berant, J. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. *arXiv preprint arXiv:1905.13453*, 2019.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

Vanderbilt University Center for Teaching. Bloom's revised taxonomy. URL https://www.flickr.com/photos/vandycft/29428436431. CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=96306577.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Xu, M., Jiang, G., Liang, W., Zhang, C., and Zhu, Y. Interactive visual reasoning under uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)*, 3, 2023.

Yadav, N., Monath, N., Angell, R., Zaheer, M., and McCallum, A. Efficient nearest neighbor search for cross-encoder models using matrix factorization. *arXiv preprint arXiv:2210.12579*, 2022.

Yeung, C.-K. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. *arXiv preprint arXiv:1904.11738*, 2019.

Yu, J., Wang, X., Tu, S., Cao, S., Zhang-Li, D., Lv, X., Peng, H., Yao, Z., Zhang, X., Li, H., et al. Kola: Carefully benchmarking world knowledge of large language models. *arXiv preprint arXiv:2306.09296*, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

A. Details on Exam Generation

In this section, we present in details the exam generation process. Figure 6 is provided as a summary and includes an overview of the exam generation, evaluation and continuous improvement.The exam generator algorithm uses a pre-trained LLM to generate a multiple choice exam with n questions on a specific task $t \in \mathcal{T}$. In this work, we relied on LLamaV2-70B to generate the questions, after a preliminary comparative study with LlamaV2-13B, ClaudeInstant and ClaudeV2. We differ an extensive analysis of the difference to follow-up work. As shown in Figure 6, the algorithm has a three-step approach: first it generates candidate questions and answers pairs from a subset of all documents, either random or topic selected. It then filters the raw questions to remove low quality ones, notably by improving the quality of incorrect "discriminator" answer choices. Finally, it filters the correct questions to ensure maximal quality and add potential constraints on diversity. In this section, we illustrate the process used for exam generation (Section A.1) and question filtering (Section A.2). We present statistics on each exam in Section A.3. Next, in Section A.4, for each of the four tasks considered in the paper, we describe the task and exam specificity. Finally, we conclude by presenting the granular accuracy results for each task in Section A.5.

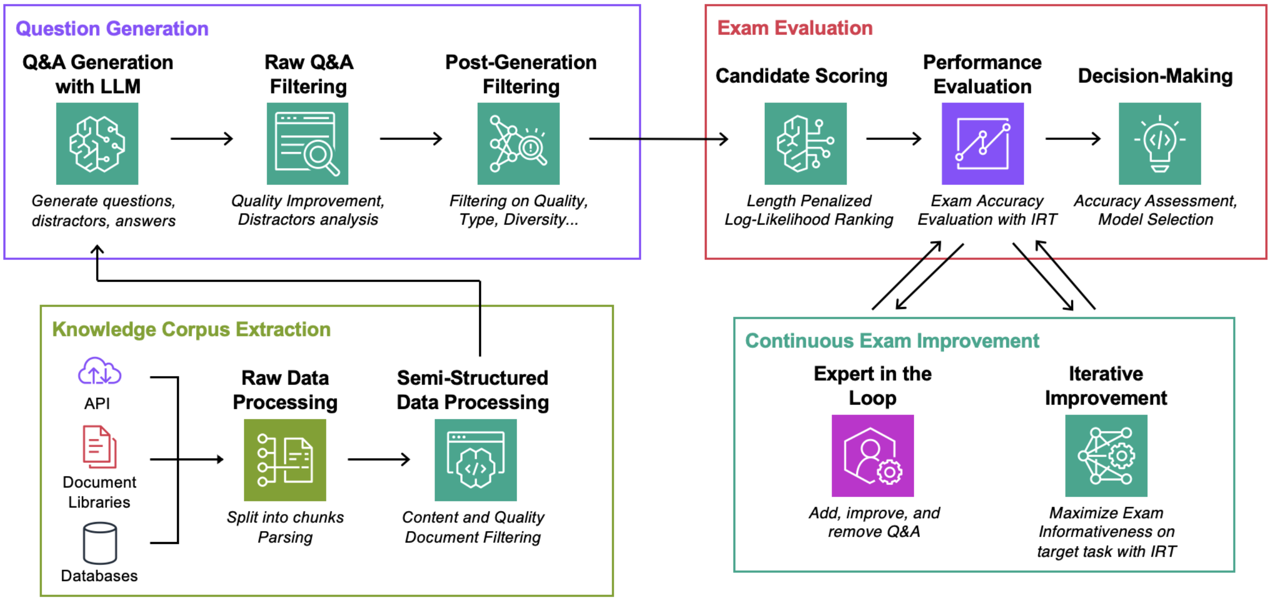
Question Generation
Q&A Generation with LLM
Generate questions, distractors, answers
Raw Q&A Filtering
Quality Improvement, Distractors analysis
Post-Generation Filtering
Filtering on Quality, Type, Diversity...
Exam Evaluation
Candidate Scoring
Length Penalized Log-Likelihood Ranking
Performance Evaluation
Exam Accuracy Evaluation with IRT
Decision-Making
Accuracy Assessment, Model Selection
Knowledge Corpus Extraction
API
Document Libraries
Databases
Raw Data Processing
Split into chunks Parsing
Semi-Structured Data Processing
Content and Quality Document Filtering
Continuous Exam Improvement
Expert in the Loop
Add, improve, and remove Q&A
Iterative Improvement
Maximize Exam Informativeness on target task with IRT

Figure 6. Summary of the exam generation, evaluation and iterative improvement processes.

A.1. Prompt for Exam Generation

There are two variables in the following prompt: `task_domain` and `documentation`. The former is one of the task names: DevOps, ArXiv, StackExchange, SEC Filings. The latter is the support documentation for the task.

```
Human: Here is some documentation from {task_domain}: {documentation}. From this, generate
a difficult multi-form question for an exam. It should have 4 candidates, 1 correct answer
and explanations.

Syntax should be:
Question: {question}
A) {candidate A}
B) {candidate B}
C) {candidate C}
D) {candidate D}
Correct Answer: {correct answer}

Assistant:
```

A.2. Question Filtering

Raw Question Parsing: To correctly extract the answer from the model generated answer, we use regular expression filters for both the question, the candidate choices and the correct answer. If one of the parsing fails, we discard the question. See Table 4 for some statistics on parsing efficiency.

Candidate Shuffling: We randomly shuffle the candidate answers order to ensure that random guessing performance is around 25%. This prevents a model bias of choosing the first candidate as the correct answer, as observed around 40% of the time during our experiments. See Table 4 for the baseline performance of (i) picking the same fixed answer for all questions, and (ii) always picking the longest answer. Aside from these two, we didn't detect any notable bias in terms of correct answer generation.

Self-contained Constraint: We use some regular expression checks to ensure that the question is self-contained, i.e. that is doesn't contain explicit references to the documentation used to generate it. In particular, we filter out questions with the following patters:

```
# term immediately followed by title in quotes
r'\b(documentation|paper|article|research|study)\b\s*\"[^\"]+\"'

# citation-like sentence followed by title
r'\b(discussed in|addressed in|described in|of the)\b\s*\"[^\"]+\"'

# fallback to original terms
r'\b(documentation|paper|article|research|study)\b'
```

Discriminator Analysis: As mentioned in Section 3.2, we note an interesting asymmetry: granted a document corpus, it is relatively easy for an LLM to generate a question and the correct answer, as this task is self-contained in the prompt in term of knowledge. However, it is considerably more difficult to create high quality incorrect answers, commonly refereed as discriminators. One bias of the model is to create either multiple rephrased correct answers or to create discriminators, which are correct but incomplete. To filter such questions out, for each question, we use Jaccard similarity at n-gram level $\mathcal{J}_n$ (where n is picked as the mean candidate answer token length), as well as embedding similarity $\mathcal{S}$. Given an exam question $q \in \mathcal{Q}$ based on documentation k, the correct answer c and the discriminators d_i, we introduce two filters:

- **Extra-Candidate Similarity**: If $\mathcal{J}(k, c) + t_1 < \max_i \mathcal{J}(k, d_i)$, or $\mathcal{S}(k, c) + t_2 < \max_i \mathcal{S}(k, d_i)$, for t_1, t_2 two threshold values, we remove the question q from the corpus. This check ensures that discriminators are not closer in meaning to the documentation compared to the original question. In practice, we obtain the best results by setting values of t_1, t_2 such that around 5% of questions are removed.
- **Intra-Candidate Similarity**: If $\max_i \mathcal{J}_n(c, d_i) \geq t_3$ or $\max_i \mathcal{S}(c, d_i) \geq t_4$, for t_3, t_4 two threshold values, we remove the question q from the corpus. This check ensures that discriminators are not too close to the original question. In practice, we obtain the best results by setting values of t_3, t_4 such that around 5% of questions are removed.

