

# KGLens : Towards Efficient and Effective Knowledge Probing of Large Language Models with Knowledge Graphs

Anonymous ACL submission

## Abstract

Large Language Models (LLMs) might hallucinate facts, while curated Knowledge Graph (KGs) are typically factually reliable especially with domain-specific knowledge. Measuring the alignment between KGs and LLMs can effectively probe the factualness and identify the knowledge blind spots of LLMs. However, verifying the LLMs over extensive KGs can be expensive. In this paper, we present KGLENS, a Thompson-sampling-inspired framework aimed at effectively and efficiently measuring the alignment between KGs and LLMs. KGLENS features a graph-guided question generator for converting KGs into natural language, along with a carefully designed importance sampling strategy based on parameterized KG structure to expedite KG traversal. Our simulation experiment compares the brute force method with KGLENS under six different sampling methods, demonstrating that our approach achieves superior probing efficiency. Leveraging KGLENS, we conducted in-depth analyses of the factual accuracy of ten LLMs across three large domain-specific KGs from Wikidata, composing over 19K edges, 700 relations, and 21K entities. 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 Nonetheless, studies have shown that LLMs can produce information that is nonfactual, hallucinated, or outdated (Perez et al., 2022; Ji et al., 2023; Lee et al., 2022; Wang et al., 2021).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, fact-checking (Thorne et al., 2018; Augenstein et al., 2023) and fact-QA (Petroni et al., 2020; Press et al., 2022; Dhingra et al., 2022) approaches are commonly used, but several challenges persist. For

fact-checking, distinguishing faithful and unfaithful statements is different from evaluating factualness of the generation. For fact-QA, scaling up the evaluation data is challenging due to the expensive nature of the annotation process. For both approaches, it is hard to exclude their data from the web-crawled pretraining corpus (Deng et al., 2023) to ensure the fairness of the evaluation. Last but not least, LLMs may respond differently to the same fact when the query is phrased in different forms, a challenge that existing fact-checking and fact-QA datasets are not equipped to handle.

In contrast, probing LLMs knowledge by transforming knowledge graph (KG) data into natural language questions addresses these limitations. First, KGs are inherently scalable, factually reliable, and tailored to specific domains. This scalability allows for extensive and efficient evaluation of LLMs across various domains. Second, question generation (QG) from KGs can be automated, enabling the rapid and large-scale creation of evaluation datasets. Moreover, generating various questions from the same set of facts allows us to assess the robustness of LLMs to different phrasings.

However, there are several challenges for QG-based KG evaluation. The first is the efficiency issue. Existing knowledge probing methods (Dong et al., 2023; Wang et al., 2023), mainly using the brute-force method which exhaustively evaluates all KG edges, are computationally expensive, time-consuming, and non-scalable. The second challenge comes to transformation of ambiguous KG triplets into natural language queries. Given a subject and predicate, the correct object may not be unique, thus the answer to the question is ambiguous. Furthermore, some subjects may lack specificity, resulting in confusing questions. Examples illustrating these issues are shown in Table. 1. In this case, both text-cloze based methods (Petroni et al., 2019; Jiang et al., 2020; Dong et al., 2023) and prompt-based QG methods (Wang et al., 2023)

043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083

<b>Multi-objects</b>
Barack Obama, educated at, Columbia University
Barack Obama, educated at, Harvard Law School
<b>Ambiguous subject</b>
Waterloo (also known as Waterloo, Ontario)
Waterloo (also known as Waterloo, Iowa)

Table 1: Examples of ambiguous KG triplets.

fail to evaluate the factualness properly.

In this study, we present a novel knowledge probing framework named KGLENS (Fig. 1), to measure the knowledge alignment between KGs and LLMs, and to identify LLM’s *knowledge blind spots*. To efficiently probe the LLM, KGLENS features a Thompson sampling inspired method to rapidly identify knowledge blind spots. More specifically, we first 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. We apply Thompson sampling to select the edges based on edge deficiency and then evaluate LLMs with the sampled edges, update the PKG with the evaluation results, and iterate this process until the running metrics converge. Our simulation experiments show that *our sampling method with PKG is more efficient than random sampling and brute-force methods*.

To accurately probe the LLM, KGLENS features a graph-guided question generator for converting KGs into natural language with GPT-4 (OpenAI, 2023). We design two types of questions (fact-checking and fact-QA) to reduce the ambiguity of the expected answer, 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 subject ambiguity. *Human evaluation results show that 97.7% of our generated questions are sensible to human annotators*.

