In Search of the Long-Tail: Systematic Generation of Long-Tail Inferential Knowledge via Logical Rule Guided Search

Huihan Li[1] **Yuting Ning**[2] **Zeyi Liao**[2] **Siyuan Wang**[3] **Xiang Lorraine Li**[4,5]
Ximing Lu[4,6] **Wenting Zhao**[7] **Faeze Brahman**[4] **Yejin Choi**[4,6] **Xiang Ren**[1,4]

[1]University of Southern California [2] Ohio State University [3]Fudan University
[4]Allen Institute for Artificial Intelligence [5] University of Pittsburgh
[6] University of Washington [7] Cornell University

{huihanl,xiangren}@usc.edu

Abstract

To effectively use large language models (LLMs) for real-world queries, it is imperative that they generalize to the *long-tail distribution*, *i.e.*, rare examples where models exhibit low confidence. In this work, we take the first step towards evaluating LLMs in the long-tail distribution of inferential knowledge. We exemplify long-tail evaluation on the Natural Language Inference task. First, we introduce **L**ogic-**In**duced-**K**nowledge-Search (**LINK**), a systematic long-tail data generation framework, to obtain factually-correct yet long-tail inferential statements. LINK uses variable-wise prompting grounded on symbolic rules to seek low-confidence statements while ensuring factual correctness. We then use LINK to curate **L**ogic-**In**duced-Long-**T**ail (**LINT**), a large-scale long-tail inferential knowledge dataset that contains 108K statements spanning four domains. We evaluate popular LLMs on LINT; we find that state-of-the-art LLMs show significant performance drop (21% relative drop for GPT4) on long-tail data as compared to on head distribution data, and smaller models show even more generalization weakness. These results further underscore the necessity of long-tail evaluation in developing generalizable LLMs.[1]

1 Introduction

Generalization, especially to unfamiliar and novel situations, is a cornerstone for the usability of large language models (LLMs) in addressing varied real-world inquiries. This imminent demand necessitates evaluation of LLMs in the *long-tail* distribution (the space consisting of unfamiliar examples on which the model has low confidence). Previous works, mostly focusing on model memorization issue, define long-tail knowledge using the frequency of entities in a knowledge base (Cao et al., 2020), in the pre-training dataset (Kandpal et al., 2023), or in Wikipedia search (Mallen et al., 2022). Godbole and Jia (2022) introduces a general definition for long-tail statements, where long-tail examples are assigned *lower likelihood by a pre-trained language model.* We follow this definition which applies to various data format and language task – for any set of statements with similar length and format, those in the long-tail distribution cannot be generated or are generated with low confidence by the models, compared to those in the head distribution.

Recent works have noticed that LLMs have a marked decline in performance when facing inputs from the long-tail (McCoy et al., 2023; Razeghi et al., 2022). Hallucination, for example, is found to be correlated with data being in the long-tail distribution (Li et al., 2024; Yu et al., 2024). LLMs' ineffective utilization of long-tail knowledge impacts its reasoning capabilities and raises reliability concerns in downstream tasks (Huang et al., 2023).

Evaluation in the long-tail distribution requires systematic generation of long-tail data. However, obtaining examples in the long-tail is non-trivial. With state-of-the-art LLMs being trained on vast volume of data on the internet (OpenAI, 2023; Touvron et al., 2023b), it is increasingly difficult to find unseen examples that can effectively test model generalization to its low-confidence end. Crowdsourcing long-tail data is also difficult because of human cognitive bias (Tversky and Kahneman, 1973, 1974), and LLMs' generation in the long-tail distribution is hindered by their pretraining task of "most likely" next token (McCoy et al., 2023).

While demonstrating long-tail evaluation across all applications and domains is not feasible within the scope of one paper, this work focuses on inferential knowledge statement in the form of Natural Language Inference (NLI) task (Bowman et al., 2015; Zellers et al., 2019): NLI requires extensive knowledge and complex reasoning about entities and events, and is one of such tasks on which LLMs

[1]https://github.com/INK-USC/LINK

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2348–2370
November 12-16, 2024

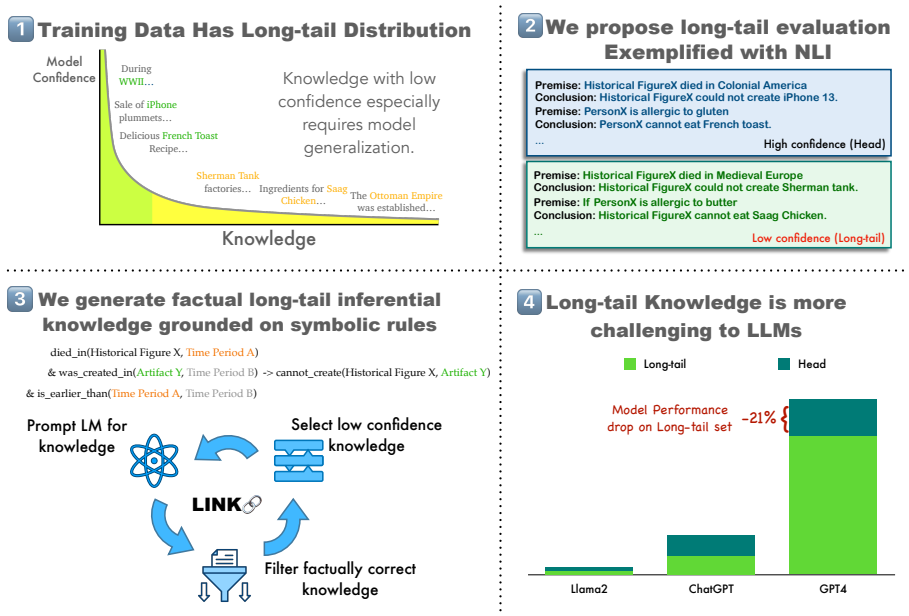


Figure 1: Overview of our motivation, long-tail data generation framework, and model evaluation.

have impressive performance (Achiam et al., 2023; Touvron et al., 2023b; Jiang et al., 2023). Following Sap et al. (2019), we structure inferential knowledge as *if-then* relations with variables, written in a *premise, conclusion* format (Table 1).

First, we make long-tail inferential knowledge generation possible. We propose a novel and lightweight long-tail inferential knowledge generation framework, **L**ogic-**In**duced-**K**nowledge-Search (**LINK**🔗) (§ 2), a variable-wise prompting framework grounded on symbolic rules. This framework enables us to obtain long-tail knowledge statements from existing LLMs. Our evaluation shows that by taking simply instructions `ChatGPT(gpt-3.5-turbo)` and `GPT4(gpt-4)` can only produce statements in the head distribution that also have lower factual correctness, but using LINK with the LLMs improves on both distribution conformity and factual correctness (§ 3).

Second, we test LLMs' long-tail generalization capability on data generated by LINK(§ 4). We produce a large-scale dataset, **L**ogic-**In**duced-Long-**T**ail (**LINT**), which contains 108k knowledge statements spanning across 4 different domains(Table 1). In the long-tail test set of LINT, `GPT4`'s capability in identifying incorrect knowledge drop by 21% relative to the test set in the head distribution, and the gap is even larger for other models we tested (`ChatGPT`, `llama2-70b`). At the same time, human performance significantly outperforms LLMs in both distributions, and stays consistent between head and long-tail test set.

Locational	Head	**P:** Organization X has a branch in Great Lakes Region. **C:** Organization X has office in North America.
	Long-tail	**P:** Organization X has a branch in Tarapaca Region. **C:** Organization X has office in South America.
Outcome and Effect	Head	**P:** Person X has Asthma. **C:** Person X should take Inhaled antiinflammatory drugs.
	Long-tail	**P:** Person X has Hepatitis. **C:** Person X should take Sofosbuvir.
Temporal	Head	**P:** Plant X vanished in Paleolithic Era. **C:** Plant X cannot surround Notre Dame de Paris.
	Long-tail	**P:** Plant X vanished in Classical Greece. **C:** Plant X cannot surround Belém Tower.
Natural Properties	Head	**P:** Bag X has trouble containing Clarinet. **C:** Upright Piano cannot fit in Bag X.
	Long-tail	**P:** Bag X has trouble containing Pandeiro. **C:** Dhak cannot fit in Bag X.

Table 1: Examples of inferential knowledge in each domain of LINT, in a *premise* (**P**), *conclusion* (**C**) format.

Our work is the first to propose a systematic framework that generates data in the long-tail distribution. Using NLI as an example, we show that generating data in the long-tail distribution is an effective way for curating evaluation examples for LLM generalization. Our work serves as a starting point for the series of research on long-tail data discovery and generation, and motivates the community to incorporate long-tail evaluation into model building pipelines.

2 Logic-Induced-Knowledge-Search (LINK🔗)

In this section, we first explain the advantages of generating inferential knowledge grounded on symbolic rules, then illustrate our process of curating symbolic rules, and lastly explain *knowledge beam*

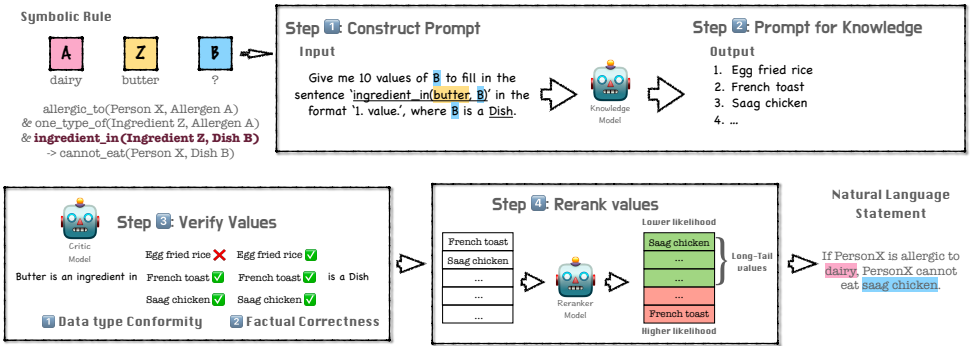


Figure 2: Overview of knowledge beam search (§ 2.3). We demonstrate searching B conditioned on the values of A and Z from previous steps. We only verbalize the predicates containing *Person X* in the final statement as all other predicates contain knowledge that the model should have.

search, our novel variable-wise search pipeline.

