KGLens 🔎 : A Parameterized Knowledge Graph Solution to Assess What an LLM Does and Doesn't Know

Anonymous ACL submission

Abstract

Measuring the alignment between a Knowledge Graph (KG) and Large Language Models (LLMs) is an effective method to assess the factualness and identify the knowledge blind spots of LLMs. However, this approach encounters two primary challenges including the translation of KGs into natural language and the efficient evaluation of these extensive and complex structures. In this paper, we present KGLENS—a novel framework aimed at measuring the alignment between KGs and LLMs, and pinpointing the LLMs' knowledge deficiencies relative to KGs. KGLENS features a graph-guided question generator for converting KGs into natural language, along with a carefully designed sampling strategy based on parameterized KG structure to expedite KG traversal. We conducted experiments using three domain-specific KGs from Wikidata, which comprise over 19,000 edges, 700 relations, and 21,000 entities. Our analysis across eight LLMs reveals that KGLENS not only evaluates the factual accuracy of LLMs more rapidly but also delivers in-depth analyses on topics, temporal dynamics, and relationships. Furthermore, human evaluation results indicate that KGLENS can assess LLMs with a level of accuracy nearly equivalent to that of human annotators, achieving 95.7% of the accuracy rate.

1 Introduction

The factualness of Large Language Models (LLMs) is crucial for their reliability and utility in various applications. Nonetheless, studies have shown that LLMs can produce information that is nonfactual, hallucinated, or outdated (Perez et al., 2022; Ji et al., 2023; Lee et al., 2022; Wang et al., 2021).

To evaluate the factualness of LLMs, researchers have developed a variety of methodologies, broadly categorized into fact-checking (Thorne et al., 2018; Augenstein et al., 2023) and fact-answering approaches (Petroni et al., 2020; Press et al., 2022; Dhingra et al., 2022). Despite these advancements,

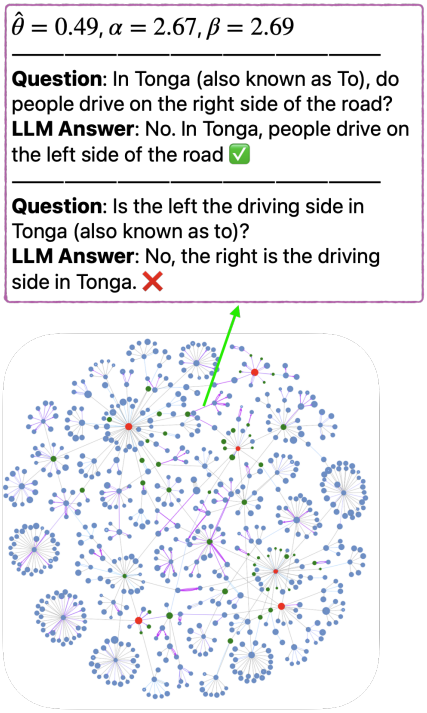


Figure 1: Parameterized knowledge graph updated by KGLENS. The edge color is associated with the expectation of $\theta \sim Beta(\alpha, \beta)$, denoting an LLM's deficiency to the corresponding fact.

several challenges persist. For facts-checking, distinguishing faithful and unfaithful facts is different from evaluating the generation of factual content. For facts-answering, scaling up the evaluation is challenging due to the expensive nature of the annotation process. And once these evaluation datasets are published, it is hard to exclude the test examples from the web-crawled LLM pretraining corpus (Deng et al., 2023). Finally, both fact-checking and fact-answering approaches assess LLMs on an instance by instance basis, overlooking the relationships among facts.

In contrast, knowledge graph (KG) encompasses a vast amount of facts, maintains connections among these facts, and can be easily updated. Once an LLM's knowledge reliability of each KG edge is

evaluated, the knowledge blind spots can be easily identified (Figure 1). Furthermore, the evaluation results for each edge can be aggregated at various levels (e.g., over time, by predicate type), offering valuable insights for model improvement.

However, there are several challenges for KG-based LLM evaluation. The first is transforming KG into natural language. Petroni et al. (2019) proposed to transform KG triplet into text-cloze task but the formulated sentences are ambiguous and unnatural. Jiang et al. (2020) alleviate this issue by mining the relation words from the web for each subject-object pair, which is impractical for large graph. Another challenge is the efficiency of the evaluation. KGs are typically large. And evaluating the robustness of an LLM's knowledge may necessitate multiple evaluation rounds using the same KG, as an LLM may respond differently to the same query.

In this study, we present a novel framework named KGLENS (Figure 2) to assess LLMs' knowledge with KG and identify the *knowledge blind spots* of LLMs. By 'knowledge blind spots', we mean specific areas or topics where the LLM's understanding is lacking, potentially leading to failures in accurately answering questions related to such knowledge. KGLENS features a graph-guided question generator for converting KGs into natural language with GPT-4 (OpenAI, 2023). We design two types of questions to support both the facts answering and facts checking, where the question type is controlled by the graph structure. We also include the entity aliases during the question generation to provide additional context and reduce the entity ambiguity. Our experiment results show that 97.7% of our generated questions are understandable to human annotators.

To improve the evaluation efficiency, we introduce a parameterized knowledge graph (PKG), where each KG edge is augmented with a beta distribution, serving as an indicator of the LLM's deficiency on that specific edge. Navigation through the PKG involves sampling and selecting the top-ranked edges globally based on their deficiency. In this way, when an LLM is unable to provide a satisfactory response to a question, the KG structure enables us to pinpoint the relevant source edge and entities. This information can then be used to update the PKG, and the process can be iteratively applied until adequate coverage is achieved. Our simulation experiments show that our sampling method with PKG is more efficient than random sampling and straightforward iteration methods. In our experiments, we collected three domain-specific KG from Wikidata, encompassing over 700 relations and 21K entities. Our evaluation of 8 LLMs shows KGLENS is capable not only of accessing the factualness of LLMs but also of pinpoints LLMs' knowledge deficiencies relative to KGs in different levels (e.g., temporal and topics). Human evaluation indicates that KGLENS can assess LLMs with a level of accuracy nearly equivalent to that of human annotators with 95.7% accuracy rate.

2 Method

Our framework is shown in Figure 2. In this section, we will introduce the parameterized KG, graph-guided question generator, and evaluation metrics.

2.1 Parameterized Knowledge Graph

A knowledge graph $\mathcal{G}$, is a set of triplets $\{(s_j, p_j, o_j)\}_{j=1\cdots i}$ where each tuple describes a relationship (predicate) p_j between a subject s_j and an object o_j.