To assess the whole framework of KGLENS, we probed 10 popular LLMs with three domain-specific KGs collected from Wikidata, encompassing over 700 relations and 21K entities. We introduce three evaluation metrics to measure the knowledge alignment between an LLM and a KG: zero-sense rate (percentage of facts that an LLM never answered correctly in all rounds), all-sense rate (percentage of facts that an LLM always answered correctly in all rounds), and win rate (percentage of facts that an LLM answered correctly more than half of the rounds). Human evaluation

shows that KGLENS can assess LLMs with a level of accuracy nearly equivalent to that of human annotators with 95.7% accuracy rate. Our key contributions are as follows:

1. We introduce a novel, efficient knowledge probing framework to efficiently identify LLMs’ knowledge blind spots across diverse topics and relationships.
2. Our QG strategy leverages KG structure to reduce ambiguity thus providing accurate and effective LLM evaluation, surpassing existing methods that focus solely on atomic triplets.
3. Human evaluation verifies our framework’s efficacy with a human-level accuracy (95.7%) in assessing LLMs’ knowledge, nearly matching human annotator performance.
4. We propose three metrics to quantify knowledge alignment between LLMs and KGs, successfully identifying knowledge gaps and reliable facts.
5. Our contributions advance the development of more reliable and factual AI systems, promoting trustworthy user experiences and efficient model improvement. We will open-source our framework and datasets.

## 2 Method

In probing the knowledge of LLMs, we face a unique challenge: while we have access to a vast corpus of KG edges, we lack insight into how well the LLM understands each of them. This scenario bears a striking resemblance to the multi-armed bandit problem in reinforcement learning, where we must balance exploration and exploitation to maximize our understanding of the LLM’s knowledge gaps.

Our approach, which we call KGLENS, aims to efficiently identify the edges where the LLM’s knowledge is weakest or most limited. This allows us to focus our efforts on the areas where the LLM needs the most improvement. The framework of KGLENS is illustrated in Fig. 1, with its workflow summarized in the caption. Key aspects of our method include Thompson Sampling Inspired Parameterized KG, Graph-Guided Question Generator, and Answer Verification.

In the following subsections, we will delve into each of these components in detail, explaining how

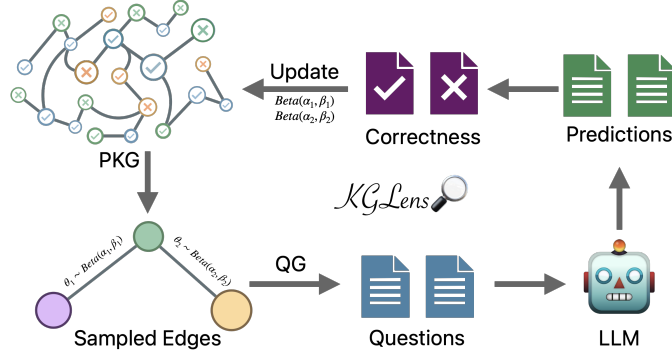


Figure 1: KGLENS Framework. KGLENS starts from the PKG initialization, where each edge is augmented with a beta distribution. Then a batch of edges is sampled based on the edge probability  $\theta$ . After that, questions are generated from these edges and an LLM will be examined with question answering 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.

they work together to create an effective knowledge probing system for LLMs.

## 2.1 Parameterized Knowledge Graph

A knowledge graph  $\mathcal{G}$ , is a set of triplets  $\{(s_j, p_j, o_j)\}_{j=1\dots i}$  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 intuition, we propose a parameterized KG (PKG), by augmenting each edge  $(s_j, p_j, o_j)$  with an additional error probability  $\theta_j$  reflecting the probability that an LLM may fail on this edge. We use a Beta distribution to model  $\theta$ :

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

where  $\alpha$  and  $\beta$  can be interpreted as the number of times the targeted LLM failed or succeeded in answering the question. The prior of each  $\theta_j$  is set to  $\text{Beta}(1, 1)$ .

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

### Edge sampling

The edge sampling process favors the edges with larger  $\theta$  values. In each iteration, we sample top- $n$  edges ranked by  $\theta$  sampled from the Beta distributions of PKG, and then send these edges to LLM for examination and verification. The signal regarding the correctness of LLM’s response is collected for each edge.

## Parameter estimation and updating

For each edge, the  $\alpha$  and  $\beta$  is updated based on the new observation of whether the response from LLM is correct, following the standard Beta distribution posterior updates. In addition, we also propagate the signal to the neighboring edges to account for the high correlation in error probability among the connected edges. To optimize the computational process, signal propagation is restricted 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  $M_j = |\text{incorrect neighborhood edges}|$  and  $N_j = |\text{correct neighborhood edges}|$ . An updated PKG (Fig. 2) is then obtained by repeating the edge sampling and parameter updating process iteratively until the running metrics (3.2) converge.

## 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 A.6. We design two types of questions for KGLENS: Yes/No Questions (judgment) 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,

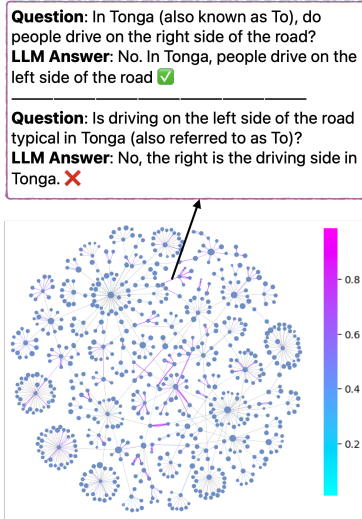


Figure 2: Parameterized KG. The edge color is an estimation in LLM’s deficiency to the associated fact.