A.3. Generated Exam Statistics

Attribute	t_{ops} **(DevOps)**	t_{arx} **(Arxiv)**	t_{stk} **(StackExchange)**	t_{sec} **(SEC Filings)**
# Candidate Q&A Generated	700	500	326	520
# Incorrectly Parsed Q&A	126	119	143	35
# Incorrect Q&A (Content)	99	32	5	41
# Surviving Q&A	275	381	148	515
Fixed Answer Baseline Accuracy	27.6%	27.6%	29.7%	27.4%
Longest Answer Baseline Accuracy	36.0%	37.3%	38.51%	41.6%
Avg. Question Length (character)	303	355	270	141

Table 4. Analysis of the exam generation process. We present both data on the exam generation and refinement process, as well as on the resulting exam characteristic.

To edge against LLM biases in question generation, we compute in Table 4 two baselines. First, the fixed answer baseline assesses the score of always picking the same given answer across all questions. By randomising the candidate answers order during the exam generation, we ensure that this baseline accuracy is around 25% as expected. Secondly, the longest answer baseline is the score obtained by a model always picking the longer answer. Although we observe a higher score than a random pick, the value is still reasonably low not to motivate a programmatic correction. No other potential biases of this type were detected during our analysis.

A.4. Knowledge Corpus and Task Creation

Task t_{ops} (DevOps). The task t_{ops} is defined over a knowledge corpus of 1249 webpages from AWS Knowledge Center [4] where each webpage troubleshoots one DevOps problem for AWS customers. Tags associated with the HTML code allows to cluster the questions on 18 overlapping topics. To create document chunks, we break down the documents in non-overlapping chunks to ensure a maximum of 10 sentences per chunk, a character length between 500 and 4500 and a token length between 200 and 900. This results in a total of 4564 document chunks. The following example is a question generated for an AWS cloud storage service called S3.

```
{
  "question": "You are an AWS engineer responsible for monitoring the storage usage of
    your company's Amazon S3 buckets. You want to track the total storage usage and number
     of objects in each bucket. Which of the following metrics in CloudWatch should you
    use to achieve this?",
  "documentation": "However, as soon as the objects are marked for deletion, you are no
    longer billed for storage (even if the object isn't removed yet). Note that the Amazon
     S3 monitoring metrics are recorded once per day. Therefore, these metrics might not
    display the most updated information. However, CloudWatch monitors your AWS resources
    and applications in real time . Also, S3 console and Storage Lens use base 2
    conversion (/1024) to report storage metrics, and CloudWatch by default uses base 10
    conversion (/1000). Resolution Daily storage metrics in CloudWatch In CloudWatch, the
    BucketSizeBytes metric captures all Amazon S3 and Amazon S3 Glacier storage types,
    object versions, and any incomplete multipart uploads. This value is calculated by
    summing up all object sizes, metadata in your bucket (both current and noncurrent
    objects), and any incomplete multipart upload sizes. For example, the BucketSizeBytes
    metric calculates the amount of data (in bytes) that's stored in an Amazon S3 bucket
    in all the following object storage classes : S3 Standard S3 Intelligent-Tiering S3
    Standard-IA S3 One Zone-IA S3 Reduced Redundancy Storage S3 Glacier Deep Archive S3
    Glacier Flexible Retrieval S3 Glacier Instant Retrieval Additionally, the
    NumberOfObjects metric in CloudWatch contains the total number of objects that are
    stored in a bucket for all storage classes. This value counts all objects in the
    bucket (both current and noncurrent), along with the total number of parts for any
    incomplete multipart uploads. The NumberOfObjects metric also calculates the total
    number of objects for all versions of objects in your bucket.",
  "choices": [
    "A) BucketSizeBytes",
    "B) ObjectVersionBytes",
    "C) IncompleteMultipartUploads",
    "D) NumberOfObjects"
  ],
  "correct_answer": "A) BucketSizeBytes"
}
```

Next, we present a random sample of 10 questions from the DevOps exam:

```
Question 1: You are an AWS engineer responsible for setting up a site-to-site VPN
    connection between your company's network and Amazon VPC. You have configured the VPN
    tunnel, but it is not establishing successfully. You have checked the AWS VPN
    configuration and found that it meets all the requirements mentioned in the
    documentation. However, you are still experiencing issues. Which of the following
    could be the cause of the problem?
Question 2: You are an AWS engineer responsible for managing an ECS cluster. You are
    receiving errors when trying to add tags to the cluster. The errors indicate that the
```

[4] *AWS Knowledge Center: https://repost.aws/knowledge-center.*

```
    IAM entity does not have the necessary permissions. Which of the following steps
    should you take to resolve this issue?
Question 3: You are an AWS engineer responsible for optimizing the performance of a
    DynamoDB table used by a web application. You've identified that the table is
    experiencing high latency, and you suspect that the DAX cache is not being effectively
     utilized. Which of the following actions would you take to optimize the cache usage
    and reduce latency?
Question 4: You are an AWS administrator and you need to ensure that the permissions for
    the Linux home directory, user's home directory, .ssh directory, and authorized_keys
    file are correct for an EC2 instance. Which of the following commands should you run
    to achieve this?
Question 5: Suppose you are an AWS engineer responsible for troubleshooting a connectivity
     issue between an application and an Amazon RDS database. The application is
    configured to use a custom parameter group, and the database instance is running in a
    Multi-AZ deployment. After reviewing the documentation, which of the following steps
    would you take FIRST to resolve the issue?
Question 6: You are the administrator of an AWS S3 bucket named 'DOC-EXAMPLE-BUCKET'. You
    have configured CloudFront to serve objects from this bucket. When a user requests the
     object 'index.html' from CloudFront, they receive an error message saying that the
    object is not found. Which of the following steps should you take to resolve this
    issue?
Question 7: You are an AWS administrator responsible for managing access to AWS resources
    across multiple accounts. You have been tasked with troubleshooting an issue where a
    user is unable to copy an object from one bucket to another. The error message
    indicates that the user lacks the necessary permissions. You have identified the
    following information:
        * The source bucket is owned by Account A.
        * The destination bucket is owned by Account B.
        * The object is owned by Account C.
        * The user attempting to copy the object is in Account D.
    Which of the following steps should you take to resolve the issue?
Question 8: A user has registered a new domain name, but it is not resolving on the
    internet. They check the domain's status using the whois command and see that it is in
     'clientHold' status. What should the user do to make the domain available on the
    internet again?
Question 9: You are a developer troubleshooting latency issues for an edge-optimized API
    endpoint in Amazon API Gateway. You have identified the following parts of the
    connection path and their corresponding durations:
        1. Start of connection to the DNS name resolution: 200 ms
        2. Start of connection to the Transmission Control Protocol (TCP) handshake to
    connect to CloudFront: 300 ms
        3. Start of connection to the Secure Sockets Layer (SSL) handshake to connect to
    CloudFront: 400 ms
        4. Start of connection to sending the client HTTP request to CloudFront: 500 ms
        5. Start of connection to the first byte transferred from CloudFront: 600 ms
        6. Total time for the request and response to the API: 1000 ms
        7. Time for API Gateway to process the request and respond to the CloudFront edge
    location: 300 ms
        8. Time for the integration endpoint to respond to the HTTP request from API
    Gateway: 400 ms
        9. Time for API Gateway to respond to the CloudFront edge location, and for
    CloudFront to respond to the client: 300 ms
    Which part of the connection path is the source of the latency for the edge-optimized
    API endpoint?
Question 10: You are a cloud engineer responsible for deploying a web application using
    AWS Lambda@Edge and CloudFront. You have associated the Lambda@Edge function with the
    CloudFront distribution, but you are experiencing 500, 502, and 503 errors. Which of
    the following steps should you take to troubleshoot the errors?
```