2.1 Advantages of Generating Inferential Knowledge from Symbolic Rules

Due to the fact that LLMs are pretrained with the task of generating the "most likely" next token, it is fundamentally challenging for them to directly generate long-tail data through prompting that are factually correct and have low likelihood. Using symbolic rules to guide the generation of knowledge statements have three benefits: (1) Symbolic rules are *designed to be correct*, so we alleviate the pressure of ensuring the deductive plausibility of the statement throughout the entire generation process. (2) The generation process can be broken down into multiple steps, each of which is conditioned on only one variable. Generating for one variable at a time will be much easier for the model, and it is easier to *manipulate the distribution* of individual values than the entire sentence. (3) From one symbolic rule, one can get abundant combinations of variable values as long as they satisfy each predicate in the rule, making the generation process *scalable*.

2.2 Curating Symbolic Rules

A symbolic rule consists of a premise and a conclusion. The conclusion is a single predicate, while the premise contains a set of predicates connected by & operators. Each predicate is a triple of a verb phrase, a subject and an object, and each variable in the symbolic rule has a designated data type.

While there are infinitely many ways to construct symbols rules, we create ours using the principles of **Compatibility and Mutual Exclusivity.** Compatibility refers to a scenario in which one or more events enables another event to occur. We construct the premise and conclusion such that if all predicates in the premise are true, then the predicate in the conclusion *can* occur. Mutual Exclusivity refers to a scenario in which two or more events or conditions cannot occur simultaneously. We construct the premise and conclusion such that if all predicates in the premise are true, then the predicate in the conclusion *will not* occur. To construct such conditions, we add constraints such as time, location, or outcomes to variables in the symbolic rules to make their interaction possible/desirable or impossible/undesirable. In other words, the conclusion describes an event (certain interaction between the variables) and the premise depicts some combination of conditions.

For example, a symbolic rule constrained by compatibility looks like:

exists(Person X, Location A) & lives(Plant Y, Climate B)
& has_climate(Location A, Climate B)
→ can_plant(Person X, Plant Y)

And a symbolic rule constrained by mutual exclusivity looks like:

exists(Person X, Time Period A) & exists(Plant Y, Time Period B)
& is_much_later_than(Time Period A, Time Period B)
→ cannot_plant(Person X, Plant Y)

Here are some additional properties of our symbolic rules:

No tautologies. For example, *allergic_to(Person X, Allergen A) → reacts_badly_to(Person X, Allergen A)* is not a valid symbolic rule. This is to ensure that the symbolic rule contains some reasoning.

The symbolic rule should contain at least 3 variables. This is to ensure some degree of complexity in the symbolic rule.

The symbolic rule should not contain predicates out of scope of LLMs' knowledge.

has_height(Tree X, Height Y) is not a valid predicate, because it is unlikely that LLMs have knowledge about the exact height of one tree. This is to avoid hallucination.

We create symbolic rules that span across four domains (of constraint type): temporal, locational, outcome and effect, and natural properties, totaling 149 person-related rules and 268 object-related rules. More about symbolic rules in Appendix B.

2.3 Knowledge Beam Search

Defining search order. Since all variables are linearly chained, we can search them one by one without repetition. We always start with the subject of the sentence – the person or the object, represented as *Datatype X*. In the rule in Table 2, for example, we start with *Person X* in the premise and find a chain of variables that connects it to the object in the conclusion: *X, A, Z, B*.

For some rules that call for factual knowledge with only one correct answer, such as age, height, year, etc., we empirically find that it increases the knowledge quality to start from the subject in the conclusion and end with the object in the premise.

Constructing Prompt. For each variable, we construct a prompt using all predicates that contain that variable and other previously searched variables. For example, to search variable *B* in the rule in Table 2, we include predicate *ingredient_in(Ingredient Z, Dish B)*. We assume *Z=butter* and construct the prompt as follows:

Give me 50 values of B to fill in the sentence "ingredient_in(butter, B)" in the format "1. value.", where B is a Dish.

Prompting for knowledge. For each partially searched beam, we obtain 200 values of the current variable from the knowledge model [2]. We call OpenAI API 4 times, generating 50 values each time (temperature=0.7 [3]). After each call, we verify the values using a critic model (see paragraph below). To prevent duplicates, we explicitly instruct the model not to generate verified correct values and set `logit_bias=-100` for incorrect values. We implement an early stop mechanism: if for two consecutive calls we do not get any correct values, we terminate the search for the beam.

[2]We use `text-davinci-003` but one can use any model.

[3]We keep `top_p=1` for maximum diversity, and `top_k` is unchangable. Ablation on temperature in Appendix D.1.

Verifying values with a critic. We use `huggingface` default implementation of `Flan-T5-XXL` (Chung et al., 2022), an instruction-tuned model that can be used zero-shot, as the critic that checks data type conformity and factual correctness of the values. We ask the model to output *yes/no* on the correctness of a given statement. For data type conformity, the statement is "*{value} is a {data type}*." For factual correctness, we convert the symbolic predicate into a natural language statement. We obtain the *yes* token probability and dynamically adjust the threshold for accepting values for different predicates. More about the critic model in Appendix C.

Pushing values to long-tail distribution with reranking. At each search step, we convert symbolic predicates into natural language statements (*ingredient_in(butter, saag chicken)* → *Butter is an ingredient in saag chicken*) and concatenate them with "and". We obtain the sentence likelihood using `huggingface` default implementation of `llama-7B` (Touvron et al., 2023a)[4] and rerank the sentences from the *lowest* likelihood to the *highest* likelihood. We take top the 75% values unless there are more than 200 values, in which case we take the top 200 values. Then we move on to the next variable. To control data distribution during evaluation, we also generate statements in the head distribution, by ranking the sentences from the *highest* to the *lowest* likelihood.

From 149 person-related rules and 268 object-related rules across four domains, we curate our dataset **L**ogic-**In**duced-Long-**T**ail(**LINT**) that consists of 54K long-tail knowledge statements. We also release 54K head distribution statements that are also searched with the LINK framework. Domain-wise statistics of LINT in Appendix H.

3 Generating Long-tail Inferential Knowledge with LINK

In this section, we compare LINK's ability to generate long-tail inferential knowledge with instruction-only LLMs, `ChatGPT` and `GPT4`, who do not use knowledge beam search.

3.1 Instruction-Only Knowledge Generation

We generate knowledge statements from a subset of 200 symbolic rules from LINT using `ChatGPT` and `GPT4` by only providing it with an instruction. We

[4]`llama2` was not released at the time of experiments.

Symbolic Rule	is_allergic_to(Person X, Food allergen A)& is_ingredient_in(Ingredient Z, Name of a dish or food B) & is_one_type_of(Ingredient Z, Food allergen A) → is_not_able_to_eat(Person P, Name of a dish or food B)
Prompt	In the following sentence, A is a Food allergen, B is a Name of a dish or food, Z is a Ingredient. Find values of A, B, Z to fill in the blank in the sentence 'If Person X is allergic to [A] and [Z] is a ingredient in [B] and [Z] is one type of [A], then Person X is not able to eat [B].' and make it a grammatical and correct sentence. Give me 50 values in the format '1. A=, B=, Z='.

Table 2: An illustration of prompts for zero-shot LLMs, containing a symbolic rule in natural language and its variables with data type specified.

prompt the LLMs with a natural language version of the symbolic rule with data types of the variables specified, and ask them to populate the rule with all variables simultaneously(Table 2). Generations from this prompt serve as the *head distribution baseline*, to which we compare model generations when they are instructed to generate in the long-tail distribution.

To instruct models to generate long-tail knowledge from symbolic rules, we append *"Use less frequent terms of A and B and C"* in the prompt.[5]

For each rule, we obtain 200 statements using default instruction and 200 statements using long-tail instruction, from `ChatGPT` and `GPT4` respectively. We present our findings below.

3.2 LINK Generations Consistently Fall in Long-tail Distribution

Following Godbole and Jia (2022)'s definition of long-tail statements, we use the most capable LLM that was producing log likelihood at the time of experiments (`text-davinci-003`) to assign likelihood to generated data. We compare how different models assign distributions to the same statements, in order to ensure that the distribution stays consistent among all models despite the absolute log likelihood difference (Appendix E.2).

We calculate the log likelihood over `InstructGPT` of all statements generated by LINK, compared with instruction-only `ChatGPT` and `GPT4`. We calculate $\delta = mean(D(H)) - mean(D(L))$ for each set of statements generated from each symbolic rule, where $D(\cdot)$ means the log likelihood distribution of the probability model, H is the set of statements as head distribution baseline, and L is the set of the statements intended to be in the long-tail distribution. For a model to successfully generate long-tail knowledge statements, the "long-tail" set of sentences should be assigned distinguishably lower probabilities than the sentence in the head distribution.

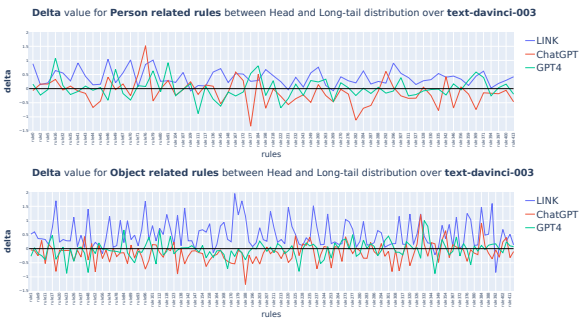


Figure 3: LINK generations have higher δ values for most rules, while δ values of `ChatGPT` and `GPT4` mostly locate around 0.

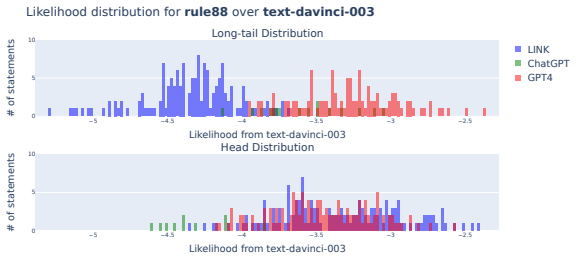


Figure 4: Only LINK generations fall in the correct distributions on the log likelihood scale of `InstructGPT`.

Figure 3 illustrates that while `ChatGPT` and `GPT4` are not able to generate long-tail statements merely from prompting, LINK is able to generate long-tail statements with much lower likelihood. Each grid on the x-axis represents a unique symbolic rule, and each grid on the y-axis represents $\delta \in [-1.5, 2]$. A δ close to 0 means that the intended long-tail distribution generations have the same log likelihoods as the statements in the head distribution, being larger than 0.3 empirically means a decent drop in likelihood, and being negative means the intended long-tail distribution data have even high likelihood than the head distribution data.