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 Wiki-data 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 queries the LLMs to generate the object(s) given the subject and the predicate, where the questions usually begin with when/where/who/what. While this question type presents greater difficulty, it is not universally applicable. Wh-questions may yield hundreds of correct objects, rendering exhaustive enumeration impractical and uninformative.

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.

## 2.3 QAV: Question Answering Verification

Our QA testing involved two difficulty levels: EASY, comprising solely Yes/No questions; and HARD, with each question type generated at a 50% chance. Few-shot in-context learning was employed 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 A.6.

## 2.4 Evaluation Efficiency Study

In this section, we conduct a simulation experiment to evaluate the efficiency of different methods in identifying the most challenging edges in a domain specific KG (NBA KG in Table 2), including: Thompson sampling, Epsilon Greedy, Monte Carlo, and the straightforward iteration method (Brute Force), which involves iterating over all edges multiple times. We first estimate the ground truth  $\theta$  by iterating over all edges 20 times using the Brute-Force method. Then, top-k difficult edges can be identified by ranking the estimated  $\theta$ . Since we are interested in the most challenging edges, Mean Square Error (MSE) for these top-k edges will be computed between the estimated  $\theta$  and the ground truth  $\theta$  for each method.

In Fig. 3, we plot different methods’ MSE score of top-200 and top-600 challenging edges over the number of API requests. The vertical epoch lines represent the cumulative number of API requests necessary to complete a full iterations over the PKG edges using the Brute-Force method. From this figure, we observe Thompson sampling method demonstrates superiority in pinpointing the most challenging edges. Notably, both Thompson sampling and Epsilon-Greedy exhibit advantages over Monte Carlo and Brute-Force methods by striking an effective balance between exploration and exploitation. For example, from Fig. 3(b), we can see that Thompson sampling with propagation achieves the accuracy of a 4.5-epoch Brute-Force method while utilizing only 65% of the compute resources. This finding underscores the potential of our approach to swiftly identify critical edges and optimize resource allocation in large-scale KG applications. In addition, we find the signal propagation mechanism further enhances the capabilities, enabling them to achieve better accuracy rapidly.

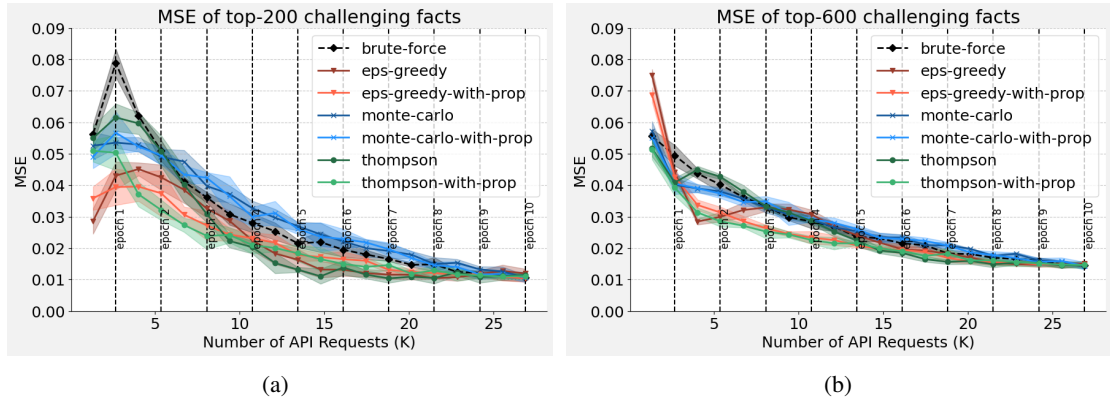


Figure 3: We measure the MSE distance between the ground truth  $\theta$  and the estimated  $\theta'$  across different sampling method. The vertical line epoch-N shows the number of API requests required for the Brute Force method to complete N full iterations over each edges. The graph consists of 2789 edges in total.

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

Table 2: Statistics of the testing knowledge graphs.

### 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 three variations of GPT-4 (GPT-4-1106-preview, GPT-4-turbo, GPT-4o). 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<sup>1</sup>. Web browsing ability is unavailable to OpenAI API calls at the time when experiments are conducted.

#### 3.1 Dataset

We prepare three test datasets (KG) with Wikidata Query Web Service<sup>2</sup> 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 Tab. 2. 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” and “domestic re-

lation”, 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. More details of KG construction are provided in the Appendix A.2.