Task t_{arx} (Arxiv). The task t_{arx} is defined over 13 000 ArXiv papers where each paper is represented by its abstract. More precisely, for each of the 13 most commonly used research tags, we randomly sample 1 000 papers published before 2021. We require the abstract to have a character length between , 000 and 1500. The following example is a question generated for the domain of astrophysics and astronomical instrumentation.

```
{
  "question": "The Swift team announced an update to the UltraViolet and Optical Telescope calibration to correct for the loss of sensitivity over time. What is the impact of this update on observations in the three near ultraviolet (UV) filters?",
  "documentation": "A Swift Fix for Nuclear Outbursts. In November 2020, the Swift team announced an update to the UltraViolet and Optical Telescope calibration to correct for the loss of sensitivity over time. This correction affects observations in the three near ultraviolet (UV) filters, by up to 0.3 mag in some cases. As UV photometry is critical to characterizing tidal disruption events (TDEs) and other peculiar nuclear outbursts, we re-computed published Swift data for TDEs and other singular nuclear outbursts with Swift photometry in 2015 or later, as a service to the community. Using archival UV, optical, and infrared photometry we ran host SED fits for each host galaxy. From these, we computed synthetic host magnitudes and host-galaxy properties. We calculated host-subtracted magnitudes for each transient and computed blackbody fits. In addition to the nuclear outbursts, we include the ambiguous transient ATLAS18qqn (AT2018cow), which has been classifed as a potential TDE on an intermediate mass black hole. Finally, with updated bolometric light curves, we recover the relationship of \\citet{hinkle20a}, where more luminous TDEs decay more slowly than less luminous TDEs, with decreased scatter as compared to the original relationship.",
  "choices": [
    "A) It decreases the sensitivity of observations in the UV filters by up to 0.3 mag.",
    "B) It corrects for the loss of sensitivity over time, but the magnitude of the correction varies depending on the filter.",
    "C) It has no impact on observations in the UV filters.",
    "D) It increases the sensitivity of observations in the UV filters by up to 0.3 mag."
  ],
  "correct_answer": "B) It corrects for the loss of sensitivity over time, but the magnitude of the correction varies depending on the filter."
}
```

Below, we present a sample of 10 questions from the ArXiv exam:

```
Question 1: Consider the following model for market inefficiency:
    $$\frac{dX_t}{dt} = \mu + \sigma X_t \sqrt{1 - \frac{1}{2} \sigma'^2} \epsilon_t$$
    where $X_t$ is the log market price, $\mu$ is the drift term, $\sigma$ is the volatility term, $\sigma'$ is the volatility of the reasonable price, and $\epsilon_t$ is a standard Brownian motion.
    Which of the following statements is true about the behavior of the market price in this model?
Question 2: A distributed energy resource management system (DERMS) is using network topology identification (TI) to organize and operate widespread distributed energy resources (DERs). The TI function relies only on the measurements available to DERMS. Which of the following approaches is used in the proposed method to improve the resiliency of TI against interruption of communication channels?
Question 3: Consider a network with a limited number of target nodes that need to be controlled. The state variables of the system are associated with the nodes of the network. The goal is to control the target variables as time functions. Which of the following approaches would be most appropriate to achieve this goal?
Question 4: Which of the following statements best describes the virial equation of state of low-density nuclear matter, as presented in the reference?
Question 5: Which of the following best describes the main idea behind the method presented in the manuscript for reconstructing the 3D structure of chromosomes from Hi-C and GAM data?
Question 6: Suppose you are analyzing a large dataset of financial variables and want to test for group-specific heterogeneity in a high-dimensional factor model. Which of the following tests would you use, and why?
Question 7: The unidirectional spin heat conveyer effect in a 200nm thin Yttrium Iron Garnet film is investigated using lock-in thermography. Which of the following statements is true regarding the observed temperature profiles?
Question 8: In the context of the evolutionary prisoner's dilemma game, what can be inferred about the impact of deceitful behavior on the evolutionary outcomes in structured populations?
Question 9: The ATLAS detector at the CERN Large Hadron Collider was used to measure the cross-section of high transverse momentum $W$ and $Z$ bosons produced in $pp$
```

```
    collisions and decaying to all-hadronic final states. Which of the following
    statements best describes the method used to reconstruct the boosted $W$ or $Z$ bosons
     in the analysis?
Question 10: Consider a network of nodes where each node represents a cluster in a
    competitive environment. The connections between nodes represent non-local
    interactions between clusters. The probability of a connection between two nodes is
    $p$. The goal is to devise a survival strategy for an arbitrarily chosen cluster.
    Which of the following strategies is most likely to be effective?
```

Task t_{stk} **(StackExchange).** The task t_{stk} is defined over 977 StackExchange question and answer pairs, randomly sampled from 18 main tags. To generate a self contained documentation corpus, we artificially join the question and the most upvoted answer, as shown in the documentation example bellow. We only requires the resulting paragraph to have a character length bellow 1500. The following example is a question generated for the domain of Salesforce development, specifically focusing on Visualforce and formula fields.

```
{
  "question": "You are working on a project that requires you to use the ‘MailingAddress‘
    field from the contact object in your visual force email templates and formula fields.
     However, you are unable to use this field directly. What should you do instead?",
  "documentation": "### User: I have to use the Contact Object’s Mailing address in my
    code. But I am not able to use it in my visual force email templates or formula fields
    . Please suggest how can this field be used\\n\\n -\\n\\n### Top Answer: The ‘
    MailingAddress‘ is a special field on the contact - it’s a group of multiple fields.
    Instead you need to use ‘MailingStreet‘, ‘MailingCity‘, ‘MailingState‘, ‘
    MailingPostcode‘ and ‘MailingCountry‘. All these fields together form the ‘
    MailingAddress‘ field.",
  "choices": [
    "A) Use the ‘MailingStreet‘, ‘MailingCity‘, ‘MailingState‘, ‘MailingPostcode‘, and ‘
    MailingCountry‘ fields separately in your code.",
    "B) Use a third-party library to format the mailing address into a single string.",
    "C) Create a custom object that contains the mailing address information and use that
    object in your code.",
    "D) Use a combination of the ‘MailingStreet‘, ‘MailingCity‘, ‘MailingState‘, ‘
    MailingPostcode‘, and ‘MailingCountry‘ fields to create a new field that represents
    the full mailing address."
  ],
  "correct_answer": "A) Use the ‘MailingStreet‘, ‘MailingCity‘, ‘MailingState‘, ‘
    MailingPostcode‘, and ‘MailingCountry‘ fields separately in your code."
}
```

Below, we present a sample of 10 questions from the StackExchange exam:

```
Question 1: Which of the following is a good resource for finding open-source projects
    related to processing primitives such as FFT, convolution, correlation, and matrix
    mathematics for machine vision?
Question 2: Which of the following commands will produce the output *M*M with proper
    spacing between the two M’s?
Question 3: You are developing an Android app that uses the PullToRefresh library. You
    have added a ‘PullToRefreshListView‘ to your layout, but you notice that the white
    separators between the items (dividers) have disappeared. What is the most likely
    solution to this problem?
Question 4: You are developing a Java application that uses JavaFX, and you want to
    distribute it to other computers. You have tried using the ‘--module-path‘ and ‘--add-
    modules‘ options to include the necessary JavaFX libraries, but it only works when you
     use the full path to the SDK library. You have also tried copying the lib file to the
     application folder and using the path to it, but it still doesn’t work. What should
    you do to make the application run on other computers?
Question 5: You are working on a project that requires you to use the ‘MailingAddress‘
    field from the contact object in your visual force email templates and formula fields.
     However, you are unable to use this field directly. What should you do instead?
Question 6: You are tasked with finding all files in a directory that do not have group
    write permissions. Which of the following commands would you use to accomplish this?
Question 7: You are given a table with date and maximum temperature data for a certain
    location over a period of several years. You are asked to find the average maximum
```

```
temperature for each day of the year over the given period.
    Which of the following SQL queries would correctly answer this question?
Question 8: You are an AWS solutions architect, and you have been tasked with designing an IAM policy that allows users from another AWS account to access a specific resource in your account. You have been given the following requirements:
        * The users in the other account should only be able to access the resource if they have been granted permission to do so by the resource owner.
        * The resource owner should be able to grant permission to the users in the other account using IAM roles.
        * The users in the other account should not be able to access any other resources in your account.
    Which of the following options meets all of the above requirements?
Question 9: Which of the following options is the best approach for handling null values when retrieving the ID of a fragment in an Android application?
Question 10: You are given a controller method that removes a user from a group. The method has a parameter ‘group_id‘ that is passed as a route parameter. The method uses the ‘where‘ method to find the group with the given ID and then calls ‘destroy‘ on the resulting object. However, the method is not working as expected and is returning an error.
    Which of the following is the correct fix for the method?
```

