FREQUENCYQA: Evaluating Factuality Robustness of LLMs Through Knowledge Frequency Control and Uncertainty Estimation

Anonymous ACL submission

Abstract

The rapid development of LLMs has sparked extensive research into their factual knowledge. Current works find that LLMs fall short on questions around low-frequency entities. However, such proofs are unreliable since the questions can differ not only in entity frequency but also in difficulty themselves. So we introduce **FREQUENCYQA** benchmark, containing **283K** abstract questions, each instantiated by a pair of high-frequency and low-frequency entities. It ensures a controllable comparison to study the role of knowledge frequency in the performance of LLMs. Because the difference between such a pair is only the entity with different frequencies. In addition, we use both correctness and uncertainty to develop a two-round method to evaluate LLMs' knowledge robustness. It aims to avoid possible semantic shortcuts which is a serious problem of current QA study. Experiments reveal that LLMs, including GPT-4o, exhibit particularly low robustness regarding low-frequency knowledge. Besides, we find that uncertainty can be used to effectively identify high-quality and shortcut-free questions while maintaining the data size. Based on this, we propose an automatic method to select such questions to form a subset called **FREQUENCYQA-Hard**, containing only hard low-frequency questions.[1]

1 Introduction

The rapid advancement of large language models (LLMs) has promoted a lot of study on their factual knowledge and reasoning ability (Wei et al., 2024a,b; Hendrycks et al., 2021).

Sun et al. (2024) and Mallen et al. (2023) compare LLMs' performance on questions around entities with different frequencies. LLMs are found struggle to handle tail knowledge. However, the questions they study are different and can vary in difficulty levels, not only in entity frequencies. As Allen-Zhu and Li (2023) emphasized, we need a more "*controlled, synthetic experiment that confirms the weakness of LLMs*" nowadays. Therefore, existing comparisons are not enough since they can not guarantee that the low frequency is the only cause of LLMs' poor performance.

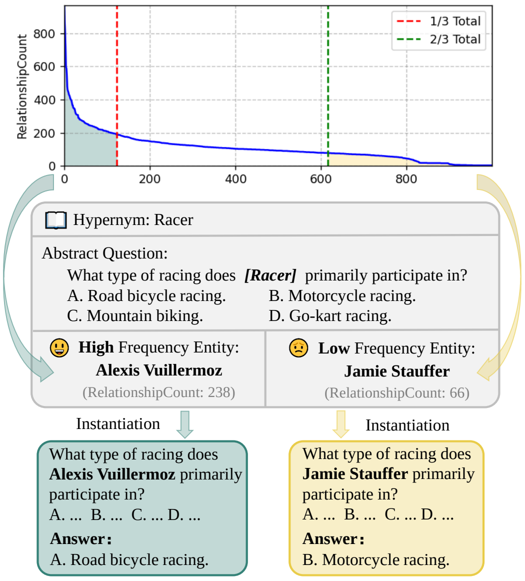


Figure 1: An example from FREQUENCYQA

To tackle the issue, we introduce our **FREQUENCYQA** benchmark. Pairs of high-frequency and low-frequency entities share the same question, as shown in Figure 1. The shared abstract question, with a hypernym to represent the two specific entities, guarantees that the difference between such a pair is only the entity. This allows for a controllable comparison between high-frequency and low-frequency entities. FREQUENCYQA is a large scale dataset containing **283K** such question pairs. It is generated through an automatic pipeline base on raw knowledge base, ensuring both diversity and scalability. Through this benchmark, we can completely compare LLM's performance on different knowledge frequencies.

[1] We provide data and code in supplementary materials, which will be released upon acceptance.

For a more robust and accurate evaluation, we use the multiple-choice format (Hendrycks et al., 2021; Geva et al., 2021) in our benchmark. But semantic shortcuts (Geirhos et al., 2020) between questions and options may help LLMs to guess the answer, which is also a common but severe problem recently. Thus, we further design a two-round method using both correctness and uncertainty to evaluate LLMs' knowledge. During experiments, we found that LLMs have very poor robustness, especially on low-frequency knowledge, where even the powerful GPT-4o also performs badly.

Recent benchmarks, like SimpleQA (Wei et al., 2024a), ensure their difficulty by collecting questions adversarially against LLMs' responses. But relying only on accuracy, they ignore the quality of their questions, and the difficulty is closely related to the models they use. Fortunately, our experiments find that uncertainty is also an effective tool in selecting both high-quality and shortcut-free questions while maintaining the benchmark size. Combining accuracy and uncertainty, we propose a new flexible method to select our subset called **FREQUENCYQA-Hard** for future study. It contains **81K** difficult low-frequency questions with high-quality and no semantic shortcuts.

In summary, we have three main contributions: **(1) [Resource]** We introduce FREQUENCYQA benchmark, where a pair of entities share the same abstract question. It enables a more controllable and reasonable proof that LLMs perform worse when the required knowledge is less frequent. (§3) **(2) [Method]** We design a two-round method using correctness and uncertainty to evaluate LLMs' robust knowledge. **[Finding]** LLMs can not stand such a test, especially on low-frequency knowledge, where even GPT-4o performs badly. (§4) **(3) [Finding]** Uncertainty is more helpful to find questions with high quality. **[Resource]** Through this, we select FREQUENCYQA-Hard benchmark containing only hard and low-frequency questions of high quality and no shortcuts. (§5)

2 Related Works

2.1 Benchmarking LLMs' Factuality

The factuality evaluation of LLMs has recently attracted significant attention (Wang et al., 2024c). Some factuality benchmarks require open-ended generation by LLMs, such as SimpleQA (Wei et al., 2024a), FreshQA (Vu et al., 2024), SelfAware (Yin et al., 2023), CLR-Fact (Zheng et al., 2024a), and HaluEval (Li et al., 2023). Such evaluations either rely heavily on expert annotation, or utilize automatic answer matching that sacrifices evaluation accuracy (Min et al., 2023; Chern et al., 2023; Wang et al., 2024b). Other benchmarks adopt the format of Yes-or-No questions (Geva et al., 2021; Zhang et al., 2024) or multiple-choice questions (MCQ) (Hendrycks et al., 2021; Wang et al., 2024a). These formats allow model responses to be easily parsed and compared with gold labels, enabling solid yet efficient evaluations.

2.2 Long-Tail Knowledge

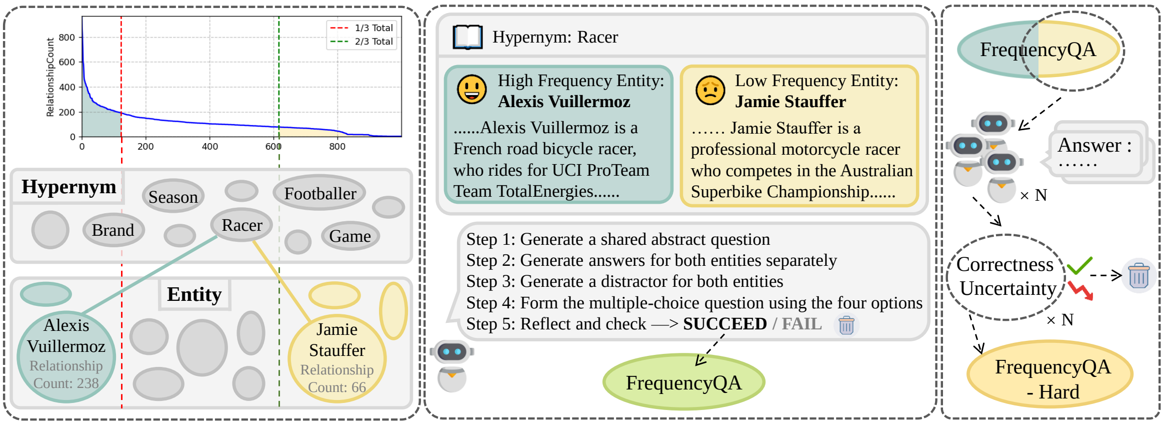
Long-tail knowledge (Wei et al., 2024b; Chen et al., 2023) is an important aspect of factuality. Kumar et al. (2024) proposes an automatic approach to generate questions for tail entities. Kandpal et al. (2023) find LLMs struggle to learn long-tail knowledge. Other works study the influence of knowledge frequency: Mallen et al. (2023) introduced PopQA, a long-tail benchmark, and found that models' performance will change with the frequency of entities in the questions. Sun et al. (2024) also proves this by constructing questions around head, torso, and tail entities. However, questions in these benchmarks are all in the form of open-ended generation, which can not be easily evaluated. They also depend on limited number of templates to produce QA questions from knowledge graphs, which will significantly harm the diversity of the benchmarks. Most importantly, the questions are different, so they may vary in difficulty levels, and thus can not provide fair comparisons.