#### 3.2 Metrics

To measure the alignment between KGs and LLMs, here we introduce three edge-level metrics, where each edge is evaluated multiple times with  $m$  successes and  $n$  failures.

**Win rate.** For each edge, LLM wins if the number of successes surpasses the number of failures, namely  $m > n$ . 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, namely  $m = 0$ . The zero-sense rate signifies the portion of edges with zero-sense.

**All-sense rate.** An LLM would has all-sense about an edge (fact) if the model has never failed to answer the edge, namely  $n = 0$ . The all-sense rate signifies the portion of edges with all-sense.

Based on the definition above, win rate is the portion of edges that an LLM has higher chance to answer them correctly, indicating the reliability of LLM. Zero-sense rate is the portion of edges that an LLM always fails to answer. All-sense rate is the portion of edges that an LLM always succeed to answer.

<sup>1</sup><https://www.01.ai>

<sup>2</sup><https://query.wikidata.org>

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-turbo	83.69	73.19	82.38	76.11	85.62	73.58	79.09
GPT-4o	<b>86.25</b>	73.71	78.45	78.46	<b>86.36</b>	<b>74.34</b>	79.59
GPT-4	84.79	<b>74.06</b>	<b>84.23</b>	<b>78.93</b>	85.14	70.80	<b>79.66</b>

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

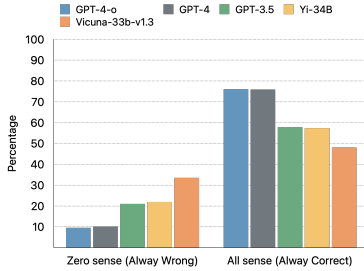


Figure 4: Percentage of facts that LLMs always answered correctly and incorrectly. Full results in Tab. 6, 7.

### 3.3 Main Results

To evaluate the KGs, we run KGLENS across LLMs with 60 iterations and 64 batch size for each graph. Tab. 3 show results of win rate. We put results of zero-sense rate and all-sense rate in Appendix Tab. 6, 7, but we visualize 5 models’ results in Fig. 4.

Across varying difficulty levels, knowledge graphs, and the tested models, GPT-4 family consistently outperforms the others in both metrics. The performance across GPT-4 variations are close, where GPT-4, GPT-4o and GPT-4-turbo exhibits the most comparable performance, followed by GPT-4-1106-preview. We find GPT-4o exhibits a more cautious disposition in disclosing personal information. It frequently refrains from providing specific details about individuals, which may be correlated with its performance disparity in NBA KG, where GPT-4 underperforms GPT-4. Further investigation is warranted. We also 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 provid-

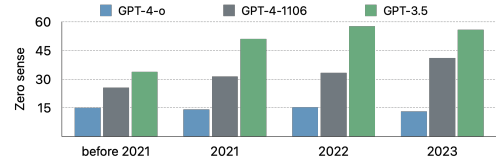


Figure 5: Zero-sense rate across years. Full results in Fig. 10.

ing 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.

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 only slightly better than the random guessing, clearly showing the gap between the legacy LLMs and the recent LLMs. We also provide examples of different error types in Tab. 5.

### 3.4 Results Analysis by Edge Attributes

Another 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 Analysis of Results

We conduct temporal analysis with movie KG. We group the results by the movie release years in Fig. 5 and full results are in Fig. 10 in Appendix. From this Fig. 5, we observe that both the GPT-3.5 and GPT-4-1106 perform worse for questions after

2020, which is reasonable as they were mainly pre-trained with data before September 2021. On the other hand, we found that GPT-4-o significantly outperforms the other models in terms of zero-sense rate and win rate. In Fig. 10, we find 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. Finally, it should be noted that it is reasonable that the rankings in Fig. 10 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 Analysis

We also show results where we group the Country KG edges by the entity type in Fig. 8 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 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(Fig. 6). 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 identify 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”.

	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.

### 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 Tab. 4, and KGLENS demonstrates robust performance in human evaluation across domains. 96% accuracy in QAV and an impressive 98% accuracy in QG. 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. Our findings show a 95.7% accuracy rate for KGLENS , indicating its ability to approximate human-level performance.

## 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 unavailability 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 (Fig. 2).

## 5 Conclusion

In this work, we introduced KGLENS, a novel and efficient method tailored for visualizing and evaluating the factual knowledge embedded in LLMs. Our proposed Thompson sampling inspired framework with parameterized KG offers a more efficient way to reveal LLM’s knowledge blind spots than existing brute force iterating methods. By evaluating various LLMs with our developed domain-specific KGs, we show KGLENS provides adaptable and customizable views of an LLM’s knowledge. Human evaluation results indicate that KGLENS can access LLMs with a level of accuracy nearly equivalent 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. For businesses employing LLMs, our contributions facilitate the development of more reliable and factual AI systems, fostering trustworthy user experiences and efficient processes for improving model knowledge.