Averaged across 200 sampled rules, LINK has a positive δ of 0.48, while `ChatGPT` and `GPT4` each has a delta of -0.14 and -0.02. The δ values for LINK (blue line) float above 0 for most of the rules, some even being above 0.5. On the other hand, δ values for `ChatGPT` (red line) and `GPT4` (green line) mostly locate around 0, with many being negative.

To better illustrate the distribution of statements

[5]We tried 10 prompts as shown in Appendix F and they have similar effect on model behavior.

Accuracy	ChatGPT	GPT4	LINK
Data Type	85.40	91.80	**94.23**
Factuality	67.50	84.82	**88.71**
Overall	56.44	78.23	**83.95**

Table 3: LINK has both the highest factual and data type accuracy in human evaluation.

generated by LINK, ChatGPT and GPT4, we plot the log likelihood of the generated statements for one symbolic rule from the three methods Figure 4. To eliminate noise from incorrect statements on the distribution, we only plot the statements that are marked as correct in human evaluation (explained in § 3.3). The likelihood distribution of more rules can be found in Figure 9.

The long-tail statements from LINK clearly fall in a much lower probability distribution than GPT4's "long-tail" generations. Moreover, GPT4's generation in the "long-tail distribution" in fact falls in the same probability distribution as its head distribution generations.

3.3 LINK Achieves Higher Data Correctness than Instruction-Only LLMs

In addition to distribution correctness, we also evaluate data type conformity and factual correctness of LLMs' long-tail knowledge generations using crowdworkers from Amazon Mechanic Turk (AMT). For data type conformity, we ask an AMT worker *Is {variable} a {data type}?* for each variable in the symbolic rule. For factual correctness, we ask an AMT worker *Does the premise entail the conclusion?* We sample 4,000 statements from LINT for human evaluation, of which 2,025 are from head distribution and 1,975 are from long-tail distribution. We take 3 annotations for each statement and take the majority vote. Annotator agreement can be found in Appendix I.3. The AMT template can be found in Appendix I.2.

Table 3 shows that instruction-only ChatGPT and GPT4 underperform LINK in both data type conformity and factual correctness. Without LINK, both models struggle more with factual correctness, a foreseeable behavior in the low likelihood realm. For domain wise performance see Table 17. For examples of failure cases, see Table 20.

3.4 Ablation Studies

Our results above show that LINK is better than instruct-only generations with ChatGPT and GPT4, in terms of both "long-tailedness" and factual correctness. In this section, we perform ablation studies on the role of reranker, critic, and knowledge models in LINK.

Ablation on reranker. Our variable-wise reranker is essential for pushing LINK generations into the long-tail distribution. Figure 6 presents the distribution comparison of generated statements by LINK and the variant without the reranker. Without the reranker, the statements for the long-tail distribution are pulled towards the head distribution, making the two completely inseparable.

Post-hoc reranking over LLM generations does not have the same effect as variable-wise reranking in LINK. Figure 7 illustrates the distribution of generated statements by LINK, compared to instruction-only GPT4 and instruction-only GPT4 reranked by InstructGPT. Post-hoc reranking barely changes the distribution of generations, even when using the same model as the evaluation.

Even though log likelihood is both used by reranker and evaluation, they are taken from different models and using different inputs. The statements we use for ranking the knowledge beams are shorter than the final statement, as they only consist of partial predicates. Despite these differences, variable-wise reranking that uses a smaller model achieves the separation that post-hoc filtering with the evaluation model cannot achieve.

Our findings highlight the importance of performing variable-wise reranking in LINK.

Ablation on the critic model. Critic models are essential for guaranteeing the generation quality, especially in the long-tail distribution. Table 8 in Appendix D.2 shows that removing the critic and removing both reranker and critic leads to significant drops of data type conformity and factual correctness of generations in the long-tail distribution. Note that LINK w/o reranker + critic has higher generation quality than LINK w/o critic. This is because without a reranker the model is only able to generate statements in the head distribution, making it easier to be factually correct than in the long-tail distribution. This observation further suggests that critic models are essential for generation qualities in the long-tail distribution.

Ablation on Knowledge Model. Our analysis above has shown that by simply providing an instruction to ChatGPT and GPT4, we cannot effectively generate knowledge statements that are both high quality and in the long-tail distribution. By

adding LINK, we can improve both distribution correctness and generation quality, as shown in Figure 8 and Table 10.

Interestingly, we find that the generations from LINK + GPT4 do not show a big improvement over LINK + InstructGPT, both on distribution correctness and generation quality. This suggests that the improvement a stronger knowledge model brings to LINK is marginal compared to that of the reranker and critic model. This finding highlights the effectiveness of LINK in facilitating long-tail generation regardless of the knowledge model.

4 LLMs' (Lack of) Generalization in the Long-tail Distribution

Using data from LINT, we evaluate LLM generalization through an *entailment classification* task on inferential knowledge in the long-tail distribution.

We use all human evaluated knowledge statements except for those with incorrect data types in LINT, with 1,925 statements in head distribution and 1,856 statements in long-tail distribution. Statements rated as factually correct has entailment between premise and conclusion, and statements rated as factually incorrect has contradiction between premise and conclusion.

In order to prevent LLMs' template biases from misleading the evaluation, we convert each statement into 13 question templates, where each question templates corresponds to a positive label (*"Yes", "True", or "Right"*) or a negative label (*"No", "False", or "Wrong"*). The question templates are summarized in Appendix G.1. We consider a model answering accurately about one statement *only if* the model answers *all* question templates correctly.

We evaluate three LLMs: `llama2-70B`, `ChatGPT` and `GPT4`. In order to enforce the model to predict the target token sets and minimize format noncompliance, we use Chain-of-Thought (CoT) (Wei et al., 2022) prompting that includes 2 in-context examples with randomly shuffled orders of positive label and negative label.

For each domain, we report aggregated (All) performance of each model as well as human baseline performance in Table 4. We also include performance on positive labels only and negative labels only. We also mark relative performance drop $\delta = \frac{t-h}{h}$, where h and t are head and tail distribution aggregated performance.

We obtain human performance on the same set of statements. We recruit 17 AMT workers who do not participate in the evaluation task (and thus have not seen the task data). The workers see the knowledge statements in *premise, conclusion* format and are asked to select "*yes/no*" to whether the premise entails the conclusion. The workers are asked to use search engines to verify their answers. See AMT templates in Appendix I.2.

We make the following observations on LLM generalization in long-tail NLI.

Performance drops in the long-tail distribution. All models exhibit a large relative drop in performance in the long-tail distribution. The most competitive model, GPT4, has a 21% overall drop from head to long-tail distribution, while other models exhibit a even larger drop.

Human Performance does not drop for long-tail distribution. Performance drop in the long-tail distribution does not occur to humans for 3 out of 4 domains. It is expected because humans can verify their knowledge using search engines, so infrequent knowledge does not challenge humans as much as models (discussion in Appendix A). The exception with the locational domain may be due to some relations being less available online(eg. *banned_in(Food, Country)*).

Brittleness towards question templates. The huge gap between model and human baseline performance indicate that LLMs cannot reliably reason on the same statement when question templates change. We find that model performance between positive and negative labels can be very different for certain domains, indicating that models are miscalibrated for positive and negative answer tokens. Although this phenomenon is not entirely due to the shift of distribution of the knowledge statements, it is caused by model's unfamiliarity of certain question templates. For example, we find that template 5 and template 12 are the same question (*"Premise: ... Conclusion: ... Does the Premise entail the Conclusion?"*) with opposite answers ("Yes" and "No"), but all models' performance on template 5 is significantly lower than that on template 12. This suggests that models are more likely to create false negatives in such context, another evidence of performance drop due to long-tail distribution.

In summary, our analysis shows that while human's inferential reasoning is not affected by their familiarity to the data (provided that they know the entities involved), model's inferential reasoning

Domain	Distribution	Llama2-70B			ChatGPT			GPT4			Human Baseline
		Pos	Neg	All	Pos	Neg	All	Pos	Neg	All	All
Natural Properties	Head	2.78	6.72	0.44	1.45	5.56	0.0	13.68	43.83	9.23	82.31
	Long-tail	2.64	3.49	0.0	1.08	5.29	0.0	10.82	39.9	7.21	82.45
	Δ	-	-	-100%	-	-	-0.0%	-	-	-21.89%	0.17%
Temporal	Head	6.58	2.6	0.46	10.72	38.28	4.13	68.3	43.95	36.91	84.69
	Long-tail	7.43	3.07	0.0	9.05	32.15	2.26	60.58	38.93	28.11	83.20
	Δ	-	-	-100%	-	-	-45.28%	-	-	-23.84%	-1.76%
Outcomes and Effects	Head	7.62	7.93	1.22	19.92	29.27	6.1	57.32	55.18	41.16	83.83
	Long-tail	8.5	9.97	0.59	19.65	26.1	2.64	55.72	46.63	33.43	85.13
	Δ	-	-	-51.64%	-	-	-56.72%	-	-	-18.78%	1.56%
Locational	Head	8.57	7.14	0.0	17.14	10.0	5.71	18.57	18.57	2.86	75.71
	Long-tail	11.24	8.99	3.37	15.73	4.49	1.12	37.08	8.99	3.37	67.42
	Δ	-	-	0.0%	-	-	-80.39%	-	-	17.83%	-10.95%
Total	Head	5.08	5.28	0.56	8.21	20.67	2.62	39.49	44.87	23.64	83.12
	Long-tail	5.69	4.67	0.27	7.76	17.86	1.28	36.58	39.34	18.66	82.44
	Δ	-	-	**-51.79%**	-	-	**-51.15%**	-	-	**-21.07%**	**-0.82%**

Table 4: Performance on the entailment classification task of three LLMs decreases on the long-tail distribution compared to the head distribution, while human performance does not.

ability drops over long-tail knowledge. Our result highlights the importance and effectiveness of long-tail evaluation for model generalization, and our dataset LINT can be used as a useful resource for testing inferential generalization of LLMs.