2.3 Abstraction Knowledge

Existing works have studied various aspects of abstraction, for example, entity abstraction (Wu et al., 2012; Song et al., 2015), event abstraction (Wang et al., 2024e,d), and conceptual abstraction (Han et al., 2024). Abstraction has been shown to be beneficial for downstream tasks like commonsense reasoning, numerical reasoning, and logical reasoning (Zhou et al., 2024; Hong et al., 2024). In this paper, we control question difficulty by sharing the same abstraction form between a pair of entities.

3 FREQUENCYQA

To ensure more fair and controllable comparisons between LLMs' performance on high-frequency and low-frequency factual knowledge, we propose a new benchmark called FREQUENCYQA.


RelationshipCount
800
600
400
200
0
0
200
400
600
800
1/3 Total
2/3 Total
Hypernym
Season
Footballer
Brand
Racer
Game
Entity
Alexis Vuillermoz
Relationship Count: 238
Jamie Stauffer
Relationship Count: 66
Hypernym: Racer
High Frequency Entity: Alexis Vuillermoz
......Alexis Vuillermoz is a French road bicycle racer, who rides for UCI ProTeam Team TotalEnergies......
Low Frequency Entity: Jamie Stauffer
...... Jamie Stauffer is a professional motorcycle racer who competes in the Australian Superbike Championship......
Step 1: Generate a shared abstract question
Step 2: Generate answers for both entities separately
Step 3: Generate a distractor for both entities
Step 4: Form the multiple-choice question using the four options
Step 5: Reflect and check —> SUCCEED / FAIL
FrequencyQA
FrequencyQA
Answer : ......
× N
Correctness
Uncertainty
× N
FrequencyQA - Hard

1) Entity Pairs Extraction from DBpedia 2) Abstract Question Generation 3) Hard High-Quality Question Selecting

Figure 2: An overview of our benchmark curation pipeline. It contains three parts. Through the first two parts, (1) Entity Pairs Extraction from DBpedia and (2) Abstract Question Generation, we can get the whole FREQUENCYQA. And through the third part, (3) Hard High-Quality Question Selecting, we can get a harder subset, containing only difficult low-frequency questions with high quality and no semantic shortcut.

3.1 Question Formulation

For an accurate evaluation, our questions are in the form of multiple choice. Although recently there are several generative QA benchmarks, like SimpleQA (Wei et al., 2024a), and also some automatic methods (Zheng et al., 2023) to evaluate generated answers. There are still significant limitations in such generative QAs since the answer should be single and indisputable. Questions like *What is the primary focus or intention behind Civil Procedure Rules* cannot be included since there are many ways to answer this question. However, multiple-choice questions do not have such limitations.

Each piece of data in FREQUENCYQA, shown in Figure 1, contains an abstract question shared by two entities having the same hypernym. One entity, having many relationships in DBpedia (Auer et al., 2007), which will be introduced next, is the high-frequency entity, and the other, with only a few relationships, is the low-frequency one. The question will have different answers for the two entities, respectively. Such data form can ensure detailed and controllable comparisons between entities with different frequencies, and that is why we call it **FREQUENCYQA**. Containing **283K** such question pairs, the benchmark is constructed through a fully automated pipeline, which is cheap and scalable.

3.2 Curation Pipeline

In this section, we discuss the data curation pipeline for our dataset. As shown in Figure 2, the pipeline contains three parts: (1) Entity Pairs Extraction from DBpedia, (2) Abstract Question Generation, and (3) Hard High-Quality Question Filtering. Here we will introduce the first two parts, used to build FREQUENCYQA, and leave the third part, used to build FREQUENCYQA-Hard, for §5.

3.2.1 Entity Pairs Extraction from DBpedia

Following Sun et al. (2024), an entity's frequency is defined by its number of relationships in DBpedia. High-frequency entities are those whose cumulative relationships account for the first $1/3$ of all sorted entities, and low-frequency entities are for the last $1/3$. Details are shown in Appendix B.

In practice, we first get all the hypernyms in DBpedia and classify the entities belonging to them into high-frequent and low-frequent. Then, we map the entities one-by-one to get entity pairs. The reason behind this is that not every random pair of entities can have a shared abstract question easily. For example, it's hard to write a shared question for *Einstein* and *Apple* even though they are all high-frequency entities. But it's easy for two specific universities to share a same abstract question. In order for our entity pairs to produce high quality abstract questions, we ensure that the two entities have the same hypernym. This allows them to have similar descriptions, making it easier to generate an abstract question.

3.2.2 Abstract Question Generation

After acquiring the entity pairs, we adopt a multi-step curation pipeline to generate high-quality abstract multiple-choice questions: (1) Generate an abstract question (without options) according to the DBpedia descriptions of both entity. (2) Separately generate the corresponding answers based on the descriptions, with length control to alleviate bias among candidate answers. (3) Generate distractors for both entities with length control. Compared to randomly selected distractors, LLM-selected distractors generation have demonstrated its effectiveness for the high relevance between distractors and designated choices (Zheng et al., 2024b). (4) Formulate the final multiple-choice question using the four answer candidates above. (5) Proofread the question according to the standards below.

The standards for questions are presented as follows: (1) Quality: The questions should have one and only one correct answer for both entities. (2) Semantic Shortcuts: The correct answer cannot be simply guessed by the names of the entities or the way the questions are asked. For example, the question: *What was the primary operational location of Sydney O-Class Tram? A. Oslo, Norway B. Sydney, Australia C. Stockholm, Sweden D. Melbourne, Australia* is not allowed since only the correct answer contains *Sydney* which is also in the entity's name. (3) Length Bias: The four options in one question should have roughly the same length to avoid length bias. In summary, the generated questions should have high-quality and no shortcuts, in semantics or length.

During the curation, we utilize *GPT-4o-mini* (OpenAI, 2024a) with few-shot expert-written Chain-of-thought (CoT) demonstrations (Wei et al., 2022), with details provided in Appendix C.

3.3 Expert Verification

We enlist the help of three postgraduate students, each with extensive experience in NLP research, to validate the quality of these generated questions through a sample of 200 question pairs. The instruction is the same as the *standard (1)* given to LLMs above. The quality of each pair is decided by majority voting. Results show that their total agreement (all 3 experts have the same judgement) is 88.0%. And 95.5% of the abstract questions are considered correct and of high quality both for the high-frequency entity and the low-frequency entity, demonstrating the reliability of our benchmark.

3.4 Main Evaluations

FrequencyQA is a large-scale benchmark comprising a total of 283,455 abstract questions, each paired with a high-frequency and a low-frequency instantiation. Detailed statistics are in Appendix A.

We experiment with a selection of LLMs on our FrequencyQA benchmark to investigate their performance on high-frequency and low-frequency questions and also the difference between them.

3.4.1 Experiment Setup

Metric: We calculate Uncertainty, Accuracy, and Macro F1-score between model predictions and ground truth labels. We apply perplexity-based uncertainty for open source LLMs and verbalized uncertainty for proprietary LLMs due to several reasons, with details explained in Appendix D. Thus, we only compare uncertainty between high and low frequency within each setting separately.

Models: We experiment with 16 different models, with a full list in Appendix E, and categorize the evaluation into three types: (1) Open Source LLM Zero-Shot (Qin et al., 2023). (2) Open Source LLM Few-Shot (Brown et al., 2020). (3) Proprietary LLM API.