Task t_{sec} (SEC Filings). The task t_{sec} is defined on 188 documents submitted yearly to the U.S. Securities and Exchange Commission (SEC) by 10 publicly traded companies, company insiders, and brokers. Each document is segmented in 20 sections, and we only select the ones on “Business Overview (Section 1)”, “Risk Factors (Section 2)” and “Management’s Discussion and Analysis of Financial Condition (Section 9)”, for a total of 493 sections. We break down these sections into documentation chunks by creating set of 6 non-overlapping sentences, with a a minimal character length of 1000 and a maximal of 1400. Finally, we modify the introduction of the documentation corpus to add some brief context on the company (e.g., “*Here is an extract of a SEC filing from* $\{$*COMPANY NAME*$\}$*:)*”).

```
{
  "question": "What is the main reason for the establishment of a valuation allowance against the remainder of AMD’s U.S. deferred tax assets, net of U.S. deferred tax liabilities, in the fourth quarter of 2002?",
  "documentation": "Here is an extract of a SEC filing from AMD (ADVANCED MICRO DEVICES):
 The 2002 income tax provision was recorded primarily for taxes due on income generated in certain foreign tax jurisdictions and the establishment of a valuation allowance against the remainder of our U.S. deferred tax assets, net of U.S. deferred tax liabilities in the fourth quarter, due to continuing substantial operating losses in the United States. As of December 26, 2004, we had federal and state net operating loss carryforwards of approximately $930 million and $45 million. We also had foreign loss carryforwards of approximately $88 million. We also had federal and state tax credit carryforwards of approximately $246 million and $86 million. The net operating loss and tax credit carryforwards will expire at various dates beginning in 2005 through 2024, if not utilized. We maintain a full valuation allowance against all our net U.S. federal and state deferred tax assets and certain of our foreign deferred tax assets ($694 million at December 26, 2004) because of our history of recent losses.",
  "choices": [
    "A) To record the tax provision for the year 2002.",
    "B) To provide for the utilization of net operating loss carryforwards.",
    "C) To reflect the continuing substantial operating losses in the United States.",
    "D) To offset the taxes due on income generated in certain foreign tax jurisdictions."
  ],
  "correct_answer": "C) To reflect the continuing substantial operating losses in the United States."
}
```

Below, we present a sample of 10 questions from the SEC Filings exam:

```
Question 1: Which of the following is NOT a factor contributing to the AAR CORP’s expected
     increase in revenue in fiscal 2016?
Question 2: What is the likelihood of the pending cases having a material adverse impact
    on the CECO Environmental Corp’s results of operations, liquidity, or financial
    condition?
Question 3: What is the primary reason for the increase in the net financial instrument
    position of Air Products and Chemicals between September 30, 2008, and September 30,
    2009?
Question 4: According to the SEC filing from Air Products and Chemicals, what is the
    vesting period for market-based deferred stock units?
Question 5: What is the primary energy source used by Air Products and Chemicals in the
    production of atmospheric gases such as oxygen, nitrogen, and argon?
Question 6: How does CECO Environmental Corp measure the cost of employee services
    received in exchange for an award of equity instruments?
Question 7: What is the measurement date for the projected benefit obligations of AAR CORP
    ’s pension plans?
Question 8: What was the total expense recorded by Air Products and Chemicals for business
     restructuring and cost reduction plans in 2012?
Question 9: According to the SEC filing from AMD, how are price reductions handled in
    relation to product cost?
Question 10: According to the SEC filing, what is the basis for the lease rental rate for
    the corporate office space leased by Adams Resources & Energy from an affiliated
    entity?
```

A.5. Granular Analysis of Results

We next present a granular representation of the exam accuracy for each task (Figure 7). Depending on the task and topic, we note that model and retrieval performance greatly varies. These insights are helpful to better understand strengths and limitations of models across several factual directions.

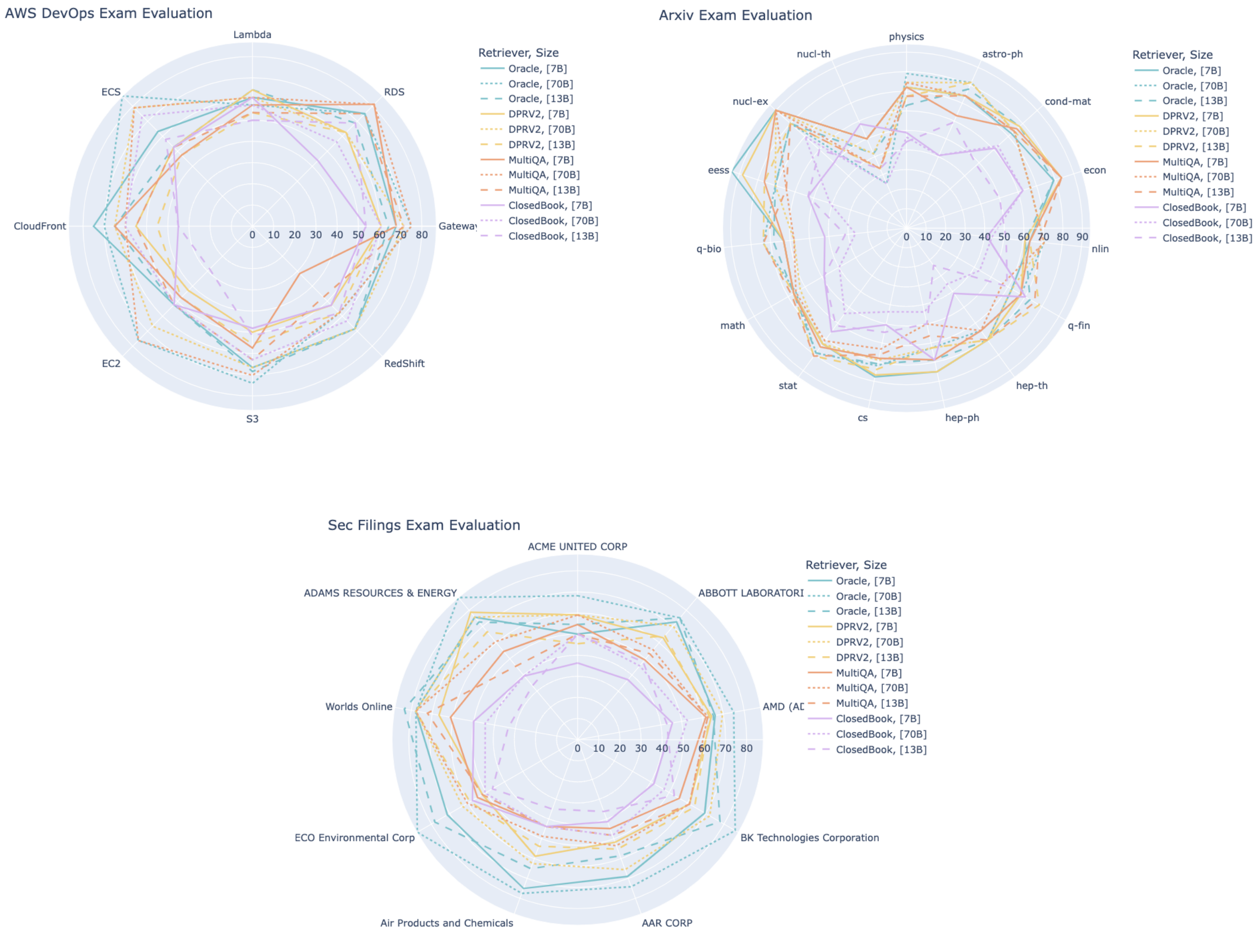

Figure 7. Granular results of our exam evaluation for different tasks: t_{ops} on top left, t_{arx} on top right, and t_{sec} on the bottom. Accuracy is reported for different retrieval approaches and retriever sizes, on a % scale. More details in captions of individual plots. Labels on the diameter shows the different companies. Colors correspond to different retrieval approaches (`Oracle`, `DPRV2`, `MultiQA`, `ClosedB`, as discussed in Section 4.2) and patterns correspond to the base LLM size (Mistral-7B, LlamaV2-13B, and LlamaV2-70B).