Intuitively, if an LLM failed in answering a question, there is a higher chance that the LLM also lacks knowledge of the related topics. To reflect this inductive bias, we propose a parameterized KG, by augmenting each edge (s_j, p_j, o_j) of the original KG with an additional error probability θ_j reflecting the probability that an LLM may fail on this edge. We use beta distribution to model θ due to the conjugacy between Bernoulli distribution and Beta distribution.

$$\theta_j \sim Beta(\alpha_j, \beta_j), \tag{1}$$

The prior of each θ_j is set to $Beta(1, 1)$.

The estimation of the posterior $\{\alpha_j, \beta_j\}_{\forall j}$ is done in an iterative manner based on the outcome from the LLM. This process consists two main stages: 1) edge ranking and sampling and 2) parameter updating.

Edge sampling The edge sampling process favors the edges with larger θ values. During the graph traversal, we sample top-n challenging edges ranked by the sampled θ values from the PKG. The top-n edges are then sent to LLM for examination and verification. The signal regarding the correctness of the output from LLM is collected for each of the edges accordingly.

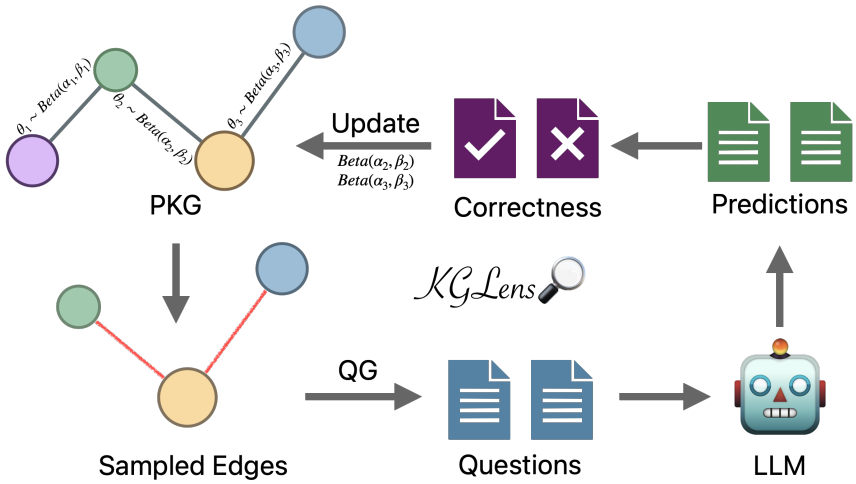


Figure 2: KGLENS Framework. Here we illustrate this framework with a simple KG example. KGLENS starts from the PKG initialization, where each edge is augmented with a beta distribution. Then a batch of edges is sampled based on the edge probability θ. After that, questions are generated from these edges and an LLM will be examined with question answersing task. Then we update the beta distribution of PKG edges based on the QA results. We iterate this process until the running metrics are converged.

Parameter estimation and updating After the signal is collect, the α and β is updated based on the new observation of whether the response from LLM is correct, following the standard Beta distribution posterior updates.

In order to account for the high correlation in error probability among the connected edges, we have additionally propagate the signal to the neighboring edges. Specifically, the signal gathered from p_j is propagated to both the incoming and outgoing edges that are connected to node s_j and o_j. To optimize the computational process, we restrict the signal propagation to one degree. Specifically,

$$\alpha_j = \alpha_j + \mathbb{I}(\text{response is incorrect}) + M_j, \quad (2)$$
$$\beta_j = \beta_j + \mathbb{I}(\text{response is correct}) + N_j, \quad (3)$$

where M_j = |incorrect neighborhood edges| and N_j = |correct neighborhood edges|.

2.2 Graph-guided Question Generation

We use GPT-4 to transform the sampled edge K_i into the natural questions with few-shot in-context learning. The prompts and demonstrations are shown in Appendix 8.4. We design two types of questions for KGLENS: Yes/No Questions (judgement) and Wh-Questions (generative), where the question type is controlled by the graph structure (out degree). In addition, to reduce the ambiguity of entities, we provide the entity alias for question generation.

2.2.1 Yes/No Questions

Each KG edge can be transformed into a question by asking if the subject's relation is the object. But in this way, the answer would always be *Yes* for all the edges. To formulate hard negative examples, we build a ground truth answer set $\mathbf{T_j}$ for each (s_j, p_j), and the candidate answer set $\mathbf{C_j}$ for each p_j. Both $\mathbf{T_j}$ and $\mathbf{C_j}$ are derived from the full Wikidata knowledge graph to ensure the completeness. Then, for a tuple $\{(s_j, p_j, o_j)\}$, we use o_j to constitute the *Yes* question, and sample a random o_x from $\mathbf{C_j} - \mathbf{T_j}$ to formulate the *No* question. Considering our QG process is on-the-fly during the evaluation, KGLENS can formulate different QA pairs for the same fact. The sampling rate between yes and no question is evenly split, with a 50-50 distribution.

2.2.2 Wh-Questions

Another type of question is to ask the LLMs to generate the object/objects given the subject and the predicate, where the questions usually begin with when/where/who/what. This question type is more challenging but cannot be applied to all edges. For example, there may be hundreds of correct objects for a Wh-Question and it makes no sense to check if a model can enumerate all of them correctly. In KGLENS, we opt to generate Wh-Questions only when the out degree of an entity is less than 10. Otherwise, the Yes/No Questions prompt is adopted.

KG	Active Edges	Dead Edges	Nodes	Predicates
Country	7844	9441	12760	338
NBA	2689	1158	805	57
Movies	8704	3053	7965	340

Table 1: Statistics of the testing knowledge graphs.

2.3 QAV: Question Answering Verification

We design the QA testing under two different difficulty levels: EASY and HARD. For EASY testing, we only use Yes/No Questions to test the LLMs. For HARD testing, we generate each type of question at a 50% chance. We use few-shot in-context learning to test the LLMs.

To verify the response, we guide the LLMs to generate either "Yes" or "No" at the beginning of the response for Yes/No Questions and subsequently generate accompanying explanations. This approach facilitates a straightforward verification process by examining the correspondence of the initial word. For Wh-Questions, we instruct the LLM to list all the correct answers. In this case, the assessment of the answer cannot be done by string matching. Therefore, we employ a GPT-4 model to check the correctness of a response given the question, the ground truth objects and their aliases. The prompts are listed in Appendix 8.4.

2.4 Metrics

To measure the alignment between KGs and LLMs, here we introduce two edge-level metrics.

Win rate. For each edge, LLM wins if the number of successes surpasses the number of failures. The win rate signifies the portion of winning edges out of all the examined edges.

Zero sense rate. An LLM would has zero sense about an edge (fact) if the model has never answered the edge correctly. The zero sense rate signifies the portion of edges with zero sense.

Based on the definition above, win rate is the portion of edges that an LLM has higher chance to answer them correctly. Zero sense rate is the portion of edges that an LLM always fails to answer. Higher win rate tells us that the LLM is more reliable for the testing KG. Higher zero sense rate tells us the LLM knows less about the testing KG.

