Benchmarking Knowledge Boundary for Large Language Model: A Different Perspective on Model Evaluation

Anonymous ACL submission

Abstract

In recent years, substantial advancements have been made in the development of large language models, achieving remarkable performance across diverse tasks. To evaluate the knowledge ability of language models, previous studies have proposed lots of benchmarks based on question-answering pairs. We argue that it is not reliable and comprehensive to evaluate language models with a fixed question or limited paraphrases as the query, since language models are sensitive to prompt. Therefore, we introduce a novel concept named knowledge boundary to encompass both prompt-agnostic and prompt-sensitive knowledge within language models. Knowledge boundary avoids prompt sensitivity in language model evaluations, rendering them more dependable and robust. To explore the knowledge boundary for a given model, we propose projected gradient descent method with semantic constraints, a new algorithm designed to identify the optimal prompt for each piece of knowledge. Experiments demonstrate a superior performance of our algorithm in computing the knowledge boundary compared to existing methods. Furthermore, we evaluate the ability of multiple language models in several domains with knowledge boundary.

1 Introduction

Recently, large language models (LLMs) have made significant advancements in a variety of tasks (Brown et al., 2020; Thoppilan et al., 2022; Bubeck et al., 2023). In order to gain deeper insights into the knowledge capabilities of different LLMs to help select appropriate LLM in practice, numerous studies have proposed various benchmarks for LLM evaluation (Guo et al., 2023; Zhong et al., 2023). The majority of previous research on model evaluation constructs a test dataset sourced from standardized examinations, such as college entrance exams and law school admission tests (Hendrycks et al., 2021). Subsequently, the questions are fed to LLMs as prompts, eliciting responses that are then scored for evaluation (Yu et al., 2023; Zhang et al., 2023).

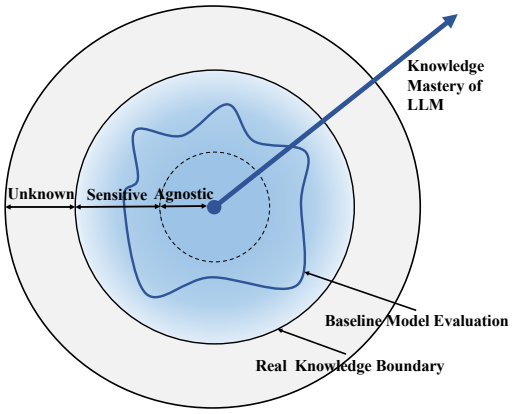


Figure 1: Illustration of three classes of knowledge based on the model's mastery of knowledge in different textual forms. Existing evaluation methods suffer from sensitivity to input prompt. Therefore, the knowledge ability depicted by these methods is irregularly shaped. We propose to evaluate the knowledge capacity with a knowledge boundary containing both Prompt-Agnostic Knowledge and Prompt-Sensitive Knowledge.

However, each piece of knowledge embodies **abstract concept** that can be expressed in a nearly infinite number of **textual forms** (Phenix, 1967). When evaluating a specific piece of knowledge, existing work only evaluated LLMs with one or several textual forms randomly sampled from the semantic space of the knowledge. However, existing LLMs are notorious for being sensitive to prompt, thereby undermining the reliability of such evaluations (Ji et al., 2023; Maharana et al., 2023; Chang and Bergen, 2023; Chen et al., 2023). Consequently, current studies on model evaluation are reasonably considered to be insufficiently robust.

As shown in Figure 1, from the perspective of the model's mastery of the textual form of knowledge, knowledge can be divided into three classes:

1) **Prompt-Agnostic Knowledge** that can be correctly answered for any textual form; 2) **Prompt-Sensitive Knowledge** that is sensitive to the form of the prompt fed into the model; 3) **Unanswerable Knowledge** that is unable to be answered by the model, regardless of the prompt employed. The majority of previous research on model evaluation ignored the presence of Prompt-Sensitive Knowledge, resorting to oversimplified binary evaluations, classifying the model's knowledge mastery merely as true or false. Dong et al. (2023) attempts to assess LLM through diverse paraphrases, yet these evaluations remain confined to limited textual forms of knowledge.

