Towards Reliable Latent Knowledge Estimation in LLMs: In-Context Learning vs. Prompting Based Factual Knowledge Extraction

Anonymous ACL submission

Abstract

We propose an approach for estimating the latent knowledge embedded inside large language models (LLMs). We leverage the in-context learning (ICL) abilities of LLMs to estimate the extent to which an LLM knows the facts stored in a knowledge base. Our knowledge estimator avoids reliability concerns with previous prompting-based methods, is both conceptually simpler and easier to apply, and we demonstrate that it can surface more of the latent knowledge embedded in LLMs. We also investigate how different design choices affect the performance of ICL-based knowledge estimation. Using the proposed estimator, we perform a large-scale evaluation of the factual knowledge of a variety of open source LLMs, like OPT, Pythia, Llama(2), Mistral, Gemma, etc. over a large set of relations and facts from the Wikidata knowledge base. We observe differences in the factual knowledge between different model families and models of different sizes, that some relations are consistently better known than others but that models differ in the precise facts they know, and differences in the knowledge of base models and their finetuned counterparts.

1 Introduction

Conversational chatbots (e.g., OpenAI's ChatGPT) built around large language models (e.g., OpenAI's GPT) are increasingly being used for a variety of information retrieval tasks such as searching for information or seeking recommendations related to real world entities like people or places (Wu et al., 2023; Zhu et al., 2023). A worrisome concern in such scenarios is the factual correctness of information generated by the LLMs (Peng et al., 2023; Hu et al., 2023a; Snyder et al., 2023; Yao et al., 2023; Ji et al., 2023; Zhang et al., 2023; Wang et al., 2023).

The latent knowledge estimation problem: To avoid making false assertions about a real-world entity, an LLM first needs to have factual (true) knowledge about the entity. Given a prompt like *"Einstein was born in the year"*, LLMs may generate both the correct answer (*"1879"*) and wrong answers (e.g., *"1878" or "1880"*) with some probabilities. If an LLM *knows* the fact, one can hope that the probability with which it would generate the correct answer would be much higher than the wrong answers (Jiang et al., 2021). As LLMs are typically pretrained over a Web corpus (including Wikipedia data) with millions of facts about real-world entities, they have the opportunity to learn factual knowledge about our world and latently embed the knowledge in their parameters. But, *how can we estimate the extent to which LLMs have knowledge of real-world facts?*

Reliability of latent knowledge estimates: Prior works (Jiang et al., 2020; Bouraoui et al., 2020) followed (Petroni et al., 2019), and represented factual knowledge in the form of triplets $\langle x, r, y \rangle$, where the subject x has a relation of type r with the object y (e.g., $\langle \text{Einstein}, \text{birth-year}, 1879 \rangle$). The central challenge of latent knowledge estimation is to infer y given x and r by *only* using information extracted from the LLM. Typically, the inference relies on probing the LLM with prompts constructed using x and r and analyzing the responses. Current approaches have few well-defined rules to avoid prompt engineering and prompt hacking, raising serious concerns about the reliability of their estimates. Against this background, in this paper, we make **four** primary contributions:

1. *A simple yet reliable latent knowledge estimator (LKE) leveraging in-context learning (ICL):* We propose a latent knowledge estimator (LKE) that leverages in-context learning (ICL), called IC-LKE, in a simple yet clever way to avoid the many reliability concerns with prompting based previous knowledge estimators.

2. *Exploring the nuances of using ICL for knowledge estimation:* We investigate the impact of dif-

ferent ICL design choices on the estimation of latent knowledge, such as the number of in-context examples, when some of the examples are unknown to the model or simply incorrect, as well as the sequence in which they appear. While we focus on knowledge estimation, our findings can inform the application of ICL in other contexts.

3. *A comparison of IC-LKE with previous approaches:* We empirically demonstrate that IC-LKE outperforms previous knowledge estimation approaches that rely on human-generated or machine-mined prompts across a variety of different open-source models and different types of factual relations. In contrast to prompting based methods, which are relation-specific and LLM-specific, IC-LKE's design is straightforward to apply.

4. *A systematic comparison of latent knowledge of open source LLMs at scale:* We use IC-LKE to evaluate the knowledge of 49 open-source LLMs spanning many families such as Llama(2), Gemma, Mistral, OPT, Pythia, etc. across a wide range of sizes, both with and without instruction-finetuning over 50 different relations and 20,000 facts from Wikidata. We find that models from some families such as Llama2, Mistral and Gemma and larger models know more facts than others, that models within the same family differ in the specific facts they know, despite being trained on the same data, and that fine-tuning reduces the amount of factual knowledge that can be extracted from the models.

Related Work: Researchers have proposed several approaches to estimate latent knowledge from LLMs, which can be categorized into two ways: (i) Model-internals based approaches leverage the LLM attention map (Wang et al., 2020), activation function (Burns et al., 2022), or model parameters (Kazemnejad et al., 2023) to decide whether factual information can be extracted from the LLM. In our study, we rely on the probability distribution of generated tokens in an LLM – thereby our method belongs to the model-responses based approach. (ii) Model-responses based approaches – generally applicable to a wide range of LLM models – often propose different prompting techniques to nudge the LLM to validate whether a target fact is stored in it (Chern et al., 2023; Sun et al., 2023; Wang et al., 2020; Petroni et al., 2019; Jiang et al., 2021; Newman et al., 2022; Jiang et al., 2020). Prompt-based methods differ subtly by the choice of prompts and evaluation criteria. Besides, the prompts are often brittle (Zamfirescu-Pereira et al., 2023; Arora et al., 2023; Sclar et al., 2023) – their success depends on the hypothesis that the LLM indeed understands the prompts. In our study, we instead seek a minimal understanding of prompts by an LLM and design a knowledge estimation method based on the in-context learning. As a test bed (Elsahar et al., 2018; Hu et al., 2023b; Sun et al., 2023; Petroni et al., 2019; Zhu and Li, 2023; Kryściński et al., 2019), we consider facts from existing knowledge graphs for performing knowledge estimation of LLMs.

2 Designing Reliable LKEs

Today, there exist many general-purpose as well as domain-specific factual knowledge bases that contain a very large number (millions to billions) of facts. The facts can be encapsulated as triplets, represented as $\langle$*subject* (x), *relation* (r), *object* $(y)\rangle$. These triplets offer a general way to represent factual knowledge about real-world entities in knowledge graphs or other structured knowledge bases. The goal of latent knowledge estimation is to infer what fraction of the facts are *known* to a LLM. We call methods that estimate the amount of latent knowledge inside an LLM *latent knowledge estimators* (LKEs).

2.1 Reliability concerns with existing LKEs

Existing approaches to estimating latent knowledge in LLMs use a variety of factual knowledge tests. Below, we identify several reliability concerns with current designs that motivate our new LKE design.

1. *LLM-specific restrictions on test topics:* Many prior works (Petroni et al., 2019; Jiang et al., 2020) limit the choice of facts that can be used in tests to those where the surface form of the objects (y) is represented by a single token by the LLM's tokenizer. As different LLMs use different tokenizers, this limitation prevents us from comparing the latent knowledge across different LLMs. Furthermore, only popular objects tend to be represented by a single token and so the resulting estimates are not representative of the LLM's knowledge of facts with multi-token object representations.

2. *Unrestricted choice of test prompts:* Many past works have attempted to use test prompts without any restrictions, including both human-generated or machine-mined prompts (Jiang et al., 2020; Zamfirescu-Pereira et al., 2023; Arora et al., 2023; Sclar et al., 2023). They typically intersperse the subject x and object y between additional relationship context-communicating tokens. Some

analyze the performance of a variety of prompts and then pick the best-performing or use an ensemble of the best-performing prompts (Jiang et al., 2020; Newman et al., 2022; Fernando et al., 2023). However, these approaches raise two important concerns: First, the generated prompts, particularly those that are machine-mined, may include tokens that can implicitly or explicitly introduce additional (side-channel) information that makes it easier to answer the question. As a specific example, in a prior work (Jiang et al., 2020), for the relation *"position held"*, the prompt *"x has the position of y"* performed worse than *"x is elected y"*. But, note that the second prompt potentially introduces a side-channel: it implicitly rules out answer choices for unelected positions like Professor and favors elected positions like President. Second, selecting from an unbounded number of potential prompt choices raises concerns about the complexity of LKEs (the size of the set of all considered prompts) and the potential for over-fitting, which in turn brings the reliability of estimates into question.

3. *Reliance on LLMs' meta-linguistic judgments:* Prior works used prompts (Chern et al., 2023; Sun et al., 2023; Wang et al., 2020; Petroni et al., 2019; Jiang et al., 2021; Newman et al., 2022; Jiang et al., 2020) for communicating the question as well as the expected format of answers. But, the scores (estimates) resulting from such prompt-based testing conflate an LLM's latent knowledge of the facts with the LLM's meta-linguistic judgments, i.e., the LLM's ability to comprehend the prompt, understand the question embedded within the prompt and output the answer in some expected format (Hu and Levy, 2023). The impact on meta-linguistic judgments can be seen from the fact that multiple semantically-equivalent prompts result in different responses from an LLM and thereby, different estimates of latent knowledge (Hu and Levy, 2023).

Motivated from the above, we derive the following three design principles for LKEs. A reliable LKE design should:

- DP1: *generate estimates for any factual topic and tokenization scheme.*
- DP2: *limit arbitrary prompt engineering to minimize over-fitting & side-channels.*
- DP3: *minimize reliance on meta-linguistic prompts.*

2.2 A new In Context learning based LKE (IC-LKE)

Our goal is to estimate whether an LLM knows a fact $f = \langle x, r, y \rangle$. The challenge is to probe the LLM and evaluate its responses in a way compatible with the design principles set in Section 2.1.

Key idea: Leverage in-context learning. LLMs have shown to exhibit In-Context Learning (ICL) abilities (Brown et al., 2020) that allow them to infer and extrapolate patterns in their inputs. We leverage this ability to communicate information about relation r without additional instructions to the LLM (DP3) by providing it with a list of facts based on r.

Example 1. Assume that we want to probe for whether an LLM knows the fact ⟨ *Einstein, birth-year, 1879* ⟩. We can use other facts for the birth-year relation such as ⟨ *Feynman, birth-year, 1918* ⟩, ⟨ *Heisenberg, birth-year, 1901* ⟩ to construct an input *"Feynman 1918 Heisenberg 1901 Einstein"*. By providing in-context examples to the model, we communicate the relation between subjects and objects. To correctly extrapolate the pattern, the model needs to retrieve Einstein's birth-year as the completion of the sequence.