3.4.2 Results and Analysis

Evaluation results are reported in Table 1. Our observations include: **(1) Huge drop in performance from high-freq to low-freq:** All models suffer a performance decrease from high-frequency questions to low-frequency questions in all three settings. For instance, the accuracy and Macro F1-score of Llama-3-8B drop up to about 14 points in the few-shot setting. Proprietary models like GPT-4o are no exception. These all prove that LLM's performance is closely related to the frequency of the knowledge in the corpus. **(2) Increased Uncertainty from high-freq to low-freq:** Similarly, the uncertainty of LLMs all increase from high-frequency questions to low-frequency questions. For example, the uncertainty difference of Gemma-2-9B is up to about 86 points when using zero-shot. The uncertainty in the zero-shot setting is generally higher than in the few-shot setting, and the difference is also more pronounced. We think it may be because LLMs find some familiar examples in the few-shot setting, which decreases their uncertainty. In spite of this, the difference is still clear between high-freq and low-freq. These all prove that LLMs not only perform better but also are more confident about high-frequency knowledge. **(3) Few-shot helps a lot only for LLMs without instruction-tuning:** Most non-instruction-tuned LLMs show a huge improvement in performance from zero-shot to few-shot, but performance is similar for those instruction-tuned LLMs. This could be because the few-shot examples only teach LLMs

Models	High Freq Question			Low Freq Question			Average			Difference (H –> T)		
	Unc.	Acc	Ma-F1	Unc.	Acc	Ma-F1	Unc.	Acc	Ma-F1	Unc.	Acc	Ma-F1
	(↓)	(↑)	(↑)	(↓)	(↑)	(↑)	(↓)	(↑)	(↑)			
Random	-	25.29	25.29	-	25.22	25.22	-	25.26	25.26	-	↓ 0.07	↓ 0.07
Majority	-	25.70	10.22	-	25.14	10.04	-	25.42	10.13	-	↓ 0.56	↓ 0.18
LLM (Open Source) + Zero-Shot												
Llama-3 *8B*	54.33	65.90	63.60	81.54	53.83	51.29	67.94	59.87	57.44	↑ 6.11	↓ 12.07	↓ 12.30
Llama-3-Instruct *8B*	77.55	80.72	80.71	117.41	69.03	68.95	97.48	74.88	74.83	↑ 39.86	↓ 11.69	↓ 11.76
Llama-3.1 *8B*	55.91	65.29	63.31	83.35	52.66	50.52	69.63	58.98	56.92	↑ 27.44	↓ 12.63	↓ 12.78
Llama-3.1-Instruct *8B*	58.97	80.06	80.08	87.05	69.99	69.94	73.01	75.03	75.01	↑ 28.08	↓ 10.07	↓ 10.14
Gemma-2 *9B*	124.97	64.50	64.80	211.34	52.24	51.23	168.16	58.37	58.01	↑ **86.37**	↓ 12.26	↓ 13.57
Phi-3.5-mini-Instruct *4B*	27.81	72.81	72.78	39.80	65.26	65.00	33.81	69.04	68.89	↑ 11.99	↓ 7.55	↓ 7.78
Falcon2 *11B*	56.93	70.72	69.80	87.31	58.07	56.70	72.12	64.40	63.25	↑ 30.38	↓ 12.65	↓ 13.10
Mistral-v0.3 *7B*	39.83	65.55	63.11	56.99	53.36	50.26	48.41	59.46	56.69	↑ 17.16	↓ 12.19	↓ 12.85
Mistral-v0.3-Instruct *7B*	44.73	73.53	73.14	66.09	63.05	62.40	55.41	68.29	67.77	↑ 21.36	↓ 10.48	↓ 10.74
LLM (Open Source) + Few-Shot												
Llama-3 *8B*	21.89	75.57	75.55	23.75	61.00	61.01	22.82	68.29	68.28	↑ 1.86	↓ **14.57**	↓ **14.55**
Llama-3-Instruct *8B*	26.20	79.94	79.92	28.70	67.98	67.95	27.45	73.96	73.93	↑ 2.50	↓ 11.96	↓ 11.96
Llama-3.1 *8B*	20.82	74.91	74.89	22.62	62.00	62.00	21.72	68.46	68.45	↑ 1.81	↓ 12.91	↓ 12.90
Llama-3.1-Instruct *8B*	20.63	79.74	79.74	22.48	69.09	69.07	21.56	74.42	74.40	↑ 1.85	↓ 10.65	↓ 10.66
Gemma-2 *9B*	20.26	80.10	80.08	22.36	68.36	68.31	21.31	74.23	74.20	↑ 2.10	↓ 11.74	↓ 11.77
Phi-3.5-mini-Instruct *4B*	11.02	73.68	73.67	**11.85**	67.46	67.33	11.44	70.57	70.50	↑ 0.83	↓ 6.22	↓ 6.34
Falcon2 *11B*	16.42	77.11	77.01	17.77	65.92	65.75	17.10	71.52	71.38	↑ 1.35	↓ 11.19	↓ 11.26
Mistral-v0.3 *7B*	13.89	75.55	75.53	14.99	62.88	62.85	14.44	69.22	69.19	↑ 1.10	↓ 12.67	↓ 12.68
Mistral-v0.3-Instruct *7B*	15.97	74.46	74.40	17.40	65.51	65.35	16.69	69.99	69.87	↑ 1.43	↓ 8.95	↓ 9.05
LLM (Proprietary) API												
GPT4o-mini (Zero-Shot)	13.52	85.61	85.58	18.34	73.85	73.73	15.93	79.73	79.66	↑ 4.82	↓ 11.76	↓ 11.85
GPT4o-mini (Few-Shot)	25.74	84.78	84.69	38.17	72.76	72.47	31.96	78.77	78.58	↑ 12.43	↓ 12.02	↓ 12.22
GPT4o-mini (CoT)	10.53	86.25	86.25	12.27	74.39	74.40	**11.40**	80.32	80.32	↑ 1.74	↓ 11.85	↓ 11.85
GPT4o (Zero-Shot)	14.18	**93.86**	**93.95**	30.98	85.76	86.69	22.58	89.81	90.32	↑ 16.80	↓ 8.10	↓ 7.26
GPT4o (Few-Shot)	28.41	93.94	**93.95**	45.81	**86.54**	**86.75**	37.11	**90.24**	**90.35**	↑ 17.40	↓ 7.40	↓ 7.20
GPT4o (CoT)	**10.39**	92.40	92.47	18.36	85.47	85.72	14.38	88.93	89.10	↑ 7.97	↓ 6.93	↓ 6.75

Table 1: Performance of various LLMs on the testing set of FrequencyQA. Unc., Acc, and Ma-F1 denote Uncertainty, Accuracy, and Macro F1-score. The Difference column shows how scores change from high-frequency questions to low-frequency questions. The best performances within each method are underlined, and the best among all methods are **bold-faced**. And for the Difference column, We underline the largest difference within each method and **bold** the one among all methods. More results can be seen in Table 10.

how to do multiple-choice questions, while those instruction-tuned ones have already learned. **(4) CoT lowers uncertainty but does not awlays aid performance:** It's obvious that after the CoT inference, LLMs are more sure about their answers. However, results show that the average accuracy of GPT-4o even drops after adding CoT, which means CoT can not always help such factual questions.

4 Robust Knowledge Measurement

With the help of FrequencyQA, we can conduct a more detailed and controllable study of the factual knowledge of LLMs.

When considering how humans tackle multiple-choice questions, it's often the case that we do not really know the correct answers. Instead, we rely on semantic shortcuts within the questions to make educated guesses. Although we intentionally exclude these shortcuts when constructing the benchmark, they are difficult to eliminate entirely from multiple-choice questions. Such a situation often occurs even in human exam questions. Therefore, we need to devise an effective method to evaluate the robustness of factual knowledge of LLMs in the form of multiple-choice questions.

4.1 The Definition of Robust Knowledge

We categorize LLM's results into four scenarios based on its uncertainty and the correctness of its answers. The uncertainty pertains to the statement based on pure *Question+Answer* (without options), which isolates the correct knowledge without the influence of the three distractors. Details are in Appendix D. Meanwhile, the correctness refers to the original multiple-choice question.

Low Uncertainty & Correct Answer: LLM shows confidence about the correct knowledge and also answers the question correctly. In this case, we consider the LLM to possess robust knowledge.