## 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



624	to neighbour edges, should we also propagate to	Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan	674
625	further edges? Instead of equally update the neigh-	Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea	675
626	bour edges, should we decay the signal? etc.	Madotto, and Pascale Fung. 2023. Survey of halluci-	676
627	Question generation currently is limited to just	nation in natural language generation. <i>ACM Comput-</i>	677
628	one hop, being able to generate complicated ques-	<i>ing Surveys</i> , 55(12):1–38.	678
629	tions that evolves multiple edge hops would enable	Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham	679
630	our method to evaluation the model not only on	Neubig. 2020. How can we know what language	680
631	factual knowledge retrieval, but also complex rea-	models know? <i>Transactions of the Association for</i>	681
632	soning capability.	<i>Computational Linguistics</i> , 8:423–438.	682
633	<b>7 Ethical Considerations</b>	Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pas-	683
634	We foresee no ethical issues originating from this	cale N Fung, Mohammad Shoeybi, and Bryan Catan-	684
635	work.	zaro. 2022. Factuality enhanced language models	685
636	<b>References</b>	for open-ended text generation. <i>Advances in Neural</i>	686
637	Oshin Agarwal, Heming Ge, Siamak Shakeri, and	<i>Information Processing Systems</i> , 35:34586–34599.	687
638	Rami Al-Rfou. 2020. Knowledge graph based syn-	Patrick Lewis, Yuxiang Wu, Lingqing Liu, Pasquale Min-	688
639	thetic corpus generation for knowledge-enhanced	ervini, Heinrich Küttler, Aleksandra Piktus, Pontus	689
640	language model pre-training. <i>arXiv preprint</i>	Stenetorp, and Sebastian Riedel. 2021. <a href="#">Paq: 65 mil-</a>	690
641	<i>arXiv:2010.12688</i> .	<a href="#">lion probably-asked questions and what you can do</a>	691
642	Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha,	<a href="#">with them</a> .	692
643	Tanmoy Chakraborty, Giovanni Luca Ciampaglia,	Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang	693
644	David Corney, Renee DiResta, Emilio Ferrara, Scott	Ren. 2019. Kagnet: Knowledge-aware graph net-	694
645	Hale, Alon Halevy, et al. 2023. Factuality challenges	works for commonsense reasoning. <i>arXiv preprint</i>	695
646	in the era of large language models. <i>arXiv preprint</i>	<i>arXiv:1909.02151</i> .	696
647	<i>arXiv:2310.05189</i> .	Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das,	697
648	Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng,	Daniel Khachabi, and Hannaneh Hajishirzi. 2023.	698
649	Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan	When not to trust language models: Investigating	699
650	Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion	effectiveness of parametric and non-parametric mem-	700
651	Stoica, and Eric P. Xing. 2023. <a href="#">Vicuna: An open-</a>	ories. In <i>Proceedings of the 61st Annual Meeting of</i>	701
652	<a href="#">source chatbot impressing gpt-4 with 90%* chatgpt</a>	<i>the Association for Computational Linguistics (Vol-</i>	702
653	<a href="#">quality</a> .	<i>ume 1: Long Papers</i> ), pages 9802–9822.	703
654	Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Ger-	OpenAI. 2023. <a href="#">Gpt-4 technical report</a> . <i>ArXiv</i> ,	704
655	stein, and Arman Cohan. 2023. <a href="#">Benchmark probing:</a>	<a href="#">abs/2303.08774</a> .	705
656	<a href="#">Investigating data leakage in large language models</a> .	Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng,	706
657	In <i>NeurIPS 2023 Workshop on Backdoors in Deep</i>	Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou	707
658	<i>Learning - The Good, the Bad, and the Ugly</i> .	Yu, Weizhu Chen, et al. 2023. Check your facts and	708
659	Jacob Devlin, Ming-Wei Chang, Kenton Lee, and	try again: Improving large language models with	709
660	Kristina Toutanova. 2018. Bert: Pre-training of deep	external knowledge and automated feedback. <i>arXiv</i>	710
661	bidirectional transformers for language understand-	<i>preprint arXiv:2302.12813</i> .	711
662	ing. <i>arXiv preprint arXiv:1810.04805</i> .	Ethan Perez, Saffron Huang, Francis Song, Trevor Cai,	712
663	Bhuwan Dhingra, Jeremy R. Cole, Julian Martin	Roman Ring, John Aslanides, Amelia Glaese, Nat	713
664	Eisenschlos, Daniel Gillick, Jacob Eisenstein, and	McAleese, and Geoffrey Irving. 2022. Red team-	714
665	William W. Cohen. 2022. <a href="#">Time-aware language mod-</a>	ing language models with language models. <i>arXiv</i>	715
666	<a href="#">els as temporal knowledge bases</a> . <i>Transactions of the</i>	<i>preprint arXiv:2202.03286</i> .	716
667	<i>Association for Computational Linguistics</i> , 10:257–	Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick	717
668	273.	Lewis, Majid Yazdani, Nicola De Cao, James Thorne,	718
669	Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang	Yacine Jernite, Vladimir Karpukhin, Jean Mail-	719
670	Sui, and Lei Li. 2023. Statistical knowledge assess-	lard, et al. 2020. Kilt: a benchmark for knowl-	720
671	ment for large language models. In <i>Thirty-seventh</i>	edge intensive language tasks. <i>arXiv preprint</i>	721
672	<i>Conference on Neural Information Processing Sys-</i>	<i>arXiv:2009.02252</i> .	722
673	<i>tems</i> .	Fabio Petroni, Tim Rocktäschel, Patrick Lewis, An-	723
		ton Bakhtin, Yuxiang Wu, Alexander H Miller, and	724
		Sebastian Riedel. 2019. Language models as knowl-	725
		edge bases? <i>arXiv preprint arXiv:1909.01066</i> .	726