More formally, given a training dataset of facts $\mathcal{F}_r = \{\langle x_i, r, y_i \rangle\}_{i=1}^{n}$ for relation r, as well as a test fact $f = \langle x, r, y \rangle$, we leverage ICL to construct prompts that elicit information about f as

$$\sigma(x, r) = x_1\ y_1\ \ldots\ x_n\ y_n\ x \qquad (1)$$

We use r to pick facts from $\mathcal{F}_r$ and concatenate the tokens corresponding to the subjects and objects, but do not include any other information about r (DP2). We use space " " as the separator token and discuss this choice in more detail in Section 4.1. We discuss other design choices for IC-LKE construction in Section 3. When further details are not needed, we simply refer to *some* input as σ.

Evaluating model outputs. We evaluate the output of model θ for input $\sigma(x, r)$ based on the probabilities θ assigns to the tokens of the corresponding object y. To allow for objects y consisting of multiple tokens and to be independent of the specific tokenization scheme (DP1), we compute the *object probability* over multiple tokens as follows:

$$P_\theta(y \mid \sigma) = \prod_{i=2}^{|y|} P_\theta(y^{(i)} \mid y^{[i-1:1]}\, \sigma) \cdot P_\theta(y^{(1)} \mid \sigma), \qquad (2)$$

where $|y|$ denotes the number of tokens in y and $P_\theta(y^{(i)} \mid y^{[i-1:1]}\, \sigma)$ is the conditional probability

of predicting the i-th token $y^{(i)}$ of y given the preceding tokens $y^{(i-1)}, \ldots, y^{(1)}$, and σ.

Multiple-choice testing. To determine whether model θ knows a fact $f = \langle x, r, y^* \rangle$, we test whether given input $\sigma(x, r)$, θ can choose the correct object y^* from among a set of M unique alternatives. Specifically, given fact f, we derive a test instance called *choice* $c = \langle x, r, y^*, \mathcal{Y} \rangle$, where $\mathcal{Y}$ is a set of M plausible but incorrect alternatives. We discuss the choice of $\mathcal{Y}$ in Section 4.

$$\text{pred}_\theta(c) \triangleq \underset{y \in \{y^*\} \cup \mathcal{Y}}{\text{argmax}} \; P_\theta(y \mid \sigma(x, r)) \quad (3)$$

denotes the prediction of θ for choice $c = \langle x, r, y^*, \mathcal{Y} \rangle$. The predicted object has the maximal object probability within $\{y^*\} \cup \mathcal{Y}$.

Evaluation Metric. We evaluate the factual knowledge of model θ over a dataset of choices $\mathcal{D} = \{c_i\}_{i=1}^n$ using multiple choice accuracy:

$$\text{acc}(\theta, \mathcal{D}) \triangleq \frac{\sum_{c \in \mathcal{D}} \delta\left(y^* = \text{pred}_\theta(c)\right)}{|\mathcal{D}|} \quad (4)$$

where $\delta(\cdot)$ is the indicator function.

The IC-LKE design satisfies the knowledge estimation design principles. The IC-LKE design proposed here satisfies the design principles from Section 2.1, since

- DP1: its relative probability comparisons between different answer-options make it applicable to arbitrary types of facts.
- DP2: it uses the same, minimal prompt design based on ICL across all relations.
- DP3: its only requirement is that the LLM is able to use ICL, no further assumptions about any metalinguistic abilities are made.

3 Exploring the design space of IC-LKE

By design, IC-LKE avoids many limitations of prior works. However, IC-LKE introduces a few design choices for the input, i.e., $\sigma(x, r)$ in Equation (1). One must decide the right n, the number of in-context examples included in $\sigma(x, r)$. Further, it is unclear how IC-LKE would be impacted when some of the chosen examples are unknown to the model or are incorrect. We study both these factors in detail by varying n and introducing unknown or incorrect examples within these n examples. These experiments allows us to better understand the number of in-context examples needed and how robust IC-LKE is to several

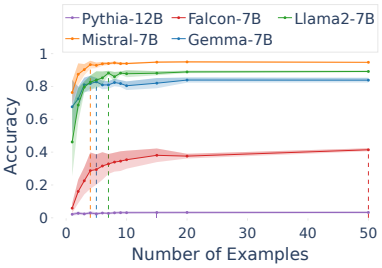
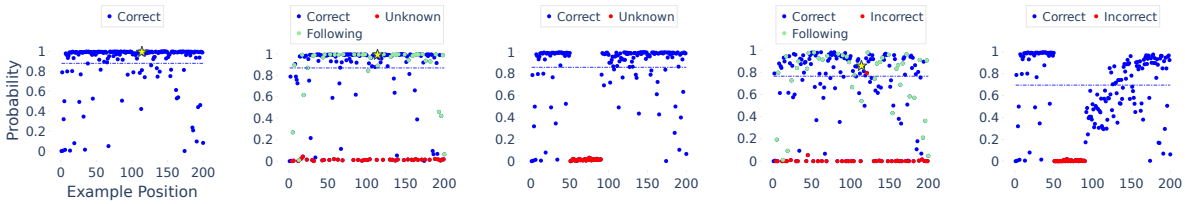


Figure 1: **[Influence of the number of in-context examples]** We examine how varying numbers of in-context examples influence the accuracy (calculated as defined in Eq 5) across different LLMs. The vertical dashed line indicates the number of examples at which the models achieve 95% of their respective stable accuracy at 50 examples.

types of noise in these in-context examples. We perform an in-depth empirical analysis on a *Nobel Laureate* dataset for the relation *'birth year'* (details in A.1). The dataset consists of facts formatted as $\langle \text{Person}(x), \text{birth-year}(r), \text{YYYY}(y) \rangle$.

More knowledgeable models need fewer in-context examples, but a small number suffices for most models. In Figure 1, we report knowledge estimation accuracy (Eq. (5)) for different LLMs evaluated on 900 test samples, with varying numbers of in-context examples (n) by randomly sampling from the training set using five random seeds. With an increasing number of in-context examples, the mean accuracy increases while the standard deviation decreases in different LLMs, i.e., the models gradually converge to a stable performance. Using dashed vertical lines, we report the minimum number of examples required by different LLMs to achieve 95% of the accuracy at 50 in-context examples. Interestingly, LLMs with higher estimation accuracy tend to require fewer in-context examples compared to those with lower accuracy. A potential explanation for this behavior is that in order to infer the relation r, models need to comprehend the examples presented in the prompt. Therefore, less knowledgeable models need to see more examples in order to infer r. To further investigate which individual facts may be known or unknown to a model, we look at the generation probability of in-context objects in 200 correct subject (x)-object (y) pairs using the Mistral-7B model, as shown in Figure 2a. Similar results for additional models are presented in Appendix E. Note that here we are only looking at probabilities of the object (y) for in-context examples given previous x y pairs in the input to understand which of these samples are known by the LLM. The Mistral-7B model demon-



(a) (Subject, object) examples in a prompt (b) Distributed unknown examples (c) Continuous unknown examples (d) Distributed incorrect examples (e) Continuous incorrect examples

Figure 2: **[Variation in object probabilities of Nobel laureate data using Mistral-7B]** Figure 2a illustrates the probability of each object at various positions in the prompt. We show the impact on probabilities after replacing objects with unknown ones at randomly distributed positions in Figure 2b and at continuous positions in Figure 2d. Similarly, we also show the impact of incorrect examples when replaced at randomly distributed positions (Figure 2d) and continuous positions (Figure 2e). In all plots, the horizontal dashed line shows the average probability of the correct examples (blue dots).

strates a gradual increase in probability for generating correct objects as we go from left to right on the x-axis (note that for a point on the x-axis, points before it are in context, thus points on the right have more context to leverage) in Figure 2a, stabilizing at a mean probability of approximately 85%. We also see that some objects at later positions have a lower generation probability. This suggests that the LLM may be less confident about its knowledge of the facts corresponding to them. We can leverage the token generation probability as a signal of LLM's confidence when evaluating LKEs (see Appendix D).

Models are robust to unknown examples. Next, we investigate the robustness of estimates to occurrence of unknown examples. We insert unknown examples in two distinct ways: one where we randomly distribute the occurrence of unknown examples throughout $\sigma(x, r)$, and another more extreme scenario where we replace a continuous block of examples with unknown ones. We chose 40 out of the 200 examples and replaced them with unknown examples created using fictitious names and birth years [1]. Our findings are shown in Figures 2b and 2c for random and continuous replacement respectively. Unknown examples are marked by red dots, examples immediately following unknown ones in cyan dots and the rest in blue dots. The unknown examples show generation probabilities close to zero, confirming the LLM's tendency to assign low probabilities to unknown data. However, interestingly, unknown examples minimally impact surrounding data in both settings.

Models are vulnerable to incorrect examples. We investigate the impact of including incorrect examples in $\sigma(x, r)$. Similar to the setup for unknown examples, we also insert 40 (out of 200) incorrect examples randomly (Figure 2d) and simultaneously (Figure 2e). In our experiments, these incorrect examples are created by altering the birth years of known Nobel laureates and are marked by red dots in the plots. In contrast to inserting unknown examples, the LLM significantly struggles with incorrect examples. Injection of such examples detrimentally affects the LLM's performance in both settings. We highlight one randomly marked yellow star example in Figure 2a, Figure 2b, and Figure 2d to show how the presence of incorrect samples brings down the probability of surrounding points.

Summary: LLMs can identify the relation pattern of subject-object pairs even with a small set of in-context examples in the prompt. LLMs are relatively robust to unknown examples, but their ability to recollect factual knowledge is vulnerable to incorrect examples, particularly when they appear in a continuous sequence. Our findings allude to the effectiveness of designing an IC-LKE, where we carefully place correct examples from a training dataset and proceed to estimate the latent knowledge of the LLM on examples from the test set. Furthermore, the findings also motivate us to design a more efficient in-context learning based LKE, called EIC-LKE, that can process multiple test examples simultaneously in a single prompt where training examples are placed preceding each test example, see more details in the Appendix F.

4 Experiments and Results

We present the empirical findings of IC-LKE (as well as the efficient version, EIC-LKE) on the knowledge-estimation task on 49 open-source (pre-trained and fine-tuned) LLMs across different LLM families and sizes. We enlist models and their simplified names used in this paper in Appendix 6, Ta-

[1] generated via https://en.namefake.com/api

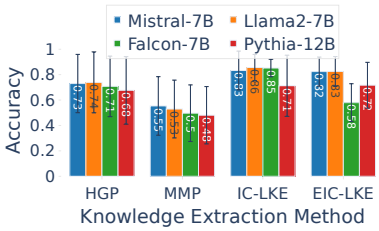


Figure 3: **[Performance comparison for different latent knowledge extractors]** We compare the accuracy of IC-LKE and EIC-LKE with the baseline method (Jiang et al., 2020) across 12 relations from T-REx-MC.