High Uncertainty & Incorrect Answer: LLM expresses uncertainty and answers incorrectly. So, we conclude that the LLM lacks the knowledge.

High Uncertainty & Correct Answer: LLM is

Models	First Round				Second Round			
	High	Low	Avg.	Diff	High	Low	Avg.	Diff
Open Source LLM								
Llama-3 *8B*	75.57	61.00	68.29	↓ **14.57**	66.09 (-9.48)	39.15 (-21.86)	52.62 (-15.67)	↓ 26.95 (+12.38)
Llama-3-Instruct *8B*	79.94	67.98	73.96	↓ 11.96	69.04 (-10.91)	43.50 (-24.49)	56.27 (-17.69)	↓ 25.54 (+13.58)
Llama-3.1 *8B*	74.91	62.00	68.46	↓ 12.91	66.27 (-8.64)	40.35 (-21.65)	53.31 (-15.15)	↓ 25.92 (+13.01)
Llama-3.1-Instruct *8B*	79.74	69.09	74.42	↓ 10.65	72.89 (-6.85)	43.91 **(-25.19)**	58.40 (-16.02)	↓ 28.98 **(+18.34)**
Llama-3.2 *3B*	68.89	57.26	63.08	↓ 11.63	59.15 (-9.73)	38.19 (-19.07)	48.67 (-14.41)	↓ 20.96 (+9.33)
Llama-3.2-Instruct *3B*	71.43	62.12	66.78	↓ 9.32	64.14 (-7.30)	40.19 (-21.93)	52.17 (-14.62)	↓ 22.68 (+14.63)
Gemma-2 *2B*	62.99	50.93	56.96	↓ 12.06	52.54 (-10.44)	35.40 (-15.53)	43.97 (-12.99)	↓ 17.14 (+5.09)
Gemma-2 *9B*	80.10	68.36	74.23	↓ 11.74	71.26 (-8.84)	**46.98** (-21.38)	**59.12** (-15.11)	↓ 24.28 (+12.54)
Phi-3.5-mini *4B*	73.68	67.46	70.57	↓ 6.22	67.08 (-6.60)	44.39 (-23.06)	55.74 (-14.84)	↓ 22.69 (+16.47)
Falcon2 *11B*	77.11	65.92	71.52	↓ 11.19	64.27 (-12.84)	43.37 (-22.55)	53.82 (-17.70)	↓ 20.90 (+9.71)
Mistral-v0.3 *7B*	75.55	62.88	69.22	↓ 12.67	68.75 (-6.80)	39.81 (-23.07)	54.28 (-14.94)	↓ 28.94 (+16.27)
Mistral-v0.3-Instruct *7B*	74.46	65.51	69.99	↓ 8.95	68.25 (-6.21)	41.19 (-24.32)	54.72 (-15.27)	↓ 27.06 (+18.10)
Proprietary LLM								
GPT4o-mini	84.78	72.76	78.77	↓ 12.02	71.00 (-13.78)	33.16 (-39.60)	52.08 (-26.69)	↓ 37.83 (+25.81)
GPT4o	**93.94**	**86.54**	**90.24**	↓ 7.40	**79.85** **(-14.09)**	25.50 **(-61.04)**	52.68 **(-37.57)**	↓ **54.34** **(+46.94)**

Table 2: Accuracy scores in LLMs' robust knowledge measurement. We also report the changes in the scores from the first round to the second round. The best performances within each method are underlined, and the best among all methods are **bold-faced**. And for the Difference column and values in parentheses, We underline the largest difference within each method and **bold** the one among all methods.

unsure but answers correctly. This indicates that it may retain the correct knowledge, but the memory is vague, or that the semantic shortcuts in the question lead to the correct answer.

Low Uncertainty & Incorrect Answer: LLM is confident yet answers incorrectly. This could result from the LLM recalling incorrect knowledge or from misleading distractors in questions.

In the first two categories, we can determine whether the LLM truly possesses the knowledge. However, in the latter two cases, multiple factors influence the final results, and the LLM's grasp of knowledge is not robust.

This classification method leverages the strengths of both multiple-choice and generative questions, since we collect the uncertainty score without the distractors. While multiple-choice questions are easy to evaluate, they may allow for shortcuts; generative questions, on the other hand, are the opposite. Our method capitalizes on the uncertainty inherent in generative questions, which do not have shortcuts, and the accuracy of easily parsed answers provided by multiple-choice questions. This approach ensures that evaluation remains straightforward while fully addressing the potential for shortcuts.

4.2 The Two-Round Measurement

We introduce our two-round measurement, which can be applied to any multiple-choice benchmarks, based on the four categories. In the first round, we present multiple-choice questions to evaluate LLM's performance. Then, results are classified into four categories. For questions falling into the latter two cases (high-uncertainty correct & low-uncertainty incorrect), we will conduct a second round questioning. Scores will be modified if the correctness of any questions changes in this round.

For questions requiring reassessment in the second round, we ask LLMs to judge whether the four statements, with details in Appendix D, are true or false. The LLM is considered to truly possess the knowledge only when all four statements are accurately judged. On one hand, the second round provides an opportunity for LLMs to correct the answer by breaking distractors into separate questions, and on the other hand, it identifies questions where the LLMs simply guess the correct answers.

Compared to breaking down the multiple-choice questions into four correctness judgments directly from the start, our two-round approach offers a comprehensive analysis of how LLMs' performance changes from the first to the second stage. It leverages both uncertainty and correctness, providing deeper insights into LLMs' confidence and robustness regarding the factual knowledge they retain. Additionally, this method enables us to determine whether a particular result is due to LLM's lack of knowledge or the shortcuts and misleading distractors in the multiple-choice questions.

Although the second round of evaluation is stricter, our FrequencyQA benchmark ensures fair classification and comparison between high-frequency and low-frequency questions under the

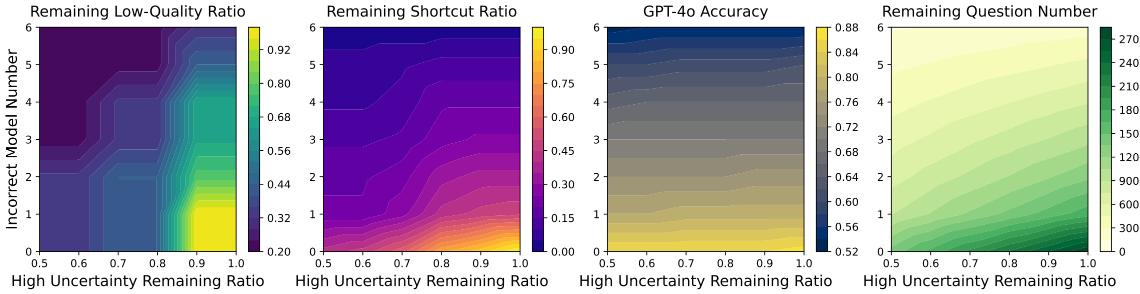


Figure 3: Heatmaps illustrating how subset quality changes with *incorrect model number* and *high uncertainty remaining ratio*. The former refers to the minimum number of times that each remaining question in the subset is answered incorrectly. The latter refers to the proportion of high-uncertainty questions that are retained in the subset.

same setting. Because we have a shared abstract question for each pair of entities, controlling other factors which may affect LLM's uncertainty and accuracy. Thus, we can rule out all the other factors to purely compare the difference between high-frequency and low-frequency knowledge.

4.3 Experiments and Analysis

Here, we conduct the experiments using this method to measure LLMs' knowledge robustness in a few-shot manner. Results are shown in Table 2

There are very interesting observations from the results: **(1) GPT-4o and GPT-4o-mini can not stand the test of robust knowledge measurement especially on low-frequency knowledge:** Their performance is still very good on high-frequency questions after the second round but drops a lot on low-frequency ones. The accuracy of GPT-4o on low-frequency questions after the second round is only 25.50, about 61 points lower than the first round. Its performance is even the worst of all the evaluated models, with GPT-4o-mini being the second-to-last. **(2) The LLMs' grasp of low-frequency knowledge is less robust than that of high-frequency knowledge:** Performance of LLMs in the second round all drop much more on low-frequency questions than on high-frequency questions. For example, in all of the open source LLMs, the accuracy of Llama-3.1-8B-Instruct in the second round, whose average performance is the best in the first round, dropped up to about 25 points on low-frequency questions while only about 7 points on high-frequency questions. **(3) LLMs which can stand the test of robust knowledge measurement must have a very accurate grasp of low-frequency knowledge:** LLMs whose average accuracy in the second round is higher than 55 are all those who performed relatively better on the low-frequency questions. This illustrates the importance of the LLMs mastering low-frequency knowledge if they want to be reliable models in the aspect of factual knowledge.

