Head-to-Tail: How Knowledgeable are Large Language Models (LLMs)? A.K.A. Will LLMs Replace Knowledge Graphs?

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, Xin Luna Dong
Meta Reality Labs
{sunkaicn,ethanxu,hwzha,yuei,lunadong}@meta.com

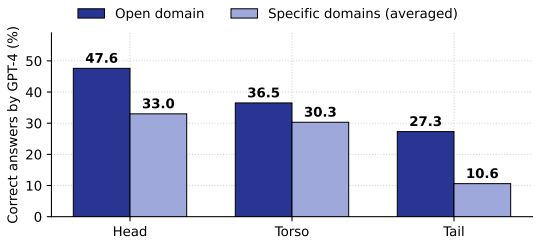
Abstract

Since the recent prosperity of Large Language Models (LLMs), there have been interleaved discussions regarding how to reduce hallucinations from LLM responses, how to increase the factuality of LLMs, and whether Knowledge Graphs (KGs), which store the world knowledge in a symbolic form, will be replaced with LLMs. In this paper, we try to answer these questions from a new angle: *How knowledgeable are LLMs?*

To answer this question, we constructed Head-to-Tail, a benchmark that consists of 18K question-answer (QA) pairs regarding head, torso, and tail facts in terms of popularity. We designed an automated evaluation method and a set of metrics that closely approximate the knowledge an LLM confidently internalizes. Through a comprehensive evaluation of 16 publicly available LLMs, we show that existing LLMs are still far from being perfect in terms of their grasp of factual knowledge, especially for facts of *torso-to-tail* entities.

1 Introduction

Pre-trained large language models (LLMs), such as ChatGPT[1], GPT-4 (OpenAI, 2023), and Llama 2 (Touvron et al., 2023b), have demonstrated impressive capabilities in internalizing knowledge and responding to common inquiries (Ouyang et al., 2022; OpenAI, 2023). Nevertheless, these models often lack knowledge of nuanced, domain-specific details and are susceptible to hallucinations (Bang et al., 2023), underscoring the significant challenges of increasing the *factuality* of LLMs and minimizing *hallucinations* from LLM responses. Conversely, the rise of LLMs has sparked debates on whether Knowledge Graphs (KGs), which store real-world factual knowledge in triplet form (subject, predicate, object), will be replaced with LLMs. This paper tries to answer these questions from a new angle: *How knowledgeable are LLMs?*



Example questions where GPT-4 gives incorrect answers

Movie

Question: What profession does Tj Singh (known for John Carter (2012)) have?
Ground Truth: Visual effects
GPT-4: Actor

Book

Question: Who authored Choke (published in 1996)?
Ground Truth: Stuart Woods
GPT-4: Chuck Palahniuk

Academics

Question: Where did Josef Kittler receive the Ph.D. (thesis: Development and application of pattern recognition techniques.)?
Ground Truth: University of Cambridge, UK
GPT-4: University of Surrey

Open

Question: What college is the sister college of Trinity College, Oxford?
Ground Truth: Churchill College, Cambridge
GPT-4: Balliol College

Figure 1: The question-answering accuracy of GPT-4 decreases in the order of head, torso, and tail entities on the Head-to-Tail benchmark, and is only 31% on average.

Finding answers to these questions is not easy. First, it is hard to directly "query" the knowledge embedded in an LLM—hallucination can be due to lack of knowledge but can also be caused by dysfunction of the generative model even if the knowledge is already parameterized in the model. We approximate the amount of knowledge in an LLM by its accuracy in answering simple-formed questions, such as *"where was the basketball player Michael Jordan born?"*; in addition, we ask the LLM to generate brief answers and admit "unsure" when its confidence is low. We chose this proxy because we found LLMs are normally very good at understand-

[1] https://openai.com/blog/chatgpt

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 311–325
June 16-21, 2024

ing simple-formed questions and produce consistent answers when regenerating answers, especially if asked to be brief (Section 3.5).

Second, there is no ready-to-use benchmark that either well represents distributions of user's interest (the query logs for major LLMs or search engines are not publicly available) or well represents the uniform distribution of the world knowledge (even the largest knowledge graphs admit sparsity of knowledge, especially towards non-popular facts). To address this challenge, we construct a benchmark of 18K QA pairs that cover various domains and various relationships in these domains. We bucket entities and relationships to *head, torso,* and *tail* according to how *popular* they are (details in Section 2) and randomly sample from each bucket; as such, we call our benchmark Head-to-Tail. This benchmark facilitates us to achieve a comprehensive view of how knowledgeable LLMs are regarding each bucket.

Through the Head-to-Tail benchmark and the experimental methodology, we answer the following three research questions (RQs):

RQ1: How reliable are LLMs in answering factual questions? (Section 3.2)

RQ2: Do LLMs perform equally well on head, torso, and tail facts? (Section 3.3)

RQ3: Do normal methods that improve LLMs, such as model size increase and instruction tuning, help LLMs to be more knowledgeable? (Section 3.4)

As shown in Figure 1, our analysis demonstrates a consistent decline in the performance of LLMs, following the order of head, torso, and tail entities, confirming our hypothesis that LLMs contain more head knowledge where training data abound. Surprisingly, even for the top-0.5% popular entities in popular domains such as *Movie*, the evaluated LLMs, at best, provide accurate answers for only ∼60% of the questions in the benchmark. Normal methods that enhance LLMs do not necessarily make them more knowledgeable, highlighting the need for more effective approaches to increase LLMs' factuality.

Our main contributions are as follows:

(i) We introduce Head-to-Tail, the first benchmark focused on comprehensively assessing the effectiveness of LLMs in incorporating factual knowledge encompassing the head, torso, and tail portions of knowledge graphs (Section 2.1). Head-to-Tail will be available at https://github.com/facebookresearch/head-to-tail.

(ii) We present an evaluation methodology accompanied by metrics designed to assess the factuality of LLMs. Our metrics allow us to distinguish hallucination and missing answers, and our evaluation method, whereas entirely automated, proves to be reliable and robust (Section 2.2-2.3).

(iii) We conducted a comprehensive evaluation and quantified the factuality of 16 LLMs regarding head, torso, and tail facts to answer the research questions (RQ1–RQ3) (Section 3). In light of these findings, we envision the future of knowledge graphs and outline a research landscape aimed at improving the overall factual reliability of LLMs (Section 4).

2 The Head-to-Tail Benchmark

We now describe the Head-to-Tail benchmark, the metrics, and our evaluation methodology.

2.1 QA pair generation

Domains and data sources. To cover a broad range of knowledge, we used the DBpedia knowledge graph (Auer et al., 2007), where the knowledge originates from Wikipedia (Denoyer and Gallinari, 2006). We used a cleaned version of the English snapshot from December 1, 2022.[2]

To better understand LLM performance on particular domains, we also selected three domains where public data are easily accessible.

- **Movie:** We used a snapshot of *IMDb*[3] from May 21, 2023.
- **Book:** We used the data of *Goodreads* scraped in 2017 released by Wan and McAuley (2018).
- **Academics:** We used a snapshot of *MAG* (Sinha et al., 2015) from September 13, 2021 and *DBLP*[4] from May 10, 2023.