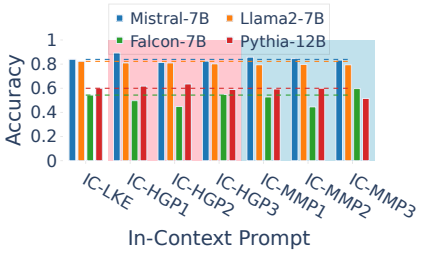


Figure 4: **[Influence of different separators]** We replace the '[space]' token separating the subject-object pairs with human-generated prompts (HGP, red background) and machine-mined prompts (MMP, blue background) for the relation 'original broadcaster'. Accuracy performance is agnostic to the separators.

ble 6, and provide a leader-board of models based on IC-LKE in Table 7.

Dataset: We evaluate the knowledge of models on a large set of facts from the T-REx dataset[2] (Elsahar et al., 2018). We selected relations from T-REx with at least 500 samples and linked to a minimum of 100 unique objects. This filtering leads to 50 distinct relations spanning categories like birth dates, directorial roles, parental relationships, and educational lineage. The resulting T-REx Multiple Choice (T-REx-MC) dataset comprises 5,000 training and 20,000 test facts. Appendix A contains detailed information on the dataset and relations.

Choosing the set $\mathcal{Y}$ & its impact on test difficulty: For each fact $\langle$*subject (x), relation (r), object (y^*)*$\rangle$, we generate alternative objects $\mathcal{Y}$ to create multiple choices. Note that the alternative objects in $\mathcal{Y}$ are viable choices and cannot be easily eliminated. Therefore, for each fact $\langle x, r, y^* \rangle$ we select $y \in \mathcal{Y}$ from other facts in the dataset that share the same relationship r. For computational feasibility, we sample $|\mathcal{Y}| = 99$ alternative objects per fact, so that a random guess between $\{y^*\} \cup \mathcal{Y}$ has a 0.01 probability of being correct.

4.1 IC-LKE vs. prompt-based approaches

We compare the performance of IC-LKE and EIC-LKE with the existing prompt-based approaches (Jiang et al., 2020) and report two key takeaways.

IC-LKE outperforms prompt-based approaches. We randomly sample three human-generated prompts (HGP) and machine-mined prompts (MMP) from (Jiang et al., 2020) for 12 common relations between T-REx-MC and (Jiang et al., 2020). The HGPs and MMPs for all relations are in Appendix G. In Figure 3, IC-LKE and EIC-LKE outperform HGP and MMP in terms of higher mean accuracy across different models and 12 relations. Also, IC-LKE and EIC-LKE have lower standard deviation than HGP and MMP, indicating a higher consistency of IC-LKE and EIC-LKE on knowledge estimation tasks. In Appendix H.2, we report relation specific results, where IC-LKE and EIC-LKE estimate higher factual knowledge than the existing works in most relations, *thereby demonstrating the superiority of IC-LKE and EIC-LKE over existing methods.*

IC-LKE is a flexible and effective knowledge estimator. We adapt IC-LKE by replacing the separator '[space]' with three separators from HGP and MMP each for the relation 'original broadcaster' and report estimation accuracy in Figure 4. We can observe that '[space]' token demonstrates an equivalent performance with semantically meaningful prompts via HGP and MMP. Therefore, *adding relation specific separators has a limited impact on factual knowledge estimation, as long as the subject-object pairs are correctly presented.* Furthermore, finding relation-specifc prompts often require hand-crafted efforts vs. an automatic in-context based approach like ours where (subject, object) pairs are used. *Therefore, IC-LKE can potentially extend to any facts from knowledge graphs over any LLM while HGP and MMP requires additional supervision and relation-specific validation.*

4.2 Evaluating Diverse Models and Relations

We investigate the performance of 35 pre-trained LLMs and 14 fine-tuned LLMs across 50 relations using the IC-LKE framework. Our analysis is designed to uncover nuanced insights into the knowledge levels and structures within these models. We will examine the results through two primary lenses: (1) the variations in knowledge across different model families, and (2) the influence of model size and fine-tuning within the same

[2] https://huggingface.co/datasets/relbert/t_rex

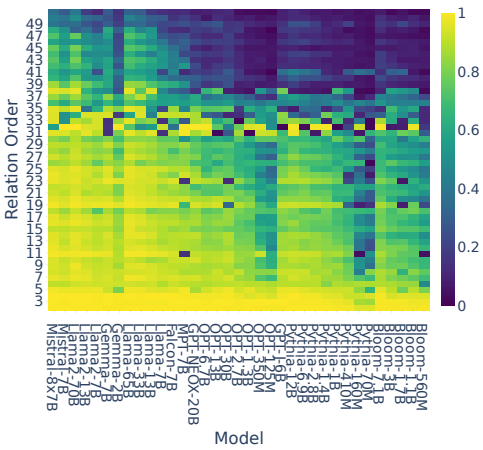


Figure 5: **[Accuracy for 35 pre-trained LLMs on the 50 different relations in T-REx-MC]** Models are grouped by family and arranged from left to right based on the accuracy of the model closest to 7 billion parameters. Within each family, models are ordered by their average accuracy.

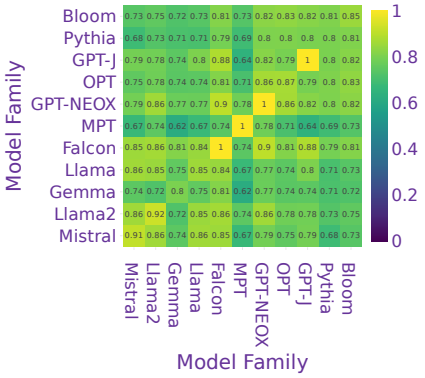


Figure 6: **[Pearson correlation coefficients between model families]** We compute the Pearson correlation coefficients between each pair of models and then compute the average correlation across the same model family.

model family on their knowledge attributes.

4.2.1 Comparing different LLMs families

Some model families are consistently more knowledgeable than the rest. We sort the model families based on the performance of the model closest to 7B parameters [3], and the models within each family based on average accuracy across 50 relations. Figure 5 shows that the Mistral, Llama2, Gemma, and Llama families have higher performance on most of the relations than Pythia, Bloom, and OPT, indicating their lower factual knowledge.

Different model families align in their relative factual knowledge. We investigate the correlations between each model pair's performance over 50 relations to assess the agreement in their knowledge levels of the 50 relations. We compute the average correlations within each model family (e.g. Llama2 7B, 13B, 70B) in Figure 6. Despite differences in architecture and training datasets among model families, there is a significant consensus (correlation > 0.6, see Figure 14) regarding the hierarchy of knowledge across various relations. We also compile the three best and worst-performing relations for each model in Table 9, illustrating the consensus among all models.

4.2.2 Comparing within the same LLM family

Larger models embed more knowledge. We show in Figure 5 that, within each model family, bigger models (e.g. Llama-65B) generally outperform their smaller counterparts (e.g. Llama-13B) in terms of accuracy with an exception in the OPT family. Models within the same family are typically pre-trained on the same datasets (Biderman et al., 2023; Zhang et al., 2022; Touvron et al., 2023). Thus, this observation suggests that, when trained on identical datasets, the larger models capture a broader set of facts.

Despite being trained on the same data, models might remember different facts. From these results, however, it is not clear if the larger models are subsuming smaller models in their factual knowledge, i.e., are the larger models also correct on the facts that the smaller models are correct on? To assess this, we compute the *subsumption rate* η:

$$\eta(\theta_1|\theta_2, \mathcal{F}) = \frac{|\phi(\theta_1, \mathcal{F}) \cap \phi(\theta_2, \mathcal{F})|}{|\phi(\theta_1, \mathcal{F})|}$$

i.e., the fraction of facts from $\mathcal{F}$ known by smaller model θ_1 that larger model θ_2 also knows. A subsumption rate of ~ 1 indicates that all of the smaller model's knowledge is also contained in the larger model. To ensure a meaningful comparison across scales, we only consider models that were pre-trained using the same training data. Table 1 shows the average subsumption rate (η) between the largest and smallest models in a family, as well as the average accuracy, over all relations for different model families. Interestingly, η is relatively low (< 0.5) for OPT, Pythia and Bloom (i.e., the larger models know less than 50% of what the smaller models know) and only reaching up to 0.8 for Gemma, Llama and Llama-2. Therefore, even though models within each family are trained on the same datasets and generally agree on the relative knowledge of different relations (Figure 6),

[3] 7B parameters is a good reference point since all model families except GPT-NEO-X have models within a gap of $\leq$ 1B parameters: Mistral-7B, Gemma-7B, Llama-7B, Falcon-7B, MPT-7B, OPT-6.7B, GPT-J-6B, Pythia-6.9B, and Bloom-7.1B.

Table 1: Average subsumption rate (η) for different model families over the relations in T-REx-MC. Despite being trained on the same datasets, models of different sizes differ in the specific facts that they know (low η).

	Smallest Model		Largest Model		
Family	#Parameters	Accuracy	#Parameters	Accuracy	η
Llama	7B	0.699	65B	0.836	0.769
Llama-2	7B	0.741	70B	0.846	0.801
Gemma	2B	0.666	7B	0.750	0.710
OPT	125m	0.430	30B	0.588	0.481
Pythia	70m	0.334	12B	0.648	0.403
Bloom	560m	0.410	7.1B	0.548	0.498

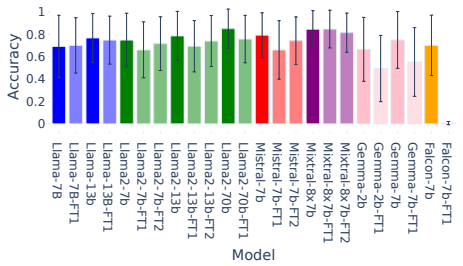


Figure 7: **[Accuracy of base vs chat-finetuned models]** We see that finetuned versions (in lighter shades) obtain lower accuracy across the relations in T-REx-MC than pre-trained models (in darker shades).

there are differences in the knowledge of specific facts they retain from their training data.