5 FrequencyQA-Hard

Directly collected questions, especially the high-frequency part, are a bit simple for today's LLMs. More importantly, they may have low quality and semantic shortcuts. So we introduce a new method to select a subset called FrequencyQA-Hard, containing only difficult, low-frequency questions that have high quality and no semantic shortcuts.

5.1 Hard High-Quality Question Filtering

The filtering method utilizes both correctness and uncertainty, similar to the last section. Previous benchmarks, like SimpleQA (Wei et al., 2024a) and MMLU-Pro (Wang et al., 2024a), were collected adversarially based on LLMs' responses to ensure question difficulty. We enhance this method by also considering LLMs' uncertainty to achieve the selection of high-quality, shortcut-free questions.

As illustrated in the third part of Figure 2, we collect the correctness and uncertainty from six open-source LLMs, listed in Appendix E, for questions with low-frequency entities. Different from above, both metrics here are about the entire multiple-choice questions, considering all the four options' quality. We choose the questions that many models answer incorrectly and exhibit high uncertainty for our hard subset. We define two hyperparameters: *incorrect model number* and *high uncertainty remaining ratio*. The former is the minimum number of times that each remaining question is answered incorrectly. The latter is the proportion of high-uncertainty questions (sort by the sum of the uncertainty of all models) retained in the subset.

The method is designed based on this assumption: Low-quality questions and those with semantic shortcuts are often associated with correct an-

Method	Backbone	Unc.	Acc	Ma-F1
Open Source LLM	Llama-3 *8B*	25.04	27.95	27.74
	Llama-3-Instruct *8B*	30.57	22.38	22.32
	Llama-3.1 *8B*	23.87	28.76	28.40
	Llama-3.1-Instruct *8B*	23.77	27.06	26.59
	Llama-3.2 *3B*	26.97	27.10	27.05
	Llama-3.2-Instruct *3B*	27.04	28.68	28.65
	Gemma-2 *2B*	31.07	22.80	22.35
	Gemma-2 *9B*	23.67	24.71	24.64
	Phi-3.5-mini *4B*	**<u>12.42</u>**	<u>31.87</u>	<u>31.19</u>
	Falcon2 *11B*	18.65	23.70	23.64
	Mistral-v0.3 *7B*	15.69	23.96	23.74
	Mistral-v0.3-Instruct *7B*	18.36	23.65	23.25
Proprietary LLM	GPT4o-mini	<u>46.49</u>	38.13	37.82
	GPT4o	54.66	**<u>69.98</u>**	**<u>70.02</u>**

Table 3: Performance of various LLMs on the testing set of FrequencyQA-Hard. Unc., Acc, and Ma-F1, denote Uncertainty, Accuracy, and Macro F1-score. The best performances within each method are <u>underlined</u> and the best among all methods are **bold-faced**.

swers and low uncertainty across different models, as our benchmark is constructed by LLMs-generated questions. The rationale behind this is as follows: (1) For low-quality questions, if the question or answer is incorrect, it is likely that the LLM did not generate it from the descriptions but rather from its internal knowledge. Consequently, models may display high confidence and yield correct answers. (2) For questions containing shortcuts, models may cheat through shortcuts, resulting in lower uncertainty and higher accuracy. These will all be proved in the following experiments.

5.2 Parameters Chosen by Expert Verification

To validate our filtering method, we invite experts to annotate the shortcuts in the 200 randomly sampled questions in the same setting mentioned in §3.3. The instruction is the same as the *standard (2)* given to LLMs above. Results show that 9.4% of the 95.5% correct and high quality questions are identified as having semantic shortcuts.

Then, we examine how the following metrics change with different settings regarding correctness and uncertainty, which are illustrated in Figure 3: **(1) Remaining Low-Quality Ratio** (the proportion of remaining low-quality problems relative to the original number of low-quality problems), **(2) Remaining Shortcut Ratio** (the proportion of remaining problems with shortcuts relative to the original number of problems with shortcuts), **(3) GPT-4o Accuracy** (the accuracy of GPT-4o on the remaining questions), and **(4) Remaining Question Count** (the size of the remaining questions). The results suggest that both uncertainty and correctness contribute to the selection of high-quality, shortcut-free questions, thereby demonstrating the effectiveness of our method. While it is expected that different models' correctness would aid in identifying more challenging questions, it is noteworthy that uncertainty proved to be more effective in selecting high-quality and shortcut-free questions while maintaining the dataset size, validating the inclusion of uncertainty in our filtering method.

Finally, we choose to set *incorrect model number* to 3 and *high uncertainty remaining ratio* to 0.8 according to Figure 3. It is a trade-off between quality, difficulty, and subset size. In this setting, only 1.5% of the total questions are of low quality, and 2.1% with shortcuts. Finally, the subset size is 81K, with a GPT-4o accuracy of 70%. Detailed statistics are shown in Appendix A.

5.3 Experiments and Analysis

Then we conduct experiments on FrequencyQA-Hard in a few-shot manner, with results shown in Table 3. In the multiple-choice question format, the open-source LLMs all significantly underperform, indicating the difficulty of our benchmark. For the proprietary LLMs, GPT-4o-mini also has a poor performance with an accuracy of about 38, even though we do not use it when constructing the subset. GPT-4o is better with an accuracy of about 70, but still has a huge room for future enhancement. And, predictably, its knowledge robustness will drop much more on this benchmark according to our experiments in §4.3.

6 Conclusions

In this paper, we first introduce FrequencyQA benchmark to evaluate LLMs' factual knowledge, with a fully automatic pipeline. This benchmark allows for more controllable and detailed comparisons between high-frequency and low-frequency knowledge of LLMs. Then, we propose a two-round method utilizing correctness and uncertainty to measure LLMs' knowledge robustness. And we are surprised to find that even powerful LLMs like GPT-4o can not stand such a test, especially on the low-frequency knowledge. At last, we discover that uncertainty is more effective in filtering out questions with low quality and shortcuts compared with correctness, which is often used by recent works. Based on this method, we provide a subset called FrequencyQA-Hard, which contains only difficult low-frequency questions of high quality and no shortcuts for future study.

Limitations

While we contribute valuable resources, methods, and findings to advance the probing of LLMs' factual knowledge, several limitations still exist that cannot be covered in this single work.

In this paper, we provide a shared abstract question with the entities being the only varying part. However, due to the sharing of such abstract questions, it is difficult to contain more entities with different frequencies in the same question because many entities lack sufficient shared features to generate common questions.

Our approach ensures that the difference between a pair is only the entity. Future research could investigate methods for measuring the entire knowledge frequency required to solve the questions, rather than limiting it to entity frequency. We believe it is a challenging but valuable task.

Additionally, our focus is on fixed knowledge, but fast-changing factual knowledge (Do et al., 2024b) and other knowledge (Do et al., 2024a; Wang et al., 2022, 2023) also deserves attention. Specifically, exploring how knowledge frequency can assist LLMs in acquiring new information (Choi et al., 2023; Zong et al., 2023) is a worthwhile area for further study.

Ethics Statement

Offensive Content Elimination. Our benchmark curation pipeline, which involves generating content using LLMs, requires stringent measures to ensure that generated responses are free from offensive material. We manually review a random sample of 200 data instances from FREQUENCYQA for any offensive content. Based on our annotations, we have not detected any offensive content. Therefore, we believe our dataset is safe and will not yield any negative societal impact.