Entities. An important contribution of the Head-to-Tail benchmark is the bucketing of head, torso,

[2] https://databus.dbpedia.org/dbpedia/mappings/mappingbased-objects

[3] https://developer.imdb.com/non-commercial-datasets/

[4] https://dblp.org/

	IMDb		Goodreads	MAG			DBLP	DBpedia
	Title	Person	Book	Article	Conference	Journal	Scholar	-
Head	767 (0.01)	34,903 (0.48)	3,150 (2.31)	1,827,710 (0.70)	257 (1.63)	225 (0.46)	79,521 (2.44)	103,564 (1.30)
Torso	4,113 (0.05)	87,645 (1.21)	7,304 (5.35)	9,386,034 (3.60)	965 (6.12)	1,266 (2.58)	500,778 (15.36)	1,255,113 (15.77)
Tail	7,536,482 (99.94)	7,111,496 (98.31)	126,134 (92.35)	249,311,539 (95.70)	14,550 (92.25)	47,546 (96.96)	2,680,704 (82.20)	6,600,206 (82.93)

Table 1: The number (%) of head, torso, and tail entities. The distribution follows the power law.

and tail entities, decided by the *popularity* of the entities (we will also discuss how the popularity of the predicates affect results in Section 3.3). We use two ways to approximate popularity: *traffic* and *density*. When there is traffic information, such as views and votes, we conveniently use traffic to measure the popularity; otherwise, we use density as a proxy, such as the number of facts or authored works about the entity. We often observe a correlation between density and traffic (e.g., the more popular a person is, the more we know about her), but as we will see soon from the benchmark statistics (Table 1), they can still lead to slightly different distributions of head, torso, and tail. We give details on how we decide the popularity of different types of entities from each data source in Appendix A.2.

We bucketed head, torso, and tail entities in three steps. First, we sorted the entities by their popularity, measured as above. Second, for each entity, we computed the cumulative popularity score up to the top-1 entity in the sorted list. Third, we bucketed the entities such that head entities comprise entities whose cumulative popularity score is up to 1/3 of that of all entities, torso entities comprise entities with cumulative scores ranging from 1/3 to 2/3, and tail entities from 2/3 to 1. (See Appendix A.7 for an example.) We determined the partitioning separately for different entity types for each domain.

To make the popularity score fair, we filtered out entities that are likely too new to have sufficient statistical data for popularity measurement. For IMDb, MAG, DBLP, and Goodreads, we kept only entities by the year of 2020, 2020, 2020, and 2015, respectively. The cut-off years are all before the cut-off time of the LLM training data, so the benchmark avoids questions that require *recent* knowledge. We did not perform similar filtering for DBpedia because the year attribute is unavailable or non-applicable for most entities, and our pilot study shows that very few (if at all) of the questions generated from DBpedia require knowledge after 2020.

Table 1 summarizes the distribution of head, torso, and tail entities. The distribution follows the *power law*, where very small percentages of entities fall in the head and torso buckets, and the majority of entities fall in the tail bucket; for example, over 99.9% of movies fall in the tail bucket, according to IMDb vote counts. We also observe that this phenomenon is more pronounced when we measure by traffic than by density; for the latter, the torso buckets are often larger (∼15% of entities), and the tails are slightly smaller (∼82%).

Questions. We generated questions using a template-based approach, where each generated question asks for an attribute of an entity. We filtered out the following types of attributes: (i) unspecific (e.g., `seeAlso` in DBpedia), (ii) dynamic (e.g., `lastLaunchRocket` in DBpedia), (iii) data source specific (e.g., `averageRating` in IMDb), and (iv) non-textual (e.g., `picture` in DBpedia). We discuss a stricter filtering criterion in Appendix A.4. For each specific domain (Movie, Book, Academics), we manually designed the question template for each attribute. DBpedia contains a large set of attributes, so we first employed ChatGPT to draft the templates (using Prompt 1 in Appendix A.1), then proofread them manually and made necessary edits. Each question template corresponds to a distinct predicate.

The answer for each question is the object of the relevant triple; when there are multiple answers (e.g., a book may have multiple authors), we included all in the answer. When necessary, we included extra information for an entity to avoid potential ambiguities (e.g., we included the publication year for a book to distinguish books of highly similar names).

We generated an equal number of questions for randomly sampled head, torso, and tail entities using each template. For each specific domain, we generated ∼1K questions for each of the head, torso, and tail buckets. As DBPedia contains more domains and relationship types, we generated ∼3K questions for each bucket. Table 2 summarizes the overall statistics of Head-to-Tail in the number of questions and templates.

Domain	Sources	# Templates	# Questions
Movie	IMDb	13	3,093
Book	Goodreads	4	3,000
Academics	MAG, DBLP	13	2,946
Open	DBpedia	393	9,132
Total		423	18,171

Table 2: The overall statistics of Head-to-Tail.

2.2 Metrics

Metrics. We find that oftentimes LLMs are intelligent enough to admit that it does not have enough information to answer a question. As such, we used three metrics: *accuracy* (**A**), *hallucination rate* (**H**), and *missing rate* (**M**), measuring the percentage of questions that an LLM gives the correct answer, gives a wrong or partially incorrect answer, or admits it cannot answer, respectively; by definition, $\mathrm{A} + \mathrm{H} + \mathrm{M} = 100\%$.

Manually deciding the correctness of answers can be cumbersome. We next describe a few different ways to automatically decide if an answer is correct.

LLM-based. We ask ChatGPT to check whether an answer is correct given the question and ground truth (Prompt 2 in Appendix A.1). Thus, accuracy $\mathbf{A_{LM}}$ is defined as the percentage of answers that ChatGPT judges as correct; hallucination rate $\mathbf{H_{LM}}$ is defined as the percentage of time when (i) an attempted answer is not missing, and (ii) ChatGPT judges the answer as incorrect (i.e., $\mathrm{H_{LM}} = 100\% - \mathrm{A_{LM}} - \mathrm{M}$).

To understand the reliability of the LLM-based metrics, we randomly sampled 840 answers from the evaluated LLMs and manually checked whether human judgment agrees with the LLM-based metrics. The agreement is 98%, which we view as reliable. Hence, we use $\mathrm{A_{LM}}$ and $\mathrm{H_{LM}}$ as the primary metrics in this study.

Rule-based. In addition, we adopt popular metrics, including *exact match (EM)*, *token F1 (F1)*, and *ROUGE-L (RL)* (Lin, 2004; Rajpurkar et al., 2016); in other words, we use rule-based methods to judge the correctness of an answer. Specifically, $\mathbf{A_{EM}}$ is computed as the percentage of answers that exactly match the ground truth; $\mathbf{A_{F1}}$ is computed as the average harmonic mean of precision and recall when comparing tokens in the returned answers and in the ground truth answers; $\mathbf{A_{RL}}$ is computed as the average normalized longest common subsequence (LCS) between the returned answers and the ground truths. For common answer types, we additionally expand the set of ground-truth answers with their variants using hand-crafted rules (e.g., "*W Shakespeare*" is a variant of "*William Shakespeare*"); when a given question has multiple expanded ground-truth answers, we take the maximum score.

Correspondingly, we measure hallucination rate by $\mathbf{H_{EM}}$ ($= 100\% - \mathrm{A_{EM}} - \mathrm{M}$), $\mathbf{H_{F1}}$ ($= 100\% - \mathrm{A_{F1}} - \mathrm{M}$), and $\mathbf{H_{RL}}$ ($= 100\% - \mathrm{A_{RL}} - \mathrm{M}$). As we will show later in Section 3.5, we observe high correlations between rule-based and LLM-based metrics.

2.3 Evaluation methodology

We prompted the LLM as shown in Prompt 3 in Appendix A.1. First, we asked LLMs to give as concise answers as possible. Second, we prompted LLMs to respond "unsure" when the LLM is not confident in the answer. We applied few-shot learning and included in the prompt two examples that are not in Head-to-Tail: one is a simple, answerable question with the corresponding answer as the response; the other is an unanswerable question with "unsure" as the response.