B. Item Response Theory Model Estimation

B.1. Item Response Theory Model Fit

To minimize the negative log-likelihood, we leverage L-BFGS-B solver. We initialize at 0 all the values of θ_m (either for RAG model in the classical IRT or latent variables for the hierarchical model), at 1 the discrimination $(a_i)_{i \in Q}$, at 0 the difficulty $(b_i)_{i \in Q}$ and at 0.25 the guessing $(c_i)_{i \in Q}$. We enforce the following constraints: $0.1 \leq a_i \leq 1.5$, $0.01 \leq b_i \leq 1$, $0.2 \leq c_i \leq 0.4$ and $-3 \leq \theta_k \leq 3$. Finally, as a form of regularization, we experimented with adding a log-normal prior on a, a normal prior on b and a beta prior on c, as commonly done in the literature. Yet, these priors led to marginal impact and were removed in the final version. We refer the reader to the source code at https://github.com/amazon-science/auto-rag-eval for the detailed implementation.

Attribute	t_{ops} **(DevOps)**	t_{arx} **(Arxiv)**	t_{stk} **(StackExchange)**	t_{sec} **(SEC Filings)**
Average Exam Accuracy	$50.8\% \pm 11.7$	$60.0\% \pm 12.2$	$56.1\% \pm 12.2$	$53.5\% \pm 11.1$
RMSE - Mean Prediction Baseline	0.49 ± 0.03	0.47 ± 0.06	0.48 ± 0.04	0.49 ± 0.03
RMSE - IRT	0.44 ± 0.05	0.42 ± 0.06	0.43 ± 0.05	0.42 ± 0.04
Discrimination - IRT	1.10 ± 0.33	1.09 ± 0.37	1.16 ± 0.44	1.01 ± 0.07
Difficulty - IRT	0.47 ± 0.39	0.40 ± 0.44	0.49 ± 0.46	0.22 ± 0.17
Guessing - IRT	0.30 ± 0.01	0.30 ± 0.09	0.30 ± 0.09	0.30 ± 0.09
Theta - IRT	-0.61 ± 1.11	0.14 ± 0.77	-0.10 ± 1.04	-0.79 ± 0.87

Table 5. Statistics on Hierarchical IRT Model Fit. We present the average value (plus/minus a standard deviation) either across questions (Discrimination/Difficulty/Guessing), models (Theta) or both (RMSE). As baseline, we compare to a fixed prediction: picking the average exam performance (Mean Prediction Baseline). RMSE refers to Root Mean Square Error.

Next, Table 5 presents results on the Hierarchical Model fit resulting from the estimation procedure. More precisely, the table highlights:

- The average exam scores of all RAG models, including standard deviations (as a column-wise average from Table 2), allowing readers to easily assess and compare the difficulty of different tasks.
- A comparison of model fit, specifically $P(X = 1|\theta)$, between the Hierarchical IRT model and a baseline across various questions, using RMSE. This comparison, showing a consistent improvement in fit for the Hierarchical IRT model, enables readers to gauge the IRT model's performance on specific tasks.
- The average of inferred question-specific parameters from the IRT model, as described in Equation 1, across all exams. This helps elucidate the unique characteristics of individual questions within the broader question corpus. Notably, these values necessitate further normalization for meaningful comparisons across tasks, elucidating, for instance, why the t_{ops} task appears less difficult than t_{stk}, despite higher average exam scores.
- The average of the inferred model ability θ, across all RAG models for each task. This aggregation aids in understanding how individual model capabilities stack up against the broader model set. Again, direct task comparisons between tasks may not be applicable.

To conclude this section, a noteworthy follow-up question is on the exact relationship between model performance and the length and size of the corpus. More precisely, are models performing better if the size of a corpora to be searched with is smaller/larger? A thorough answer to this question would involve considering the length and chunk size of the corpus as design variables and study the downstream impact on the performance, for a fixed exam. As seen in Table 1, we chose 4 tasks with a representative variation in the corpus size (from 977 to 13,000 documents) and document length (from 144 to 254 words). For our initial experiments, we noticed that the variations around these values led to second order differences in performance and thus decided to commit to a given value. For this reasons and given space and focus constraints, we decided to differ an extensive analysis to follow-up work.

Yet, as discussed in Section 3.3, our hierarchical framework naturally allows to answer these question in a quantitative way and for any task of interest. More precisely, if we were to address this, we would postulate first a parabolic relation between the ability level variable θ and the corpus length and assess the model fit. This epitomizes a model where the accuracy first increases with the corpus size, due to documentation overlap and then decreases above a certain size.

B.2. Task-based Item Response Theory Model Results

In this section, we present for all four tasks both individual question characteristic curves $p_i(\theta)$, item response functions $I(\theta|g_i, d_i, b_i)$ of $p_i(\theta)$ and aggregated aggregated Information function $\bar{I}_{cat}(\theta)$.