Licenses. We will share our code under the MIT license, allowing other researchers free access to our resources for research purposes. Our dataset will be released under a CC license, also providing scholars with free access. We take full responsibility for any rights violations or issues related to the data license. The DBpedia dataset used in this paper is shared under the CC BY-SA license, permitting its use for research. As for language models, we access all open-source LMs via the Huggingface Hub (Wolf et al., 2020). All associated licenses permit user access for research purposes, and we have agreed to adhere to all terms of use.

Annotations. For expert verifications, we have obtained IRB approval and support from our institution's department, enabling us to invite expert graduate students to validate the quality of our data. They all agree to participate voluntarily without being compensated. We have made significant efforts to eliminate offensive content, thereby ensuring that no annotators are offended.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.2, knowledge manipulation. *CoRR*, abs/2309.14402.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Lihu Chen, Simon Razniewski, and Gerhard Weikum. 2023. Knowledge base completion for long-tail entities. *CoRR*, abs/2306.17472.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative AI - A tool augmented framework for multi-task and multi-domain scenarios. *CoRR*, abs/2307.13528.

Sehyun Choi, Tianqing Fang, Zhaowei Wang, and Yangqiu Song. 2023. KCTS: knowledge-constrained tree search decoding with token-level hallucination detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14035–14053. Association for Computational Linguistics.

Quyet V. Do, Tianqing Fang, Shizhe Diao, Zhaowei Wang, and Yangqiu Song. 2024a. Constraintchecker: A plugin for large language models to reason on commonsense knowledge bases. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 714–731. Association for Computational Linguistics.

Quyet V Do, Junze Li, Tung-Duong Vuong, Zhaowei Wang, Yangqiu Song, and Xiaojuan Ma. 2024b. What really is commonsense knowledge? *arXiv preprint arXiv:2411.03964*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, and Kevin Stone. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Kaiqiao Han, Tianqing Fang, Zhaowei Wang, Yangqiu Song, and Mark Steedman. 2024. Concept-reversed winograd schema challenge: Evaluating and improving robust reasoning in large language models via abstraction. *CoRR*, abs/2410.12040.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Ruixin Hong, Hongming Zhang, Xiaoman Pan, Dong Yu, and Changshui Zhang. 2024. Abstraction-of-thought makes language models better reasoners. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 1993–2027. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Rohan Kumar, Youngmin Kim, Sunitha Ravi, Haitian Sun, Christos Faloutsos, Ruslan Salakhutdinov, and Minji Yoon. 2024. Automatic question-answer generation for long-tail knowledge. *CoRR*, abs/2403.01382.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6449–6464. Association for Computational Linguistics.

Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. 2023. Vera: A general-purpose plausibility estimation model for commonsense statements. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1264–1287. Association for Computational Linguistics.

Quentin Malartic, Nilabhra Roy Chowdhury, Ruxandra Cojocaru, Mugariya Farooq, Giulia Campesan, Yasser Abdelaziz Dahou Djilali, Sanath Narayan, Ankit Singh, Maksim Velikanov, Basma El Amel Boussaha, Mohammed Al-Yafeai, Hamza Alobeidli, Leen Al Qadi, Mohamed El Amine Seddik, Kirill Fedyanin, Réda Alami, and Hakim Hacid. 2024. Falcon2-11b technical report. *CoRR*, abs/2407.14885.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. *OpenAI*.

OpenAI. 2024b. Hello gpt-4o. *OpenAI*.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1339–1384. Association for Computational Linguistics.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul

Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Yangqiu Song, Shusen Wang, and Haixun Wang. 2015. Open domain short text conceptualization: A generative + descriptive modeling approach. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 3820–3826. AAAI Press.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-tail: How knowledgeable are large language models (llms)? A.K.A. will llms replace knowledge graphs? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 311–325. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Lyudmila Rvanova, Sergey Petrakov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2024. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *CoRR*, abs/2406.15627.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry W. Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, and Thang Luong. 2024. Freshllms: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 13697–13720. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024a. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *CoRR*, abs/2406.01574.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2024b. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 14199–14230. Association for Computational Linguistics.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024c. Factuality of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 19519–19529. Association for Computational Linguistics.

Zhaowei Wang, Quyet V. Do, Hongming Zhang, Jiayao Zhang, Weiqi Wang, Tianqing Fang, Yangqiu Song, Ginny Y. Wong, and Simon See. 2023. COLA: contextualized commonsense causal reasoning from the causal inference perspective. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5253–5271. Association for Computational Linguistics.

Zhaowei Wang, Wei Fan, Qing Zong, Hongming Zhang, Sehyun Choi, Tianqing Fang, Xin Liu, Yangqiu Song, Ginny Y. Wong, and Simon See. 2024d. Absinstruct: Eliciting abstraction ability from llms through explanation tuning with plausibility estimation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 973–994. Association for Computational Linguistics.

Zhaowei Wang, Haochen Shi, Weiqi Wang, Tianqing Fang, Hongming Zhang, Sehyun Choi, Xin Liu, and Yangqiu Song. 2024e. Abspyramid: Benchmarking the abstraction ability of language models with a unified entailment graph. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 3991–4010. Association for Computational Linguistics.

Zhaowei Wang, Hongming Zhang, Tianqing Fang, Yangqiu Song, Ginny Y. Wong, and Simon See. 2022. Subeventwriter: Iterative sub-event sequence generation with coherence controller. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1590–1604. Association for Computational Linguistics.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024a. Measuring short-form factuality in large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024b. Long-form factuality in large language models. *CoRR*, abs/2403.18802.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 481–492. ACM.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8653–8665. Association for Computational Linguistics.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024. How language model hallucinations can snowball. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tianshi Zheng, Jiaxin Bai, Yicheng Wang, Tianqing Fang, Yue Guo, Yauwai Yim, and Yangqiu Song. 2024a. Clr-fact: Evaluating the complex logical reasoning capability of large language models over factual knowledge. *Preprint*, arXiv:2407.20564.

Tianshi Zheng, Weihan Li, Jiaxin Bai, Weiqi Wang, and Yangqiu Song. 2024b. Assessing the robustness of retrieval-augmented generation systems in k-12 educational question answering with knowledge discrepancies. *Preprint*, arXiv:2412.08985.

Ben Zhou, Hongming Zhang, Sihao Chen, Dian Yu, Hongwei Wang, Baolin Peng, Dan Roth, and Dong Yu. 2024. Conceptual and unbiased reasoning in language models. *CoRR*, abs/2404.00205.

Qing Zong, Zhaowei Wang, Baixuan Xu, Tianshi Zheng, Haochen Shi, Weiqi Wang, Yangqiu Song, Ginny Y. Wong, and Simon See. 2023. TILFA: A unified framework for text, image, and layout fusion in argument mining. In *Proceedings of the 10th Workshop on Argument Mining, ArgMining 2023, Singapore, December 7, 2023*, pages 139–147. Association for Computational Linguistics.

Appendices

A Benchmark Statistics

FrequencyQA is a large-scale benchmark comprising a total of 283,455 abstract question pairs, each paired with a high-frequency and a low-frequency entity. We guarantee that each entity corresponds to a single question sourced from 9166 hypernyms, to ensure no overlap. We partition our data into training, validation, and testing splits following an 8:1:1 ratio, ensuring that entities of different frequency intervals are evenly distributed in each split. Specific details are shown in Table 4.

And for FrequencyQA-Hard, there are 81,136 high-quality and shortcut-free questions with low-frequency entities from 4,876 hypernyms in total, with details shown in Table 5.

Split	#Q. Pair	#Entity	#Hyper.	#H. R.	#L. R.
Train	226,762	453,524	8,430	628	76
Valid	28,345	56,690	3,316	627	76
Test	28,348	56,696	3,314	629	76
Total	283,455	566,910	9,166	628	76

Table 4: The statistics of FrequencyQA benchmark. #Q. Pair refers to the number of question pairs. Hyper. means hypernym. #H. R. and #L. R. refer to the average number of relationships of high-frequency entities and low-frequency entities respectively.

Split	#Q. = #Entity	#Hyper.	#L. R.
Train	65,057	4,396	77.45
Valid	7,978	1,466	77.23
Test	8,101	1,484	77.65
Total	81,136	4,876	77.44

Table 5: The statistics of FrequencyQA-Hard benchmark. Here the number of questions is equal to the number of entities since it only has low frequency entities.