With this prompt, rule-based metrics are more likely to reflect the factual correctness of the answers, and we can simply compute the missing rate (i.e., M) by counting "unsure" or empty answers. We observed that explicitly asking for "unsure" as an answer could significantly reduce hallucination rate (Section 3.5).

To summarize, the following three setups in the benchmark and evaluation methodology help us best approximate the existence of (confident) knowledge in the LLMs: (i) focusing on simple questions in easy-to-understand forms, (ii) asking for concise answers to ease evaluation, and (iii) hinting the LLMs to answer "unsure" to suppress unnecessary hallucinations.

3 Experimental Analysis

3.1 Models and configurations

We evaluated representative state-of-the-art LLMs of various sizes and architectures, including ChatGPT, GPT-4 (OpenAI, 2023), LLaMA (7B, 13B, 33B, 65B) (Touvron et al., 2023a), Llama 2 (70B) (Touvron et al., 2023b), Vicuna (7B, 13B) (Chiang et al., 2023), Flan-T5 (3B, 11B) (Chung et al., 2022), RWKV (7B) (Peng et al., 2023b), Falcon (7B, 40B), and Falcon-Instruct (7B, 40B) (Almazrouei et al., 2023). We

Model	All		Open		Movie		Book		Academics	
	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
GPT-4	30.9	19.7	37.1	25.3	41.7	15.5	21.3	19.4	10.0	6.8
ChatGPT	20.3	14.1	22.1	14.8	34.7	13.3	16.9	24.9	3.0	1.9
Llama 2 (70B)	11.8	34.0	7.5	24.8	27.9	34.3	10.3	54.5	9.8	41.0
LLaMA (33B)	18.2	80.0	19.0	79.1	28.7	70.1	15.8	82.9	7.1	90.3

Table 3: The best overall accuracy is only ∼31% on Head-to-Tail. All numbers are in percentage (%).

employed the most deterministic settings (i.e., `temperature=0` or `top_k=1`) for all models. We present more details in Appendix A.3.

Table 14 in Appendix A.8 gives detailed results of all LLMs. We note that our goal is NOT to compare different LLM models; rather, by examining the metrics by different LLMs, we make sure to report the common patterns among the representative LLMs. We also note that it is hard to exhaustively benchmark every recent model in this fast-moving field; we conducted evaluations up to GPT-4 (OpenAI, 2023) and Llama 2 (Touvron et al., 2023b), and detailed discussions can be based on slightly older models, where we observe similar patterns.

3.2 RQ1: How reliable are LLMs in answering factual questions?

We present in Table 3 the overall performance of GPT-4, ChatGPT, Llama 2-70B, and LLaMA-33B, which perform the best in most metrics on Head-to-Tail among all LLMs introduced in Section 3.1. The best overall accuracy is obtained by GPT-4 at 31%.

Interestingly, for questions that are not answered correctly, different LLMs show different patterns: GPT-4 and ChatGPT give unsure or empty answers for the majority of them, and the hallucination rate is <20% (still non-negligible); LLaMA-33B mostly provides hallucinated answers, resulting with high hallucination rate (∼80%); Llama 2-70B falls in-between. We suspect fine-tuning and reinforcement learning of these models may explain the different patterns when the model is unsure of the answers. Figure 1 shows examples of counterfactual answers given by GPT-4.

Finally, for all models, the overall performance varies substantially across different specific domains. All models perform the best in the *Movie* domain and worst in the *Academics* domain, likely because of the relatively low popularity of the *Academics* domain, as we will discuss soon.

Domain	Head		Torso		Tail	
	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
Movie	59.3	14.8	55.0	16.9	10.9	14.7
Book	22.8	24.4	24.3	21.8	16.9	12.0
Academics	15.8	9.9	10.5	6.8	3.9	3.7
Open	47.6	30.2	36.5	24.1	27.3	21.6
All	40.3	23.3	33.4	19.7	19.0	15.9

(a) GPT-4.

Domain	Head		Torso		Tail	
	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
Movie	39.2	28.2	33.9	29.8	10.7	44.9
Book	15.0	52.2	12.9	54.4	3.1	56.9
Academics	12.9	35.2	11.1	38.2	5.3	49.7
Open	9.9	22.3	6.9	25.4	5.7	26.8
All	16.2	30.3	13.2	33.0	6.1	38.6

(b) Llama 2-70B.

Table 4: LLMs' factuality, measured by A_{LM} (%), decreases in the order of head, torso, and tail entities from Head-to-Tail.

Model	A_{LM}	H_{LM}	M
GPT-4	46.0 (↑5.7)	21.4 (↓1.9)	32.6 (↓3.7)
Llama 2 (70B)	18.7 (↑2.5)	29.7 (↓0.6)	51.6 (↓1.9)

Table 5: Accuracy on the top-10% popular questions in the head bucket is only slightly better than overall head entities. (↑/↓: increased/decreased % compared with using all head instances.)

3.3 RQ2: Do LLMs perform equally well on head, torso, and tail facts?

The overall accuracy of GPT-4 and Llama 2-70B (A_{LM}) declines in the order of head, torso, and tail entities, as shown in Figure 1 and Table 4. We observe the same pattern for other LLMs. This verifies our hypothesis that as we lack training data for long-tail entities, it is difficult for LLMs to obtain knowledge for such entities.

Surprisingly, the QA accuracy is still low even for the head entities (e.g., GPT-4 achieves an A_{LM} of 48% in the open domain). We further retain top-10% popular questions from the head bucket. As shown in Table 5, GPT-4 and Llama 2-70B obtained slightly higher accuracy (within 6 percent point) and lower hallucination rate for these super

Model	Head & Torso		Tail	
	A_{LM}	H_{LM}	A_{LM}	H_{LM}
GPT-4	42.9	20.3	36.8	25.6
ChatGPT	18.6	14.2	22.3	14.8
Llama 2 (70B)	8.0	42.1	7.5	23.8
LLaMA (7B)	15.3	83.5	13.5	77.7
LLaMA (13B)	14.6	85.1	14.7	83.6
LLaMA (33B)	18.2	81.4	19.0	78.9
LLaMA (65B)	20.1	79.7	18.3	81.4
Vicuna (7B)	12.5	82.0	9.3	77.4
Vicuna (13B)	13.0	70.3	8.6	55.5
Flan-T5 (3B)	4.4	13.0	3.4	10.4
Flan-T5 (11B)	9.2	11.1	5.0	8.1
RWKV (7B)	6.9	28.7	6.4	29.7
Falcon (7B)	11.3	51.0	8.1	43.7
Falcon (40B)	14.4	34.1	8.5	29.2
Falcon-Instruct (7B)	8.8	48.3	7.2	47.1
Falcon-Instruct (40B)	12.8	15.3	7.9	15.2

Table 6: Comparison of LLMs' factuality about head, torso, and tail predicates in A_{LM} (%) and H_{LM} (%) using open-domain instances from Head-to-Tail.

popular entities, but the accuracy is still disappointingly low (46% for GPT-4 and 19% for Llama 2-70B), and the missing rate is notable. We have a further discussion in Appendix A.6.

The QA accuracy on tail entities is significantly lower in most of the domains. Notably, *Academics* intuitively is a long-tail domain, and we observe $\sim$10% overall accuracy and very low accuracy (16% for GPT-4 and 13% for Llama 2-70B) even for head entities in this domain.

Finally, hallucination rate drops from head to torso to tail for GPT-4, but increases for Llama 2-70B. We hypothesize that there is at least one more factor that affects the hallucination rate—the internal assessment of the confidence. When an LLM "knows" what is unknown to it, it is likely to reduce confidence when answering related questions and produce fewer hallucinations.