Fine-tuning reduces latent knowledge. Finally, we investigate the effects of chat-based fine-tuning on the factual knowledge of models. Base language models are often fine-tuned (using a mix of supervised and reinforcement learning (Ouyang et al., 2022)) to make them better at following instructions. While prior works have shown that this makes the models better at various benchmarks, it's unclear how such fine-tuning affects latent knowledge. Figure 7 illustrates the comparative accuracy of pre-trained models and their fine-tuned counterparts. In almost all cases, the fine-tuned models obtain lower accuracy than their base versions. This suggests that fine-tuning reduces the amount of extractable latent knowledge in the models. A similar observation was also made by Yu et al. (2024). We observe a similar trend using EIC-LKE in Appendix H.6, Figure 15. Additional results on evaluating generated outputs (using 50 tokens) in Figure 16 reveal the same pattern. To further assess if the fine-tuned models are acquiring new knowledge, we compute the subsumption rate between pre-trained and fine-tuned versions (Table 10). We find that most of the latent knowledge in fine-tuned models is already present in base models (high η), thus indicating, that fine-tuned models may not be obtaining additional knowledge.

5 Concluding Discussion

In this work, we investigate a new way to estimate latent factual knowledge from an LLM. Unlike prior approaches that use prompting, our method relies on in-context learning. Our method not only addresses many reliability concerns with prompting, but it also recollects (at time significantly) more factual knowledge than prompting. In contrast to prompting, which requires relationship-specific and LLM-specific prompt engineering, our method can be applied with minimal effort to test factual knowledge of relations across a variety of structured knowledge bases and LLMs. This ability enables us to compare the latent knowledge captured by many different families of open-source LLMs; we expect our results to be of interest to designers of these LLMs. Finally, to design our in-context learning based LKE, we explore the impact of the number and ordering of correct, incorrect, and unknown examples used as inputs; our findings may be of independent interest to developing a better understanding of in-context learning.

A fundamental question posed by our and prior work on estimating latent knowledge in LLMs: *What does it mean for an LLM to know a fact?* Suppose we tried to infer if an LLM knows the capital of Germany using the input "France Paris; Spain Madrid; Germany " and suppose the answer were *Berlin*. What we have learnt is that the LLM knows that the relationship r between Germany and Berlin is similar to that between France and Paris or Spain and Madrid. What we have not learned is whether the LLM knows that the relation r is called "capital" in English or "hauptstadt" in German. The latter is revealed by prompts such as "The capital of Germany is ". But, such prompts don't reveal whether the LLM knows that what Berlin means to Germany is similar to what Paris means to France.

Is one type of knowing facts better than other? It is difficult to answer in general. Neither type of knowing guarantees that the knowledge can be put to use in different contexts and tasks, such as when we ask the LLM where the parliament of Germany is located. Nevertheless, one clear takeaway from our study is related to *how factual knowledge is latently embedded in an LLM*. We show that more factual knowledge can be recollected using in-context learning, i.e., the representations of subjects and objects that share the same relationship, than by prompting with the name of their relationship.

6 Limitations

This study contributes to advancing our understanding of latent factual knowledge in LLMs through an innovative in-context learning approach. However, it is essential to acknowledge the inherent limitations of our work. While the use of in-context learning aims to mitigate the influence of prompt engineering and the reliability issues associated with previous prompting methods, it introduces its own biases based on the selection and formulation of in-context examples. We discus these in detail in Section 3. For example, the choice of which examples to include, their order, and their factual accuracy can influence model responses, and thus these in-context examples must be carefully curated for reliable latent knowledge estimation. Additionally, our study's limitation in testing simple-format facts underlines a critical gap in assessing LLMs' complex reasoning abilities. The knowledge estimation framework employed predominantly hinges on the LLM's capacity to correctly recall or recognize factual information from a given set of triplets or structured prompts. This narrows the scope of evaluation to straightforward factual recall, thereby overlooking the models' capability to engage in more sophisticated cognitive processes such as reasoning, synthesis, and inference, which we leave as open avenues for future work.

References

Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel J. Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Ré. 2023. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering Latent Knowledge in Language Models Without Supervision. *arXiv preprint*. ArXiv:2212.03827 [cs].

I.-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. FacTool: Factuality Detection in Generative AI – A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. *arXiv preprint*. ArXiv:2307.13528 [cs] version: 2.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *Preprint*, arXiv:2309.16797.

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5040–5060.

Xiangkun Hu, Dongyu Ru, Qipeng Guo, Lin Qiu, and Zheng Zhang. 2023a. RefChecker for fine-grained hallucination detection.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S Yu, and Zhijiang Guo. 2023b. Do large language models know about facts? *arXiv preprint arXiv:2310.05177*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Amirhossein Kazemnejad, Mehdi Rezagholizadeh, Prasanna Parthasarathi, and Sarath Chandar. 2023. Measuring the knowledge acquisition-utilization gap

in pretrained language models. *arXiv preprint arXiv:2305.14775*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the Factual Consistency of Abstractive Text Summarization. *arXiv preprint*. ArXiv:1910.12840 [cs].

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Benjamin Newman, Prafulla Kumar Choubey, and Nazneen Rajani. 2022. P-adapters: Robustly extracting factual information from language models with diverse prompts. In *International Conference on Learning Representations*.

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, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2023. On early detection of hallucinations in factual question answering. *arXiv preprint arXiv:2312.14183*.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-Tail: How Knowledgeable are Large Language Models (LLM)? A.K.A. Will LLMs Replace Knowledge Graphs? *arXiv preprint*. ArXiv:2308.10168 [cs].

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

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

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.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*.

Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. LLM lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Kaifeng Yun, Linlu GONG, Nianyi Lin, Jianhui Chen, Zhili Wu, Yunjia Qi, Weikai Li, Yong Guan, Kaisheng Zeng, Ji Qi, Hailong Jin, Jinxin Liu, Yu Gu, Yuan Yao, Ning Ding, Lei Hou, Zhiyuan Liu, Xu Bin, Jie Tang, and Juanzi Li. 2024. KoLA: Carefully benchmarking world knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

JD Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Jeff Huang, Chuyue Sun, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2023. Efficiently programming large language models using sglang. *Preprint*, arXiv:2312.07104.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

Zeyuan Allen Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.

A Dataset

A.1 Creation of Nobel laureates dataset from Wikidata

The Nobel Dataset is a collection of biographical information about all Nobel laureates up until the year 2022, totaling 954 individuals. This dataset was curated using data obtained from Wikidata's querying service[4]. The following attributes are included for each laureate:

- **Name:** The full name of the Nobel laureate.
- **Birth Year:** The year in which the laureate was born.
- **Award Year:** The year(s) in which the laureate was awarded the Nobel Prize.
- **Nature of Award:** A brief description of the reason for the award, including the field of the Nobel Prize (e.g., Physics, Peace).
- **Gender:** The gender of the laureate.

Here are some examples from the Nobel Dataset:

Table 2: Excerpt from the Nobel Dataset

Name	Birth Year	Award Year	Nature of Award	Gender
Albert Einstein	1879	1921	Physics	male
Louis de Broglie	1892	1929	Physics	male
Carl D. Anderson	1905	1936	Physics	male
Polykarp Kusch	1911	1955	Physics	male
Melvin Schwartz	1932	1988	Physics	male
Jerome I. Friedman	1930	1990	Physics	male

A.2 Creation of multiple choices from T-REx: TREx-MC

T-REx (Elsahar et al., 2018) is a large-scale alignment dataset that aligns between Wikipedia abstracts and Wikipedia triples. We have utilized the processed version of T-REx available on HuggingFace [5] for our experiments. We filtered out the relations that have more than 500 facts and 100 unique object entities. The unique objects ensure having 100 feasible multiple-choices for each fact in each relation. We curated 50 relations for our dataset TREx-MC that essentially consists of $<$ *subject, relation, multiple choices* $>$. The multiple choices comprise the correct answer along with 99 other potential choices. We list the 50 relations in Table 3 below.

The following attributes are included in TREx-MC dataset for each relation:

- **Subject** : The subject entity for each fact.
- **Object**: The object entity or the correct answer for each fact.
- **Multiple choices**: The list of other potential choices for each fact.
- **Title** : The Wikipedia title for each fact.
- **Text**: The Wikipedia abstract corresponding to each fact.

Some examples from the T-REx-MC dataset for 2 relations are listed in Table 4

[4] https://query.wikidata.org/

[5] https://huggingface.co/datasets/relbert/t_rex

Table 3: List of 50 relations from T-REx-MC

date of birth	date of death	director	father	spouse	child	sibling	composer	is a tributary of	student of
instance of	cast member	genre	contains the administrative territorial entity	educated at	parent taxon	screen writer	performer	capital	producer
is made by	named after	developer	publisher	founded by	drafted by	has played at	part of the series	manufacturer	production company
mother	cause of death	has subsidiary	creates	point in time	inception	publication date	languages spoken, written or signed	original language of film or TV show	official language
native language	position played on team / speciality	original broadcaster	record label	author	discoverer or inventor	characters	lyrics by	distributed by	home venue

Table 4: Excerpts from T-REx-MC Dataset

Subject	Object	Multiple choices	Title	Text
		Date of birth		
Giovanni Bia	24 October 1968	['26 September 1981', '20 February 1981', ..,'20 September 1960']	Giovanni Bia	Giovanni Bia (born 24 October 1968) is a former Italian footballer...
Brian May	19 July 1947	['24 December 1931', '1 December 1976', ... '23 August 1964]	Brian May	Brian Harold May, CBE (born 19 July 1947) is an English musician...
		Composer		
Mexico Trilogy	Robert Rodriguez	['Fred Schneider', 'Brandy', .., 'Tommaso Traetta']	Mexico Trilogy	The Mexico Trilogy or Mariachi Trilogy (also Desperado Trilogy on some DVD releases) is a series of American..
Chelsea Walls	Jeff Tweedy	['Carmine Coppola', 'Jimmy Chi', ...'Maurice Ravel']	Chelsea Walls	Chelsea Walls is a 2001 independent film directed by Ethan Hawke and released by Lions Gate Entertainment.

B Inference Setup

We experiment with and use three different inference setups:

1. Transformers Based Setup: This setup utilizes the utilities present in the transformers library (Wolf et al., 2020) to obtain the log probabilities for generating the different options.

2. vLLM Based Setup: vLLM ((Kwon et al., 2023)) is a fast inference library for large language models (LLMs). It efficiently manages attention key and value memory using PagedAttention. We observed considerable speed boosts for all 3 LKEs compared to the standard Transformers API.

3. SGLang Based Setup: SGLang (Zheng et al., 2023) is a structured generation language designed for large language models (LLMs). It speeds up LLM interactions and provides enhanced control through tight integration of its frontend language and backend runtime system. SGLang also leverages Radix Attention to cache common components across queries in the KV cache, enabling substantial speedups. We observed sizable speed boosts for IC-LKE and EIC-LKE over vLLM. However, we are constrained by SGLang's limited model family support at the moment, and only utilize it for the Llama, Mistral, and Mixtral families.