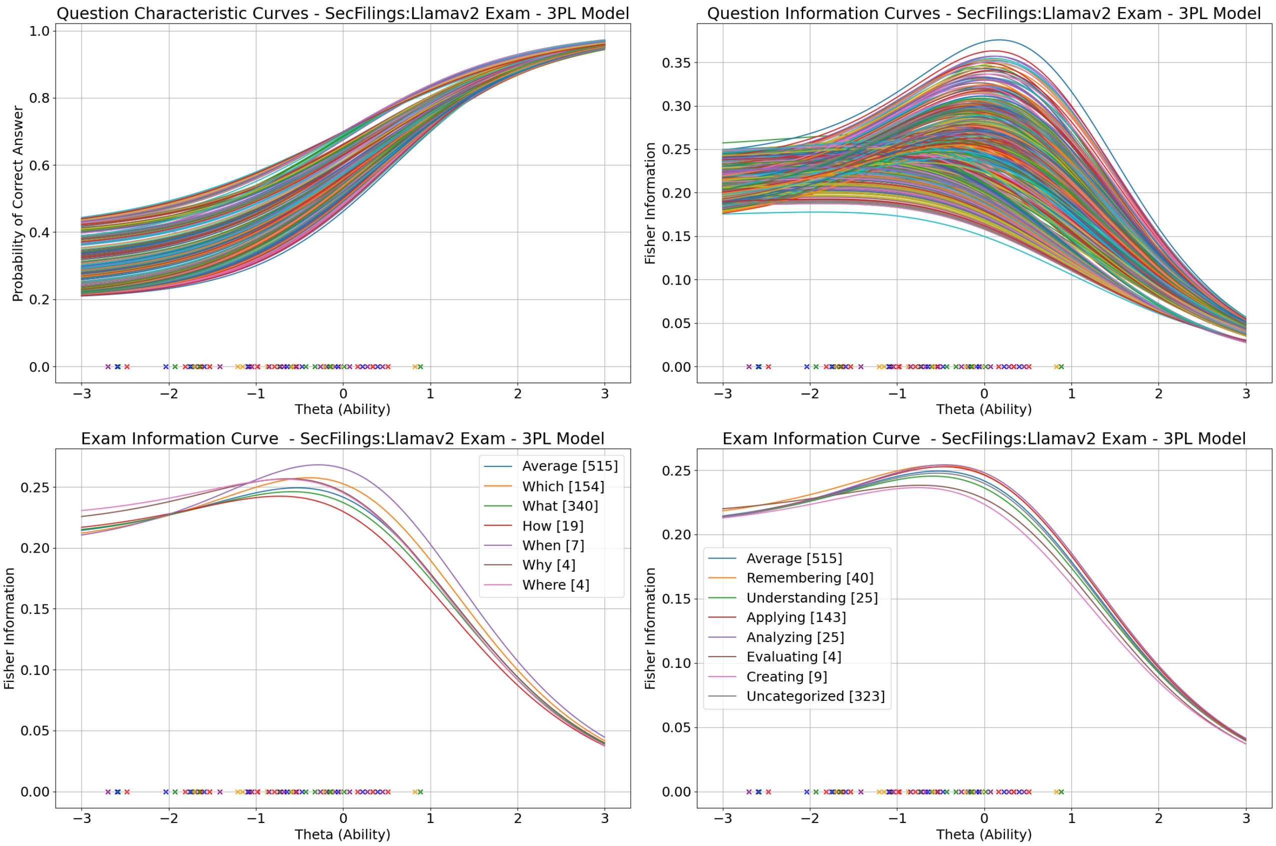

Figure 8. Hierarchical IRT Analysis Results for Sec Filings task t_{sec}. (*Upper-Left*) Modeling $p_i(\theta)$, where each line corresponds to the Item Response Function for a given question $q_i \in \mathcal{Q}$. (*Upper-Right*) Aggregated Information function $I(\theta|g_i, d_i, b_i)$ of $p_i(\theta)$, where each line to a given question $q_i \in \mathcal{Q}$. (*Lower-Left*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to semantic taxonomy. (*Lower-Right*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to Bloom taxonomy. For all graphs, each cross on the x-axis correspond to a given model ability θ_m.

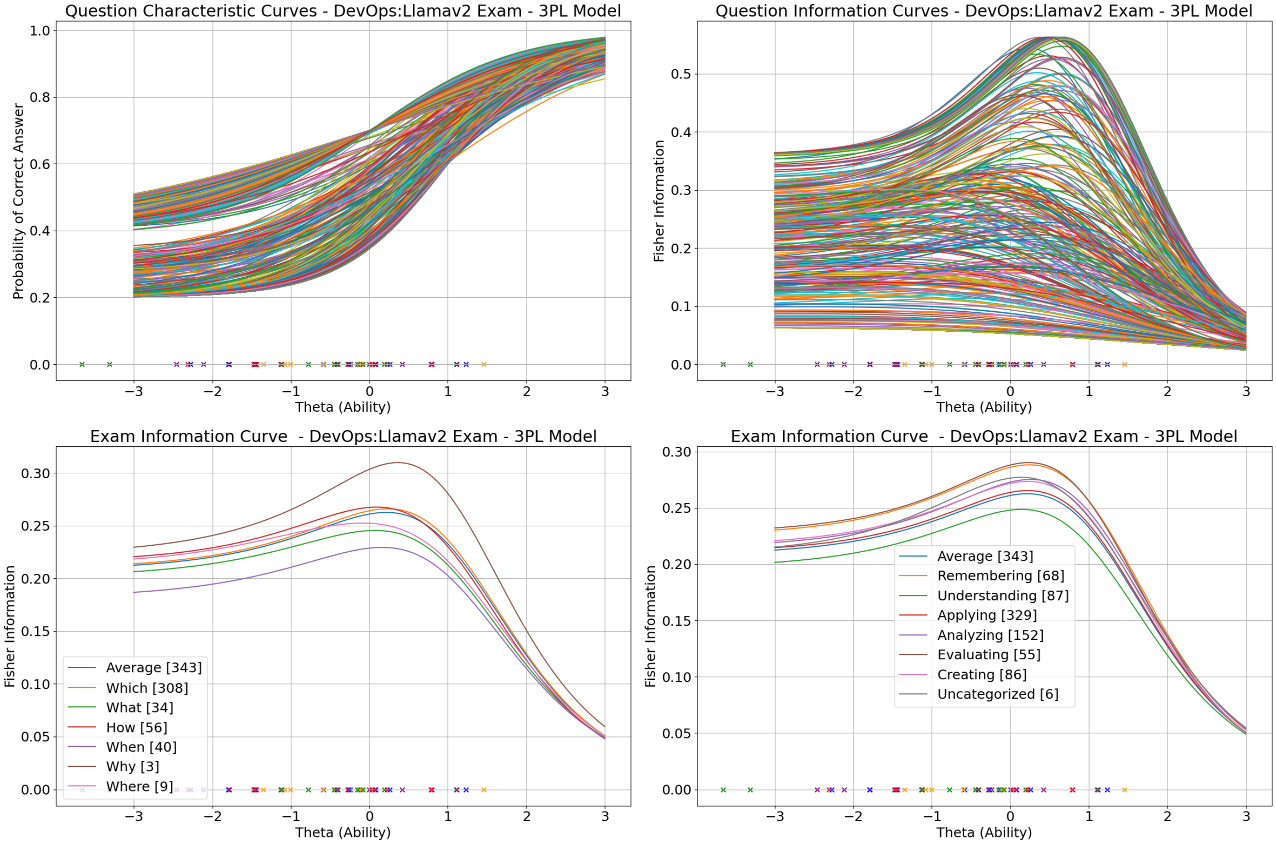

Figure 9. Hierarchical IRT Analysis Results for DevOps task t_{ops}. (*Upper-Left*) Modeling $p_i(\theta)$, where each line corresponds to the Item Response Function for a given question $q_i \in \mathcal{Q}$. (*Upper-Right*) Aggregated Information function $I(\theta|g_i, d_i, b_i)$ of $p_i(\theta)$, where each line to a given question $q_i \in \mathcal{Q}$. (*Lower-Left*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to semantic taxonomy. (*Lower-Right*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to Bloom taxonomy. For all graphs, each cross on the x-axis correspond to a given model ability θ_m.