3 Experiments

In this paper, we develop three domain-specific KGs using Wikidata to evaluate the knowledge accuracy and reliability of two widely used LLM APIs (GPT-3.5-turbo and GPT-4), two legacy LLMs (Babbage-002 and Davinci-002), together with an preview version of GPT-4 (GPT-4-1106-preview). We also evaluated three open source LLMs including Vicuna-33b-v1.3 (Chiang et al., 2023), Xwin-LM-13B-V0.2 (Team, 2023), and Yi-34B-Chat [1]. We narrow down the evaluation to English language in three domains: country, NBA (The National Basketball Association), and movie. It should be noted that KGLENS is not design for any specific domain or language.

3.1 Building Knowledge Graphs

We prepare the testing knowledge graphs with Wikidata Query Web Service [2] in three topics: country, NBA, and movie. The country KG includes knowledge about 16 countries. The NBA KG contains the knowledge related to 30 NBA teams. And the movies are sampled from films after 2015.

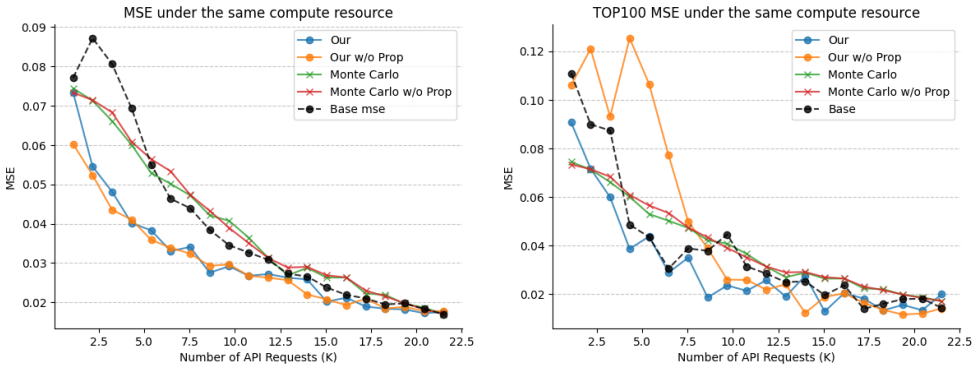
The statistics of our KGs are shown in Table 1. The term **"dead edges"** refers to edges that are less intriguing to inquire about but are still crucial for displaying entity relations. For example, certain predicates such as "member of", "domestic relation", or "contains the administrative territorial entity", exemplify links between entities, but they are less captivating to inquire about and are too prevalent. Conversely, significant and meaningful edges are referred to as **"active edges"**, and we use them to generate questions. Active edges represent the essential and noteworthy connections in the knowledge graph, from which we extract information to formulate insightful questions.

Developing and cleaning these domain specific KG is not trivial. More details of KG construction are provided in the Appendix 8.1.

3.2 Evaluation Efficiency Study

Before presenting the evaluation results, here we first show the efficiency investigation of our proposed method. We performed a simulation study to evaluate how different methods perform under the same computational resource. We compare the following approches: 1) our method, 2) our method without propagation, 3) the Monte Carlo method, 4) Monte Carlo without propagation and 5) the straightforward iteration method (base), which involves iterating over all edges multiple times.

[1] https://www.01.ai
[2] https://query.wikidata.org



(a) (b)

Figure 3: We measure the MSE distance between the ground truth θ and the estimated θ across different sampling method. X-axis denotes the number of API request. Figure 3(a) shows the MSE of the whole graph, and Figure 3(b) of the top 100 difficult edges (100 edges with largest θ). 22.5K API requests corresponds to iterate over the entire edge set 8 times.

We used a NBA PKG pretrained with the base method as the ground truth PKG. This ground truth was established without edge sampling nor signal propagation, using only a straightforward iteration over all edges 20 times. We then computed the mean square error (MSE) between the estimated θ and the ground truth θ (only examined edges are selected). We show the simulation results in Figure 3, where the amount of compute resource allocated is represented as the number of API requests.

Based on Figure 3(a), our proposed sampling method appears to be advantageous. Notably, our method converges to the ground truth θ faster than both Monte Carlo and Base method. Random sampling from Monte Carlo approach only helped at the very beginning when the compute resource is limited. In term of signal propagation, there is no significant benefit seen when the MSE is computed across all the examined edges.

We also plot the MSE across the top 100 most challenging edges, as shown in Figure 3(b). We can see that our method with signal propagation demonstrates its capability to swiftly identify difficult edges, compared to other methods. The simulation results indicate our sampling method is more efficient than the others and our propagated parameter updating method can identify those challenging edges earlier.

3.3 Main Results

To evaluate the KGs, we run KGLENS across LLMs with 60 iterations and 64 batch size for each graph. Table 2 and Table 3 show results of win rate and zero sense rate over different knowledge graphs under EASY and HARD evaluation modes.

Across varying difficulty levels, knowledge graphs, and the tested models, GPT-4 consistently outperforms the others in both metrics. Also, we find the recent released GPT-4-1106-preview performs worse than GPT-4, which is reasonable for a preview version.

We find the gap between GPT-3.5-turbo and GPT-4 relatively larger across all domains and all difficulty levels, and GPT-3.5-turbo is even worse than the legacy LLMs under NBA KG EASY mode. Upon investigating the evaluation logs, GPT-3.5 exhibits a conservative approach, abstaining from generating answers when lacking confidence rather than providing speculative responses. Responses following this protocol consistently begin with the phrases, "I am sorry, but I couldn't find any information on/about...", "I'm sorry, but as an AI assistant, I do not have the capability to provide real-time information ...". In such cases, the edge would be marked as failed when the model declines to answer a question. We also observed such behavior in Yi-34B-Chat and Vicunna-33b-v1.3.

We find the open sourced model Yi-34B-Chat is comparable to the GPT-3.5-turbo model and outperforms GPT-3.5-turbo in the NBA KG dataset on both Easy and Hard modes. This is a remarkable achievement for Yi-34B-Chat, considering its smaller size compared to GPT-3.5-turbo. A smaller-sized LLM called Xwin-LM-13B-V0.2, also did something very interesting. It followed the trend set by Yi-34B-Chat and outperformed Vicuna-33b in our experiments. Given Vicuna-33b-v1.3 is an

LLMs	Country		NBA		Movie		Average
	EASY	HARD	EASY	HARD	EASY	HARD	
Babbage-002	57.46	34.39	58.32	27.65	57.48	31.00	44.38
Davinci-002	58.85	38.36	58.21	30.57	55.66	34.72	46.06
Vicuna-33b-v1.3	66.51	55.87	36.60	41.66	50.56	46.22	49.57
Xwin-LM-13B-V0.2	54.77	49.13	53.52	50.51	53.59	47.84	51.56
Yi-34B-Chat	66.72	56.16	65.66	62.06	59.86	55.78	61.04
GPT-3.5-turbo	74.43	63.42	57.98	56.95	62.80	57.70	62.21
GPT-4-1106-preview	82.27	72.42	79.09	70.57	83.15	66.95	75.74
GPT-4	**84.79**	**74.06**	**84.23**	**78.93**	**85.14**	**70.80**	**79.66**

Table 2: Win rate results for different LLMs evaluated under EASY and HARD modes.