5 Related Work

Works on model generalization analysis have focused on **generating adversarial examples** for model evaluation (Zhang and Li, 2019; Ziegler et al., 2022; Perez et al., 2022; Casper et al., 2023), flagging abnormal inputs that are likely to trigger bad behavior. Recently, the community has realized the importance of **testing language models' abilities in the long-tail distribution** (Godbole and Jia, 2022). Works reveal that LLM performance is affected by input data probability. (McCoy et al., 2023; Razeghi et al., 2022), and more works have focused on **generating less common data for probing LLMs**. RICA (Zhou et al., 2020) proposes to include novel entities in self-contained commonsense statements to evaluate robust inference capabilities. UnCommonSense (Arnaout et al., 2022) proposes to evaluate models on informative negative knowledge about everyday concepts in addition to positively expressed commonsense knowledge. Razeghi et al. (2022) observe a correlation between the model performance on math problems and the frequency of numeric and temporal terms from those instances in the pretraining data.

In addition to probing models on less common data, recent works also **test LLMs generating less common data**. Chen et al. (2023) propose a negative knowledge generation task where models generate uncommon knowledge with negation conditioned on constrained keywords. Tang et al. (2023) introduce the "less likely brainstorming" task that asks a model to generate outputs that humans think are relevant but less likely to happen.

Generating uncommon data is challenging not only for LLMs, but also for **humans because of our cognitive bias**. Tversky and Kahneman (1974) observe that humans are prone to more systematic errors when facing uncertain events, and Tversky and Kahneman (1973) reveal that humans tend to evaluate the frequency of classes or the probability of events by availability, i.e., by the ease with which relevant instances come to mind. These traits make it difficult for humans to come up with novel associations (Kray et al., 2006), a crucial ability to create data in the long-tail distribution.

6 Conclusion

Using NLI as a case study, we illustrated the significant potential of long-tail data in uncovering the generalization limitations of LLMs. We introduced the first systematic framework designed to generate inferential data within the long-tail distribution, and then demonstrated a noteworthy performance drop of LLMs in the long-tail examples. Our work initiates a new line of research focused on long-tail data discovery and generation, urging the research community to adopt long-tail evaluation in the development of LLMs.

Checklist

Limitation

Limitation on knowledge statement format. Long-tail knowledge statements may come in multiple shapes and forms. Our work focuses only on *premise, conclusion* format, as the first step towards the generation of knowledge statements. The symbolic rules do not have high complexity, due to the limited number of variables and predicates, and being under the constraint for the symbolic rules to be linearly chained. Therefore, the effectiveness of our framework on generating more complex knowledge statements has not been tested.

Limitation on testing with open-source models. Our work did not include open-source models in evaluations of long-tail statement generation and entailment classification task. While `ChatGPT` and `GPT4` are arguably the strongest models, open-source models may exhibit new behaviors in the long-tail realm that are worth exploring.

Limitation on ablating with different critic and reranker model settings. While we performed extensive ablation studies on the critic and reranker models and established their importance in the LINK framework, we did not explore a diverse set of model options as well as hyperparameter settings. Using other models may or may not affect the performance of LINK.

Limitation on sample size. Due to constraint from human annotation resources, we were only able to evaluate models on 200 rules uniformly sampled from the LINT. Although the general trend should remain the same, model performance evaluated on all rules may result in some deltas.

Risk

Generation of harmful values. LINK might be used on mal-intention-ed rules or searching for toxic and harmful values, where researchers may replace our reranker with another model trained to prefer more harmful values.

Environmental tax. Another potential risk is increasing environmental burdens because we extensitvely call OpenAI APIs to large language models during search; however, one can replace the large language models with smaller open source models with less environmental tax.

Factual errors in generations. Because LINK operates in the long-tail realm, its generations are no guaranteed to be correct 100% of the time. If one uses the generations directly without verification, one my introduce false information into their system.

Use and Distribution

All data we collected through LLMs in our work are released publicly for usage and have been duly scrutinized by the authors. Data for all human studies that we conduct are also publicly released with this work, with appropriate annotator anonymizations.

Our framework LINK may only be used for generations that follow the ethics guideline of the community. Using LINK on mal-intention-ed rules or searching for toxic and harmful values is a potential threat, but the authors strongly condemn doing so.

References

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

Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z Pan. 2022. Uncommonsense: Informative negative knowledge about everyday concepts. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 37–46.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Ermei Cao, Difeng Wang, Jiacheng Huang, and Wei Hu. 2020. Open knowledge enrichment for long-tail entities.

Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. 2023. Explore, establish, exploit: Red teaming language models from scratch. *arXiv preprint arXiv:2306.09442*.

Jiangjie Chen, Wei Shi, Ziquan Fu, Sijie Cheng, Lei Li, and Yanghua Xiao. 2023. Say what you mean! large language models speak too positively about negative commonsense knowledge. *arXiv preprint arXiv:2305.05976*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi

Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Ameya Godbole and Robin Jia. 2022. Benchmarking long-tail generalization with likelihood splits. *arXiv preprint arXiv:2210.06799*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv: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*, pages 15696–15707. PMLR.

Laura J Kray, Adam D Galinsky, and Elaine M Wong. 2006. Thinking within the box: The relational processing style elicited by counterfactual mindsets. *Journal of personality and social psychology*, 91(1):33.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. *arXiv preprint arXiv:2401.03205*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. 2023. Embers of autoregression: Understanding large language models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.

Liyan Tang, Yifan Peng, Yanshan Wang, Ying Ding, Greg Durrett, and Justin F Rousseau. 2023. Less likely brainstorming: Using language models to generate alternative hypotheses. *arXiv preprint arXiv:2305.19339*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Amos Tversky and Daniel Kahneman. 1973. Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, 5(2):207–232.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Qifan Yu, Juncheng Li, Longhui Wei, Liang Pang, Wentao Ye, Bosheng Qin, Siliang Tang, Qi Tian, and Yueting Zhuang. 2024. Hallucidoctor: Mitigating hallucinatory toxicity in visual instruction data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12944–12953.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Jiliang Zhang and Chen Li. 2019. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems*, 31(7):2578–2593.

Pei Zhou, Rahul Khanna, Seyeon Lee, Bill Yuchen Lin, Daniel Ho, Jay Pujara, and Xiang Ren. 2020. Rica: Evaluating robust inference capabilities based on commonsense axioms. *arXiv preprint arXiv:2005.00782*.

Daniel Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Benjamin Weinstein-Raun, Daniel de Haas, et al. 2022. Adversarial training for high-stakes reliability. *Advances in Neural Information Processing Systems*, 35:9274–9286.

A More Discussions

Our main experiments on entailment classification task assume that the LLMs have the specific knowledge necessary to answer the questions. While this is a fair implication, given that their training data are known to contain essentially anything available on the web, one may also worry that the model's handling of such knowledge is impacted by the memorization of this knowledge. As knowledge gets more long-tail , it means it appears in the training data less and is thus harder to memorize. Given the imperfect memorization of current LLMs, this may impact their performance on such knowledge, and our current experiments suggest that. To verify this, for future experiments one can count the frequency of this long-tail knowledge in the training data to measure how imperfect memorization makes certain knowledge long-tail .

B Symbolic Rule Creation

Following the criteria mentioned in § 2.2, we curated 417 symbolic rules using the following steps:

1. Select Compatibility (conclusion will be positive) or Mutual Exclusivity (conclusion will be negative).

2. Defining Constraints. Our constraints can be categorized as temporal, locational, natural properties, and desirable/undesirable outcomes and effects. *Temporal* constraints refer to time period or age, *locational* constraints refer to geographic location such as countries or cities as well as climates (tropical or polar, etc), *natural properties* refer to physical properties of objects such as temperature, size, density, speed, and *outcomes and effects* include allergies, cure of disease, etc.

3. Selecting argument types. We either use Person-Object or Object-Object as key arguments.

4. Defining interactions between arguments. Person-Object interactions include "using", "operating", "buying", "consuming", and Object-Object are more type-specific (eg. "scratching" when the constraint is "hardness" in natural property).

5. Expanding data types for Person or Object. We prompt `InstructGPT` to generate more

specific data types under Person (eg. Historical Figure) or Object (eg. Vehicle, Tool).

6. Optimize the verb describing the interaction. We prompt `InstructGPT` to generate a more accurate verb for the expanded data types.

Table 5 shows how an example symbolic rule is constructed.

Constraint	Temporal (Age)
Arguments	Person-Object
Constraint Predicate	is_of_age(Person A, Age X) & requires_a_minimal_age_of(Object B, Age Y) & is_smaller_than(Age X, Age Y)
Principle	Mutual Exclusivity
Interaction	cannot_**operate**(Person A, Object B)
Prompt for Data Type Expansion	In rule "requires_a_minimal_age_of(Object B, Age Y) & cannot_operate(Person A, Object B)", B is a variable representing an object. List 10 subcategories of object that B could be that also make the rule true.
Expanded Data Types	Vehicle, Machinery, Alcohol, Firearm, Tattoo Equipment, Tobacco Product
Prompt for Verb Optimization	cannot_operate(Person A, Object B) is equal to [mask](Person A, Vehicle B). Write the best predicate that could fit in [mask] token.
Expanded Rule Example	is_of_age(Person A, Age X) & requires_a_minimal_operating_age_of(Object B, Age Y) & is_smaller_than(Age X, Age Y) → cannot_drive(Person A, Vehicle B)

Table 5: Illustration of our process for creating an example symbolic rule.

C Critic Model

We find that while the critic model usually verifies data type conformity with high accuracy, it often creates false negatives when verifying factual correctness. Moreover, even within false negatives that result from the same predicate, the correct values get higher `yes` token probabilities than the incorrect values. We hypothesize that while the critic model is less confident about certain knowledge because it is trained on a smaller portion of the knowledge than `text-davinvi-003`, it can still rank the values inherently. Therefore, we extract the probability of the `yes` token instead of taking the argmax. We also implement a dynamic critic threshold that adjusts the threshold for accepting values for different predicates. The algorithm is as follows:

1. We start with a threshold of 0.85.

2. If no correct values are found, we decrease the threshold by 0.05.

3. If some correct values are found, we set the threshold for the predicate to the current threshold and do not decrease it in further calls.

4. If the threshold is set but we find some values with a higher `yes` token probability than the threshold, we increase the threshold by an increment of 0.05 to accommodate the higher probability. Then we retrospectively reject previous accepted values with a lower `yes` token probability than the new threshold.

5. For data type conformity, we set a minimum threshold of 0.65 because we expect the model to be more confident.

In this way, we can find the maximum available threshold for each beam, which guarantees precision while reducing false negatives.