Head-to-tail predicates. We investigated whether the performance still correlates with the head-to-tail order regarding the popularity of *predicates* instead of entities. We sorted the predicates from DBpedia by popularity (measured by the number of relational triples with the predicate) and partitioned the sorted predicates into head, torso, and tail in a similar fashion. We then re-partitioned the open-domain questions into head, torso, and tail predicate buckets, each containing 72, 450, and 8, 610 questions, respectively. Since the number of questions in the head bucket is low, we merged the head and torso buckets.

Table 6 compares the performance on head & torso vs. on tail. We observe no consistent correlation among different LLMs between the performance and the head-to-tail predicate ordering, and the differences in accuracy are not very high. This is not too surprising for two reasons. First, the semantics of each predicate is mostly consistent with the semantics of the predicate names, which can be well understood by LLMs. Second, when facts are present for tail predicates, they are often about the head entities, and factual information for head entities is likely to be more abundant in the training data.

3.4 RQ3: Does normal methods that improve LLMs increase the factuality?

Table 7 compares LLMs in different sizes and with or without instruction tuning. First, we observe that an increased model size does not automatically translate to a better grasp of factual knowledge. For example, LLaMA-33B modestly outperforms LLaMA-65B across the head, torso, and tail subsets ($+0.4\%$ in A_{LM} and -1.9% in H_{LM} on average) while they share the same training dataset and hyperparameters. This provides additional evidence for our hypothesis that once the model is sufficiently large, the abundance of training data plays a more critical role in the factuality of the LLMs.

Second, compared with LLaMA and Falcon, the instruction-tuned counterparts (i.e., Vicuna and Falcon-Instruct) have lower accuracy, as they learned to be more conservative in providing factual answers and thus generate "unsure" more often (e.g., Vicuna-13B is 26.9% higher in M than LLaMA-13B). Despite so, they still have high hallucination rate.

3.5 Robustness of our evaluation methodology

Finally, we evaluate the robustness of our evaluation methodology.

Correlations between rule- and LLM-based metrics. For each combination of popularity (head, torso, tail) and domain (movie, book, academics, open), we calculate Spearman's rank and Pearson correlation coefficients between rule- and LLM-based metrics over all LLMs. We report the aggregated results (minimum, mean) in Table 8. The correlation scores suggest that A_{LM} (resp. H_{LM}) strongly correlates with A_{EM}, A_{F1}, and A_{RL} (resp. H_{EM}, H_{F1}, and H_{RL}), indicating that rule-based metrics are good alternatives for lower-cost or faster evaluation.

Model	Head-to-Tail			Head		Torso		Tail	
	A_{LM}	H_{LM}	M	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
LLaMA (7B)	12.1	80.0	7.9	19.0	74.4	11.7	81.0	5.4	84.8
LLaMA (13B)	14.4	84.3	1.3	22.0	77.2	14.8	83.8	6.3	91.9
LLaMA (33B)	18.2	80.0	1.8	26.0	72.8	19.8	78.7	8.8	88.6
LLaMA (65B)	17.8	81.9	0.3	25.9	73.8	18.7	81.0	8.7	90.9
Vicuna (7B)	10.1	79.2	10.8	16.2	72.7	9.6	79.8	4.3	85.0
Vicuna (13B)	9.2	62.6	28.2	14.0	55.0	8.8	62.8	4.7	70.0
Flan-T5 (3B)	2.3	17.4	80.3	3.9	19.7	1.5	17.1	1.3	15.5
Flan-T5 (11B)	4.2	20.0	75.7	7.6	23.7	3.2	19.9	2.0	16.5
Falcon (7B)	9.5	57.9	32.6	14.5	53.8	9.2	57.9	4.8	62.0
Falcon (40B)	10.8	41.0	48.2	16.2	36.4	11.2	40.0	4.9	46.6
Falcon-Instruct (7B)	6.8	56.7	36.5	11.5	56.0	5.6	57.2	3.4	56.7
Falcon-Instruct (40B)	10.8	32.2	57.0	16.7	30.5	11.5	31.1	4.3	34.8

Table 7: Comparison of different LLMs with different sizes. All numbers are in percentage (%).

LLM-Based	Rule-Based	ρ		r	
		Min.	Mean	Min.	Mean
A_{LM}	A_{EM}	0.721	0.915	0.921	0.966
	A_{F1}	0.775	0.951	0.781	0.969
	A_{RL}	0.730	0.947	0.775	0.969
H_{LM}	H_{EM}	0.968	0.991	0.993	0.998
	H_{F1}	0.976	0.995	0.998	0.999
	H_{RL}	0.976	0.995	0.998	0.999

Table 8: The minimum and mean Spearman's rank correlation coefficients (ρ) and Pearson correlation coefficients (r) show high correlation between LM- and rule-based metrics.

	Domain	Few-shot		Zero-shot		In-domain	
		A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
Head	Open	32.7	**20.8**	32.6	24.7	**45.0**	27.8
	All	29.4	**17.2**	29.2	18.6	**38.3**	24.7
Torso	Open	19.7	**13.3**	21.6	17.9	**30.1**	23.0
	All	21.9	**14.6**	22.8	16.7	**29.8**	22.8
Tail	Open	13.8	**10.2**	14.9	14.5	**23.0**	19.5
	All	9.5	**10.5**	10.3	12.7	**15.4**	20.2

Table 9: Performance of ChatGPT with different prompts on Head-to-Tail. All numbers are in percentage (%).

Effect of brief and "unsure". We randomly sampled 1.2K questions and tested the stability of answers if we call ChatGPT to regenerate answers. When not requiring brief or "unsure" answers, for 18% of questions, ChatGPT regenerated different answers. Adding the requirement for brief answers (Prompt 6 in Appendix A.1) reduced the percentage to 4%, and further asking "unsure" answers with few-shot examples (Prompt 3) reduced the percentage to 1%. In addition, according to manual evaluation on 150 randomly sampled questions, removing "unsure" as an option increases ChatGPT's hallucination rate by 13 percentage points.

Robustness of prompts. We explore two other prompts. Compared with the original prompt that conducts few-shot learning (Section 3.1), denoted as **Few-shot**, the **Zero-shot** prompt does not provide examples and thus is zero-shot learning (Prompt 4 in Appendix A.1), and the **In-domain** prompt has the answerable example swapped out for an in-domain example generated by the same question template as the target question (Prompt 5 in Appendix A.1).

As shown in Table 9, **Few-shot** and **Zero-shot** show very similar results, but performance differences are noticeable between **Few-shot** and **In-domain**. In particular, in-domain examples help get more correct answers (+8.9%, +7.9%, +5.9% in A_{LM} for head, torso, tail) but at the cost of more hallucinations (+7.5%, +8.2%, +9.7% in H_{LM} for head, torso, tail). We suspect that the in-domain examples boost the confidence of ChatGPT in answering a question, so it answers questions even when the real confidence is not that high, causing both higher accuracy and higher hallucination rate.

Despite the fluctuation, our original prompt template (**Few-shot**) appears to be better at approximating the (confident) factuality of LLMs with the QA accuracy, and the *relative* performance among the head, torso, and tail remains stable over different prompts.

4 Discussions

4.1 The future of knowledge graphs

The experimental analysis indicates that although LLMs have incorporated factual knowledge within their parameters, the amount of this encoded knowledge remains limited. Knowledge of long-tail entities is already sparse in KGs and is even more deficient in LLMs.