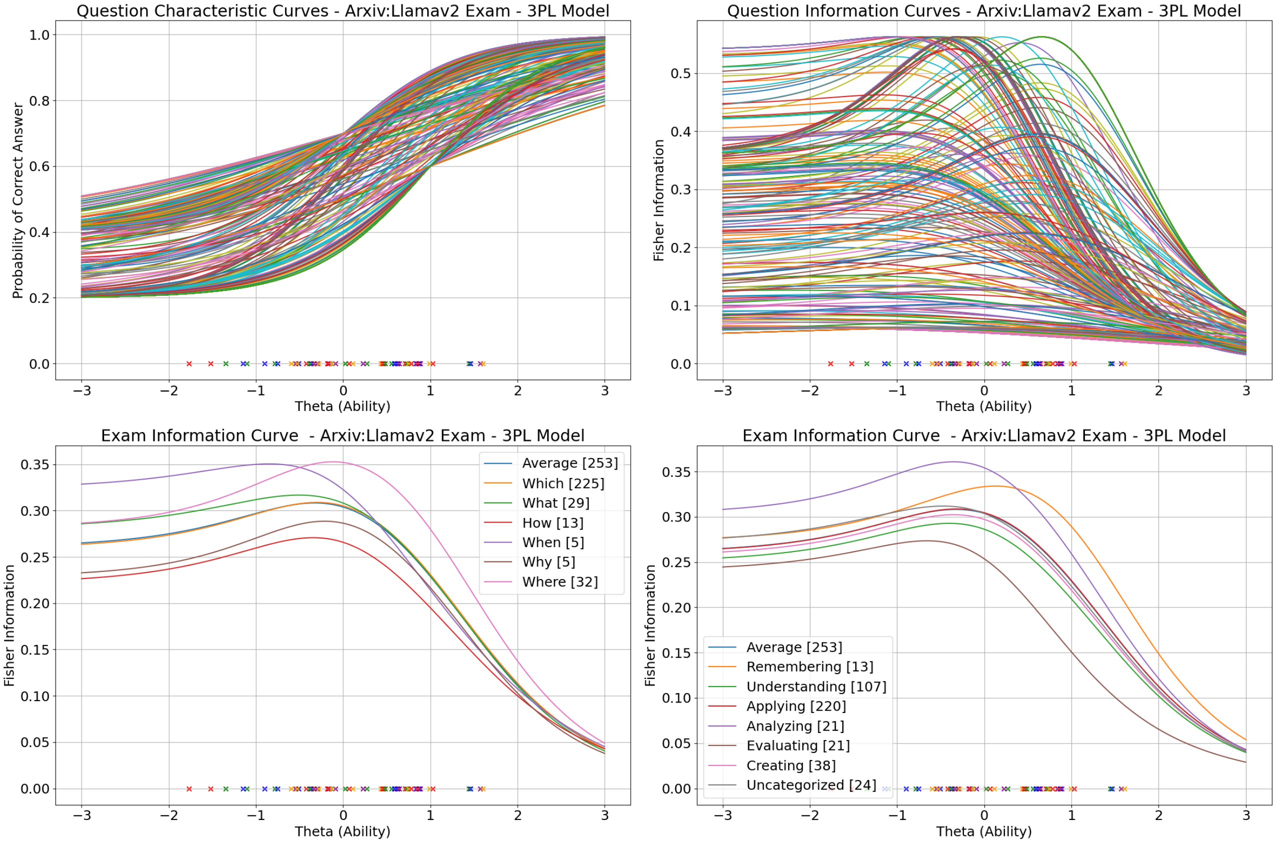

Figure 10. Hierarchical IRT Analysis Results for Arxiv task t_{arx}. (*Upper-Left*) Modeling $p_i(\theta)$, where each line corresponds to the Item Response Function for a given question $q_i \in \mathcal{Q}$. (*Upper-Right*) Aggregated Information function $I(\theta|g_i, d_i, b_i)$ of $p_i(\theta)$, where each line to a given question $q_i \in \mathcal{Q}$. (*Lower-Left*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to semantic taxonomy. (*Lower-Right*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to Bloom taxonomy. For all graphs, each cross on the x-axis correspond to a given model ability θ_m.

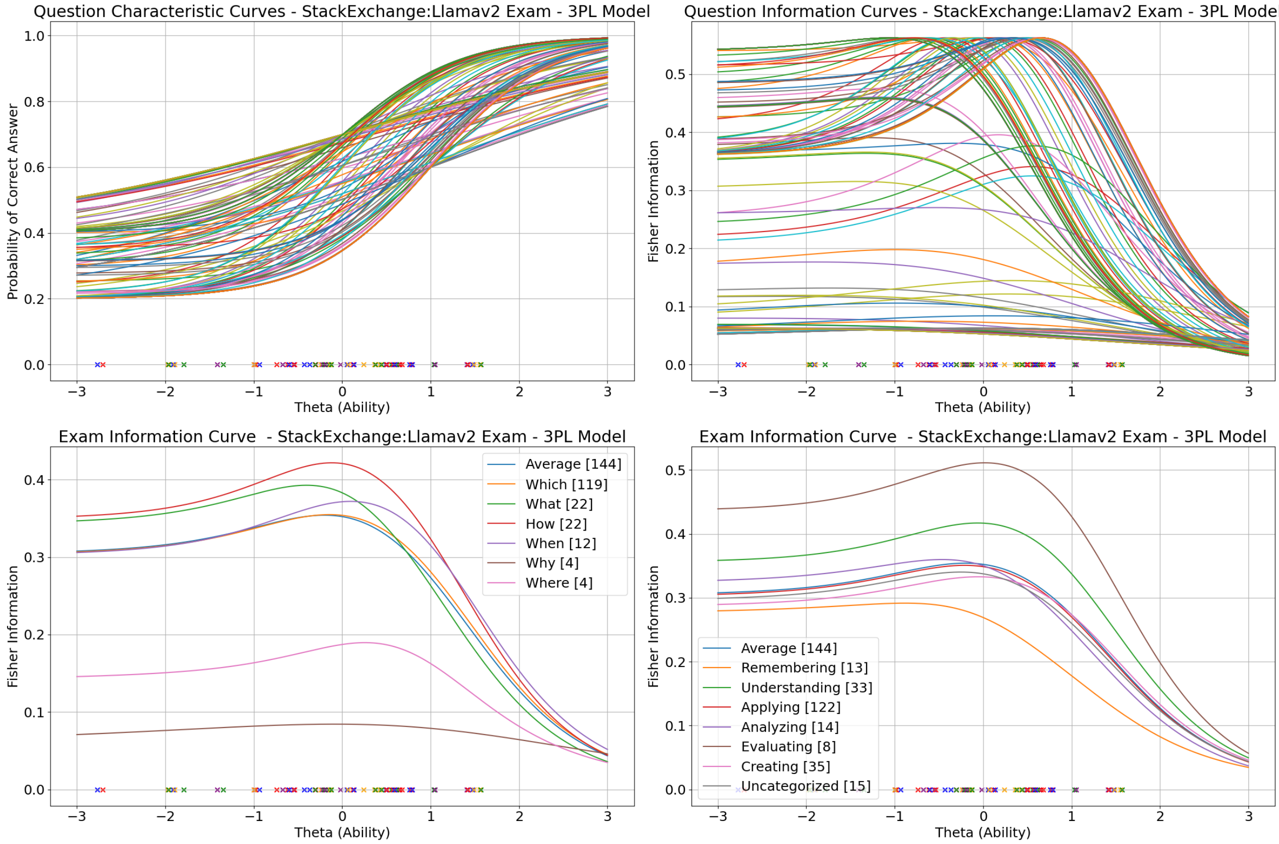

Figure 11. Hierarchical IRT Analysis Results for StackExchange task t_{stk}. (*Upper-Left*) Modeling $p_i(\theta)$, where each line corresponds to the Item Response Function for a given question $q_i \in \mathcal{Q}$. (*Upper-Right*) Aggregated Information function $I(\theta|g_i, d_i, b_i)$ of $p_i(\theta)$, where each line to a given question $q_i \in \mathcal{Q}$. (*Lower-Left*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to semantic taxonomy. (*Lower-Right*) Modeling aggregated Information function $\bar{I}_{cat}(\theta)$, averaged across questions according to Bloom taxonomy. For all graphs, each cross on the x-axis correspond to a given model ability θ_m.

B.3. Iterative Item Response Theory Model

We next present in Algorithm 1 the procedure used to iteratively maximise the informativeness of our exam corpus, by alternatively fitting the IRT model and updating the exam corpus. Here, we simply discard the least discriminative r^{th} quantile of questions (where r is chosen in practice around 10). Using more sophisticated update technique relying on newer question generation is a fascinating open question and direction for follow-up work.

Algorithm 1 Iterative Exam Improvement with IRT Model

```
**Input:** Initial Exam $\mathcal{Q}_1$, Maximal Step Count $k$, Drop Ratio $r$
**Initialize:** Fit IRT($\mathcal{Q}_1$), infer $(\theta_m)_{m \in \mathcal{M}}$ and $(g_i, d_i, b_i)_{i \in \mathcal{Q}_1}$
**for** $j = 1$ **to** $k - 1$ **do**
    **Exam Update:** Discard the $r^{th}$ first quantile of $(d_i)_{i \in \mathcal{Q}_j}$, the least discriminative questions of $\mathcal{Q}_j$ to generate $\mathcal{Q}_{j+1}$
    **Model Fiting:** Fit IRT($\mathcal{Q}_{j+1}$), using $(\theta_m)_{m \in \mathcal{M}}$ and $(g_i, d_i, b_i)_{i \in \mathcal{Q}_j}$ as initilization.
**end for**
**Output:** $\mathcal{Q}_k$
```

Figure 12 illustrate the evolution of the exam aggregated Information function $\bar{I}_{\mathcal{Q}_j}(\theta)$ alongside the evolution of the exam corpus. For some tasks such as t_{arx} or t_{ops}, we witness a continuous strictly dominating improvement, although mostly in the low to medium ability levels. Such improvement is also witnessed for t_{stk}, although convergence happens faster. Finally, for t_{sec}, the evolution is non-monotonic and interestingly mostly happens in high ability regions.