C Implementation Details

C.1 IC-LKE

IC-LKE leverages 50 randomly chosen samples from the training data as in-context examples but does not use the relation name. The base prompt is now composed of 50 different examples followed by the name of the entity being tested. A sample would be “Albert Einstein 14 March 1879 Ernest Rutherford 30 August 1871 ... J.J. Thomson 18 December 1856 Max Planck."

The subsequent process is the same as PB-LKE . The process involves adding 100 different choices to the base prompt. A single forward pass is conducted for each sequence, generating log probabilities for the entire sequence. The common part, represented by the tokens for the base prompt is then removed from the tokens of the concatenated base prompt and option resulting in the log probabilities for the option. Similar to PB-LKE , if the option is tokenized into multiple tokens, a single probability value is obtained by multiplying the individual token probabilities. The resulting values are normalized across multiple choices, and the option with the highest probability is selected as the correct answer. We use the vLLM Based & SGLang Based Setup for this LKE.

C.2 EIC-LKE

The EIC-LKE retrieves all 100 samples from our training dataset, initially maintaining them in a single sequence. Then, starting from the 50th training sample, we intersperse our test sample with all the choices every 5 examples. This results in a sequence that includes both the correct and incorrect choices. To determine the probability of each choice, we first use a tokenizer to tokenize all the subjects and choices separately. Then, we combine their token IDs, using a space token to separate the subject and object, and a comma to differentiate between different tuples. After obtaining the sequence’s token IDs, we input these token IDs into a simple forward pass. We use the token length of each subject and object to locate the probability of their corresponding tokens. Finally, we calculate the probability of all the choices by multiplying the probabilities of all their tokens. The resulting values are normalized across the choices, and the choice with the highest probability is selected as the correct answer. We use a vLLM Based Setup for this LKE.

D Different Metrics

The evaluation metric can readily be adapted to existing classification metrics. For example, we introduced the metric Accuracy@K, a calibrated measure that assesses a model’s confidence in its predictions. This metric quantifies how accurately the model identifies knowledge at specified confidence levels for a given relation. We filter the instances that have their confidence levels > threshold K and form the set $\mathcal{D}_K = \{c_i | \, \text{pred}_\theta(c_i) \geq K \; \forall c \in \mathcal{D}\}$. Following this, we use our accuracy measure to compute Accuracy@K for varying values of K, the results of which are shown in Figure 8.

$$\text{acc}_K(\theta, \mathcal{D}_K) \triangleq \frac{\sum\limits_{\langle x,r,y^*,\mathcal{Y}\rangle \in \mathcal{D}_K} \delta\left(y^* = \text{pred}_\theta(x, r, y^*, \mathcal{Y})\right)}{|\mathcal{D}_K|} \tag{5}$$

.

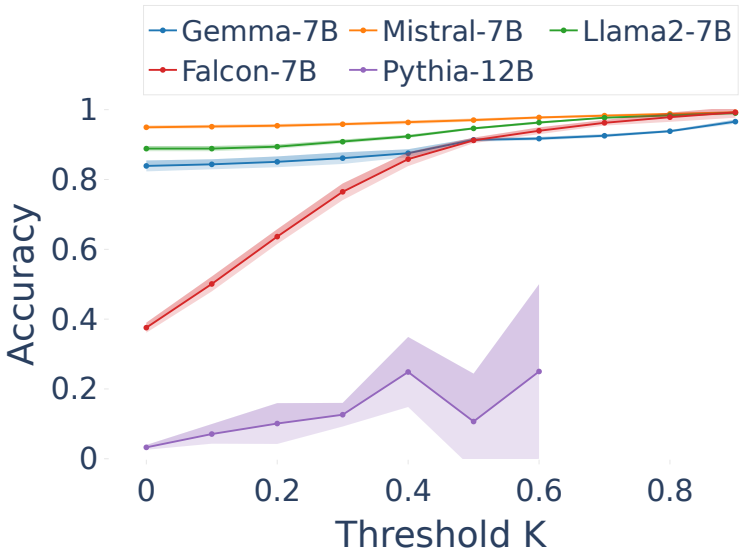
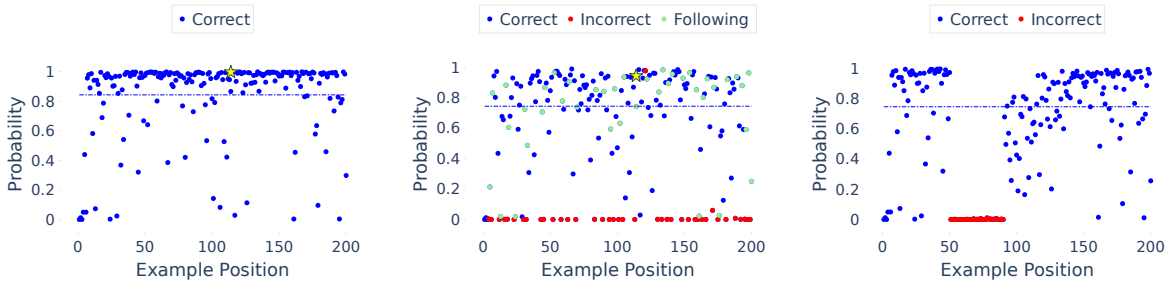


Figure 8: **Accuracy@K for different models** We evaluated five models on the Nobel dataset, which consists of 50 examples. Each model's performance was measured using the Accuracy@K metric at various thresholds.

E Probabilities of objects in sequence

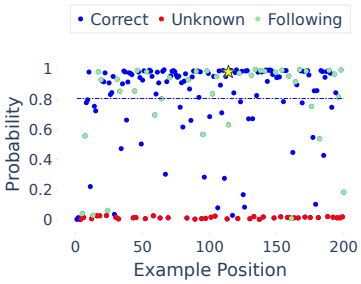
We first consider 200 correct examples (subject-object pairs) and report the *absolute* generation probability of objects in corresponding examples. We showed the results for Llama2-7B, Falcon-7B, Gemma-7B, and Pythia-12B in Figure 11, Figure 9 and Figure 10. Figure 11a, Figure 9a, and Figure 10a illustrates the probability of each object at various sequence positions; Figure 11b, Figure 9b, and Figure 10b shows the impact on probabilities after substituting 40 objects dispersed within the sequence with incorrect ones. Figure 11c, Figure 9c, and Figure 10c visualizes the effect of replacing objects at simultaneous positions. Figures 11d, Figure 9d, Figure 10d, Figure 11e, Figure 9e, and Figure 10e present the outcomes of using unknown subject-object pairs as replacements. We used a horizontal dashed line showing an average probability of the correct examples. The yellow star notated the example at position 114 in the sequence.



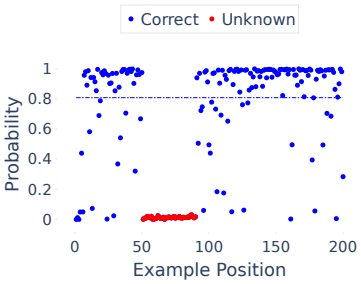
(a) Correct Subject-Object Pairs

(b) Distributed Replace

(c) Simultaneous Replace



(d) Distributed Replace



(e) Simultaneous Replace

Figure 9: **[Analysis of object probability in one sequence of Nobel laureate data using Llama2-7b]**

F Efficient In Context learning based LKE (EIC-LKE)

We improve the efficiency of IC-LKE to perform knowledge extraction of multiple test facts in a single prompt. Leveraging the context length in LLMs, the efficient version, namely EIC-LKE, places multiple test facts surrounded by training facts into the same prompt. We measure the object probability of each of the (alternative) test facts in the seuqence to determine whether the LLM assigns higher probability to the correct fact than the others.

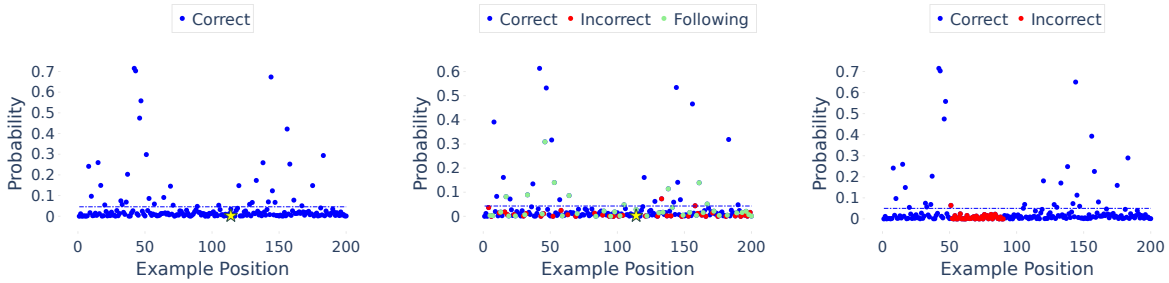
Example 2. Considering the training facts in Example 1, we evaluate two test choices (highlighted in yellow) for the birth-year relation: $\langle$ Einstein, birth-year 1879, $\mathcal{Y}_1 = \{1880\}\rangle$ and $\langle$ Louis birth-year, 1892, $\mathcal{Y}_2 = \{1850\}\rangle$ *using two prompts instead of four as in IC-LKE.*

Feynman 1918 *Einstein 1879** *Heisenberg 1901* *Louis 1850*

Feynman 1918 *Einstein 1880* *Heisenberg 1901* *Louis 1892**

G Details about the human-generated prompts and machine-mined prompts

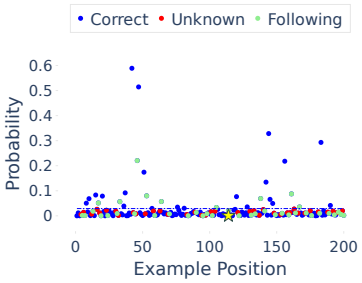
We list the used human-generated and machine-mined prompts from (Jiang et al., 2020) in Table 5 with subjects denoted as <'head'>.