For benchmark quality, we further compute the correctness of the annotated date in §3.3 coming from the train, validation, and test set separately. The results presented in Table 6 show the high correctness in all splits of our benchmark. Besides, our labeling is very strict. If either of the questions in one pair is wrong, we will judge the whole question pair incorrect. Since each pair contains two questions, actually only about 2.25% of the total questions are incorrect.

Split	Correctness	Agreement
Train	95.68	88.27
Valid	93.75	87.50
Test	95.45	90.91

Table 6: The statistics of FrequencyQA-Hard benchmark. Here the number of questions is equal to the number of entities since it only has low frequency entities.

B Definition of High and Low Frequency

First, we randomly sample 1K entities from DBpedia and compute their relationship count separately. This process is to study the distribution of entities' relationship counts in DBpedia and find boundaries for high-frequency and low-frequency entities. Then we sort these entities in order of their relationship count from highest to lowest. High-frequency entities possess cumulative relationship count of up to 1/3 of all entities, while low frequency entities range from 2/3 to 1. We exclude those between 1/3 and 2/3 to make comparison clear.

Repeating this process 3 times, we get the number of relationships to distinguish high-frequency and low-frequency entities, which are higher than 185 and lower than 107 respectively. (Of the 1K randomly sampled entities, 119 entities are identified as high frequent and 621 entities as low frequent.) These two numbers are then used to classify all the high-frequency and low-frequency entities.

It is necessary to randomly select 1K entities first to calculate the two boundaries, since the time to compute the cumulative relationship count for all the sorted entities in DBpedia is unaffordable.

C Details in FrequencyQA Construction

In the process of entity pairs extraction, both hypernyms that do not have enough entities and low-frequency entities that do not have enough high-frequency entities to pair with are all discarded. Finally, we get 293K entities pairs from 9,261 hypernyms in total.

These pairs are then fed into the LLM to generate questions. Each pair has a shared abstract question respectively. Since a hypernym can have several entity pairs, the final number of abstract questions, which is equal to the number of pairs, is more than that of hypernyms. After LLM's proofread stage, there are 283K abstract questions left, which are used to build our FrequencyQA benchmark.

Task	Prompt
Question Generation	You will be given two entities belonging to the same hypernym. Generate a shared multiple choice question for both entities based on their descriptions according to the following 5 steps. Requirements: First, the questions should have one and only one correct answer for both entities. Second, the correct answer cannot be simply guessed by the names of the entities or the way the questions are asked. For example, the question: "What was the primary operational location of Sydney O-Class Tram? A. Oslo, Norway B. Sydney, Australia C. Stockholm, Sweden D. Melbourne, Australia" is not allowed since only the correct answer contains "Sydney" which is also in the entity's name. Third, the four options should have roughly the same length. Step 1, generate the shared question containing "[entity_name]" which can be replaced by the two entity names and then have different answers accordingly. Step 2, use roughly the same amount of words to answer the questions for both entities separately. Step 3, generate a misleading distractor for both entities separately. The distractors should have roughly the same length with the correct answers. Step 4, form the final multiple choice question using the above four answer candidates, and make sure the answer for Entity1 is 'A' , the answer for Entity2 is 'B', and their misleading answer candidates are 'C' and 'D'. Step 5, check whether the final question and answer candidates meet the above requirements. If yes, then output **SCUUEED**, otherwise output **FAIL**. Follow these examples: ......*(Examples written by experts)* **Hypernym:** *[Hypernym]* **Entity1:** *[Entity1]* **Entity1 Description:** *[Entity1 Description]* **Entity2:** *[Entity2]* **Entity2 Description:** *[Entity2 Description]*

Table 7: The prompt used to generate questions in FrequencyQA. Placeholders *[Hypernym]*, *[Entity1]*, *[Entity1 Description]*, *[Entity2]*, *[Entity2 Description]* will be replaced with real hypernym, high-frequency entity, low-frequency entity, and their descriptions accordingly.

Entities belonging to these questions come from 9,166 different hypernyms.

In the process of Abstract Question Generation, the prompt we use to generate questions is shown in Table 7. In questions generated by LLMs, the answer for the high-frequency entity is A, low-frequency entity is B, and their distractors are C and D respectively. And in the end, we randomly shuffle the 4 options in one question to guarantee the balance of the correct options.

Task	Prompt
Uncertainty Generation	Question: [question] Answer: **[option].** Uncertainty: **[uncertainty percentage]** Answer the following multiple choice question. Select only one correct answer from the choices and give your uncertainty score, following the above format. *[Question]* A. *[OptionA]*. B. *[OptionB]*. C. *[OptionC]*. D. *[OptionD]*.

Table 8: The prompt used to generate uncertainty score for proprietary LLMs. Placeholders *[Hypernym]*, *[Entity1]*, *[Entity1 Description]*, *[Entity2]*, *[Entity2 Description]* will be replaced with real hypernym, high-frequency entity, low-frequency entity, and their descriptions accordingly.

D Calculation of Uncertainty

Following Liu et al. (2023), we first combine the questions with each of their options, and use GPT-4o-mini to transform them into four statements. The data can also be found in our benchmark.

Then we compute the uncertainty for each statement. For open-source LLMs, uncertainty refers to their perplexity for generating the correct statement. For proprietary LLMs, we allow them to generate their uncertainty scores, with prompt shown in Table 8. Then, the threshold between high and low uncertainty for each model is determined by its own average uncertainty on the testing set of each benchmark respectively.

There are several reasons for choosing different uncertainties for each setting. On one hand, perplexity-based uncertainty is unavailable for proprietary LLMs. On the other hand, open source LLMs often fail to understand the instructions and can not generate uncertainty scores. We have conducted experiments on verbalized uncertainty of open source LLMs. However, experiments show that Llama-3-8B-Instruct has a 18.5% chance of not generating an uncertainty score, which makes the results unreliable and will also affect further robustness evaluations. In addition, Vashurin et al.

(2024) mentioned that, for multiple choice QA, information-based methods such as perplexity are substantially superior to quantifying model uncertainty.

Method	Prompt
Zero-Shot	*[Question]* A. *[OptionA]*. B. *[OptionB]*. C. *[OptionC]*. D. *[OptionD]*. The correct answer is:
Few-Shot	*[Examples]* Answer the multiple choice question. Select only one correct answer from the choices, following above examples. *[Question]* A. *[OptionA]*. B. *[OptionB]*. C. *[OptionC]*. D. *[OptionD]*.
CoT	Question: [question] Rational: [rationale] Answer: **[option].** Answer the multiple choice question. Think step by step and generate a short rationale to support your reasoning. Choose one best answer based on the generated rational, following the above format. Keep your whole response in 50 tokens. *[Question]* A. *[OptionA]*. B. *[OptionB]*. C. *[OptionC]*. D. *[OptionD]*.

Table 9: The prompt used when evaluating LLMs on our benchmark. Placeholders *[Examples]*, *[Question]*, *[OptionA]*, *[OptionB]*, *[OptionC]*, *[OptionD]* will be replaced with the real examples, questions and their options accordingly.

E Experiment Details

For the main evaluations on FREQUENCYQA, we categorize the evaluation of different models into three types: **(1) OPEN SOURCE LLM ZERO-SHOT:** We first evaluate Llama3, Llama3.1, Llama3.2 (Touvron et al., 2023; Dubey et al., 2024), Gemma2 (Mesnard et al., 2024; Riviere et al., 2024), Phi3.5 (Abdin et al., 2024), Falcon, Falcon2 (Malartic et al., 2024), Mistral (Jiang et al., 2023), and their instruction versions accordingly in a zero-shot manner (Qin et al., 2023). **(2) OPEN SOURCE LLM FEW-SHOT:** Then we evaluate the above models in a few-shot manner (Brown et al., 2020). Since our benchmark is in the form of four options multiple-choice questions, the shot number is set to four to minimize bias, where each of the four examples corresponds to a different correct answer in (a, b, c, d). **(3) PROPRIETARY LLM API:** Finally, we evaluate the performance of GPT-4o (OpenAI, 2023, 2024b) and GPT-4o-mini (OpenAI, 2024a), using zero-shot, few-shot, and Chain-of-Thought (CoT; Wei et al., 2022).