Nevertheless, LLMs have been revolutionizing the way people seek information and calling for reconsideration of the best representation of factual knowledge. We term the forthcoming generation of KGs as *Dual Neural KGs*: knowledge can reside explicitly as triples (similar to KGs) and implicitly as embeddings (like in LLMs); the symbolic form caters to human understanding and explainability, while the neural form benefits machine comprehension and seamless conversations. A piece of knowledge can exist in both formats or in the one that is more appropriate. The harmonious blend of the two forms, capitalizing on the latest LLM innovations, is an exciting research area as we elaborate next.

Head knowledge. This involves popular entities where training data are ample. Ideally, LLMs could be taught such knowledge for efficient retrieval, meaning head knowledge shall exist in both forms. Currently, LLMs still have a mediocre QA accuracy for popular entities (see Table 5), so a critical research area is to infuse head knowledge into LLMs through model training or fine-tuning. Early work in this line includes knowledge infusion (Liu et al., 2021; Wang et al., 2021; Zhen et al., 2022).

Torso-to-tail and recent knowledge. This involves non-popular entities and emerging knowledge, where training data are typically sparse or absent. This type of knowledge might be best represented as triples. Serving such knowledge requires effectively deciding when external knowledge is essential, efficiently retrieving the relevant knowledge, and seamlessly integrating it into the answers. Early attempts in this direction involve knowledge-augmented LLMs (Asai et al., 2023; Nakano et al., 2022; Shi et al., 2023; Borgeaud et al., 2022).

4.2 Limitations and extensions

Taxonomy. Our work does not discuss the effectiveness of LLMs in capturing taxonomy or type hierarchies, which could be an extension of this study. Specifically, we hypothesize that LLMs can effectively incorporate type relationships (e.g., hypernyms and synonyms), even for the fine-granularity sub-types. Hence, it may no longer be worth manually constructing a very deep and complex hierarchy in the future.

Robustness to question formulation. This paper primarily aims to evaluate how much an LLM "knows" a fact with high confidence; we thus tested various ways of formulating factual questions and selected the least ambiguous form for this study. However, this approach does not assess the model's robustness to paraphrasing or consider the diverse ways models can be queried, such as entailment or cloze-style prompts. Our supplementary experiment in Appendix A.5 suggests that varying the form of questions does not significantly impact the evaluation results. A more thorough evaluation of robustness is beyond the scope of this paper and left for future research.

5 Related Work

Benchmarks. Most works studied the factuality of LLMs using existing QA benchmarks such as WebQuestions (Berant et al., 2013), TriviaQA (Joshi et al., 2017), LC-QuAD (Trivedi et al., 2017; Dubey et al., 2019), QALD-9 (Usbeck et al., 2018), Natural Questions (Kwiatkowski et al., 2019), and EntityQuestions (Sciavolino et al., 2021). A recent line of work has been constructing new QA benchmarks to assess LLMs' factuality, especially for long-tail knowledge (Mallen et al., 2023; Kim et al., 2023). Compared with these benchmarks, Head-to-Tail is the first to specifically assess how well LLMs incorporate head, torso, and tail factual information.

LLM Evaluation. Recent years have seen a proliferation of research on assessing the factuality of LLMs (Roberts et al., 2020; Petroni et al., 2021; Shuster et al., 2021; Mielke et al., 2022; Tan et al., 2023; Hu et al., 2023; Peng et al., 2023a; Omar et al., 2023; Kandpal et al., 2023; Mallen et al., 2023; Chen et al., 2023). Most of these works focus on a single knowledge source, such as Freebase or Wikipedia, and they have yet to systematically perform the evaluation explicitly regarding head/torso/tail entities or attributes. One work close to ours is Omar et al. (2023), which evaluated ChatGPT using facts collected from diverse knowledge sources; however, their evaluation was carried out manually on only 450 QA instances.

There are three works that also showed the correlation between the QA accuracy of language models and fact popularity (Mallen et al., 2023; Kandpal et al., 2023; Kim et al., 2023). Our work, conducted in parallel, focuses on a different angle—how knowledgeable are LLMs? For this purpose, we systematically designed experimental methodology, including the definition of head, torso, and tail entities, the design of metrics, and the evaluation method. Our benchmark is comprehensive in containing different knowledge sources, different

domains, and rich relations. Compared with these three works, we gave more quantified answers for research questions RQ1–RQ3.

6 Conclusion

We introduce Head-to-Tail, the first benchmark designed to assess the ability of LLMs to internalize head, torso, and tail facts. Alongside the dataset, we present a new evaluation methodology with appropriate metrics for automatically evaluating LLMs' factuality. Our evaluation shows that even the most advanced LLMs have notable limitations in representing factual knowledge, particularly for the torso and tail entities. Accordingly, we suggest new research areas to seamlessly blend knowledge in the symbolic form and neural form.

Acknowledgements

We would like to thank the anonymous ARR reviewers and meta reviewer for their constructive and insightful feedback.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46, Toronto, Canada. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, and etc. Jordan Hoffmann†. 2022. Improving language models by retrieving from trillions of tokens. *arXiv*.

Lihu Chen, Simon Razniewski, and Gerhard Weikum. 2023. Knowledge base completion for long-tail entities. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pages 99–108, Toronto, ON, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Ludovic Denoyer and Patrick Gallinari. 2006. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II*, page 69–78, Berlin, Heidelberg. Springer-Verlag.

Nan Hu, Yike Wu, Guilin Qi, Dehai Min, Jiaoyan Chen, Jeff Z Pan, and Zafar Ali. 2023. An empirical study of pre-trained language models in simple knowledge graph question answering. *World Wide Web*, pages 1–32.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15696–15707. PMLR.

Youngmin Kim, Rohan Kumar, Sunitha Ravi, Haitian Sun, Christos Faloutsos, Ruslan Salakhutdinov, and Minji Yoon. 2023. Automatic question-answer generation for long-tail knowledge. In *Second Workshop on Knowledge Augmented Methods for Natural Language Processing (KDD-KnowledgeNLP)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *AAAI*.

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)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *arXiv*.

Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots.

OpenAI. 2023. Gpt-4 technical report.

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

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023a. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Xiangru Tang, Bolun Wang, Johan S. Wind, Stansilaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. 2023b. RWKV: Reinventing rnns for the transformer era.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 243–246, New York, NY, USA. Association for Computing Machinery.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *The Semantic Web – ISWC 2017*, pages 210–218, Cham. Springer International Publishing.

Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 2018. 9th challenge on question answering over linked data (QALD-9). In *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018.*, pages 58–64.

Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 86–94, New York, NY, USA. Association for Computing Machinery.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-adapter: Infusing knowledge into pre-trained models with adapters. In *ACL*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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*, pages 38–45, Online. Association for Computational Linguistics.

Chaoqi Zhen, Yanlei Shang, Xiangyu Liu, Yifei Li, Yong Chen, and Dell Zhang. 2022. A survey on knowledge-enhanced pre-trained language models.