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

731 Adam Roberts, Colin Raffel, and Noam Shazeer. 2020.  
732 [How much knowledge can you pack into the param-](#)  
733 [eters of a language model?](#) In *Proceedings of the*  
734 *2020 Conference on Empirical Methods in Natural*  
735 *Language Processing (EMNLP)*, pages 5418–5426,  
736 Online. Association for Computational Linguistics.

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

741 Xwin-LM Team. 2023. [Xwin-lm](#).

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

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

750 Weixuan Wang, Barry Haddow, Alexandra Birch,  
751 and Wei Peng. 2023. Assessing the reliability of  
752 large language model knowledge. *arXiv preprint*  
753 *arXiv:2310.09820*.

# Appendix

754

## A Appendix

755

### A.1 Cost Analysis

756

Here we highlight that the cost of GPT-4 is not counted by the number of queries, but the number of tokens. After doing a cost analysis, we conclude *GPT-4 only cost about \$20 per graph*, which is acceptable.

757

758

GPT-4 in KGLENS is only used to construct the question and verify LLM answers, and all the generation is based on knowledge graph triplets. And for yes/no questions, we simply use string matching to verify the answer. Our QG prompt using GPT-4 is around 60-100 input tokens + 10-30 output tokens, and 60 input token plus 8 output tokens for answer verification. Currently based on OpenAI website, gpt-4-0125-preview and gpt-4-1106-preview costs \$10 per 1M input tokens and \$30 per 1M output tokens. If we take an upper bound cost of our request 100 input tokens and 30 output tokens, each request costs <\$0.002. The final cost of course also depends on the size of the knowledge graph, but in our case for example, NBA knowledge graph has 2689 active edges, 22.5K API requests is sufficient to iterate over the entire edge set 8 times, and this translates to \$45 (in reality if using KGLENS, using less than half the number of API requests can get to a decent theta estimation \$20 per evaluation)

759

760

761

762

763

764

765

766

767

768

The major cost is actually the hosting cost of the target LLM. For reference 1 A100 GPU cost is around \$5 per hour, and this cost scale up easily when evaluating larger LLMs.

769

770

In addition, we believe for answer verification step, it is not a requirement to use GPT-4 (we choose it for simplicity), maybe a lower cost model could be used to replace GPT-4.

771

772

### A.2 Knowledge Graph Building and Cleaning

773

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.

774

775

776

#### A.2.1 Sampling Strategies and Preserving Data Distribution

777

Maintaining the original data distribution is important when cleaning a knowledge graph. To achieve this, random walk with both forward A.7.1 and backward A.7.2 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.

778

779

780

781

782

783

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.

784

785

786

#### A.2.2 Focus and Curated Relevance

787

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.

788

789

790

791

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.

792

793

794

795

#### A.2.3 Filtering Less Relevant Entities

796

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.

797

798

799

800

- 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.

### A.3 Uncovered Error Types

Error Type	Question	Response	Answer
<b>Factual Error</b>	On which side does the railway traffic run in Israel?	Railway traffic in Israel runs on the right-hand side.	Left side
<b>Obsolete Knowledge</b>	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
<b>Self-contradiction</b>	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
<b>Inconsistent Response</b>	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.

### A.4 Zero-sense rate and all-sense rate

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-turbo	7.26	12.73	6.77	9.44	8.51	17.52	10.37
GPT-4	7.42	12.99	6.07	<b>8.13</b>	8.35	17.67	10.11
GPT-4o	<b>5.84</b>	<b>11.86</b>	<b>7.43</b>	8.24	<b>7.63</b>	<b>16.33</b>	<b>9.56</b>

Table 6: Zero-sense rate results for different LLMs evaluated under EASY and HARD modes.

### A.5 Human Evaluation

We conduct human evaluation with an internal paid crowdsourcing 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.