In this paper, we aim to reduce the contingency when evaluating LLMs. Different from previous paradigms of LLM evaluation, we attempt to explore these Unanswerable Knowledge of the model to be evaluated, thereby illuminating the knowledge boundaries of LLMs. How can we find Unanswerable knowledge for the model? It is obvious that trying all prompts for the knowledge to query the model is too resource-intensive. Therefore, we choose to make efforts to search for the optimal prompt. We formalize optimal prompt searching as a discrete optimization problem: given some question paraphrases, we search for a prompt to maximize the probability of generating the correct answer. We propose Projected Gradient Descent method with Constraints (PGDC), a new algorithm that updates prompt with gradient descent and implements proximal projection to search discrete prompts. To ensure that the optimized prompt has the same semantics as the original prompt, we introduce the semantic loss, which is a measure of the distance between the semantic representations of the optimized prompt and the original prompt.

Experimental results demonstrate that our proposed PGDC can outperform baselines in depicting knowledge boundaries. In addition, results on counterfactual datasets demonstrate that our approach is reasonable and robust. Human evaluation also reveals that our optimized prompts generally have the same semantics as the original questions. Moreover, we delineate models' knowledge boundaries in different domains using PGDC to evaluate LLMs. The size of the model's domain knowledge boundaries is strongly associated with the performance of downstream tasks in the domain. The optimal prompts also have some patterns that can give some inspiration for designing prompts when using corresponding LLMs.

In summary, our contributions are: (1) We propose a new evaluation paradigm for benchmarking knowledge boundaries to compare models' capabilities, which can reduce the randomness in current evaluations. (2) We design PGDC, projected gradient descent method with constraints, to optimize prompts and obtain knowledge boundaries of LLMs which achieves the best results on four datasets. (3) We evaluate five models using knowledge boundaries and obtain some valuable findings.

Our code and data will be released to the community to facilitate future research.

2 Preliminaries

2.1 Knowledge Boundary Requirements

We attempt to calculate the knowledge boundary of LLM by automatically constructing the optimal prompt. As various methods for prompt engineering have been proposed to obtain better prompt as query (Dong et al., 2022; Wei et al., 2023), not all of them are suitable for calculating the knowledge boundary of LLM. In this section, we propose four basic requirements for the algorithm applied to the calculation of knowledge boundaries: Universality, Truthfulness, Robustness and Optimality.

Universality When searching for an optimal prompt, the method should work for most current LLMs, regardless of its size and architecture.

Truthfulness The constructed prompt should share the same semantics as the original question, and not be allowed to change subject or relation.

Robustness When searching for the optimal prompt for a piece of knowledge, the effectiveness of the method should be relevant to the knowledge capacity of LLM. In other words, the algorithm should tend not to find appropriate prompt for unanswerable knowledge.

Optimality The algorithm should search for as much prompt-sensitive knowledge in the LLM as possible.

2.2 Problem Formulation

In this section, we give a formal problem formulation of searching for the optimal prompt. For a given piece of knowledge, assume we have an LLM that models next-token probability $P(x_i|x_1, x_2, ..., x_{i-1})$ with an input sequence

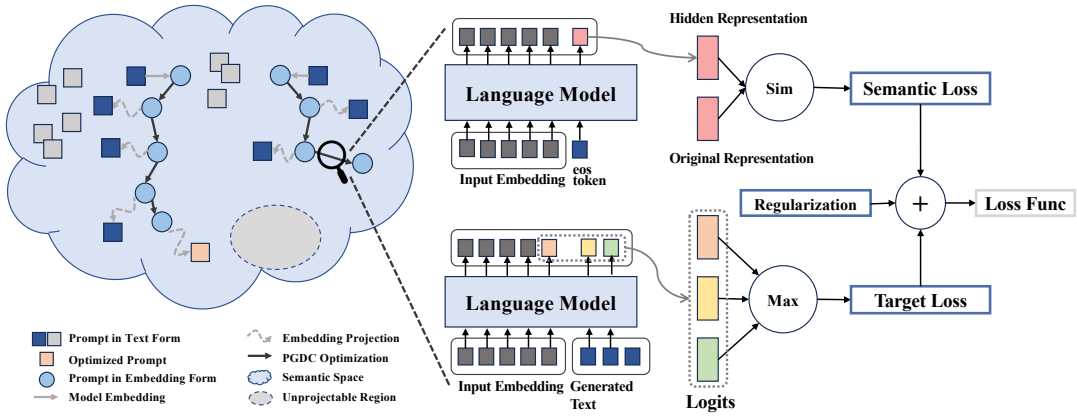


Figure 2: An illustration of our PGDC method, projected gradient descent method with constraints. The left part of the image shows the overall framework of our method: we start from a few labeled prompts, perform gradient descent with the target answer as the optimization goal and try to project the embedding into text form, while ensuring that the whole search process is in the same semantic space of the expression of the target knowledge. The right side of the image shows how our loss function is calculated at each step of gradient descent.