To verify the effectiveness of our critic model, we use crowd workers from AMT to evaluate the data type conformity and factual correctness of predicates. Specifically, for each symbolic predicate that contains two variables (e.g., *exist_during(Location X, Historical Time Period Y)*), we will present a statement in natural language (e.g., *Saigon* existed during *The Cold War.*) with 3 types of questions: (1) clear reference: Q1 and Q2. (2) factual correctness: Q3. (3) data type conformity: Q4 and Q5.

- **Q1:** Does "*Value A*" in the Statement "*Statement*" have a clear reference?

- **Q2:** Does "*Value B*" in the Statement "*Statement*" have a clear reference?

- **Q3:** Is the Statement "*Statement*" factually correct, with very high probability?

- **Q4:** Is the Statement "*Value A* is a *Data Type A*." factually correct, with very high probability?

- **Q5:** Is the Statement "*Value B* is a *Data Type B*." factually correct, with very high probability?

We sample 3 rules from our data and requested human annotators to rate the data type conformity and factual correctness of statements. Table 6 shows the error rate of each question. Only if all the questions are answered with "Yes" do we consider the statement as correct. The overall correctness of statements in head distribution and long-tail distribution are 0.8567 and 0.8467 respectively, which indicates a high quality of statements accepted by our critic model.

	Q1	Q2	Q3	Q4	Q5
Error Rate	0.0004	0	0.0639	0.0011	0

Table 6: The error rate of each question in human verification. Most errors occur on factual correctness.

D Ablation Studies on LINK

D.1 Hyperparameters on Knowledge Model

When constructing LINT, we used `InstructGPT` as the knowledge model with temperature=0.7 and top_p=1. Since top_p=1 maximizes sampling diversity and top_k is hidden from the OpenAI API, we conduct ablation studies on whether temperature affects the result of knowledge search, comparing temperature of 0.5 (low diversity), 0.7 (medium diversity) and 1.0 (high diversity), using a few sampled rules. In this ablation study, we use `gpt-3.5-turbo-instruct` checkpoint as knowledge model and `llama-2-70b` as the approximation of the language distribution.

Temperature	Data Type	Factuality	Overall
0.5	89.05	93.21	82.94
0.7	89.00	92.33	82.25
1.0	88.08	91.75	81.00

Table 7: Different temperatures result in similar data type conformity and factual correctness.

Table 7 shows similar data type conformity and factual correctness among the three ablated temperature, with temperature=1.0 having the lowest accuracy among the three settings.

Figure 5 shows that all three temperature settings can successfully generate knowledge statements in the long-tail distribution, except for when temperature=0.5 in one of the six sampled rules.

This phenomenon reflects that higher temperature helps generating more diverse values and therefore more likely to generate long-tail values, while risking lowering factual salience.

D.2 Effect of Critic on LINK

To investigate the effectiveness of the critic, we provide an ablation study on a few sampled rules by removing the critic in LINK. Table 8 shows the generation quality of LINK and several variants in long-tail distribution. Without the critic, the generation quality decreases significantly. However, the performance drop is less significant in the head distribution Table 9. Besides, if we replace the reranker with a random sampling method, the generated statements cannot lie in the long-tail distribution (which will be further explained in § D.3) and have higher correctness without the critic. It indicates that it is harder for models to generate correct statements from the long-tail distribution than the head distribution without LINK.

	Data Type	Factuality	Overall
LINK	**93.42**	**97.50**	**91.33**
w/o critic	52.58	52.08	33.00
w/o critic+reranker	75.42	73.92	58.25
LINK with `GPT4`	92.75	96.17	89.25
w/o critic	63.00	58.17	40.00
w/o critic+reranker	88.33	83.50	74.50

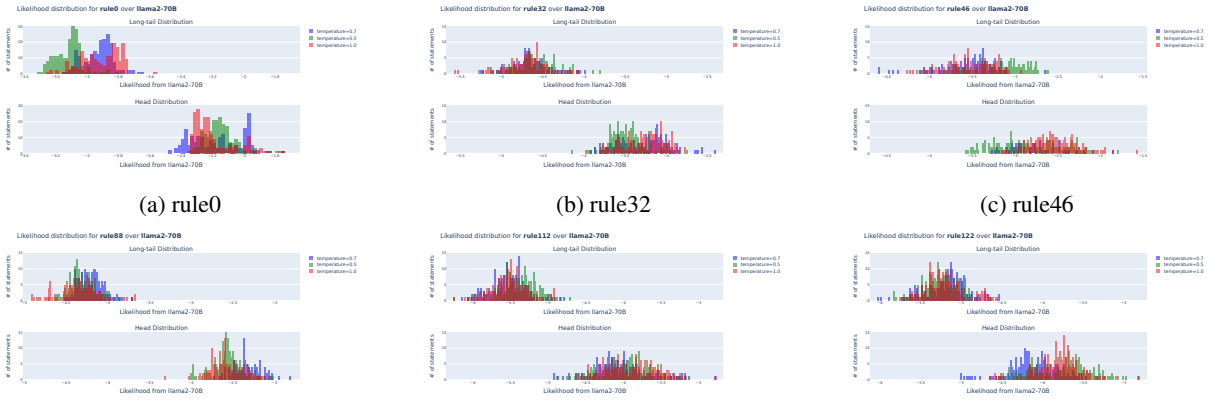
Table 8: Ablation study on the critic model in the long-tail distribution. Removing the critic from LINK will significantly decrease the generation quality. Using a critic is necessary to guarantee the correctness of generated statements, especially in the long-tail distribution.

	Data Type	Factuality	Overall
LINK	**95.17**	**97.75**	**93.00**
w/o critic	80.33	73.50	59.75
w/o critic+reranker	76.66	74.83	59.33
LINK with `GPT4`	92.17	98.08	90.83
w/o critic	91.00	80.42	72.08
w/o critic+reranker	88.17	85.33	75.58

Table 9: Ablation study on the critic model in the head distribution. Removing the critic decreases the data quality, but not as much as in the long-tail distribution. LINK w/o critic+reranker has the same performance between head and long-tail distribution, demonstrating that without a reranker all generations are in the same distribution.

D.3 Effect of Reranker on LINK

To investigate the effectiveness of the reranker, we provide an ablation study on a few sampled rules by replacing the reranker step with a random sampling method. Figure 6 presents the distribution comparison of generated statements by LINK and the



(a) rule0 (b) rule32 (c) rule46

(d) rule88 (e) rule112 (f) rule122

Figure 5: All three temperature settings of LINK can successfully generate knowledge statements in the long-tail distribution, except for when temperature=0.5 in one of the six sampled rules.

variant without the reranker. Without the reranker, the generated statements for both head distribution and long-tail distribution are pulled towards the center of the distribution, making them completely inseparable.

D.4 Ineffectiveness of Post-hoc Reranking for LLM generated knowledge.

To further highlight the importance of performing step-wise reranking in LINK, we confirm that applying a post-hoc reranker on the GPT4 generations from instructions does not have the same effect as LINK. We use InstructGPT to rerank the GPT4 generations from the *lowest* to the *highest* likelihood and take the top 75% results as the long-tail distribution. For the head distribution, we rerank the generations from the *highest* to the *lowest* likelihood and take the top 75% results.

We evaluate on the same set of rules as in § 3.2 as an example. Figure 7 illustrates the distribution of generated statements by LINK, prompt-based GPT4 and prompt-based GPT4 with reranker. We observe that using post-hoc reranker still cannot achieve a separation between the generation of the head distribution and the long-tail distribution, even with the same model as the evaluation. It demonstrates that maneuvering the distribution during the searching process is necessary and more effective than post-hoc filtering.

D.5 Applying GPT4 as the knowledge model

Table 10 shows the generation quality of GPT4 using baseline prompting method, LINK and LINK with GPT4 as the knowledge model over 6 sample rules. Using a stronger model as the knowledge model has marginal effect on the quality of generations compared to LINK. Figure 8 shows that whatever the knowledge model is, the distribution of generations by LINK can correctly fall in the long-tail distribution.

	Data Type	Factuality	Overall
Zero-shot GPT4	85.44	88.42	74.39
LINK	**93.42**	**97.50**	**91.33**
LINK with GPT4	92.75	96.17	89.25

Table 10: Using a stronger model as the knowledge model does not improve generation qualities for LINK, but using LINK with a language model has significant improvement over zero-shot performance.

E Addendum on Distribution

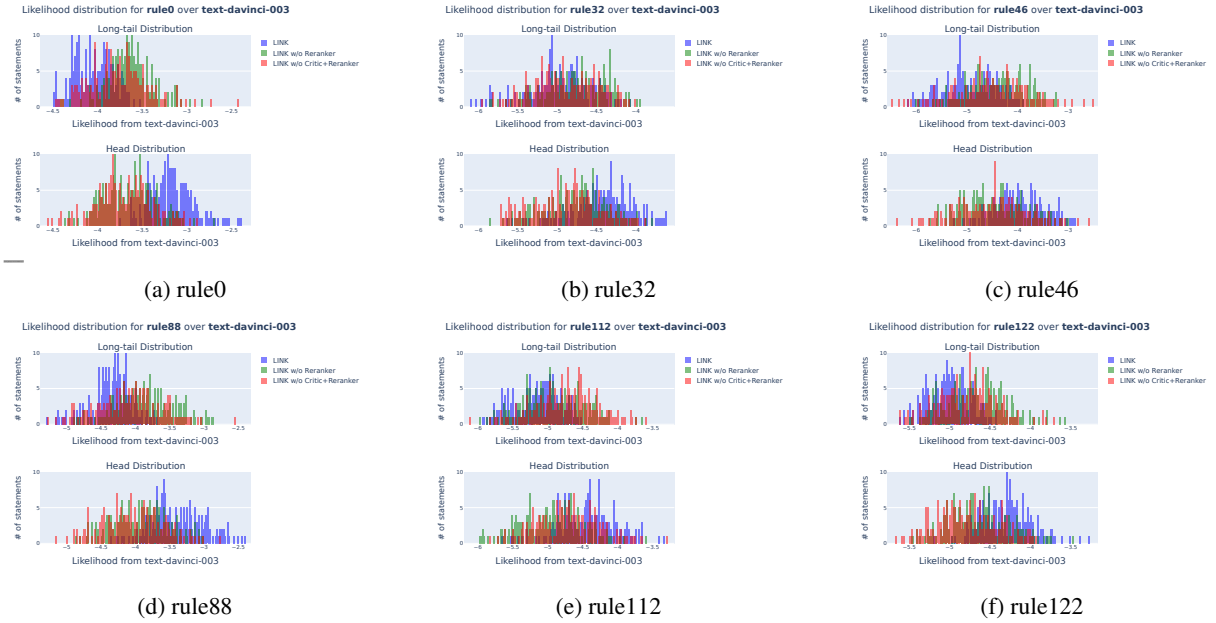
E.1 Additional distribution plots for symbolic rules

As an extension on § 3.2, we show the distribution of statements sampled by LINK, ChatGPT and GPT4 from 6 symbolic rules on InstructGPT in Figure 9.