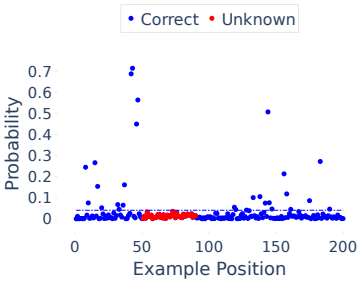
(a) Correct Subject-Object Pairs

(b) Distributed Replace

(c) Simultaneous Replace

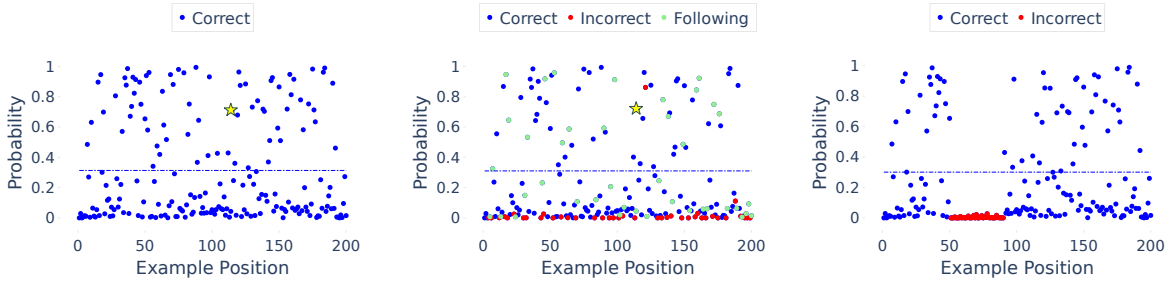


(d) Distributed Replace



(e) Simultaneous Replace

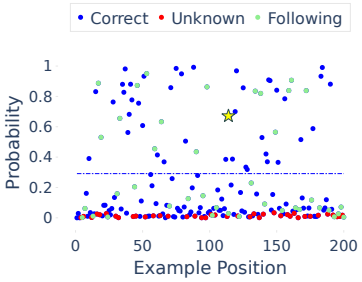
Figure 10: **[Analysis of object probability in one sequence of Nobel laureate data using Pythia-12B]**



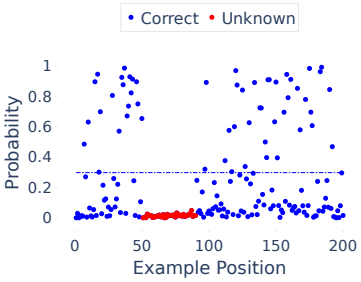
(a) Correct Subject-Object Pairs

(b) Distributed Replace

(c) Simultaneous Replace



(d) Distributed Replace



(e) Simultaneous Replace

Figure 11: **[Analysis of object probability in one sequence of Nobel laureate data using Falcon-7B]**

Table 5: Templates for Selected Relations

Relation Name	Index	HGP Template	MMP Template
Instance of	1	{subject} means	{subject} is a small
	2	{subject} is one	{subject} and liberal
	3	{subject} is a	{subject} artist
Genre	1	{subject} is playing music	{subject} series of
	2	{subject} play	{subject} favorite
	3	{subject} performs	{subject} is an american
Position played on team / speciality	1	{subject} plays in position	{subject} substitutions :
	2	{subject} plays at position	{subject} substitutes :
	3	{subject} is in the position	
Original language of film/TV show	1	The original language of {subject} is	{subject} a. r. rahman
	2	The source language of {subject} is	
	3	The default language of {subject} is	
Capital	1	The capital of {subject} is	{subject} united states embassy in
	2	The capital city of {subject} is	{subject} representative legislature
	3	Its capital {subject} is	{subject} rock band from
Native language	1	{subject} is a native language of	{subject} descent
	2	The mother tongue of {subject} is	{subject} speak the
	3	{subject} means	{subject} population or a widely spoken
Named after	1	{subject} is named after	{subject} and produces
	2	{subject} is named for	{subject} variety of standard)
	3	{subject} is called after	{subject} official
Official language	1	The official language {subject} is	{subject} professor of
	2	{subject} is	{subject} is the official language in
	3	{subject} is officially	{subject} is the official language spoken in
Developer	1	{subject} is developed by	{subject} was developed by
	2	{subject} is created by	{subject} 2008
	2	{subject} is designed by	{subject} references external links
Original broadcaster	1	{subject} was originally aired on	{subject} premiered on
	2	{subject} was originally broadcast on	{subject} aired on
	3	{subject} was originally shown in	{subject} 2021
Record label	1	{subject} is signed to	{subject} signed with
	2	{subject} is a recording artist for	{subject} sohmed a recording contract with
	3	{subject} is a recording artist on	{subject} released by
Manufacturer	1	{subject} is represented by music label	{subject} attributed to the
	2	{subject} is represented by the record label	{subject} 113
	3	{subject} is represented by	{subject} cedar point

H Additional results

H.1 Model Name Simplification

We list all the models and their simplified names we evaluated in the paper in Table 6.

H.2 Additional results on baseline comparison

We compare IC-LKE and EIC-LKE on 12 relations from T-REx-MC: *capital, named after, developer, manufacturer, genre, instance of, native language, original broadcaster, language spoken written or signed, original language of film / TV show, official language, position played on team/speciality.* We chose those 12 relations from T-REx-MCthat are found to be in common with (Jiang et al., 2020) where they define the templates for HGP and MMP. We evaluated 4 models (Mistral-7B, Llama-7B, Falcon-7B, and Pythia-12B) and showed all the results in Figure 12.

H.3 Full order of models and relations

We evaluated 49 models on 50 relations by our IC-LKEand EIC-LKE. Table 7 shows the ordered models by the average accuracy of all the 50 relations. Table 8 shows the ordered relations by the average accuracy of all the 49 models.

H.4 Full evaluation on EIC-LKE

We evaluated all the pre-trained models using EIC-LKE, but didn't evaluate GPT-NEOX-20B due to the limitation of its context window size. Figure 13 shows the heatmap of models vs. relations, ordered in the same way as in Figure 5.

H.5 Relation accuracy correlation of all the pre-trained models

In Table 14, we show the Pearson correlation coefficients between each model pair's performance across the 50 relations.

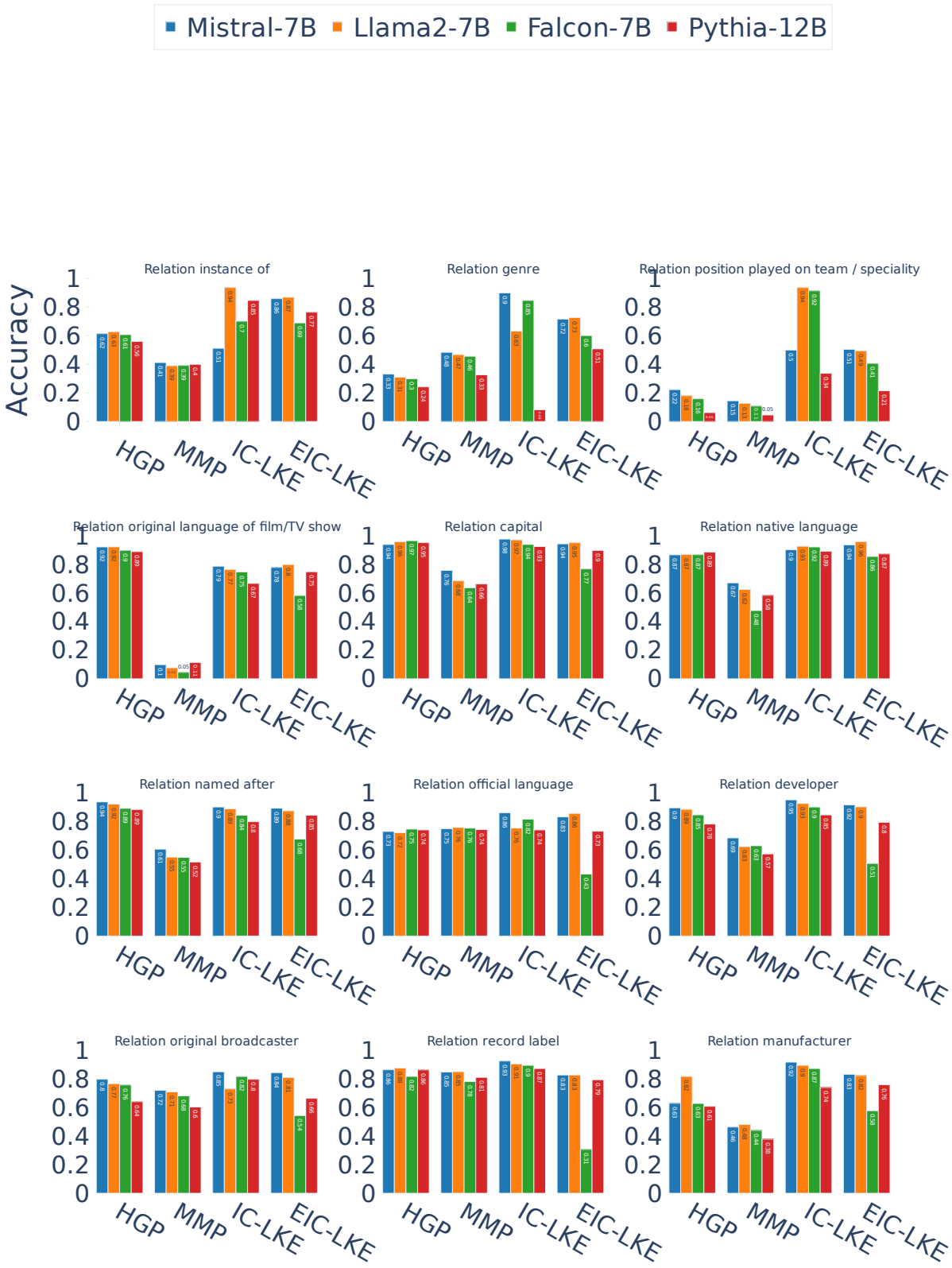


Figure 12: Accuracy for different latent knowledge estimators on all 12 relations