$(x_1, x_2, ..., x_{i-1})$. The piece of knowledge is expressed in various textual forms to construct a QA set of multiple questions and answers. Different questions in the QA set are paraphrases, while answers are aliases. We believe that if the model is able to answer one of the questions correctly, it is possible for the model to "know" this piece of knowledge. Therefore, if the model is able to generate one of the correct answers with prompt semantically similar to one of the questions, we consider the knowledge within its knowledge boundary.

To illustrate the problem, we start from the simpler case with only one question $Q = \{q_1, q_2, ..., q_n\}$ and one answer $A = \{a_1, a_2, ..., a_m\}$. Prompt X is initialized with Q and optimized to maximize the probability of generating A while remaining semantically similar to Q. We formalize optimal prompt searching as the problem:

$$\min_X \Phi(X) = L(X, A) + \lambda R(X, Q), \quad (1)$$

where $L(\cdot)$ denotes the loss function to penalize unsuccessful generation. $R(\cdot)$ indicates the semantic distance between the optimized prompt and the initial prompt while λ is the penalty factor.

3 Method

To obtain a better knowledge boundary for LLM, our effort is directed towards identifying the optimal prompt within the semantic space. As illustrated in Figure 2, PGDC optimizes prompt in the neighbour semantic space of the original question. The prompt in text form is first mapped to prompt in embedding form as continuous text embedding. During PGDC optimization, the text embedding is updated through gradient descent with direction of the loss function $\Phi(\cdot)$. After each update, if the text embedding is close to a discrete prompt, it is projected to the discrete prompt through embedding projection. To avoid the text embedding from entering unprojectable region where there are no close discrete prompts to project, we introduce a regularization to force the embedding not to enter these regions. After multiple iterations of update, we get the final optimized prompt.

3.1 PGDC Optimization

In PGDC algorithm, we do not specify the positions of the answer in our LLM-generated output, which relaxes constraints in the model output and leaves space for the LLM to generate reasoning process and do inference. Therefore, we define the target loss of generating a specific answer A with a slicing window method:

$$L = \min_{j<=t-k_i+1} -\log P(O_{j:j+k_i} = A), \quad (2)$$

where $O = \{o_1, o_2, ..., o_t\}$ denotes the output of the LLM given X as the input. In this way, PGDC automatically searches for the target position in the model output and optimizes the probability of generating answer. When there exist multiple answers in the answer set A^*, we optimize the answer with the highest probability to be generated:

$$L = \min_{A \in A^*} \min_{j<=t-k_i+1} -\log P(O_{j:j+k_i} = A). \quad (3)$$

We separately optimize prompts with PGDC if there are multiple paraphrases of questions in the piece of knowledge.

Since PGDC optimizes prompts in the continuous embedding space while text space is discrete, it is hard for methods of automatically searching for prompts to constrain semantic information (Shin et al. (2020); Jones et al. (2023)). To combat the challenge, we introduce a semantic constraint to the loss function, which is defined as:

$$R(X, Q) = ||h(X) - h(Q)||_2, \quad (4)$$

where $h(\cdot)$ is the hidden representation of prompt and $|| \cdot ||_2$ denotes the L2 distance between two items. As illustrated in Figure 2, the hidden representation is obtained with the last hidden layer output of the LLM given the concatenation of prompt and a <eos> character.