A Appendix

A.1 List of Prompts

```
You are given a few samples of a relation in the format of <X, relation, Y>. You need to write a question *template* about the relation, which can be used to generate questions. The template needs to have one blank such that a question about Y can be generated by filling the blank with X.

#Example 1
Samples: <!Hero, musicBy, Eddie DeGarmo>, <9 to 5 (musical), musicBy, Dolly Parton>, <All About Us (musical), musicBy, John Kander>
Template: The music of _ is by whom?

#Example 2
Samples: <10,000 Maniacs, bandMember, Dennis Drew>, <16bit (band), bandMember, Eddie Jefferys>, <1TYM, bandMember, Teddy Park>
Template: Name a band member of _?

#Example 3
Samples: {SAMPLES}
Template:
```

Prompt 1: Question template drafting.

```
You need to check whether the prediction of a question-answering system to a question is correct. You should make the judgment based on a list of ground truth answers provided to you. Your response should be "correct" if the prediction is correct or "incorrect" if the prediction is wrong.

Question: Who authored The Taming of the Shrew (published in 2002)?
Ground truth: ["William Shakespeare", "Roma Gill"]
Prediction: W Shakespeare
Correctness: correct

Question: Who authored The Taming of the Shrew (published in 2002)?
Ground truth: ["William Shakespeare", "Roma Gill"]
Prediction: Roma Gill and W Shakespeare
Correctness: correct

Question: Who authored The Taming of the Shrew (published in 2002)?
Ground truth: ["William Shakespeare", "Roma Gill"]
Prediction: Roma Shakespeare
Correctness: incorrect

Question: What country is Maharashtra Metro Rail Corporation Limited located in?
Ground truth: ["India"]
Prediction: Maharashtra
Correctness: incorrect

Question: What's the job of Song Kang-ho in Parasite (2019)?
Ground truth: ["actor"]
Prediction: He plays the role of Kim Ki-taek, the patriarch of the Kim family.
Correctness: correct

Question: Which era did Michael Oakeshott belong to?
Ground truth: ["20th-century philosophy"]
Prediction: 20th century.
Correctness: correct

Question: Edward Tise (known for Full Metal Jacket (1987)) is in what department?
Ground truth: ["sound department"]
Prediction: 2nd Infantry Division, United States Army
Correctness: incorrect

Question: What wine region is Finger Lakes AVA a part of?
Ground truth: ["New York wine"]
Prediction: Finger Lakes AVA
Correctness: incorrect

Question: {QUESTION}
Ground truth: {GROUND_TRUTH}
Prediction: {PREDICTION}
Correctness:
```

Prompt 2: Correctness checking.

```
Answer the following questions in as few words as possible. Say "unsure" if you don't know.

Question: What is the capital of China?
Answer: Beijing

Question: What is the captical of Wernythedia?
Answer: unsure

Question: {QUESTION}
Answer:
```

Prompt 3: Question answering (Few-shot).

```
Answer the following question in as few words as possible. Say "unsure" if you don't know. {QUESTION}
```

Prompt 4: Question answering (Zero-shot).

```
Answer the following questions in as few words as possible. Say "unsure" if you don't know.

Question: What is the captical of Wernythedia?
Answer: unsure

Question: {QUESTION#}
Answer: {ANSWER#}

Question: {QUESTION}
Answer:
```

Prompt 5: Question answering (In-domain) (#: the in-domain instance described in Section 3.5).

```
Answer the following questions in as few words as possible. {QUESTION}
```

Prompt 6: Question answering (simply asking for concise answers).

```
Answer the following questions in as few words as possible. Return your best guess if you don't know.

Question: What is the capital of China?
Answer: Beijing

Question: {QUESTION}
Answer:
```

Prompt 7: Question answering (returning its best guess instead of "unsure" when the confidence is low).

A.2 Popularity measure in head-to-tail partition

- **IMDb** (traffic): The number of votes (i.e., `numVotes`) the *title* (e.g., movie, short, TV series, etc.) has received; we do NOT consider whether the vote is high or low in the counting. For person entities, we use the total number of votes received by the titles the person is known for.
- **Goodreads** (traffic): The count of ratings (i.e., `ratings_count`) the book has received; similarly, we do NOT take into consideration whether the rating is high or low.

- **MAG** (traffic): The number of citations (i.e., `CitationCount`) the entity (i.e., scholarly article, conference, or journal) has received.
- **DBLP** (density): The number of works the scholar has authored.
- **DBpedia** (density): The number of relational triples in DBPedia that contain the entity.

A.3 Implementation details

We interacted with ChatGPT and GPT-4 through OpenAI API[5]. The employed version of ChatGPT and GPT-4 is `gpt-3.5-turbo-0301` and `gpt-4-0613`, respectively. We used Transformers (Wolf et al., 2020) to interact with the other LLMs on A100 (80GB) GPUs, and we used 16-bit floating point formats (i.e., float16 for Flan-T5 and RWKV, bfloat16 for LLaMA, Llama 2, Vicuna, Falcon, and Falcon-Instruct). We employed the original LLaMA, Llama 2, Flan-T5, Falcon, and Falcon-Instruct versions. The employed version of RWKV and Vicuna is `v4 Raven` and `v1.1`, respectively.

A.4 Impact of less naturally occurring questions

Model	Movie		Book		Academics	
	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
GPT-4	43.8	12.6	39.1	24.6	11.2	9.8
ChatGPT	37.8	14.5	31.0	21.5	2.3	1.6
Llama 2 (70B)	30.4	31.6	19.7	10.0	4.5	59.0
LLaMA (7B)	18.7	71.9	21.4	59.1	2.4	94.9
LLaMA (13B)	25.3	73.2	24.4	74.5	4.6	93.7
LLaMA (33B)	31.2	67.5	31.7	65.9	5.2	91.7
LLaMA (65B)	27.1	72.4	32.3	66.5	8.2	91.8
Vicuna (7B)	21.1	68.5	19.2	62.9	2.6	91.3
Vicuna (13B)	20.3	58.5	11.4	38.7	3.1	77.4
Flan-T5 (3B)	1.7	15.1	2.8	4.5	0.2	4.3
Flan-T5 (11B)	6.3	22.1	6.6	10.1	0.8	16.0
RWKV (7B)	4.5	24.3	13.3	37.0	0.1	9.5
Falcon (7B)	20.4	62.3	15.0	45.1	3.5	80.7
Falcon (40B)	26.0	43.1	11.4	7.1	4.7	55.0
Falcon-Instruct (7B)	13.4	66.4	9.9	34.1	1.9	58.5
Falcon-Instruct (40B)	28.4	37.5	11.9	1.9	3.9	51.2

Table 10: Comparison of LLMs' factuality on Head-to-Tail without relatively less naturally occurring questions. All numbers are in percentage (%).

When constructing Head-to-Tail, we include all predicates that allow reasonable factual questions. Table 10, instead, shows metrics on predicates that users are more likely to ask about. In general we observed higher performance on the *Movie* and *Book* domains, but the accuracy is still fairly low and we observe similar patterns regarding head, torso, and tail entities.

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

A.5 Asking questions in different forms

	Head		Torso		Tail	
	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
Original	51.3	11.5	46.4	16.6	6.4	11.8
Cloze-style	50.8	8.9	46.1	14.3	6.8	12.6

Table 11: ChatGPT's factuality in A_{LM} (%) and H_{LM} (%) obtained by the cloze-style queries closely mirrors that of the simple-formed questions in the *Movie* domain.

We explored the influence of question formulation on the evaluation results using ChatGPT in the *Movie* domain. We rewrote all questions as cloze-style questions (e.g., "What's the release year of Mr. & Mrs. Smith" was transformed to "The release year of Mr. & Mrs. Smith is _"). As shown in Table 11, the performance obtained by the cloze-style queries is very similar to that obtained by simple-formed questions.

A.6 Further discussions on the missing rate

Domain	A_{LM}	H_{LM}	M
Movie	63.5	13.5	22.9
Academics	25.3	14.7	60.0

Table 12: Performance of GPT-4 on the top-10% popular questions in the head bucket. All numbers are in percentage (%).

Prompt	A_{LM}	H_{LM}	M
Original (Prompt 3)	63.5	13.5	22.9
Prompt 7	68.8	16.7	14.6

Table 13: Performance of GPT-4 on the top-10% popular questions in the head bucket in the *Movie* domain. All numbers are in percentage (%).