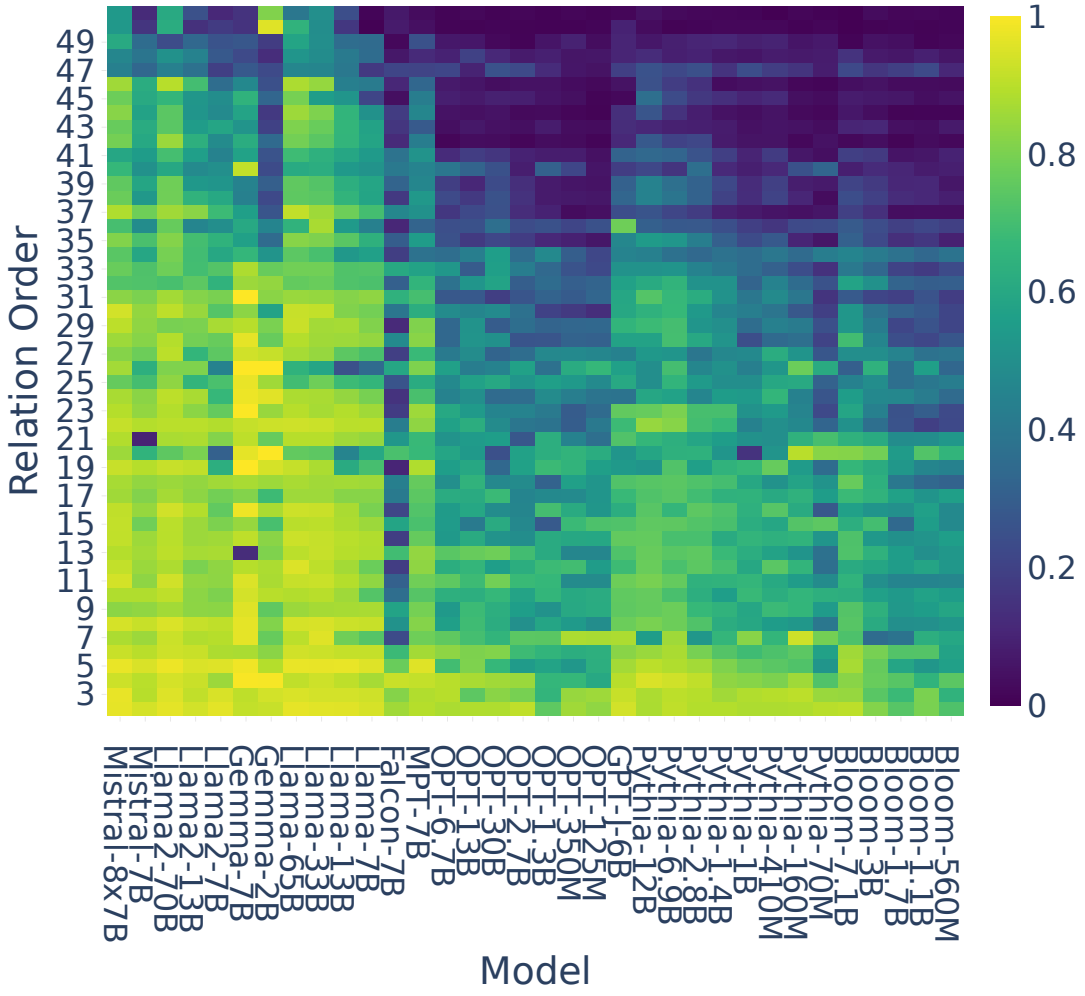


Figure 13: Accuracy for 35 pre-trained LLMs on the 50 different relations in T-REx-MC, evaluated by EIC-LKE.

Table 6: Model Name Simplifications

Original Name	Simplified Name in Paper
mistral-mixtral-8x7B-v0.1	Mixtral-8x7B
Nous-Hermes-2-Mixtral-8x7B-SFT	Mixtral-8x7B-FT1
Nous-Hermes-2-Mixtral-8x7B-DPO	Mixtral-8x7B-FT2
mistral-7b	Mistral-7B
mistral-instruct-7b	Mistral-7B-FT1
openhermes-2.5-mistral-7b	Mistral-7B-FT2
llama2-70b	Llama2-70B
llama2-70b-chat	Llama2-70B-FT1
llama2-13b	Llama2-13B
llama2-13b-chat	Llama2-13B-FT1
vicuna-13b-v1.5	Llama2-13B-FT2
llama2-7b	Llama2-7B
llama2-7b-chat	Llama2-7B-FT1
vicuna-7b-v1.5	Llama2-7B-FT2
gemma-7b	Gemma-7B
gemma-7b-it	Gemma-7B-FT1
gemma-2b	Gemma-2B
gemma-2b-it	Gemma-2B-FT1
llama-65b	Llama-65B
llama-33b	Llama-33B
llama-13b	Llama-13B
vicuna-13b-1.3	Llama-13B-FT1
llama-7b	Llama-7B
vicuna-7b-1.3	Llama-7B-FT1
falcon-7b	Falcon-7B
falcon-instruct-7b	Falcon-7B-FT1
mpt-7b	MPT-7B
gpt-neox-20b	GPT-NEOX-20B
opt-30b	OPT-30B
opt-13b	OPT-13B
opt-6.7b	OPT-6.7B
opt-2.7b	OPT-2.7B
opt-1.3b	OPT-1.3B
opt-350m	OPT-350M
opt-125m	OPT-125M
gpt-j-6b	GPT-J-6B
pythia-12b	Pythia-12B
pythia-6.9b	Pythia-6.9B
pythia-2.8b	Pythia-2.8B
pythia-1.4b	Pythia-1.4B
pythia-1b	Pythia-1B
pythia-410m	Pythia-410M
pythia-160m	Pythia-160M
pythia-70m	Pythia-70M
bloom-7.1b	Bloom-7.1B
bloom-3b	Bloom-3B
bloom-1.7b	Bloom-1.7B
bloom-1.1b	Bloom-1.1B
bloom-560m	Bloom-560M

Table 7: Model Performance Comparision

Model	Average Accuracy	Standard Deviation
Llama2-70B	0.8511	0.17591
Mixtral-8x7B-SFT	0.84765	0.16919
Mixtral-8x7B	0.84605	0.16653
Llama-65B	0.84185	0.17528
Mixtral-8x7B-DPO	0.81535	0.17580
Llama-33B	0.81255	0.19088
Mistral-7B	0.79310	0.20000
Llama2-13B	0.78692	0.21892
Llama-13B	0.76845	0.21796
Llama2-70B-chat	0.75815	0.21272
Llama2-7B	0.74945	0.24069
Vicuna-13B	0.74940	0.21427
Gemma-7B	0.74717	0.25668
Openhermes-2.5	0.74365	0.21241
Vicuna-13B-2	0.74080	0.22807
Vicuna-7B-2	0.71695	0.24016
Falcon-7B	0.70190	0.27052
Vicuna-7B	0.70155	0.24724
Llama2-13B-chat	0.69387	0.22966
Llama-7B	0.69260	0.27912
Gemma-2B	0.66600	0.28627
GPT-NEOX-20B	0.66145	0.30972
Llama2-7B-chat	0.66130	0.24996
Mistral-instruct-7B	0.66120	0.26173
MPT-7B	0.64545	0.30638
Pythia-12B	0.63325	0.32412
OPT-6.7B	0.62110	0.31313
GPT-J-6B	0.60965	0.32319
OPT-13B	0.60845	0.31017
Pythia-6.9B	0.59185	0.32359
Bloom-7.1B	0.58270	0.31404
OPT-30B	0.57925	0.31813
Pythia-2.8B	0.57580	0.32773
Pythia-1.4B	0.56330	0.33600
Gemma-7B-instruct	0.55327	0.30689
OPT-2.7B	0.55109	0.33260
Bloom-3B	0.54375	0.29199
Pythia-1B	0.54220	0.31560
OPT-1.3B	0.53610	0.33335
Bloom-1.1B	0.51115	0.29346
OPT-350M	0.50735	0.30716
Gemma-2B-instruct	0.49474	0.29628
Pythia-410M	0.47995	0.29598
Bloom-1.7B	0.47660	0.29658
OPT-125M	0.45195	0.29330
Bloom-560M	0.38465	0.28747
Pythia-160M	0.37145	0.28505
Pythia-70M	0.31260	0.27404
Falcon-instruct-7B	0.00605	0.01459

Table 8: Relations and their average accuracies

Order	Relation	Average Accuracy
1	publication date	0.992071428571429
2	inception	0.983214285714286
3	point in time	0.975714285714286
4	drafted by	0.922214285714286
5	native language	0.8825
6	production company	0.873428571428571
7	languages spoken, written or signed	0.865071428571429
8	performer	0.831142857142857
9	has played at	0.826642857142857
10	capital	0.815857142857143
11	is made by	0.815357142857143
12	producer	0.794714285714286
13	record label	0.794571428571429
14	named after	0.791071428571429
15	developer	0.786928571428571
16	publisher	0.7835
17	original broadcaster	0.781214285714286
18	cast member	0.777
19	home venue	0.771714285714286
20	has subsidiary	0.754142857142857
21	manufacturer	0.749928571428571
22	screenwriter	0.732285714285714
23	contains the administrative territorial entity	0.7255
24	creates	0.721214285714286
25	official language	0.709857142857143
26	mother	0.697857142857143
27	part of the series	0.692214285714286
28	founded by	0.684714285714286
29	original language of film or TV show	0.6825
30	date of birth	0.668857142857143
31	date of death	0.641594184576485
32	instance of	0.588990518331226
33	position played on team / speciality	0.537642857142857
34	genre	0.536
35	distributed by	0.522785714285714
36	parent taxon	0.488428571428571
37	director	0.432928571428571
38	author	0.331285714285714
39	father	0.309214285714286
40	educated at	0.306285714285714
41	characters	0.282857142857143
42	composer	0.276785714285714
43	child	0.259142857142857
44	lyrics by	0.258428571428571
45	sibling	0.250285714285714
46	spouse	0.238785714285714
47	is a tributary of	0.212142857142857
48	cause of death	0.206
49	discoverer or inventor	0.173142857142857
50	student of	0.123357142857143