As the optimization process is implemented in the continuous space, it is necessary to project the embedding into discrete tokens. The optimized embedding obtained might enter the unprojectable region shown in Figure 2, which makes the projection hard. Therefore, we add a regularization in the loss function to punish prompt embedding far from discrete tokens:

$$\delta(X) = \Sigma_{i=1}^{N} \min_{v \in \mathcal{V}} ||\hat{x}_i - Wv||_2, \quad (5)$$

where $\mathcal{V}$ denotes the vocabulary of LLM and W is the projection from vocab to embedding space. The lowercase letters such as x represent tokens while $\hat{x}$ represents its embedding. N in Equation 5 denotes the length of the prompt X which varies in different iterations.

In general, the final loss function of PGDC is formulated as:

$$\Phi(X) = L(X, A) + \lambda_1 R(X, Q) + \lambda_2 \delta(X) \quad (6)$$

3.2 Proximal Projection

Instead of projecting the prompt into text space after the overall optimization (Guo et al., 2021) or conducting projection after each iteration (Cheng et al., 2020), PGDC achieves flexible transformation of embedding space to text space with a threshold of the vector distance. Formally, the transformation can be written as:

$$\hat{x}_i = \begin{cases} Wv, & \min_{v \in \mathcal{V}} ||\hat{x}_i - Wv||_2 < c \\ \hat{x}_i, & \min_{v \in \mathcal{V}} ||\hat{x}_i - Wv||_2 >= c, \end{cases} \quad (7)$$

where c represents the threshold of the $L2$ distance. As illustrated in Figure 2, the dashed line shows the proximal projection process.

3.3 Algorithm Summary

In general, PGDC iteratively optimizes prompt in the embedding space with gradient descent to minimize the loss function in Equation 5 and do proximal projection after each iteration. A detailed pseudocode is shown in Appendix C.

4 Experiments

In this section, we perform comparisons between our method and baseline methods which are commonly used in model evaluation on common knowledge benchmarks and unanswerable knowledge benchmarks. We also conduct a manual evaluation to check whether the semantics of the prompts we obtained are consistent with the original question.

4.1 Datasets and Models

Common Knowledge Benchmarks In order to evaluate the performance of different methods on common knowledge, we choose to use KAssess (Dong et al., 2023) and PARAREL (Elazar et al., 2021) for our evaluation. Both of them consist of knowledge tuples and hand-curated prompt templates, where all subjects, relations, and objects exist as entities in WikiData.

Unanswerable Knowledge Benchmarks To test whether our optimized prompts leak answers or induce hallucinations that cause LLMs to answer knowledge that they could not answer originally, we perform evaluations on two counterfactual datasets, COUNTERFACT (denoted as CFACT) (Meng et al., 2022) and ALCUNA (Yin et al., 2023). CFACT containes 20K counterfactual knowledge records with a diverse set of subjects, relations, and linguistic variations. ALCUNA is a biological dataset for evaluating the ability of the model in face of new knowledge.

The above datasets have multiple expressions for each knowledge query, except for ALCUNA.

Models Our experiments use GPT-2 (774M) (Radford et al., 2019), GPT-J (6B) (Wang and Komatsuzaki, 2021), LLaMA2 (7B) (Touvron et al., 2023b), and Vicuna (7B) (Chiang et al., 2023).

4.2 Baseline Methods

There are several common methods of assessing the model's mastery of knowledge, and we use the

Dataset	Model	Method						
		zero	few	dis	P-zero	P-few	P-dis	**PGDC(ours)**
PARAREL↑	LLaMA2	34.43%	58.23%	17.96%	54.78%	66.95%	44.16%	**71.36%**
	Vicuna	34.19%	59.56%	8.40%	54.97%	**69.69%**	23.06%	69.63%
	GPT-J	23.23%	44.84%	2.40%	40.78%	54.06%	7.25%	**55.95%**
	GPT-2	9.27%	12.61%	3.13%	18.18%	20.46%	9.01%	**47.68%**
KAssess↑	LLaMA2	23.69%	32.03%	6.73%	50.00%	50.75%	24.73%	**69.84%**
	Vicuna	23.21%	33.37%	9.90%	51.15%	53.66%	38.20%	**57.63%**
	GPT-J	15.95%	20.47%	12.67%	40.23%	38.20%	2.26%	**48.62%**
	GPT-2	4.03%	3.64%	2.46%	13.66%	11.44%	15.75%	**24.71%**
CFACT↓	LLaMA2	**1.32%**	4.56%	13.88%	3.30%	9.31%	36.46%	3.41%
	Vicuna	**1.40%**	3.08%	4.95%	3.36%	6.91%	14.28%	3.50%
	GPT-J	**1.39%**	3.18%	2.30%	3.75%	6.12%	6.74%	4.82%
	GPT-2	**1.10%**	1.77%	3.28%	3.00%	3.92%	9.32%	2.81%
ALCUNA↓	LLaMA2	**0.00%**	0.63%	30.48%	**0.00%**	0.63%	30.48%	**0.00%**
	Vicuna	**0.00%**	0.80%	0.90%	**0.00%**	0.80%	0.90%	**0.00%**
	GPT-J	**0.00%**	0.08%	0.72%	**0.00%**	0.08%	0.72%	**0.00%**
	GPT-2	**0.00%**	0.30%	2.10%	**0.00%**	0.30%	2.10%	**0.00%**