LLMs	Country		NBA		Movie		Average
	EASY	HARD	EASY	HARD	EASY	HARD	
Babbage-002	54.09	33.63	43.54	23.98	53.96	29.91	39.85
Davinci-002	55.70	37.16	43.87	25.93	52.73	33.93	41.55
Vicuna-33b-v1.3	64.27	54.95	35.05	39.41	48.99	45.64	48.05
Xwin-LM-13B-V0.2	53.93	48.81	46.60	45.68	52.26	47.10	49.06
Yi-34B-Chat	64.22	55.45	56.55	55.60	57.97	54.34	57.36
GPT-3.5-turbo	68.43	60.65	50.10	51.33	59.89	56.07	57.75
GPT-4-1106-preview	79.32	69.64	71.78	65.29	80.39	64.47	71.81
GPT-4	82.24	71.19	77.44	73.63	82.44	68.58	75.92
GPT-4-turbo	80.60	71.21	<b>75.93</b>	72.84	83.42	71.91	75.99
GPT-4o	<b>82.95</b>	<b>71.86</b>	71.98	<b>73.34</b>	<b>83.91</b>	<b>72.81</b>	<b>76.14</b>

Table 7: All-sense rate results for different LLMs evaluated under EASY and HARD modes.

### 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.

816

## A.6 Prompt

817

### 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>*

818

### 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>

## A.7 Wikidata Web Query

### A.7.1 Forward Walk

```
1 SELECT DISTINCT ?subject ?subjectLabel ?subjectDesc ?predicate ?predicateLabel ?
826 predicateDesc ?object ?objectLabel ?objectDesc
827 2 WHERE {{
828 3   VALUES ?subject {{
829 4     {values}
830 5   }}
831 6   ?subject ?predicate ?object .
832 7   ?subject rdfs:label ?subjectLabel .
```

8	?subject schema:description ?subjectDesc .	833
9	?property wikibase:directClaim ?predicate .	834
10	?property rdfs:label ?predicateLabel .	835
11	?property schema:description ?predicateDesc .	836
12	?object rdfs:label ?objectLabel .	837
13	?object schema:description ?objectDesc .	838
14	FILTER (lang(?subjectLabel) = "en")	839
15	FILTER (lang(?subjectDesc) = "en")	840
16	FILTER (lang(?predicateLabel) = "en")	841
17	FILTER (lang(?predicateDesc) = "en")	842
18	FILTER (lang(?objectLabel) = "en")	843
19	FILTER (lang(?objectDesc) = "en")	844
20	}}	845
21	ORDER BY UUID()	846
22	LIMIT {limit}	847

## A.7.2 Backward Walk

1	SELECT DISTINCT ?subject ?subjectLabel ?subjectDesc ?predicate ?predicateLabel ?	849
	predicateDesc ?object ?objectLabel ?objectDesc	850
2	WHERE {{	851
3	VALUES ?object {{	852
4	{values}	853
5	}}	854
6	?subject ?predicate ?object .	855
7	?subject rdfs:label ?subjectLabel .	856
8	?subject schema:description ?subjectDesc .	857
9	?property wikibase:directClaim ?predicate .	858
10	?property rdfs:label ?predicateLabel .	859
11	?property schema:description ?predicateDesc .	860
12	?object rdfs:label ?objectLabel .	861
13	?object schema:description ?objectDesc .	862
14	FILTER (lang(?subjectLabel) = "en")	863
15	FILTER (lang(?subjectDesc) = "en")	864
16	FILTER (lang(?predicateLabel) = "en")	865
17	FILTER (lang(?predicateDesc) = "en")	866
18	FILTER (lang(?objectLabel) = "en")	867
19	FILTER (lang(?objectDesc) = "en")	868
20	}}	869
21	ORDER BY UUID()	870
22	LIMIT {limit}	871