All the open-source models are run on 4 NVIDIA A6000 (40G) GPUs with BF32. And for proprietary LLM, we access them via OpenAI API [2]. The different kinds of prompts we use are shown in Table 9. And all the evaluation results are reported in Table 10.

For the question filtering of FREQUENCYQA-Hard, the six open-source LLMs we use are Llama-3-8B-Instruct, Llama-3.1-8B-Instruct, gemma-2-9b, Phi-3.5-mini-instruct, Falcon-11B, and Mistral-7B-Instruct-v0.3.

[2] https://platform.openai.com/docs/api-reference

Models	High Freq Question			Low Freq Question			Average			Difference (H –> T)		
	Unc. (↓)	Acc (↑)	Ma-F1 (↑)	Unc. (↓)	Acc (↑)	Ma-F1 (↑)	Unc. (↓)	Acc (↑)	Ma-F1 (↑)	Unc.	Acc	Ma-F1
Random	-	25.29	25.29	-	25.22	25.22	-	25.26	25.26	-	↓ 0.07	↓ 0.07
Majority	-	25.70	10.22	-	25.14	10.04	-	25.42	10.13	-	↓ 0.56	↓ 0.18
LLM (Open Source) + Zero-shot												
Llama-3 *8B*	54.33	65.90	63.60	81.54	53.83	51.29	67.94	59.87	57.44	↑ 6.11	↓ 12.07	↓ 12.30
Llama-3-Instr *8B*	77.55	80.72	80.71	117.41	69.03	68.95	97.48	74.88	74.83	↑ 39.86	↓ 11.69	↓ 11.76
Llama-3.1 *8B*	55.91	65.29	63.31	83.35	52.66	50.52	69.63	58.98	56.92	↑ 27.44	↓ 12.63	↓ 12.78
Llama-3.1-Instr *8B*	58.97	80.06	80.08	87.05	69.99	69.94	73.01	75.03	75.01	↑ 28.08	↓ 10.07	↓ 10.14
Llama-3.2 *3B*	67.51	57.93	53.65	99.55	48.10	44.66	83.53	53.02	49.16	↑ 32.04	↓ 9.83	↓ 9.00
Llama-3.2-Instr *3B*	74.27	71.47	71.51	108.24	62.17	62.18	91.26	66.82	66.85	↑ 33.97	↓ 9.30	↓ 9.33
Gemma-2 *2B*	133.57	46.48	43.40	213.40	37.76	34.07	173.49	42.12	38.74	↑ 79.83	↓ 8.72	↓ 9.33
Gemma-2 *9B*	124.97	64.50	64.80	211.34	52.24	51.23	168.16	58.37	58.01	↑ **86.37**	↓ 12.26	↓ 13.57
Phi-3.5-mini-Instr *4B*	27.81	72.81	72.78	39.80	65.26	65.00	33.81	69.04	68.89	↑ 11.99	↓ 7.55	↓ 7.78
Falcon *7B*	62.33	18.19	18.55	92.92	17.93	18.19	77.63	18.06	18.37	↑ 30.59	↓ 0.26	↓ 0.36
Falcon-Instr *7B*	88.88	25.68	14.49	128.75	25.71	14.01	108.82	25.70	14.25	↑ 39.87	↑ 0.02	↓ 0.48
Falcon2 *11B*	56.93	70.72	69.80	87.31	58.07	56.70	72.12	64.40	63.25	↑ 30.38	↓ 12.65	↓ 13.10
Mistral-v0.3 *7B*	39.83	65.55	63.11	56.99	53.36	50.26	48.41	59.46	56.69	↑ 17.16	↓ 12.19	↓ 12.85
Mistral-v0.3-Instr *7B*	44.73	73.53	73.14	66.09	63.05	62.40	55.41	68.29	67.77	↑ 21.36	↓ 10.48	↓ 10.74
LLM (Open Source) + 4-shot												
Llama-3 *8B*	21.89	75.57	75.55	23.75	61.00	61.01	22.82	68.29	68.28	↑ 1.86	↓ **14.57**	↑ **14.55**
Llama-3-Instr *8B*	26.20	79.94	79.92	28.70	67.98	67.95	27.45	73.96	73.93	↑ 2.50	↓ 11.96	↓ 11.96
Llama-3.1 *8B*	20.82	74.91	74.89	22.62	62.00	62.00	21.72	68.46	68.45	↑ 1.81	↓ 12.91	↓ 12.90
Llama-3.1-Instr *8B*	20.63	79.74	79.74	22.48	69.09	69.07	21.56	74.42	74.40	↑ 1.85	↓ 10.65	↓ 10.66
Llama-3.2 *3B*	23.58	68.89	68.84	25.59	57.26	57.17	24.59	63.08	63.01	↑ 2.01	↓ 11.63	↓ 11.67
Llama-3.2-Instr *3B*	23.66	71.43	71.43	25.67	62.12	62.09	24.67	66.78	66.76	↑ 2.01	↓ 9.32	↓ 9.34
Gemma-2 *2B*	26.96	62.99	62.93	29.43	50.93	50.91	28.20	56.96	56.92	↑ 2.47	↓ 12.06	↓ 12.02
Gemma-2 *9B*	20.26	80.10	80.08	22.36	68.36	68.31	21.31	74.23	74.20	↑ 2.10	↓ 11.74	↓ 11.77
Phi-3.5-mini-Instr *4B*	11.02	73.68	73.67	**11.85**	67.46	67.33	11.44	70.57	70.50	↑ 0.83	↓ 6.22	↓ 6.34
Falcon *7B*	21.69	28.40	21.25	23.42	28.25	20.63	22.56	28.33	20.94	↑ 1.73	↓ 0.15	↓ 0.62
Falcon-Instr *7B*	21.69	26.27	18.32	23.42	25.67	18.15	22.56	25.97	18.23	↑ 1.73	↓ 0.59	↓ 0.17
Falcon2 *11B*	16.42	77.11	77.01	17.77	65.92	65.75	17.10	71.52	71.38	↑ 1.35	↓ 11.19	↓ 11.26
Mistral-v0.3 *7B*	13.89	75.55	75.53	14.99	62.88	62.85	14.44	69.22	69.19	↑ 1.10	↓ 12.67	↓ 12.68
Mistral-v0.3-Instr *7B*	15.97	74.46	74.40	17.40	65.51	65.35	16.69	69.99	69.87	↑ 1.43	↓ 8.95	↓ 9.05
LLM (Proprietary) API												
GPT4o-mini (Zero-Shot)	13.52	85.61	85.58	18.34	73.85	73.73	15.93	79.73	79.66	↑ 4.82	↓ 11.76	↓ 11.85
GPT4o-mini (Few-Shot)	25.74	84.78	84.69	38.17	72.76	72.47	31.96	78.77	78.58	↑ 12.43	↓ 12.02	↓ 12.22
GPT4o-mini (CoT)	10.53	86.25	86.25	12.27	74.39	74.40	**11.40**	80.32	80.32	↑ 1.74	↓ 11.85	↓ 11.85
GPT4o (Zero-Shot)	14.18	**93.86**	**93.95**	30.98	85.76	86.69	22.58	89.81	90.32	↑ 16.80	↓ 8.10	↓ 7.26
GPT4o (Few-Shot)	28.41	93.94	**93.95**	45.81	**86.54**	**86.75**	37.11	**90.24**	**90.35**	↑ 17.40	↓ 7.40	↓ 7.20
GPT4o (CoT)	**10.39**	92.40	92.47	18.36	85.47	85.72	14.38	88.93	89.10	↑ 7.97	↓ 6.93	↓ 6.75

Table 10: Performance of various LLMs on the testing set of FREQUENCYQA. Unc., Acc, and Ma-F1, denote Uncertainty, Accuracy, and Macro F1-score. And the Difference column shows how scores change from high-frequency questions to low-frequency questions. The best performances within each method are underlined and the best among all methods are **bold-faced**. And for the Difference column, We underline the largest difference within each method and **bold** the one among all methods.