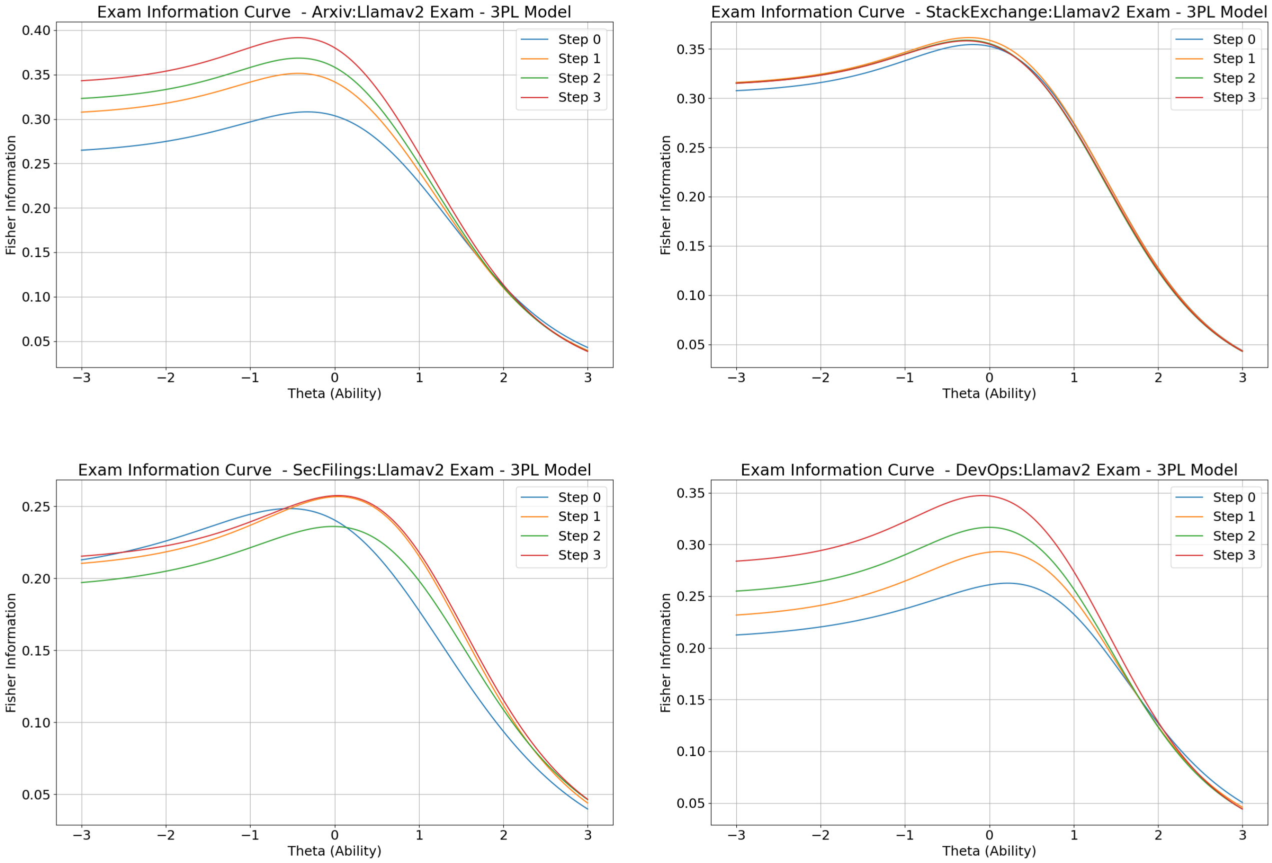

Figure 12. Evolution of the Exam aggregated Information function $\bar{I}_{\mathcal{Q}_j}(\theta)$ during the maximization steps j. (*Upper-Right*) DevOps Task (*Upper-Left*) StackExchange Task (*Lower-Right*) Sec Filings Task (*Lower-Right*) DevOps Task

C. Bloom's Revised Taxonomy of Cognitive Domains

Bloom's Revised Taxonomy, an update to the original educational framework by Benjamin Bloom, restructures the classification of cognitive objectives in education. Developed by Lorin Anderson and David Krathwohl, it emphasizes dynamic cognitive processes over static knowledge categories. The revised model, articulated in six progressive levels (Remember, Understand, Apply, Analyze, Evaluate, and Create) moves from basic recall of facts to the sophisticated synthesis of new ideas. This hierarchy guides the design of educational curricula and assessments, focusing on developing higher-order thinking skills and critical analysis.

In this revised framework, the original noun-based categories are replaced with verbs, highlighting active learning processes. For instance, "Analyze" involves deconstructing information to understand its components, while "Create" represents the peak of cognitive skill, where learners develop new constructs or solutions. This taxonomy is crucial in modern education, providing a foundation for teaching strategies that challenge students' cognitive abilities and prepare them for complex problem-solving across various disciplines.

Table 6 illustrates the levels of the revised Bloom's Taxonomy, from the lowest to the highest, along with a brief description and examples from the tasks considered of how they might translate into multiple-choice questions.

To cluster the questions in the correct Bloom taxonomy category, we use an algorithm detecting keyword usage, relying on a list extending the one presented in Section C. Note that one question can be classified in several categories.

Level	Description	Example Question	Keywords
Remembering	Retrieving, recalling, or recognizing knowledge from memory.	What is the reason for AMD being named in the settlement agreement with Philips Semiconductors Corporation? (SecFilings task)	list, identify, name, define, state, mention, recall, label, repeat, recognize
Understanding	Explaining ideas or concepts, understanding the meaning, translating knowledge into new contexts.	Consider a two-dimensional nonlinear Langevin equation with a dissipative, non-potential force that creates a line of stable fixed points (attracting line) touching a line of unstable fixed points (repelling line). In the low-noise limit, the stationary distribution of this system satisfies a large deviation principle with two competing terms. Which of the following statements best describes the nature of these two terms? (Arxiv task)	explain, describe, summarize, predict, interpret, paraphrase, translate, illustrate, rephrase, clarify, check, find, experience, suspect, review, notice, assume, interact, observe, understand
Applying	Using information in new situations or applying knowledge to solve problems.	You are an AWS engineer responsible for scaling a Redis (cluster mode disabled) cluster. You want to reduce costs but ensure minimal downtime for your application. Which of the following scaling actions should you choose, and why? (DevOps task)	demonstrate, apply, use, write, illustrate, solve, show, execute, implement, operate, practice, set, configure, use, try, follow, take, use, run, serve, task, read, operate, work, enable, exist
Analyzing	Breaking information into parts, examining relationships, differentiating between parts.	You are an AWS engineer responsible for deploying and managing an application using Amazon Elastic Container Service (ECS). You have created an ECS service with a task definition, and you want to add tags to the service. However, you are encountering issues with adding tags. Which of the following is the most likely reason for this issue? (DevOps task)	analyze, distinguish, compare, differentiate, examine, test, question, inspect, debate, investigate, manage, differentiate, optimize, troubleshoot, resolve, investigate, compare
Evaluating	Making judgments based on criteria, checking and critiquing.	Which of the following activities contributed the most to AAR CORP's net cash provided by investing activities in fiscal 2015? (SecFilings task)	evaluate, rate, justify, critique, decide, rank, measure, validate, test, assess, evaluate, decide, choose, verify, test, monitor, recommend
Creating	Designing, constructing, planning, producing, or inventing something new based on existing information.	You are developing a Node.js application using the AWS SDK, and you need to use the guard module from the StorefrontControllers package. However, when you try to require the module, you get an error saying that it cannot find the module. What should you do to resolve this issue? (StackExchange task)	design, construct, produce, invent, devise, formulate, originate, assemble, generate, create, design, develop, compose, generate, implement, produce, build, customize, formulate

Table 6. Bloom's Taxonomy.