LLMs	Country		NBA		Movie		Average
	EASY	HARD	EASY	HARD	EASY	HARD	
Babbage-002	24.51	51.56	15.34	38.61	26.70	56.77	35.58
Davinci-002	24.44	47.27	17.69	37.89	28.54	52.71	34.76
Vicuna-33b-v1.3	17.19	26.09	41.75	37.30	36.01	42.47	33.47
Xwin-LM-13B-V0.2	28.96	35.12	19.06	26.92	34.59	38.48	30.52
Yi-34B-Chat	16.15	25.17	14.16	18.79	26.58	30.90	21.96
GPT-3.5-turbo	14.98	20.32	17.17	21.09	22.70	29.36	20.94
GPT-4-1106-preview	7.59	14.16	8.19	12.42	9.21	21.43	12.17
GPT-4	**7.42**	**12.99**	**6.07**	**8.13**	**8.35**	**17.67**	**10.11**

Table 3: Zero sense rate results for different LLMs evaluated under EASY and HARD modes.

instruction-fine-tuned model, it only has slightly edged out legacy OpenAI completion models. In fact Vicuna-33b-v1.3 only performs better in answering Country related questions.

Lastly, we find the two legacy models exhibit comparable performance across evaluations. The random guessing baseline of the win rate is 50% for EASY evaluation, and 25% for HARD evaluation. We find Babbage-002 and Davinci-002 results are just slightly better than the random guessing, clearly showing the gap between the legacy LLMs and the recent LLMs.

In addition to the quantitative analysis, we have identified four categories of common errors within LLMs: Factual errors, Obsolete Knowledge errors, Self-contradiction errors, and Inconsistent Response errors. We provide examples of each error type in Table 5.

3.4 Results Analysis by Edge Attributes

One advantage of evaluating LLM with KG is that the results can be aggregated by different edge attributes. In this section, we show KGLENS can be used for two different focuses of evaluation including the temporal groups and entity groups.

3.4.1 Temporal Groups

We first show the HARD mode Movie KG results which are grouped by the movie release years in Figure 4. The EASY mode results are in Appendix 8. From this figure, we observe that both the GPT-3.5 and GPT-4 perform worse for questions after 2020, which is reasonable as they were mainly pretrained with data before September 2021. Also, we found that GPT-4 significantly outperform the other models in terms of zero-sense rate and win rate. All models get worse when evaluated in HARD mode, but GPT-3.5 is more robust. This is because a big portion of GPT-3.5's failures are caused by refusing to answer the questions, instead of providing the wrong answers, which explains its results in EASY and HARD testing. Interestingly, we find all three recent LLMs perform worse for movies released in 2018, which might related to the pretraining data collection but need further investigation as their pretraining data are not publicly available. It should be noted that it is reasonable that the rankings in Figure 4 are not strictly aligned with the years, as the temporal difference is not the only factor that affect the evaluation results.

3.4.2 Entity Groups

In addition to the temporal group evaluation, we also show results where we group the Country KG edges by the entity type in Figure 7 in appendix.

The proficiency levels across countries can be visualized using a color coded table, where a darker color signifies higher zero sense rate and thus a

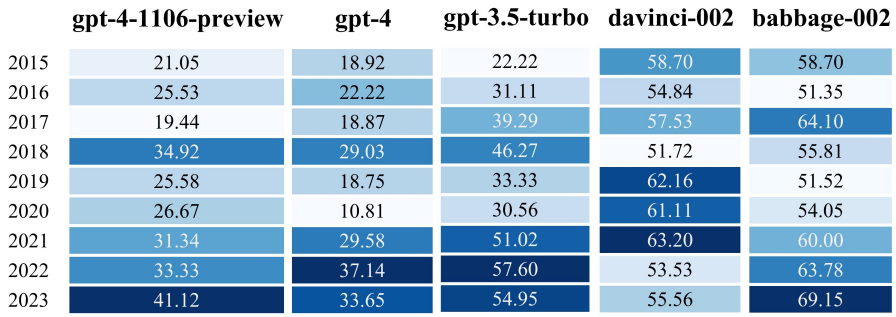


Figure 4: Zero sense rate grouped by years for moive KG in HARD mode. The three recent LLMs perform worse for knowledge after 2020, while the behaviors of the two legacy LLMs are more randomly.

lower level of proficiency. Taking GPT-4 evaluated against country KG under HARD level difficulty for example, GPT-4 exhibits a recognition accuracy where the Austria, Mexico, and Italy are identified and ranked as 1, 2, and 3 respectively. In contrast, countries such as Canada, Philippines, and the United Kingdom are positioned at the lower end of the ranking scale.

The rationale behind the ranking can be elucidated by examining the dotted heatmap in the appendix(Figure 5). In this figure, the size of each dot corresponds to the number of edges within the predicate sub-group, normalized by the total size of edges in the entire group. Additionally, the color of each dot serves as an indicator of the knowledge proficiency associated with the predicate sub-group pertaining to the respective country. Contrary to the table color theme, the darker color here indicates lower zero sense rate and thus higher level of proficiency.

We find KGLENS can easily tells where the errors came from for each country group. Concentrating on the Austria and the Canada, which represent the highest and lowest ranked countries, respectively, it becomes evident that GPT-4 exhibits enhanced proficiency pertaining to specific predicate sub-groups. Notably, these sub-groups include "located in time zone", "located in the administrative territorial entity", "electrical plug type," "emergency phone number," and "head of state".