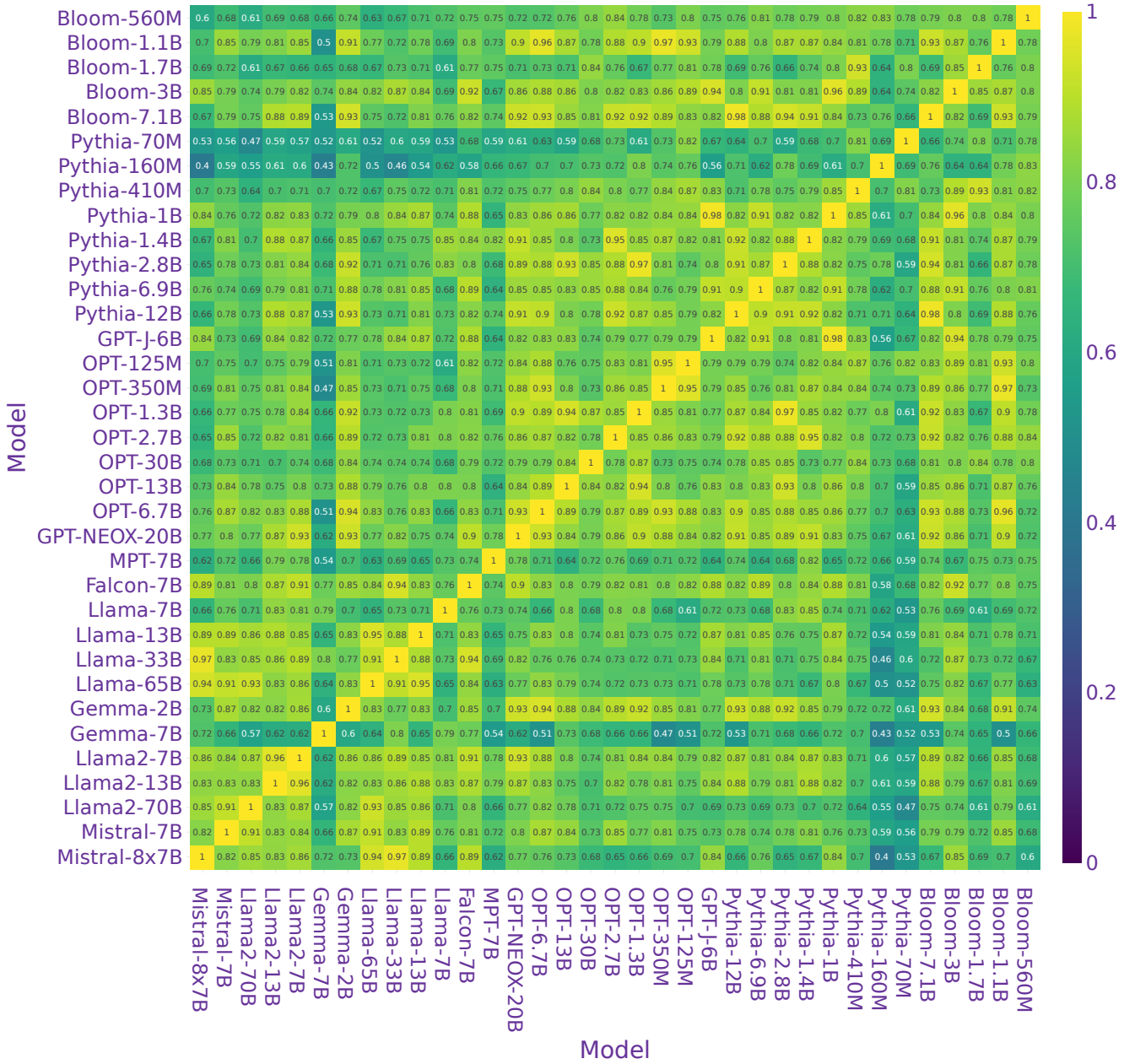


Figure 14: **[Pearson Correlation Coefficients Between All Pre-trained Models]** We calculated the Pearson correlation coefficients for each model pair among 49 models across 50 relations.

Order/Model	**Mistral-8x7B**	**Mistral-7B**	**Llama2-70B**	**Llama2-13B**	**Llama2-7B**	**Gemma-7B**	**Gemma-2B**
1	publication date	point in time	point in time	publication date	publication date	point in time	point in time
2	point in time	date of death	inception	point in time	inception	inception	inception
3	inception	publication date	publication date	inception	point in time	publication date	publication date
…	……	……	……	;	;	;	
48	discoverer or inventor	discoverer or inventor	student of	discoverer or inventor	educated at	date of death	position played on team / speciality
49	cause of death	cause of death	cause of death	cause of death	cause of death	instance of	discoverer or inventor
50	student of	student of	is a tributary of	student of	student of	date of birth	student of
Order/Model	**Llama-65B**	**Llama-33B**	**Llama-13B**	**Llama-7B**	**Falcon-7B**	**MPT-7B**	**GPT-NEOX-20B**
1	publication date	publication date	publication date	publication date	point in time	publication date	publication date
2	point in time	point in time	point in time	point in time	inception	inception	inception
3	inception	inception	inception	inception	publication date	point in time	date of death
…	……	……	……	;	;	;	
48	discoverer or inventor	discoverer or inventor	discoverer or inventor	student of	discoverer or inventor	student of	discoverer or inventor
49	cause of death	cause of death	cause of death	instance of	is a tributary of	is a tributary of	lyrics by
50	student of	student of	student of	date of birth	student of	educated at	student of
Order/Model	**OPT-30B**	**OPT-13B**	**OPT-6.7B**	**OPT-2.7B**	**OPT-1.3B**	**OPT-350M**	**OPT-125M**
1	inception	publication date	publication date	inception	publication date	inception	inception
2	publication date	inception	inception	publication date	inception	publication date	publication date
3	point in time	point in time	date of death	point in time	drafted by	point in time	point in time
…	……	……	……	;	;	;	
48	position played on team / speciality	composer	lyrics by	discoverer or inventor	director	is a tributary of	student of
49	discoverer or inventor	student of	student of	student of	student of	spouse	is a tributary of
50	director	date of birth	discoverer or inventor	instance of	date of birth	student of	educated at
Order/Model	**GPT-J-6B**	**Pythia-12B**	**Pythia-6.9B**	**Pythia-2.8B**	**Pythia-1.4B**	**Pythia-1B**	**Pythia-410M**
1	inception	point in time	publication date	publication date	publication date	publication date	publication date
2	publication date	publication date	inception	inception	inception	inception	inception
3	point in time	inception	point in time	point in time	date of death	point in time	drafted by
…	……	……	……	……	……	……	……
48	lyrics by	lyrics by	lyrics by	student of	discoverer or inventor	lyrics by	student of
49	student of	genre	director	date of birth	lyrics by	student of	discoverer or inventor
50	date of death	director	date of death	lyrics by	director	date of death	is a tributary of
Order/Model	**Pythia-160M**	**Pythia-70M**	**Bloom-7.1B**	**Bloom-3B**	**Bloom-1.7B**	**Bloom-1.1B**	**Bloom-560M**
1	publication date	publication date	publication date	publication date	publication date	publication date	publication date
2	point in time	point in time	inception	inception	inception	inception	inception
3	date of death	native language	date of death	point in time	point in time	date of death	point in time
…	……	……	……	……	……	……	……
48	student of	official language	lyrics by	is a tributary of	screenwriter	is a tributary of	is a tributary of
49	capital	instance of	student of	spouse	student of	spouse	director
50	director	date of death	spouse	student of	spouse	student of	student of

Table 9: Top 3 and Bottom 3 relations for each pre-trained model

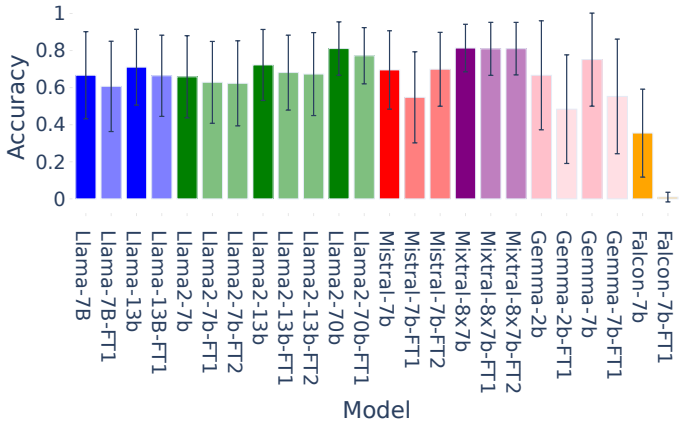


Figure 15: **[Base vs chat-finetuned models]** We see that finetuned versions (depicted in lighter shades) obtain lower accuracy across the relations in T-REx-MC than pre-trained models (shown in darker shades), evaluated by IC-LKE.

Family	Model Type	Accuracy	Model Type	Accuracy	η
Llama-7B	Base	0.699	FT-1	0.693	0.779
Llama-13B	Base	0.770	FT-1	0.735	0.854
Llama2-7B	Base	0.741	FT-1	0.712	0.808
Llama2-7B	Base	0.741	FT-2	0.664	0.790
Llama2-13B	Base	0.771	FT-1	0.748	0.831
Llama2-13B	Base	0.771	FT-2	0.692	0.801
Llama2-70B	Base	0.846	FT-1	0.739	0.811
Mistral-7B	Base	0.793	FT-1	0.639	0.793
Mistral-7B	Base	0.793	FT-2	0.750	0.869
Mixtral-7Bx8	Base	0.832	FT-1	0.835	0.928
Mixtral-7Bx8	Base	0.832	FT-2	0.817	0.911
Gemma-2B	Base	0.666	FT-1	0.488	0.577
Gemma-7B	Base	0.749	FT-1	0.511	0.557

Table 10: Average subsumption rate (η) for base models and fine-tuned models over the relations in T-REx-MC. Despite being fine-tuned on smaller datasets, fine-tuned models (low η). The results are based on IC-LKE.

H.6 Impact of finetuning

We show the results evaluated by EIC-LKE for all the pre-trained models and fine-tuned models in Figure 15 from the relations in TREx-MC, which also conveys the message about reduced knowledge in fine-tuned models. We also show the results for the average subsumption rate (η) for base models and fine-tuned models over the relations in T-REx-MC.

H.7 Evaluation of Generated Output

We also evaluated the generated output, where we used greedy searching(temperature=0), and asked both pre-trained and fine-tuned models to generate 50 tokens using different prompts from HGP and MMP. Following this, we checked for the presence of the ground truth in the generated output of 50 tokens. The generation is correct if present, and incorrect otherwise, then we compute the generation accuracy on the test dataset. We report the average generation accuracy based on 12 relations and the HGP/MMP templates shown in Table 5.

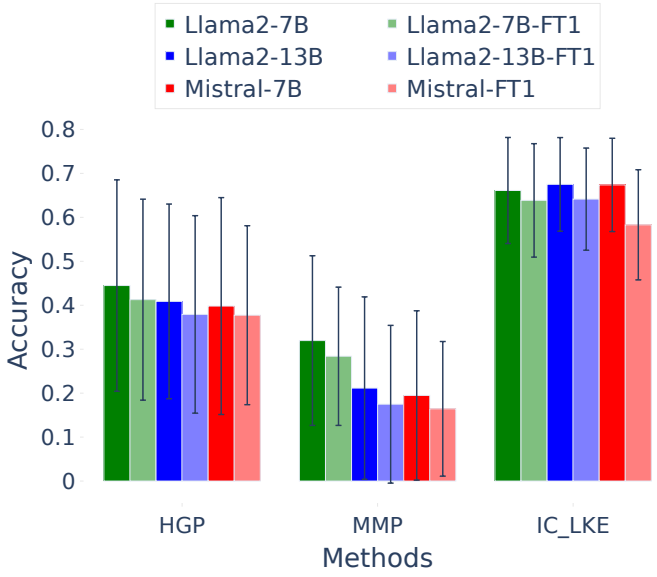


Figure 16: Accuracies computed over generated outputs (50 tokens) for pre-trained and fine-tuned models using HGP, MMP, and IC-LKE.