## A.8 Additional Figures

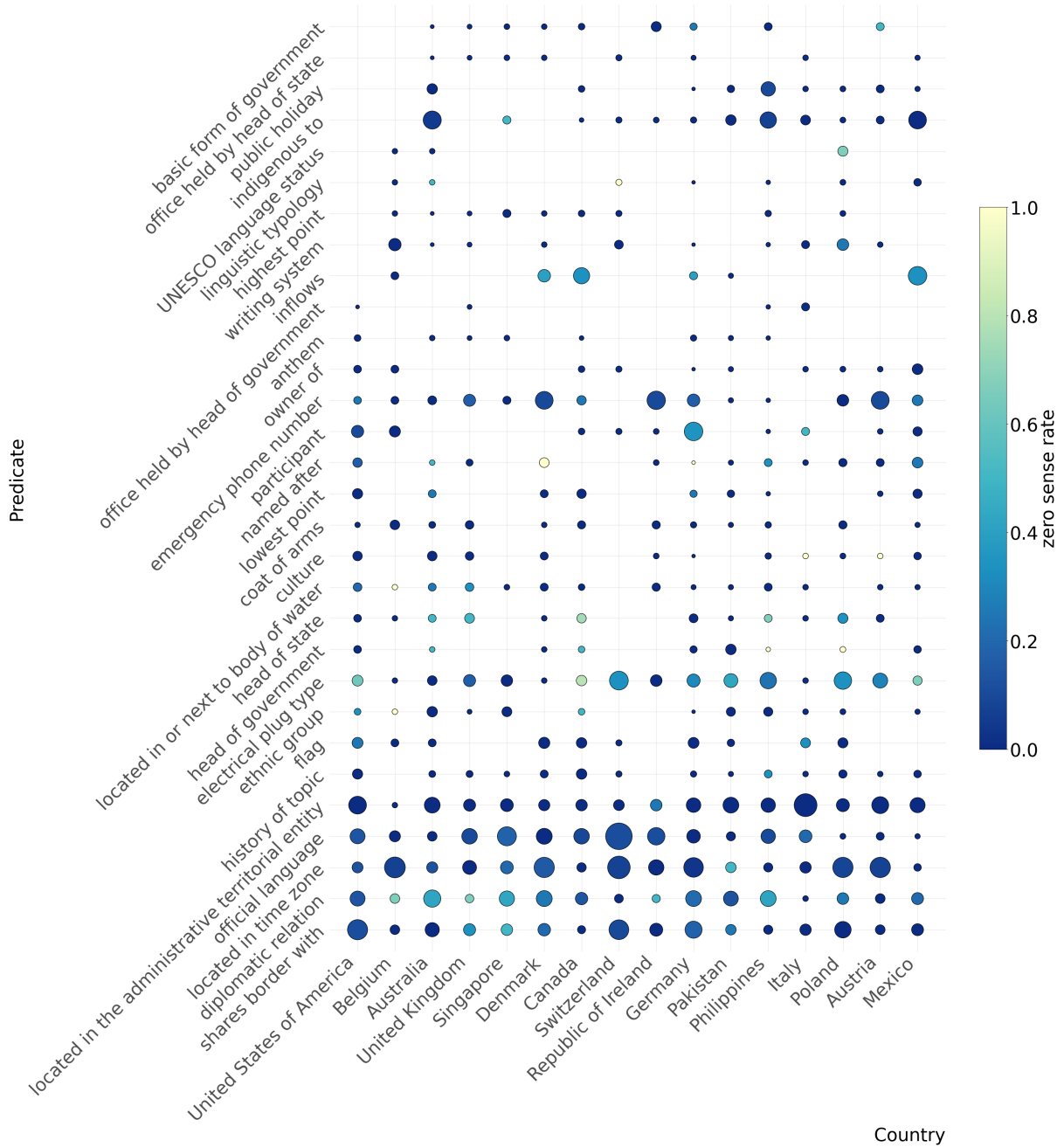


Figure 6: Predicate level knowledge proficiency of GPT-4 evaluated under HARD difficulty. The darker color indicates a lower zero-sense rate. The dot size shows the proportional size of the number of edges in the predicate sub-group.



	<b>gpt-4-1106-preview</b>	<b>gpt-4</b>	<b>gpt-3.5-turbo</b>	<b>davinci-002</b>	<b>babbage-002</b>
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 7: Country KG EASY-level zero-sense rate grouped by countries.

	<b>gpt-4-1106-preview</b>	<b>gpt-4</b>	<b>gpt-3.5-turbo</b>	<b>davinci-002</b>	<b>babbage-002</b>
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 8: Country KG HARD-level zero-sense rate grouped by countries.

	<b>gpt-4-1106-preview</b>	<b>gpt-4</b>	<b>gpt-3.5-turbo</b>	<b>davinci-002</b>	<b>babbage-002</b>
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 9: Movie KG EASY-level zero-sense rate grouped by years.

	<b>gpt-4-1106-preview</b>	<b>gpt-4</b>	<b>gpt-3.5-turbo</b>	<b>davinci-002</b>	<b>babbage-002</b>
2015	21.05	18.92	22.22	58.70	58.70
2016	25.53	22.22	31.11	54.84	51.35
2017	19.44	18.87	39.29	57.53	64.10
2018	34.92	29.03	46.27	51.72	55.81
2019	25.58	18.75	33.33	62.16	51.52
2020	26.67	10.81	30.56	61.11	54.05
2021	31.34	29.58	51.02	63.20	60.00
2022	33.33	37.14	57.60	53.53	63.78
2023	41.12	33.65	54.95	55.56	69.15

Figure 10: Zero-sense rate grouped by years for moive KG in HARD mode.

	<b>gpt-4-1106-preview</b>	<b>gpt-4</b>	<b>gpt-3.5-turbo</b>	<b>davinci-002</b>	<b>babbage-002</b>
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 11: NBA EASY-level zero-sense rate grouped by teams

	<b>gpt-4-1106-preview</b>	<b>gpt-4</b>	<b>gpt-3.5-turbo</b>	<b>davinci-002</b>	<b>babbage-002</b>
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 12: NBA HARD-level zero-sense rate grouped by teams