3.5 Human Evaluation

We conduct human evaluation to verify the question generation module and the question answering module of KGLENS. A random sample of 300 instances was obtained (100 per domain, 50 per question type), and human annotations were acquired through five rounds of rating. The assessment process is conducted instance by instance, where the annotators were tasked with evaluating two aspects (QG and QA): firstly, the clarity of the generated question's intent, and secondly, the correctness of the LLM's response in relation to the ground truth answer and its synonymous expressions. These second objective is to verify if the annotator's judgement agrees with KGLENS's judgement, and is only conducted for Wh questions, as there is no need to verify Yes/No by human. After collected the ratings, a majority voting mechanism was employed for each instance, wherein a label was assigned as "True" if at least three annotators concurred on the evaluation criterion. The evaluation results are presented in Table 4, and KGLENS demonstrates robust performance in human evaluation across domains. It achieves a remarkable 96% accuracy in question answering verification and an impressive 98% accuracy in question generation. We also report the overall accuracy of KGLENS . For the purpose of this evaluation, we define an instance as correct when two conditions are met: the generated question is marked as correct by human; and the QA correctness judged by KGLENS aligns with human judgment. The results indicate when using KGLENS to assess LLMs

	Country	NBA	Movie	Average
QG	96%	98%	99%	97.7%
- Wh-question	100%	100%	98%	99.3%
- Yes/No-question	92%	96%	100%	96%
QAV	96%	96%	96%	96%
QG+QAV	94%	96%	97%	95.7%

Table 4: Human assessment of question generation (QG) module and question answering verification (QAV) module. Majority voting among five annotators was employed as the method for rendering a final judgment.

with knowledge graph, it can approximate human-level performance, achieving an accuracy rate of 95.7%.

4 Related Work

It's an established fact that pre-trained models have the ability to learn and retain knowledge. For example, Petroni et al. (2019) discovered that BERT (Devlin et al., 2018), even without finetuning, harbors relational knowledge comparable to traditional NLP methods. With LLMs showcasing superior in-context learning and knowledge retention, evaluating their knowledge becomes pivotal to bolster performance and mitigate hallucination.

The knowledge assessment often tests the model with specific knowledge-related datasets (Lewis et al., 2021; Petroni et al., 2020; Roberts et al., 2020; Peng et al., 2023; Press et al., 2022; Mallen et al., 2023). However, given the fact that LLMs are trained on web-crawled corpora and the data is constantly evolving, it is hard to exclude the test examples from the pretraining corpus. For example, Deng et al. (2023) use fill-in probing and multi-choice probing to check the data leakage of pretrained LLMs. Their results show that GPT-3.5-turbo exhibited a noteworthy ability to guess the missing option. Another concern is that the knowledge is dynamic, and the evaluation datasets remain fixed, which makes it challenging to evaluate the LLMs' knowledge accurately. Dhingra et al. (2022) propose a diagnostic dataset that pairs the text and timestamp together and jointly models text and time. However, their dataset is static and designed for 2010 to 2020, which is not suitable for evaluating the LLMs' knowledge in the future. Finally, the predominant metric employed by these datasets revolves around the test set accuracy, making it challenging to identify solutions for enhancing the LLM and reducing the hallucination.

On the other hand, knowledge graphs have the advantages of customization to specific domains, evolving knowledge, and reduced potential for test set leakage, which has been employed as a structured knowledge source for LLMs (Lin et al., 2019; Agarwal et al., 2020; Rosset et al., 2020) and also been employed as a tool to probe knowledge in LLMs. LAMA (Petroni et al., 2019) is the first work to probe a pretrained model with KGs, where they use the KG to generate the cloze statement and evaluate the LM's knowledge with accuracy. However the cloze statement is not a natural question, and the correct answer is not unique in many cases, making the evaluation inaccurate. LPAQA (Jiang et al., 2020) propose to mine the relation words from the web for each subject-object pair, which is impractical for large knowledge graph. In addition, these methods mainly focus on the accuracy but neglect that LLMs may respond differently to the same fact, where reliability should also be considered. KaRR (Dong et al., 2023) proposes to solve this issue by using multiple prompts for each KG edge and using the output logits of LLMs to measure the knowledge reliability. However, KaRR is inefficient for large graphs, and it is not generalizable due to the unavailable of LLM's output logits. Moreover, transforming KG triplets into questions is more natural than the text cloze task, but previous works mainly adopt the text cloze task for simplicity. Finally, to our best knowledge, there is no existing work that visualizes the LLM's knowledge with KG (Figure 1).

5 Conclusion

In this work, we introduced KGLENS, a novel and efficient method tailored for visualizing and evaluating the factual knowledge embedded in LLMs. By evaluating various LLMs with our developed domain-specific KGs, we show KGLENS provides adaptable and customizable views of an LLM's knowledge. In addition to evaluating the accuracy of facts, our proposed parameterized KG offers an efficient way to assess the knowledge reliability of LLMs. Human evaluation results indicate that KGLENS can access LLMs with a level of accuracy nearly qeuivalent to that of human annotators, achieving 95.7% of the accuracy rate. Furthermore, our tool KGLENS, together with our assessment KGs, sourced from Wikidata, will be available to the research community, fostering collaboration and serving as a valuable resource for future investigations into language models.

6 Limitation

KG plays a pivotal role in our approach, and its quality significantly impacts the effectiveness of this method. A high-quality KG is essential not only for the Question Generation step to generate meaningful questions but also for signal propagation. If the KG is fragmented and scattered, signal propagation then becomes less beneficial.

While our current method incorporates counting updates for alpha and beta, we acknowledge the

potential for improvement. Exploring alternative methods for updating these parameters is an area of active research for us.

The signal propagation method is another direction that we can dive into, instead of only propagate to neighbour edges, should we also propagate to further edges? Instead of equally update the neighbour edges, should we decay the signal? etc.

Question generation currently is limited to just one hop, being able to generate complicated questions that evolves multiple edge hops would enable our method to evaluation the model not only on factual knowledge retrieval, but also complex reasoning capability.

7 Ethical Considerations

We foresee no ethical issues originating from this work.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv preprint arXiv:2010.12688*.

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, et al. 2023. Factuality challenges in the era of large language models. *arXiv preprint arXiv:2310.05189*.

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

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2023. Benchmark probing: Investigating data leakage in large language models. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang Sui, and Lei Li. 2023. Statistical knowledge assessment for large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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, 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.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151*.

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

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

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*.

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*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

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*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

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

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*.

Xwin-LM Team. 2023. Xwin-lm.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Cunxiang Wang, Pai Liu, and Yue Zhang. 2021. Can generative pre-trained language models serve as knowledge bases for closed-book qa? *arXiv preprint arXiv:2106.01561*.

8 Appendix

8.1 Knowledge Graph Building and Cleaning

Given Wikidata's vastness and inherent noise, we implement multiple strategies to maintain focus, relevance, and precision. Following techniques empower us to delve into specialized domains and ensure us a targeted and reliable exploration of the data.

8.1.1 Sampling Strategies and Preserving Data Distribution

Maintaining the original data distribution is important when cleaning a knowledge graph. To achieve this, random walk with both forward 8.5.1 and backward 8.5.2 dimension are employed. Sorting by random value of each queried edges, the sub-knowledge graph contains the representative samples that mirror the diversity of the original knowledge graph, we can preserve the inherent distribution of entities and relationships. This approach guarantees that our cleaned knowledge graph remains a faithful representation of the underlying data, enabling us to draw accurate conclusions from our research.