Table 1: The success rate of constructing prompts to elicit specific knowledge on four Datasets. We conduct experiments on four different LLMs to illustrate the performance of our proposed PGDC. Dataset PARAREL and KAssess provide true knowledge to characterize the ability of different methods on obtaining knowledge boundary while the pieces of knowledge in dataset CFACT and ALCUNA are fake which shows the robustness of PGDC.

following as baselines:

Zero-Shot (zero) Zero shot prompting is the simplest and most common approach used in previous evaluation work. We directly query models using questions from benchmarks.

Few-Shots (few) Few shots prompting is commonly used to enhance model performance by utilizing the contextual learning capabilities of LLMs. We retrieve similar knowledge in the dataset as exemplars to feed to the model.

Discriminator (dis) We can also use the judgment question format to assess whether a model knows one piece of knowledge. So we provide LLM with one knowledge statement and let it determine whether this statement is correct or incorrect.

Since there are several paraphrases for each knowledge query in the benchmarks, for each above baseline, we will use two different metrics to simulate previous work on model evaluation: 1) For each knowledge query, we will randomly select one of its expressions for evaluation. 2) For each knowledge query, as long as one of its paraphrases can be answered correctly, the knowledge is considered to be inside the boundaries, and the baseline method using this metric is denoted as P-baseline.

A more detailed description of the dataset as well as the implementation of the baseline methods, are shown in Appendix A. The hyperparameter settings for the PGDC are shown in Appendix B.

4.3 Results and Analysis

Table 1 summarizes the experimental results on four different LLMs. Our proposed PGDC achieves the highest performance on common knowledge benchmarks on almost all LLMs. The results indicate that the knowledge boundary found by our method is more comprehensive than baseline methods, which shows the *Optimality* and *Universality* of PGDC. The experimental results on unanswerable knowledge benchmarks including CFACT and ALCUNA reflect the *Robustness* of different prompt methods. PGDC only slightly raises the amount of unanswerable knowledge over zero-shot baseline, which shows that our proposed method will only introduce relatively limited fake knowledge and meets the *Robustness* requirement. PGDC considers knowledge to be within the boundaries if any of its paraphrases can be answered correctly for each knowledge query and shows comparative results with P-zero while outperforming P-few and

P-dis. Moreover, the prompts generated by PGDC are generally consistent semantically with the original questions (as shown in Section 4.5). Therefore, PGDC meets all four fundamental criteria (Universality, Truthfulness, Robustness, and Optimality) in calculating knowledge boundaries.

In addition, we can observe that:

Evaluating LLMs with a fixed question or limited paraphrases as the query is not reliable and comprehensive According to the zero and P-zero results of the PARAREL and the KAssess dataset, we can see that different queries yield different results, and different prompting methods may result in different inter-system rankings. This suggests that assessing LLMs using a predetermined question or restricted paraphrases as the query lacks reliability and comprehensiveness. Evaluating LLMs with a fixed question or limited paraphrases may lead to the selection of suboptimal LLMs for practical applications, demonstrating the necessity of optimizing prompt design to seek more realistic knowledge boundaries.

Discrimination format is much less reliable than cloze-style format P-dis has a similar proportion of responses as true on common and unanswerable knowledge benchmarks, which correlates with the model's preference for true. This observation aligns with previous findings by Wu and Aji (2023); Wang et al. (2023).