E.2 Distribution Comparison of Different Models

In this section, we show that the long-tail distribution of different language models overlap, and that this evidence supports our assumption that a universal natural language distribution exists; subsequently, the long-tail distribution of a language model can be used to approximate the long-tail distribution of other language models.

We sample knowledge statements generated by LINK from six rules and calculate their proba-



(a) rule0 (b) rule32 (c) rule46

(d) rule88 (e) rule112 (f) rule122

Figure 6: When we remove reranker from LINK, the distribution of the resulting head and long-tail statements are pulled towards the center. Using reranker is essential for separating the head and long-tail distribution.

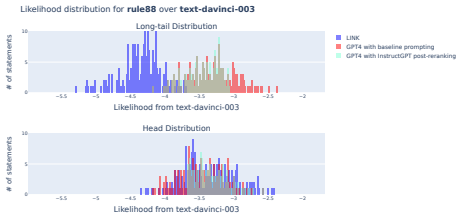


Figure 7: Post-hoc reranking of GPT4 does not help move the distribution towards the long-tail distribution.

bilities with `llama-7b`, `llama2-7b`, `llama-2-70b`, and `InstructGPT`. Figure 10, Figure 11 and Figure 12 respectively show the distribution comparison between `InstructGPT` and the three open-source models over the sampled statements from each rule.

For every rule, we note that if a set of statements falls into the low-probability distribution of `InstructGPT`, it also falls into the low-probability distribution of the open-source model. Therefore, the categorization on long-tail distribution by one language model can effectively approximate the categorization on long-tail distribution by other models; hence, we use `InstructGPT` as the approximation of the written natural language distribution in our distribution evaluation.

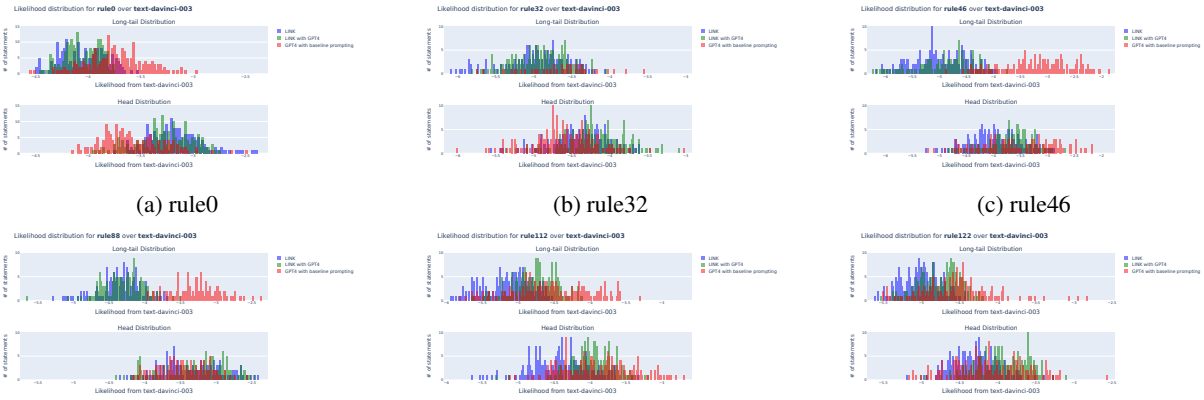
F Long-tail Prompts for LLMs

We tried 10 prompts when prompting LLMs to generate knowledge statements in the long-tail distribution directly with instructions. Table 11 shows the 10 prompts which are appended to the original instruction.

	Prompt
1	Use less frequent terms of A and B and Z.
2	Use terms of A and B and Z that are less common.
3	Use terms with lower frequency for A and B and Z.
4	Use terms of A and B and Z that have lower probability in language model distribution.
5	Use less frequent words of A and B and Z.
6	Use words of A and B and Z that are less common.
7	Use words with lower frequency for A and B and Z.
8	Use less frequent entities of A and B and Z.
9	Use entities of A and B and Z that are less common.
10	Use entities with lower frequency for A and B and Z.

Table 11: An illustration of 10 prompts that instruct LLMs to generate knowledge statements in the long-tail distribution.

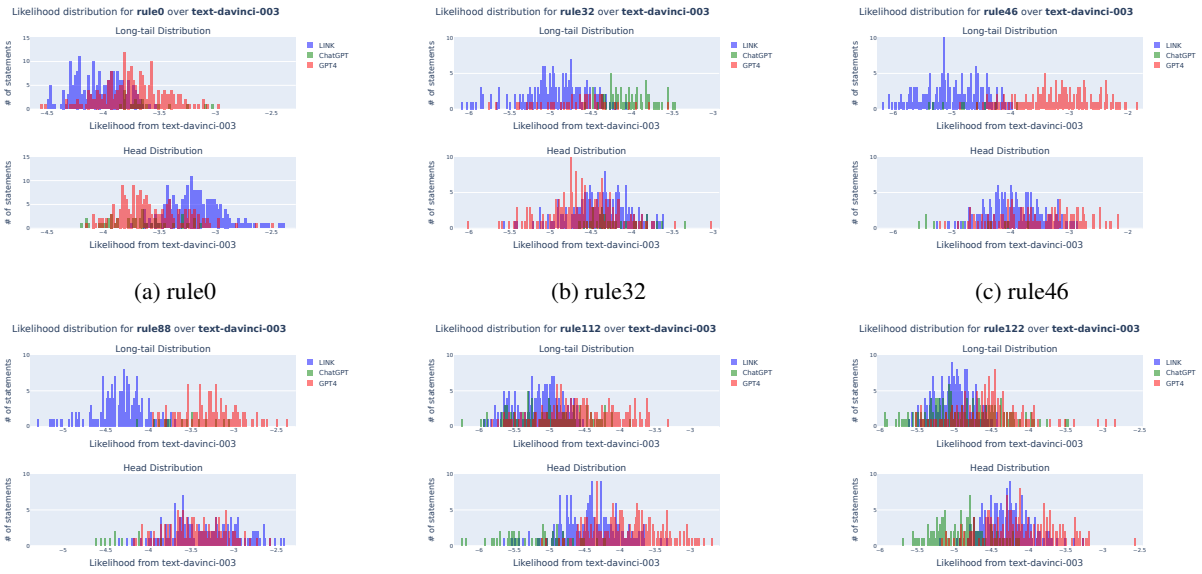
Following Section 3.2, Figure 13 shows the δ value for `GPT4` baselines with these prompts and LINT. These prompts have a similar effect on the distribution of generated statements on most rules and LINT consistently has a higher δ, indicating



(a) rule0 (b) rule32 (c) rule46

(d) rule88 (e) rule112 (f) rule122

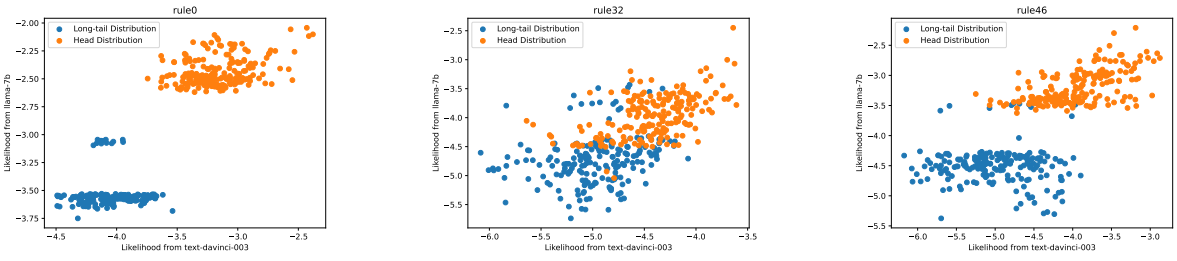
Figure 8: LINK using GPT4 creates statements that fall in a roughly similar long-tail distribution as the original LINK with InstructGPT.



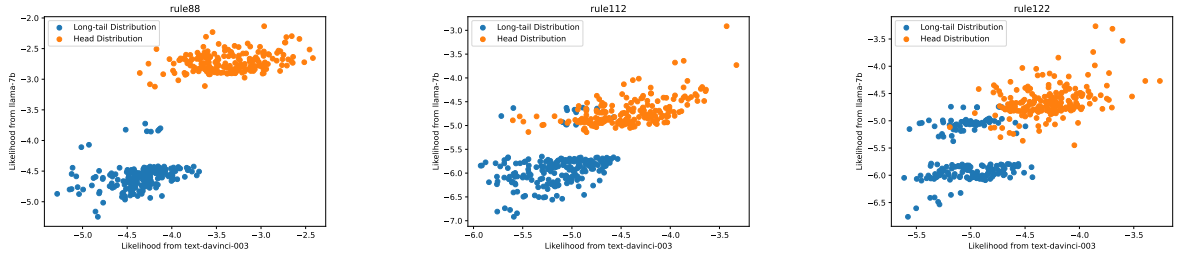
(a) rule0 (b) rule32 (c) rule46

(d) rule88 (e) rule112 (f) rule122

Figure 9: An illustration on the distribution of generated statements by LINK, `ChatGPT` and `GPT4`. While LINK's long-tail generations fall into a lower probability distribution than those of `GPT4`, `GPT4`'s "long-tail distribution" overlaps with the head distribution, indicating that these generations are not truly long-tail.