The extent of the random walk distance is flexible and tailored to specific requirements. Within our sub knowledge graphs, we conduct random walks spanning three steps, encompassing both nodes and edges within this range for analysis.

8.1.2 Focus and Curated Relevance

In the realm of knowledge graphs, Wikidata stands out as a repository of extensive information. However, our research necessitates a more nuanced approach. While Wikidata offers comprehensive knowledge, our focus lies in curated topics and entities tailored for specific purposes. This distinction is vital as it allows us to delve deeper into specialized domains, ensuring the precision and relevance of the data we analyze.

To address this issue, the parameterized knowledge graph begins by establishing a set of human selected central entities, from which it initiates random walks to explore neighboring entities. Additionally, we perform predicate analysis to discern and exclude predicates of lesser importance or those that are overly common. This approach ensures the focus on pertinent data while filtering out less relevant information.

8.1.3 Filtering Less Relevant Entities

The other challenges we encounter in Wikidata pertains to the noise within its knowledge graph. This noise manifests in the form of entities that are rarely mentioned or of lesser importance in the context of our research objectives. To maintain the integrity of our analysis, it is important to identify and filter out these less relevant entities.

- Filtering by language count: entities mentioned in multiple languages are often more significant and relevant to a broader audience. By focusing on such multilingual entities, we ensure the inclusion of globally relevant information in our analysis.
- Filtering by word frequency: entities that are frequently mentioned in various contexts are likely to hold greater importance. By considering word frequency, we prioritize entities that are central to discussions, thereby enhancing the relevance and significance of the data included in our analysis.
- Filtering out entities with no alias: entities without aliases are less likely to be widely recognized or referenced. By excluding these entities, we focus our analysis on well-known and frequently mentioned entities, aligning our research with more meaningful and impactful data points.

8.2 Uncovered Error Types

Error Type	Question	Response	Answer
Factual Error	On which side does the railway traffic run in Israel?	Railway traffic in Israel runs on the right-hand side.	Left side
Obsolete Knowledge	Who is the current head of state in England?	The current head of state in England is Queen Elizabeth II.	Charles III of the United Kingdom
Self-contradiction	Is the Australian dollar the currency of Nauru (also known as NR)?	No, the Australian dollar (AUD) is the official currency of Nauru, a small island nation in the Pacific Ocean.	Yes
Inconsistent Response	In Tonga (also known as TO), do people drive on the right side of the road?	No, in Tonga, people drive on the left side of the road.	No
	Is the left the driving side in Tonga (also known as TO)?	No, the right is the driving side in Tonga.	Yes

Table 5: Error types uncovered from the country KG.

8.3 Human Evaluation

We conduct human evaluation with an internal paid crowdsouring service, where 5 annotators participated the annotation process with their consent of using the data. All the annotators are from English speaking countries. The annotation instruction is shown below.

Instruction

In this survey, you are provided with a question, the answer (and answer aliases separated by comma), and an AI's response to this question.
Background
This survey is to check if the question is understandable, and if the AI's response is matched with the given answer.
Annotation Guideline
As an annotator, you are asked to judge whether the intent of the question is clear enough to be understood (if you can understand the question, please select the option 'Yes', whether you know the answer or not).
For some questions, you won't see the answer and AI's response.
For some questions, you will see the answer and AI's response. Then you need to check if the AI's response is correct or not. Please select the option "correct" when the answer, or any of the answer alias appears in AI's response. Otherwise, select "Incorrect".
Note: You don't need to do any research to find the answer. Just use the provided answer.

8.4 Prompt

Answer Verification

Given a question and its correct answers, please judge whether a response is correct or not, only reply with yes or no.
User:
QUESTION: What writing system does the Ixcatec language use?
ANSWERS: latin script, latn, roman script
Assistant:
RESPONSE: the ixcatec language uses the latin alphabet for writing.
User:
QUESTION: On which side does railway traffic run in the philippines?
ANSWERS: left, left side, left-hand side
Assistant: RESPONSE: Railway traffic in the philippines runs on the right-hand side.
User:
QUESTION: <question>
ANSWERS: <answers>

Yes/No Question Generation

Given a semantic triple (SUBJECT, PREDICATE, OBJECT), generate a QUESTION that asks if the SUBJECT's PREDICATE is the OBJECT, or includes the OBJECT. The alias of the SUBJECT can be used in the question.
User: SUBJECT is Joroud, PREDICT is place of birth, OBJECT is Gamla Uppsala, SUBJECT_ALIAS is Jörundr
Assistant: Is Gamla Uppsala the birth place of Jorund (also known as Jörundr)?
User: SUBJECT is Asia, PREDICT is located in time zone, OBJECT is Nepal Standard Time, SUBJECT_ALIAS is Asian continent
Assistant: Is Nepal Standard Time included in the time zones of Asia (also known as the Asian continent)? User: <s, p, o, [subject alias]>

Wh-Question Generation

Given a semantic triple SUBJECT, PREDICATE, OBJECT, generate a QUESTION that can be answered by the OBJECT. The OBJECT must not occur in the question.
User: SUBJECT is Jorund, PREDICT is place of birth, OBJECT is Gamla Uppsala
Assistant: What is the birth place of the legendary Swedish king Jorund?
User: SUBJECT is Yellow Emperor, PREDICT is father, OBJECT is Shaodian, SUBJECT_ALIAS
Assistant: Who is the father of Yellow Emperor?
User: <s, p, o>

Yes No Question Answer

You are a helpful assistant, please answer Yes or No to the user's questions.
User:Is Belgium located in the continent of Europe?
Assistant: Yes.
User: Is Andrzej Duda the head of state of Belgium?
Assistant: No.
User: <yes no question>

Generative Question Answer

You are a helpful assistant, please give short and accurate answers to the user's question. If there are multiple answers, please list as much as possible.
User:What is the birth place of Jorund?
Assistant: Gamla Uppsala.
User: Who is the father of Yellow Emperor?
Assistant: Shaodian
User: <generative question>

8.5 Wikidata Web Query

8.5.1 Forward Walk

```
SELECT DISTINCT ?subject ?subjectLabel ?subjectDesc ?predicate ?predicateLabel ?
    predicateDesc ?object ?objectLabel ?objectDesc
WHERE {{
  VALUES ?subject {{
    {values}
  }}
  ?subject ?predicate ?object .
  ?subject rdfs:label ?subjectLabel .
  ?subject schema:description ?subjectDesc .
  ?property wikibase:directClaim ?predicate .
  ?property rdfs:label ?predicateLabel .
  ?property schema:description ?predicateDesc .
  ?object rdfs:label ?objectLabel .
  ?object schema:description ?objectDesc .
  FILTER (lang(?subjectLabel) = "en")
  FILTER (lang(?subjectDesc) = "en")
  FILTER (lang(?predicateLabel) = "en")
  FILTER (lang(?predicateDesc) = "en")
  FILTER (lang(?objectLabel) = "en")
  FILTER (lang(?objectDesc) = "en")
}}
ORDER BY UUID()
LIMIT {limit}
```