It is observed that even for the top-10% popular questions in the head bucket, the missing rate of GPT-4 is still over 30% (Table 5). Although this might seem counterintuitive, there are two reasons. First, the performance reported in Table 5 is based on all the studied domains, including the tail domain *Academics*. Table 12 compares GPT-4's performance on the top-10% head entities in the *Academics* and the *Movie* domains. The missing rate on the more popular domain *Movie* is much lower (23%). Second, if we explicitly ask the LLM to return the best guess (Prompt 7) instead of responding "unsure", GPT-4's missing rate on the top 10%

of head entities in the *Movie* domain would further drop to 15% (Table 13). However, this is with the price of higher hallucination rate, showing that the confidence of this part of knowledge is low. Interestingly, even after the above change, GPT-4 still admits to being "unsure" for 15% of questions (e.g., GPT-4's answers are "unknown" given the questions "What is the death year of Debbi Datz-Pyle (known for The Matrix (1999))?", "What movie is Alan R. Kessler known for?"). This further confirms that LLMs are not good at memorizing (internalizing) factual information.

A.7 An example of entity bucketing

Suppose there are 12 entities $A, B, C, \ldots, L$, and their popularity scores are $A = 8$, $B = 4$, $C = D = 2$, $E = F = \ldots = L = 1$. The total popularity scores add up to 24 ($= 8 + 4 + 2 + 2 + 1 \times 8$). Top-$1/3$ traffic (a total score of 8) is contributed by $\{A\}$, thus the head; mid-$1/3$ traffic is contributed by $\{B, C, D\}$ (a total score of $4 + 2 + 2 = 8$), thus the torso; bottom-$1/3$ traffic is contributed by $\{E, F, \ldots, L\}$ (a total score of $1 \times 8 = 8$), thus the tail.