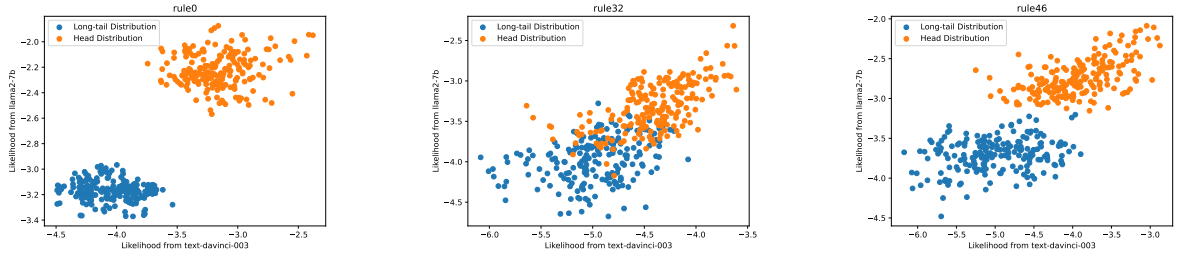


(a) rule0 (b) rule32 (c) rule46

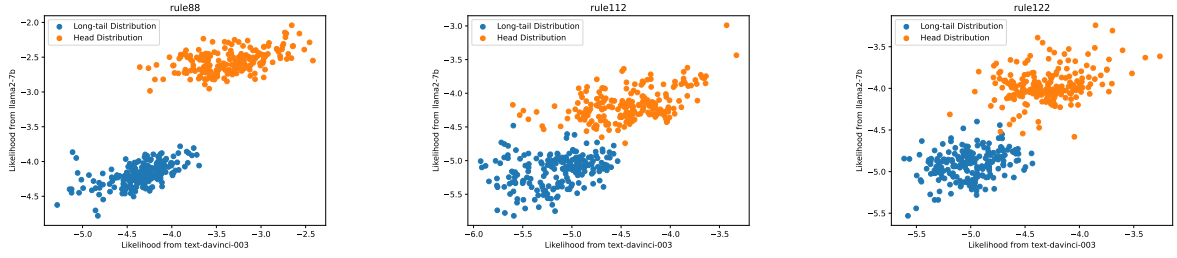


(d) rule88 (e) rule112 (f) rule122

Figure 10: An illustration of the distribution comparison between `llama-7B` and `InstructGPT` of generated statements by LINK.

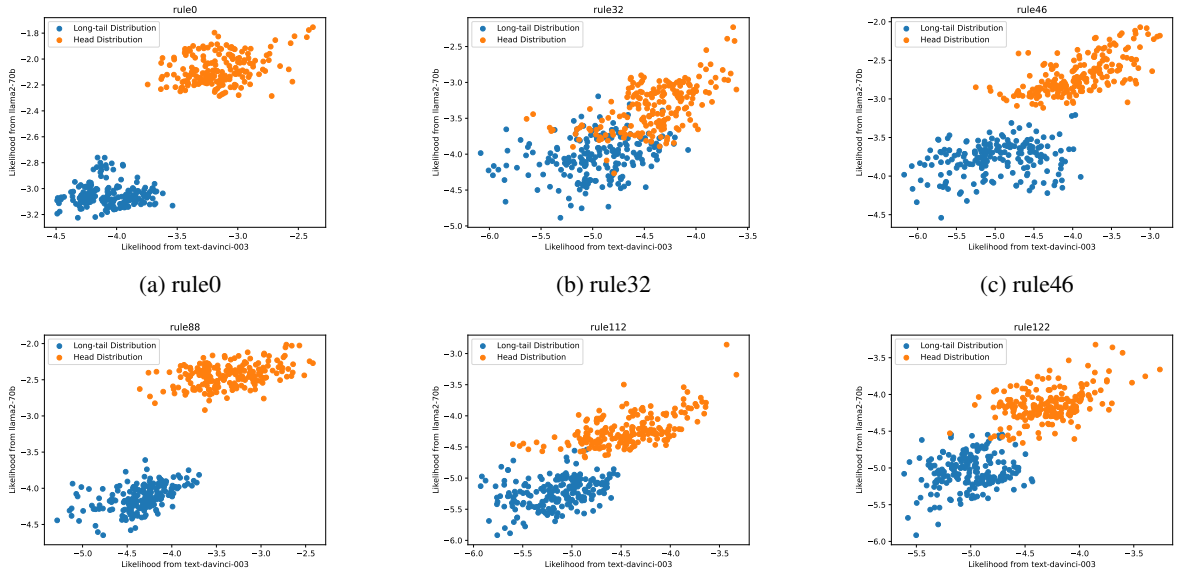


(a) rule0 (b) rule32 (c) rule46



(d) rule88 (e) rule112 (f) rule122

Figure 11: An illustration of the distribution comparison between `llama2-7B` and `InstructGPT` of generated statements by LINK.



(a) rule0 (b) rule32 (c) rule46

(d) rule88 (e) rule112 (f) rule122

Figure 12: An illustration of the distribution comparison between `llama2-70B` and `InstructGPT` of generated statements by LINK.

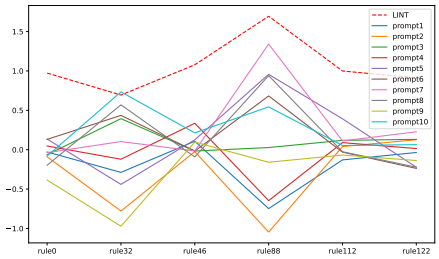


Figure 13: LINT has a higher δ than `GPT4` baselines with 10 different prompts.

that no matter the prompt we use, LLMs cannot directly generate long-tail statements by following instructions. Though these prompts have different δ values on some rules, no one can consistently achieve a higher δ among them.

G Entailment Classification Probing

G.1 Probing template

Table 12 shows templates we used for the entailment classification task. As mentioned in § 4, we divide the templates into positive templates and negative templates. Positive templates are those with a positive label (i.e., *Yes*, *Right* and *True*) and negative templates are those with a negative label (i.e., *No*, *Wrong* and *False*).

Most of the templates have definite labels across all rules. However, the label of Template 7 depends on the rules. If the rule has a positive conclusion (e.g., *Person X can use ChatGPT*), the answer to the question should be positive, i.e., *Yes*. On the contrary, if the rule has a negative conclusion(e.g., *Person X cannot use ChatGPT*), the answer to the question should be negative, i.e., *No*.

G.2 Accuracy averaged over templates

One main concern of not using accuracy of our main metric is because accuracy is not favorable for label imbalance. Because we have 13 templates where for each knowledge statement we will have an uneven number of positive or negative answers, the negative label rate in total is around 51.58%. Therefore, we decide to consider the template biases and deem a model correct only if it answers all templates of a statement correctly, as our goal is to test the true knowledge of models.

When evaluating the accuracy across all templates, the model's performance also drops over long-tail knowledge. Table 13 shows the accuracy across all templates of `GPT4`. Even if LLMs are brittle to templates, they exhibit a performance drop in the long-tail distribution among all domains.

G.3 Model and human error analysis

Table 14 shows probing examples in which models and humans make error. For those from the head distribution, the entailment can be easy if humans can use search engines. For example, humans can

	Template	Label
1	Is it true that if premise, conclusion.	Yes
2	Yes or no: if premise, conclusion.	Yes
3	True or false: if premise, conclusion.	True
4	Right or Wrong: if premise, conclusion.	Right
5	Premise: premise. Conclusion: conclusion. Does premise entail conclusion?	Yes
6	Premise: premise. Conclusion: conclusion_negation. Does premise contradict the conclusion?	Yes
7	Answer the question with yes or no: if premise, conclusion_question?	*Depends*
8	Is it true that if premise, conclusion_negation.	No
9	Yes or no: if premise, conclusion_negation.	No
10	True or false: if premise, conclusion_negation.	False
11	Right or Wrong: if premise, conclusion_negation.	Wrong
12	Premise: premise. Conclusion: conclusion_negation. Does premise entail conclusion?	No
13	Premise: premise. Conclusion: conclusion. Does premise contradict the conclusion?	No

Table 12: Templates used for machine entailment classification task.

	Head	Long-tail
Natural Properties	67.07	63.40
Temporal	83.14	78.97
Outcomes and Effects	81.79	80.19
Locational	68.90	67.07

Table 13: Accuracy over all templates, GPT4

search the periods of "The Paleolithic Era" and "Lion Gate of Mycenae", and answer easily. Thus the human errors in the head distribution may be due to carelessness. For those containing long-tail knowledge, even with search engines, it is not so easy to infer the answer for human annotators. It is also likely that models do not have such knowledge either.

G.4 Rationale analysis

As mentioned in § 4, we examine the rationale the model generated during COT in the entailment classification task and found that the models tend to avoid drawing a "definite conclusion". Table 15 shows an example.

H Domain-wise statistics of LINT

Table 16 shows the detailed statistics of LINT.

H.1 Domain-wise human evaluation

As mentioned in § 3.3, we uniformly sample 4,000 statements from LINT for human evaluation. Table 17 provides more detailed domain-wise statistics on the data type conformity and factual correctness performance of LINT long-tail knowledge generation. While "Natural Properties" has the highest overall accuracy and factuality, model performance on positive templates in Table 4 is the lowest while model performance on negative templates is the highest in this domain. This suggests that these LLMs might have been most aligned in this domain during pre-training.

H.2 Rule definitions

Table 18 shows the definitions of the 6 sampled rules.

I Amazon Mechanic Turk

I.1 Recruiting Workers

We recruit workers from all English-speaking countries (US, UK, New Zealand, Australia, Canada), although AMT workers are mostly US-based. We use a qualification task to recruit AMT workers. In the qualification task, all workers will be presented with three manually selected statements, which are clear and representative. Each statement has five related questions as described in Appendix C. Only workers who answer all the questions correctly will be recruited. In the end, we recruited 38 workers to evaluate the quality of generation and 17 workers as human baselines for the entailment classification task. We paid the workers $0.47 per annotation for evaluating the quality of generations and $0.11 per annotation for the entailment classification task, to match $15 per hour based on their working time.

I.2 Templates

Figure 14 and Figure 15 show the template we use for the evaluation of generation quality and the entailment classification task.

I.3 Agreement statistics

Table 19 shows the agreement of annotations in the evaluation task. The high agreement of the data type conformity and factual correctness for LINT ensures the reliability of our results. The

Statement:	Yes or no: if Plant X vanished in Paleolithic Era, Plant X cannot surround Lion Gate of Mycenae. (Head)
True label:	Yes
Human answer:	No
Model answer (GPT4 + COT):	Yes. The Paleolithic Era ended around 10,000 BC and the Lion Gate of Mycenae was built around 1250 BC. If PlantX vanished in the Paleolithic Era, it could not have surrounded the Lion Gate of Mycenae which was built much later.
Statement:	Yes or no: if Bag X has trouble containing Pandeiro, Dhak cannot fit in Bag X. (Long-tail)
True label:	Yes
Human answer:	No
Model answer (GPT4 + COT):	No. The size of the Pandeiro does not directly relate to the size of the Dhak. They are different objects and may have different shapes and sizes. Therefore, just because BagX has trouble containing a Pandeiro doesn't necessarily mean that a Dhak cannot fit in BagX.