8.5.2 Backward Walk

```
SELECT DISTINCT ?subject ?subjectLabel ?subjectDesc ?predicate ?predicateLabel ?
    predicateDesc ?object ?objectLabel ?objectDesc
WHERE {{
  VALUES ?object {{
    {values}
  }}
  ?subject ?predicate ?object .
  ?subject rdfs:label ?subjectLabel .
  ?subject schema:description ?subjectDesc .
  ?property wikibase:directClaim ?predicate .
  ?property rdfs:label ?predicateLabel .
  ?property schema:description ?predicateDesc .
  ?object rdfs:label ?objectLabel .
  ?object schema:description ?objectDesc .
  FILTER (lang(?subjectLabel) = "en")
  FILTER (lang(?subjectDesc) = "en")
  FILTER (lang(?predicateLabel) = "en")
  FILTER (lang(?predicateDesc) = "en")
  FILTER (lang(?objectLabel) = "en")
  FILTER (lang(?objectDesc) = "en")
}}
ORDER BY UUID()
LIMIT {limit}
```

8.6 Additional Figures

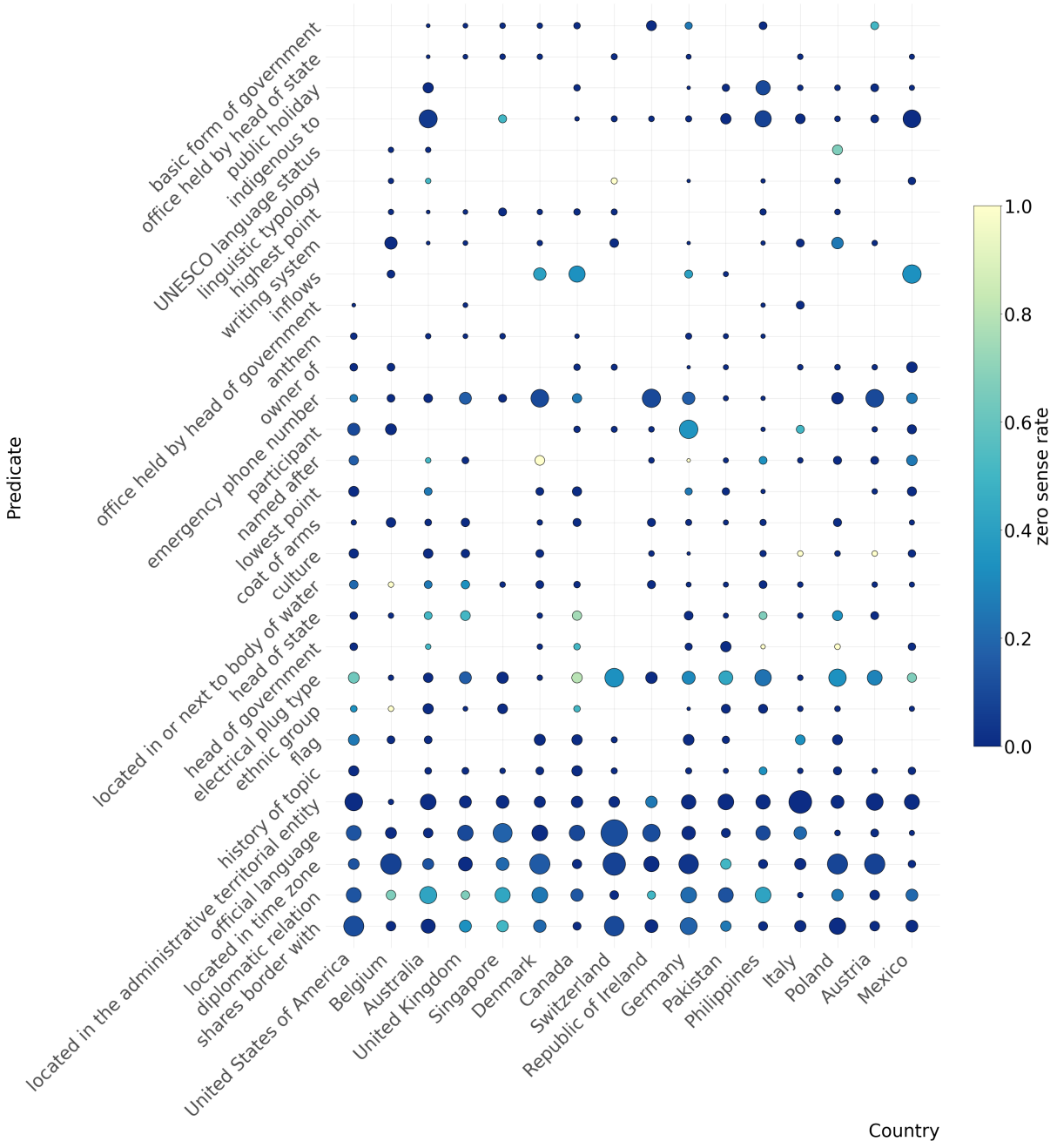


Figure 5: Predicate level knowledge proficiency of GPT-4 evaluated under HARD difficulty. The darker color indicates a lower zero sense rate. The dot size shows the proportional size of the number of edges in the predicate sub-group.

	gpt-4-1106-preview	gpt-4	gpt-3.5-turbo	davinci-002	babbage-002
Australia	11.20	11.31	18.45	26.57	27.63
Austria	7.64	7.19	20.98	30.81	31.32
Belgium	9.04	12.34	16.11	26.96	29.52
Canada	10.60	13.66	22.86	29.38	28.65
Denmark	7.33	9.88	20.38	30.68	35.71
Germany	10.48	11.54	19.47	26.97	31.02
Italy	6.16	8.23	17.42	30.98	28.73
Mexico	11.33	7.73	10.60	24.10	28.29
Pakistan	8.97	7.48	15.92	25.18	27.03
Philippines	5.62	7.74	23.47	31.13	21.20
Poland	6.52	12.87	19.51	26.00	33.33
Republic of Ireland	8.76	5.13	22.48	24.83	25.00
Singapore	7.09	10.19	21.05	26.40	30.39
Switzerland	10.16	9.16	20.77	30.61	26.83
United Kingdom	12.05	14.85	22.16	26.94	26.54
United States of America	9.07	7.41	18.65	27.27	28.06