Different models prefer different prompts Since traditional model evaluation methods use fixed queries, the model's preference for prompt affects the score. The difference between P-zero and zero then reflects the fact that the model is sensitive to prompt. Even for queries with the same meaning, different ways of asking can produce different results. GPT-2 also acquires a fair amount of knowledge, but is overly sensitive and thus scores lower on traditional assessment methods.

Manual design of a good prompt is difficult Few shots prompting induces more knowledge than zero-shots. However, it is difficult to verify how to select good examples and whether a good enough prompt has been designed.

PGDC, on the other hand, uses cloze-style problem and automatically searches for the optimal prompt for different models, so it is a much better approach to model evaluation.

We also analyze the knowledge detected by PGDC as well as the knowledge detected by baselines on KAssess. We categorize relations according to KAssess, and analyze the accuracy of PGDC and baseline methods on various relation categories. The results of PGDC and the strong baseline method P-few on the strong LLaMA2 model are shown in Figure 3, while the coverage results of other baseline methods are presented in the Appendix D. We find that the knowledge boundaries we obtained can almost cover baselines. Moreover, we also record the iterations on KAssess to find the optimized prompt using PGDC in Figure 4. We observe that PGDC can find the optimal prompt for the majority of queries within 15 iterations.

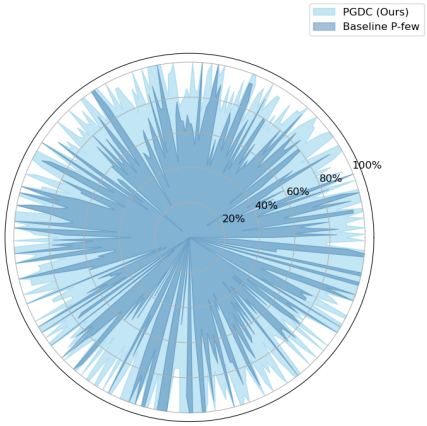


Figure 3: Knowledge boundaries of PGDC and baseline method P-few on KAssess using LLaMA2 model.

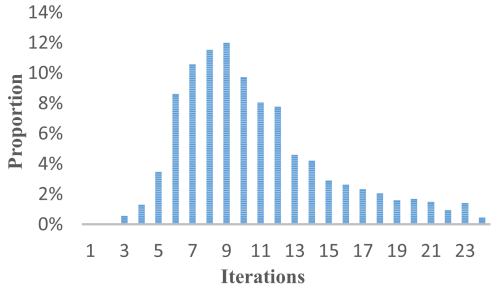


Figure 4: Iterations on KAssess to find the optimized prompt using PGDC with LLaMA2 model.

4.4 Comparison with Prompt Optimization Method

Since our method is a prompt optimization type of method, we conduct experiments to compare the robustness of PGDC and Autoprompt (Shin et al., 2020), a representative method of prompt optimization. Autoprompt is a Hotflip-based algorithm (Ebrahimi et al., 2018) in optimizing prompt, which employs several trigger tokens to elicit the target

[Original Prompt] The associated item of source code is;	[Answer] program
[PGDC Prompt] The early item of source code is a simple;	[Optimal Paraphrase]
[Original Prompt] Isatis tinctoria is a source of;	[Answer] indigo
[PGDC Prompt] Isatis tinctoria, a source of the natural dye;	[Reasoning and Inference]
[Original Prompt] The host country of Australian Capital Territory is;	[Answer] Australia
[PGDC Prompt] <s> host country of Australian Capital Territory is;	[Format and Stop Words]

Table 2: Cases that PGDC successfully updates the prompt. We summarize the advantages of PGDC into three aspects : 1) Optimal Paraphrase; 2) Reasoning and Inference; 3) Format and Stop Words.

Dataset	CFACT↓			
Model	GPT-2	GPT-J	LLaMA2	Vicuna
Autoprompt	92.38%	85.67%	88.35%	33.09%
PGDC	**2.81%**	**4.82%**	**3.41%**	**3.50%**

Table 3: Comparison of PGDC and AutoPrompt on CFACT dataset.

output. The exact experimental setup is shown in Appendix E.

As shown in Table 3, we can find that Autoprompt induces the model to output target answers on counterfactual datasets in a large percentage. This result suggests that Autoprompt is more similar to an adiversarial attack algorithm that are committed to getting the target answer, while PGDC optimizes the prompt within the semantic constraint.