Table 14: Examples that human and model make mistakes.

Input	**Premise**: PersonX is of age 17 years 8 months. **Conclusion**: PersonX cannot consume Bourbon. Does premise entail conclusion?
Label	Yes
Answer	No
Rationale	In many countries, the legal drinking age is 18 or 21. If we do not know the specific country PersonX is in, we cannot definitively conclude that they cannot consume Bourbon.

Table 15: A rationale generated by GPT4 with COT shows that even though the model is able to cite the relevant information, it inclines to answer "No" because it cannot "definitively conclude".

	Rule	Head	Long-tail
Temporal	81	15,143	15,317
Outcomes and Effects	132	14,966	15,010
Natural Properties	139	16,788	16,669
Locational	65	7,323	7,370
Total	417	54,220	54,366

Table 16: The number of symbolic rules and knowledge statements in different domains in LINT.

agreement for baselines is lower, which also indicates that the generated statements of baselines are of low quality and confusing for human annotators.

I.4 Failure Case Examples

We analyze some failure cases that are labeled as incorrect in the human evaluation. Table 20 presents some examples.

	Data Type	Factuality	Overall
Temporal	90.18	85.27	77.38
Outcomes and Effects	94.97	80.73	75.98
Natural Properties	96.61	96.61	93.81
Locational	98.88	70.79	70.79

Table 17: The factual and data type accuracy of each domain in human evaluation.

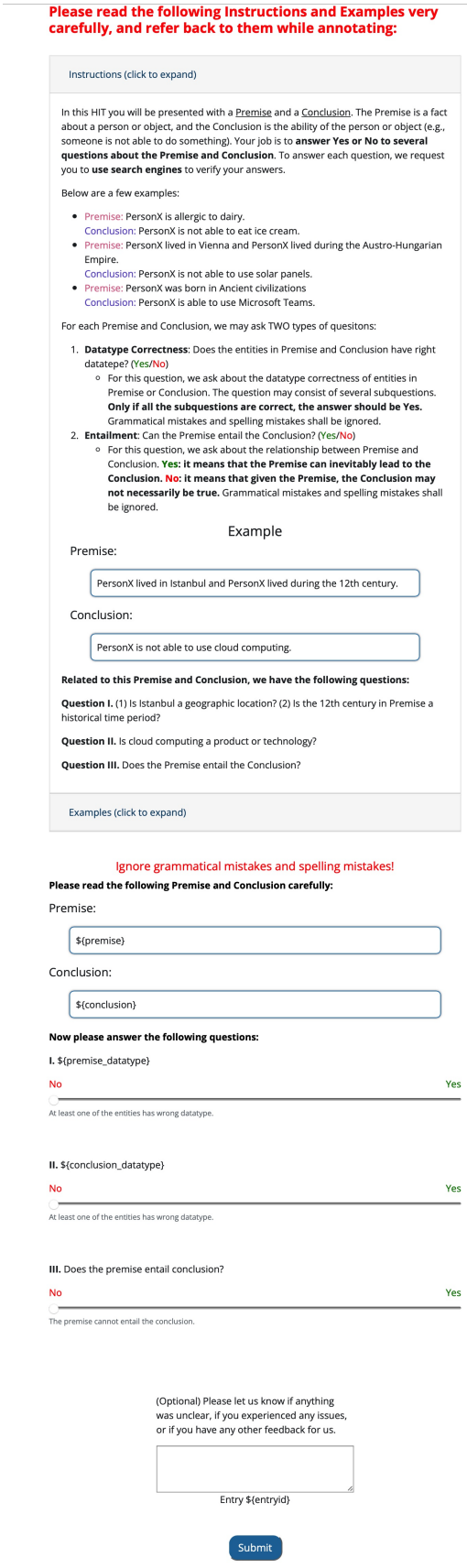


Figure 14: AMT template for the evaluation of generation quality.

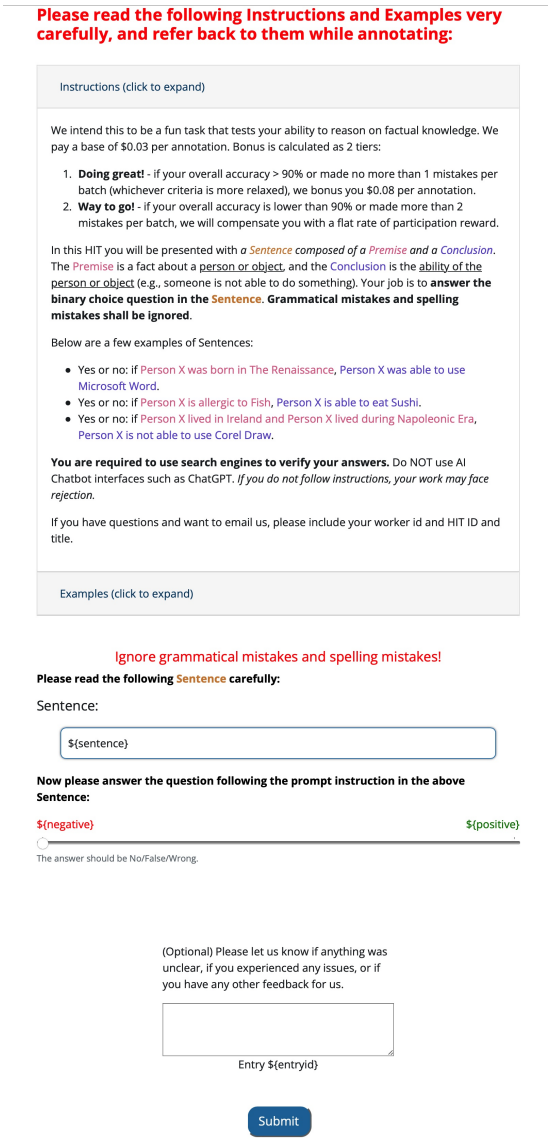


Figure 15: AMT template for the human baseline of the entailment classification task.

Rule0	lived_in(Person P, Geographic Location A) & lived_during(Person P, Historical Time Period D) & existed_during(Geographic Location A, Historical Time Period D) & was_invented_in(Product or Technology C, Year Y) & is_more_than_a_century_earlier_than(Historical Time Period D, Year Y) → is_not_able_to_use(Person P, Product or Technology C)
Rule32	has_trouble_lifting(Person X, Name of Appliance B) & is_heavier_than(Object A, Name of Appliance B) → cannot_lift(Person X, Object A)
Rule46	is_allergic_to(Person A, Substance X) & includes(Name of Cosmetics B, Substance X) → cannot_use(Person A, Name of Cosmetics B)
Rule88	died_in(Historical Figure A, Historical Time Period X) & was_created_during(Artifact B, Historical Time Period Y) & is_earlier_than(Historical Time Period X, Historical Time Period Y) → cannot_create(Historical Figure A, Artifact B)
Rule112	has_trouble_containing(Room B, Furniture C) & is_larger_than(Furniture A, Furniture C) → cannot_fit_in(Furniture A, Room B)
Rule122	has_trouble_containing(Trunk B, Furniture C) & is_larger_than(Furniture A, Furniture C) → cannot_fit_in(Furniture A, Trunk B)

Table 18: Rule definitions of six sampled rules.

Accuracy	ChatGPT	GPT4	LINK
Data Type	79.29	83.16	87.54
Factuality	38.35	58.48	75.10
Overall	65.64	74.93	83.39

Table 19: Agreement of annotations in the evaluation task.

Rule 172	Locational	**Rule:** is_located_in(Person A, Location X) & is_forbidden_in(Food Item B, Location X) → cannot_eat(Person A, Food Item B) **Premise:** Person X is located in Houston **Conclusion:** Person X cannot eat Chocolate **Is Houston a location? Annotation:** Yes **Is Chocolate a food item? Annotation:** Yes **Does the premise entail the conclusion? Annotation:** No **Reason:** It is a factual error. Chocolate is not actually forbidden in Houston, so People in Houston can eat chocolate.
Rule 371	Capability and Advice	**Rule:** can_treat(Drug B, Name of Disease X) & has(Person A, Name of Disease X) → should_take(Person A, Drug B) **Premise:** Person X has Hepatitis **Conclusion:** Person X should take Sofosbuvir **Is Hepatitis a name of disease? Annotation:** Yes **Is Sofosbuvir a drug? Annotation:** Yes **Does the premise entail the conclusion? Annotation:** No **Reason:** It is a factual error. There are different types of hepatitis viruses. Sofosbuvir is a medication used primarily for the treatment of hepatitis C. For other types of hepatitis, different medications or treatments may be necessary.
Rule 274	Temporal	**Rule:** vanished_in(Plant A, Historical Time Period X) & was_invented_in(Weapon B, Historical Time Period Y) & is_earlier_than(Historical Time Period X, Historical Time Period Y) → cannot_be_used_to_conceal(Plant A, Weapon B) **Premise:** Plant X vanished in Mongol **Conclusion:** Plant X cannot be used to conceal M92 Zolja **Is Mongol a historical time period? Annotation:** No **Is M92 Zolja a weapon? Annotation:** Yes **Does the premise entail the conclusion? Annotation:** Yes **Reason:** It is a data type error. The Mongols are an East Asian ethnic group native to Mongolia, not a time period. The Mongol Empire may refer to a period of the 13th and 14th centuries, but Mongol cannot.
Rule 204	Natural Properties	**Rule:** has_trouble_containing(Drawer B, Tool C) & is_larger_than (Tool A, Tool C) → cannot_be_placed_in(Tool A, Drawer B) **Premise:** Drawer X has trouble containing Scroll saw **Conclusion:** Car cannot be placed in Drawer X **Is Scroll saw a Tool? Annotation:** Yes **Is Car a Tool? Annotation:** No **Does the premise entail the conclusion? Annotation:** Yes **Reason:** It is a data type error. Car is a vehicle instead of a tool.

Table 20: Examples that are labeled as incorrect during human evaluation. Note that the reasons are analyzed by the authors instead of annotators.