A.8 Supplemental Results

	Model	All									Movie		Book		Academics		Open	
		A_{EM}	H_{EM}	A_{F1}	H_{F1}	A_{RL}	H_{RL}	A_{LM}	H_{LM}	M	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}	A_{LM}	H_{LM}
Head	GPT-4	**31.1**	32.6	**37.2**	26.5	**37.1**	26.5	**40.3**	23.3	36.3	**59.3**	14.8	22.8	**24.4**	15.8	9.9	**47.6**	30.2
	ChatGPT	21.8	24.9	25.6	21.1	25.6	21.1	29.4	**17.2**	53.3	51.3	**11.5**	20.1	26.3	5.9	**3.0**	32.7	20.8
	Llama 2 (70B)	13.9	32.7	16.5	30.0	16.5	30.1	16.2	30.3	53.5	39.2	28.2	15.0	52.2	12.9	35.2	9.9	22.3
	LLaMA (7B)	10.4	83.0	15.5	77.9	15.4	78.0	19.0	74.4	6.6	27.2	69.5	21.6	74.8	5.0	90.6	19.9	70.7
	LLaMA (13B)	12.7	86.6	18.1	81.1	18.0	81.2	22.0	77.2	0.8	36.5	63.1	20.1	79.9	9.7	89.8	21.7	77.1
	LLaMA (33B)	16.7	82.0	22.3	76.5	22.2	76.5	26.0	72.8	1.3	42.9	57.0	**24.2**	75.8	10.8	87.5	25.8	72.3
	LLaMA (65B)	14.9	84.8	21.7	78.0	21.6	78.1	25.9	73.8	0.3	37.0	62.3	23.1	76.9	**16.5**	83.5	26.1	73.7
	Vicuna (7B)	9.4	79.6	13.7	75.3	13.6	75.4	16.2	72.7	11.0	30.1	59.7	18.6	75.6	3.9	91.4	14.8	70.2
	Vicuna (13B)	8.7	60.3	11.9	57.1	11.9	57.1	14.0	55.0	31.0	29.9	52.0	10.8	64.0	5.4	74.3	12.5	46.8
	Flan-T5 (3B)	2.5	**21.1**	3.3	**20.3**	3.3	**20.3**	3.9	19.7	76.4	2.3	19.9	3.7	52.1	0.1	7.9	5.7	12.8
	Flan-T5 (11B)	5.8	25.5	7.4	23.9	7.4	23.9	7.6	23.7	68.7	10.7	30.1	7.9	59.5	0.4	21.0	8.8	**10.6**
	RWKV (7B)	6.2	35.3	8.3	33.2	8.2	33.2	9.6	31.9	58.5	9.8	26.7	15.4	49.8	0.2	13.2	10.7	33.7
	Falcon (7B)	10.0	58.3	12.9	55.4	12.8	55.5	14.5	53.8	31.7	28.1	51.6	14.8	68.3	8.1	80.4	11.9	41.1
	Falcon (40B)	12.1	40.5	14.5	38.1	14.6	38.0	16.2	36.4	47.4	36.0	30.3	10.2	52.9	11.5	61.8	13.1	24.8
	Falcon-Instruct (7B)	6.6	60.9	9.3	58.2	9.3	58.3	11.5	56.0	32.4	20.8	60.1	11.4	65.7	1.6	67.7	11.6	47.7
	Falcon-Instruct (40B)	12.4	34.8	15.2	32.0	15.2	32.0	16.7	30.5	52.7	39.4	27.6	9.8	51.0	10.9	60.0	13.2	15.3
Torso	GPT-4	**25.6**	27.5	**30.9**	22.3	**30.8**	22.3	**33.4**	19.7	46.9	**55.0**	16.9	**24.3**	**21.8**	10.5	6.8	**36.5**	24.1
	ChatGPT	16.6	20.0	19.0	17.5	19.0	17.5	21.9	**14.6**	63.5	46.4	16.6	22.5	29.2	2.3	**1.7**	19.7	13.3
	Llama 2 (70B)	11.4	34.8	13.7	32.5	13.6	32.5	13.2	33.0	53.8	33.9	29.8	12.9	54.4	**11.1**	38.2	6.9	25.4
	LLaMA (7B)	5.7	87.0	9.7	83.0	9.6	83.1	11.7	81.0	7.3	21.2	72.3	9.5	80.4	2.6	93.7	12.2	80.0
	LLaMA (13B)	8.5	90.1	12.6	86.0	12.5	86.1	14.8	83.8	1.4	28.8	70.5	14.2	85.3	6.3	92.2	12.9	85.2
	LLaMA (33B)	12.7	85.7	17.5	80.9	17.5	81.0	19.8	78.7	1.5	36.0	63.8	19.2	80.5	7.9	88.4	18.4	79.9
	LLaMA (65B)	11.4	88.2	16.5	83.2	16.4	83.3	18.7	81.0	0.3	32.5	66.9	20.0	79.1	9.4	90.6	16.7	83.2
	Vicuna (7B)	5.4	84.0	8.7	80.7	8.6	80.8	9.6	79.8	10.6	22.9	64.6	8.7	82.1	2.6	91.6	7.7	80.3
	Vicuna (13B)	5.4	66.2	8.1	63.5	8.1	63.6	8.8	62.8	28.4	20.7	57.3	4.7	66.7	4.0	75.2	7.7	59.4
	Flan-T5 (3B)	0.7	**17.9**	1.3	**17.3**	1.3	**17.3**	1.5	17.1	81.4	1.2	**14.2**	0.5	51.4	0.1	7.1	2.5	10.0
	Flan-T5 (11B)	2.0	21.1	3.3	19.8	3.3	19.8	3.2	19.9	76.9	4.0	24.7	1.4	53.7	0.8	19.6	4.3	**7.3**
	RWKV (7B)	2.0	26.9	3.4	25.5	3.3	25.5	3.8	25.0	71.1	2.3	24.4	4.1	33.8	0.1	7.6	5.4	28.0
	Falcon (7B)	5.9	61.3	8.3	58.8	8.3	58.9	9.2	57.9	32.9	20.5	54.6	6.5	74.0	6.2	83.4	7.3	45.6
	Falcon (40B)	8.4	42.8	10.1	41.1	10.1	41.1	11.2	40.0	48.9	28.0	34.6	5.8	52.1	9.3	62.7	7.9	30.5
	Falcon-Instruct (7B)	2.8	60.0	5.0	57.8	5.0	57.9	5.6	57.2	37.2	11.2	67.6	2.8	67.8	1.4	67.8	5.9	46.8
	Falcon-Instruct (40B)	8.7	33.9	11.0	31.6	10.9	31.7	11.5	31.1	57.4	31.7	31.2	7.1	50.4	9.4	61.5	6.7	14.9
Tail	GPT-4	**13.4**	21.6	**17.1**	17.8	**17.1**	17.9	**19.0**	15.9	65.1	**10.9**	14.7	**16.9**	**12.0**	3.9	3.7	**27.3**	21.6
	ChatGPT	5.9	**14.0**	7.7	**12.2**	7.7	**12.2**	9.5	**10.5**	80.1	6.4	11.8	8.0	19.2	0.8	**0.9**	13.8	10.2
	Llama 2 (70B)	4.3	40.4	6.7	38.0	6.6	38.0	6.1	38.6	55.4	10.7	44.9	3.1	56.9	**5.3**	49.7	5.7	26.8
	LLaMA (7B)	1.5	88.7	4.5	85.7	4.4	85.8	5.4	84.8	9.8	3.2	80.4	2.0	82.5	1.1	95.8	8.7	83.4
	LLaMA (13B)	1.9	96.3	5.0	93.1	5.0	93.2	6.3	91.9	1.8	4.8	92.1	2.4	96.5	1.9	96.6	9.5	88.7
	LLaMA (33B)	4.1	93.3	7.8	89.5	7.8	89.6	8.8	88.6	2.6	7.3	89.4	4.1	92.5	2.6	95.1	12.8	84.9
	LLaMA (65B)	3.5	96.2	7.4	92.2	7.4	92.2	8.7	90.9	0.4	5.4	94.4	5.9	93.2	3.3	96.6	12.5	87.1
	Vicuna (7B)	1.4	87.9	3.9	85.4	3.9	85.4	4.3	85.0	10.7	5.1	85.6	1.5	86.7	1.5	90.4	5.9	82.4
	Vicuna (13B)	1.6	73.1	3.9	70.8	3.9	70.8	4.7	70.0	25.3	5.7	72.6	1.7	77.2	1.6	81.8	6.4	62.8
	Flan-T5 (3B)	0.6	16.2	1.1	15.7	1.1	15.7	1.3	15.5	83.2	1.1	**8.0**	0.2	53.0	0.4	6.0	2.1	8.8
	Flan-T5 (11B)	1.2	17.3	2.3	16.2	2.3	16.2	2.0	16.5	81.5	2.8	9.6	0.6	51.2	0.6	18.5	2.6	**6.8**
	RWKV (7B)	0.5	22.1	1.6	21.1	1.6	21.1	1.8	20.9	77.4	0.3	16.4	0.4	16.5	0.0	10.3	3.3	27.2
	Falcon (7B)	2.1	64.6	4.2	62.6	4.2	62.6	4.8	62.0	33.2	7.6	71.2	1.4	75.2	1.9	89.6	5.8	45.7
	Falcon (40B)	2.7	48.8	4.3	47.2	4.3	47.2	4.9	46.6	48.5	7.6	54.2	1.3	55.5	3.7	71.3	5.6	33.2
	Falcon-Instruct (7B)	1.5	58.6	2.9	57.2	2.9	57.2	3.4	56.7	39.9	5.1	65.7	0.9	67.3	1.0	67.1	4.3	46.8
	Falcon-Instruct (40B)	2.3	36.8	4.3	34.9	4.2	34.9	4.3	34.8	60.9	7.0	44.6	1.0	51.3	3.9	67.7	4.6	15.5
Head-to-Tail	GPT-4	**23.3**	27.2	**28.4**	22.2	**28.3**	22.2	**30.9**	19.7	49.4	**41.7**	15.5	**21.3**	**19.4**	**10.0**	6.8	**37.1**	25.3
	ChatGPT	14.7	19.6	17.4	**16.9**	17.4	**16.9**	20.3	**14.1**	65.6	34.7	**13.3**	16.9	24.9	3.0	**1.9**	22.1	14.8
	Llama 2 (70B)	9.8	35.9	12.3	33.5	12.3	33.5	11.8	34.0	54.2	27.9	34.3	10.3	54.5	9.8	41.0	7.5	24.8
	LLaMA (7B)	5.9	86.2	9.9	82.2	9.8	82.3	12.1	80.0	7.9	17.2	74.1	11.0	79.2	2.9	93.4	13.6	78.0
	LLaMA (13B)	7.7	91.0	11.9	86.8	11.8	86.8	14.4	84.3	1.3	23.3	75.3	12.2	87.2	6.0	92.9	14.7	83.7
	LLaMA (33B)	11.2	87.0	15.9	82.3	15.8	82.4	18.2	80.0	1.8	28.7	70.1	15.8	82.9	7.1	90.3	19.0	79.1
	LLaMA (65B)	9.9	89.8	15.2	84.5	15.1	84.5	17.8	81.9	0.3	25.0	74.5	16.3	83.1	9.7	90.3	18.4	81.3
	Vicuna (7B)	5.4	83.8	8.8	80.5	8.7	80.5	10.1	79.2	10.8	19.4	70.0	9.6	81.5	2.7	91.2	9.5	77.6
	Vicuna (13B)	5.2	66.5	8.0	63.8	7.9	63.8	9.2	62.6	28.2	18.8	60.7	5.7	69.3	3.7	77.1	8.9	56.4
	Flan-T5 (3B)	1.3	**18.4**	1.9	17.8	1.9	17.8	2.3	17.4	80.3	1.5	14.0	1.5	52.2	0.2	7.0	3.4	10.5
	Flan-T5 (11B)	3.0	21.3	4.3	20.0	4.3	20.0	4.2	20.0	75.7	5.8	21.5	3.3	54.8	0.6	19.7	5.2	**8.3**
	RWKV (7B)	2.9	28.1	4.4	26.6	4.4	26.6	5.1	25.9	69.0	4.1	22.5	6.6	33.4	0.1	10.4	6.5	29.6
	Falcon (7B)	6.0	61.4	8.5	58.9	8.4	59.0	9.5	57.9	32.6	18.7	59.1	7.6	72.5	5.4	84.5	8.3	44.1
	Falcon (40B)	7.7	44.0	9.7	42.1	9.7	42.1	10.8	41.0	48.2	23.9	39.7	5.8	53.5	8.1	65.3	8.8	29.5
	Falcon-Instruct (7B)	3.6	59.8	5.7	57.8	5.7	57.8	6.8	56.7	36.5	12.4	64.5	5.0	66.9	1.4	67.5	7.3	47.1
	Falcon-Instruct (40B)	7.8	35.2	10.2	32.8	10.1	32.9	10.8	32.2	57.0	26.0	34.5	6.0	50.9	8.0	63.1	8.2	15.3

Table 14: Comparison of LLMs' factuality about head, torso, and tail entities using all instances and instances of each domain from Head-to-Tail. All numbers are in percentage (%).