4.5 Semantic Preservation Evaluation

In order to check whether the prompt obtained by PGDC is semantically consistent with the original questions (*Truthfulness*), we perform a manual evaluation. We randomly select 200 samples from the PARAREL dataset. Specifically, we enlisted three college students who hold English qualification certificates. Initially, they were given an evaluation guideline, which is detailed in Appendix F. Each evaluator underwent a training process to improve their comprehension of the annotation procedure. Prior to annotation, we administered a qualification test comprising 10 samples; only annotators who passed this test were deemed qualified and permitted to proceed with annotation.

The human evaluation results show that the semantic preservation rates of GPT-2, GPT-J, LLaMA2, and Vicuna are respectively 80.5%, 85.1%, 83.3%, and 86.2%. This indicates that the prompts generated by PGDC are generally semantic consistent with the original questions, which demonstrates the general Truthfulness of PGDC. More details about the human evaluation are shown in Appendix F.

4.6 Case Study

To understand how PGDC steers question prompts to generate desired answers, we manually study cases in which PGDC successfully updates the prompt. We summarize the advantages of PGDC into three aspects and provide cases in Table 2: 1) Finding the optimal paraphrase of the original prompt. Due to human resource constraints, it is impossible to enumerate all paraphrases of the original question. PGDC automatically searches for the optimal paraphrase that elicits correct answers. 2) Leaving space for LLM to reason and infer. PGDC allows LLMs to generate some tokens to assist their reasoning and inference to achieve the answer. 3) Changing the format and stop words in the original prompt. Some special tokens and stop words vary in different LLMs, which can be hard for humans to detect. PGDC is able to optimize format and stop words on the basis of gradient.

5 Assessments of LLMs

The above experiments have demonstrated the effectiveness of PGDC in detecting knowledge and the reasonableness of the obtained optimal prompt. In this section, we apply PGDC on MMLU (Hendrycks et al., 2021) to evaluate LLMs.

5.1 Experimental Settings

We evaluate GPT-2 (774M), GPT-J (6B), LLaMA2 (7B), Vicuna (7B) and Mistral (7B) (Jiang et al., 2023) from the perspective of 30 refined domain knowledge using MMLU[1].

To be consistent with our approach, we modify the questions in MMLU from choice questions to a cloze format, which yields more reliable and stable assessment results. Following previous work

[1]MMLU covers 57 subjects. To fit the theme of our paper, here we have selected 30 topics related to knowledge, dropping topics such as computation and reasoning, for analysis.

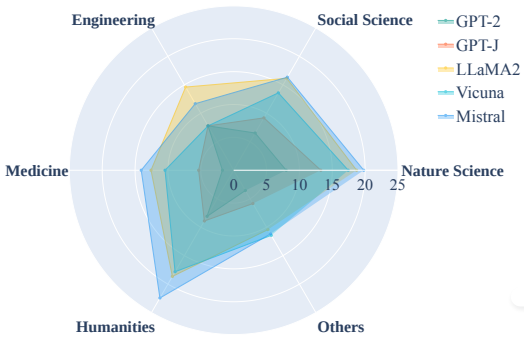


Figure 5: Knowledge boundaries of different domains of models on MMLU.

(Anil et al., 2023; Touvron et al., 2023a), we categorize the questions in MMLU into six types of topics: natural sciences, medical and biological sciences, computer science and logic, social sciences, humanities, and others. More details of the experiment are shown in Appendix G.1.

5.2 Results

The results[2] of our model evaluation on each of the broad categories are demonstrated in Figure 5, and more detailed scores are shown in Appendix G.2.

We can find that Mistral has the largest knowledge boundaries overall. LLaMA2 exceeds the other models by a lot in engineering domain. It may be because LLaMA2 uses a lot more new labeled code data for training. Vicuna performs not far behind LLaMA2 on other topics. GPT-2 has very small knowledge boundaries and performs poorly in the more specialized medical domain. By identifying more reliable knowledge boundaries, we help select the appropriate LLM in practice.

However, it is worth noting that these scores are quite low (around 20 points). This is due to the fact that we use more difficult cloze-style questions for reliability. It also reflects the fact that the results obtained from choice-style benchmark may be too high. The problems in MMLU are relatively specialized, and the current general-purpose models do not have a lot of knowledge in the relevant areas.