Figure 6: Country KG EASY-level zero sense rate grouped by countries.

	gpt-4-1106-preview	gpt-4	gpt-3.5-turbo	davinci-002	babbage-002
Australia	11.45	13.41	16.83	47.87	50.70
Austria	10.00	7.41	15.32	40.40	47.98
Belgium	5.63	9.35	16.13	42.65	47.03
Canada	12.08	19.49	24.32	45.05	46.77
Denmark	9.09	11.03	14.71	47.00	45.35
Germany	11.77	11.69	17.25	44.92	51.85
Italy	8.61	9.09	15.62	38.07	48.31
Mexico	6.29	9.33	13.14	43.68	51.93
Pakistan	15.57	13.55	16.54	42.67	55.62
Philippines	11.46	16.02	18.41	45.98	47.49
Poland	12.58	12.70	15.56	46.49	45.03
Republic of Ireland	8.16	11.81	22.39	43.65	47.06
Singapore	13.72	15.38	22.31	47.17	49.11
Switzerland	7.87	10.00	14.56	47.89	45.00
United Kingdom	10.32	16.95	25.34	45.37	51.15
United States of America	10.96	11.54	20.93	40.60	47.27

Figure 7: Country KG HARD-level zero sense rate grouped by countries.

	gpt-4-1106-preview	gpt-4	gpt-3.5-turbo	davinci-002	babbage-002
2015	17.65	9.80	17.86	20.00	26.00
2016	16.33	7.69	29.41	29.03	25.00
2017	8.00	14.58	22.64	38.09	29.55
2018	14.29	7.55	26.67	32.65	30.77
2019	15.91	5.88	32.50	19.44	21.88
2020	15.56	11.63	26.32	20.93	20.00
2021	12.99	7.89	35.92	27.78	37.21
2022	16.22	17.82	47.10	27.50	22.03
2023	26.83	20.00	43.48	34.52	30.38

Figure 8: Movie KG EASY-level zero sense rate grouped by years.

	gpt-4-1106-preview	gpt-4	gpt-3.5-turbo	davinci-002	babbage-002
Atlanta Hawks	6.09	4.27	14.95	17.73	15.60
Boston Celtics	6.27	4.09	13.07	17.23	14.40
Brooklyn Nets	6.38	4.78	14.41	17.19	15.57
Charlotte Hornets	6.38	4.35	12.53	17.77	14.33
Chicago Bulls	6.00	4.66	14.49	17.11	15.57
Cleveland Cavaliers	6.42	4.28	12.69	16.88	14.00
Dallas Mavericks	5.88	4.29	14.66	17.14	15.71
Denver Nuggets	6.17	4.51	14.63	17.61	15.50
Detroit Pistons	6.47	4.37	14.59	17.81	15.69
Golden State Warriors	6.04	4.41	14.48	17.34	15.45
Houston Rockets	6.18	3.99	13.39	17.38	14.30
Indiana Pacers	6.04	3.96	12.65	18.08	14.24
Los Angeles Clippers	6.54	4.74	14.68	17.32	15.70
Los Angeles Lakers	6.33	4.38	14.50	17.23	15.61
Memphis Grizzlies	6.02	4.18	12.20	17.41	14.35
Miami Heat	5.89	4.30	14.40	17.30	15.81
Milwaukee Bucks	6.11	4.27	14.80	17.33	15.65
Minnesota Timberwolves	5.72	4.53	14.46	17.05	15.20
New Orleans Pelicans	5.89	5.00	14.37	17.57	15.71
New York Knicks	5.85	4.82	14.98	17.17	15.57
Oklahoma City Thunder	6.05	4.23	12.43	17.65	14.39
Orlando Magic	5.62	4.17	14.11	17.36	15.61
Philadelphia 76ers	6.11	4.56	14.61	17.61	15.61
Phoenix Suns	6.27	3.94	14.89	17.25	15.46
Portland Trail Blazers	6.22	4.16	14.48	17.14	15.65
Sacramento Kings	6.11	4.39	14.45	17.35	15.62
San Antonio Spurs	6.35	4.56	14.36	17.12	15.49
Toronto Raptors	6.60	5.21	16.81	17.69	17.09
Utah Jazz	5.94	4.59	14.33	17.41	15.62
Washington Wizards	5.80	4.50	14.85	17.15	15.61

Figure 9: NBA EASY-level zero sense rate grouped by teams

	gpt-4-1106-preview	gpt-4	gpt-3.5-turbo	davinci-002	babbage-002
Atlanta Hawks	9.45	5.22	17.90	37.33	38.95
Boston Celtics	7.82	4.91	12.68	36.41	37.63
Brooklyn Nets	9.66	5.61	17.20	37.03	38.60
Charlotte Hornets	7.40	4.98	13.12	36.09	37.87
Chicago Bulls	9.35	4.96	17.30	37.51	38.61
Cleveland Cavaliers	7.56	5.27	13.87	36.33	37.23
Dallas Mavericks	9.19	5.34	17.13	37.39	38.80
Denver Nuggets	9.55	5.18	18.27	37.05	39.10
Detroit Pistons	9.65	5.67	18.14	37.36	39.26
Golden State Warriors	9.55	5.46	17.29	37.56	38.89
Houston Rockets	7.58	4.93	12.89	37.41	37.90
Indiana Pacers	6.90	4.70	12.96	36.69	38.14
Los Angeles Clippers	9.84	6.03	18.19	37.06	39.59
Los Angeles Lakers	9.91	5.77	17.99	36.98	38.75
Memphis Grizzlies	7.01	4.47	12.44	36.31	38.29
Miami Heat	8.94	5.15	16.96	37.10	38.78
Milwaukee Bucks	9.37	5.20	17.68	36.96	38.96
Minnesota Timberwolves	9.30	5.07	16.75	37.23	38.68
New Orleans Pelicans	9.25	5.42	16.65	37.25	38.86
New York Knicks	9.81	5.09	17.25	37.02	39.47
Oklahoma City Thunder	6.89	4.44	12.26	35.80	37.61
Orlando Magic	9.04	4.96	16.40	36.83	39.01
Philadelphia 76ers	9.74	5.10	17.32	37.32	38.85
Phoenix Suns	9.55	5.48	17.02	37.50	38.95
Portland Trail Blazers	8.96	5.18	16.82	37.23	39.31
Sacramento Kings	9.65	5.81	17.47	37.10	38.94
San Antonio Spurs	9.66	5.57	17.15	37.17	38.81
Toronto Raptors	10.27	5.86	20.61	39.82	39.31
Utah Jazz	9.30	4.94	17.26	37.09	38.78
Washington Wizards	9.37	5.09	17.46	37.29	38.80

Figure 10: NBA HARD-level zero sense rate grouped by teams