6 Related Work

Model Evaluation Several benchmarks have been proposed to evaluate Large Language Models (LLMs) on human exams like college entrance and law school admission tests (Suzgun et al., 2022; Srivastava et al., 2023; Choudhary and Reddy, 2023; Zhong et al., 2023). In terms of knowledge assessment, LAMA (Petroni et al., 2019) evaluates whether models can correctly predict masked object entities in a cloze-style prompt. Some studies (Onoe et al., 2021; Mallen et al., 2023; Arodi et al., 2023; Yu et al., 2023) focus on measuring LLMs' understanding and mastery of world knowledge. These benchmarks do not take into account that LLMs are sensitive to different prompts. Some works (Elazar et al., 2021; Dong et al., 2023) focus on estimating and measuring the consistency of LLMs given diverse prompts. All previous studies on model evaluation use fixed prompts, and our work pioneers prompt optimization for evaluating LLMs' knowledge boundaries.

Prompt Optimization Due to the sensitivity of language models to prompts, better prompts can help achieve higher performance in specific tasks (Deng et al. (2022); Wei et al. (2023); Yang et al. (2023)). Prompt engineering like in-context learning greatly improves the performance of prompt methods (Dong et al., 2022). Another related line of work attempts to formalize prompt searching as a discrete optimization task to achieve better performance in specific tasks (Shin et al., 2020). Some studies adopt Hotflip-based algorithms (Ebrahimi et al., 2018) to automatically construct prompts (Wallace et al. (2019); Shin et al. (2020); Jones et al. (2023);). In addition, several work tries to optimize prompts in continuous embedding space with Gumbel-softmax trick (Guo et al., 2021) and projection (Cheng et al. (2020); Wen et al. (2023)).

7 Conclusion

The sensitivity of LLMs to prompt leads to the unreliability of the results obtained from traditional model evaluation works that use fixed queries to evaluate the model. To address this problem, we propose semantics-preserving prompt optimization methods, PGDC, to find the knowledge boundaries of models for model evaluation. Our experiments demonstrate shortcomings of previous model evaluation methods and the fact that the prompt we find is superior to the fixed prompt. At the same time, the prompt found by our method maintains the original semantics and does not induce knowledge that is not captured by the model, which outperforms previous prompt optimization efforts. Moreover, we conduct experiments exploring the boundaries of the model's different domain knowledge and compare and analyze the LLM's capabilities.

[2]For display purposes, our radar graph ranges 1-25%.

Limitations

According to our definition, one can say that a model knows this knowledge when it can answer the corresponding question with the optimal prompt. In this paper, we only aim to find the Unanswerable Knowledge of the model as the knowledge boundary. In fact, for Prompt-sensitive knowledge, the model's sensitivity also reflects the model's mastery of it (the knowledge in the color gradient in Figure 1). At this stage we would like to have a clear boundary, so we have not considered this part for now. But exploring this part of knowledge is an interesting and important future work.

Ethics Statement

A potential negative impact of our approach is that malicious attackers could use our method to attack public large pre-trained language models, leading to fake knowledge generation. As pre-trained language models advance in many tasks, addressing safety concerns becomes increasingly necessary and imperative. Analyses in our paper can help enhance the evaluation of pre-trained language models.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, et al. 2023. Palm 2 technical report.

Akshatha Arodi, Martin Pömsl, Kaheer Suleman, Adam Trischler, Alexandra Olteanu, and Jackie Chi Kit Cheung. 2023. The kitmus test: Evaluating knowledge integration from multiple sources in natural language understanding systems.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, et al. 2020. Language models are few-shot learners.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Tyler A. Chang and Benjamin K. Bergen. 2023. Language model behavior: A comprehensive survey.

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

Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples.

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.

Nurendra Choudhary and Chandan K. Reddy. 2023. Complex logical reasoning over knowledge graphs using large language models.

Zhenyun Deng, Yonghua Zhu, Yang Chen, Qianqian Qi, Michael Witbrock, and Patricia Riddle. 2022. Prompt-based conservation learning for multi-hop question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1791–1800, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang Sui, and Lei Li. 2023. Statistical knowledge assessment for large language models.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

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

Joseph L Fleiss, Bruce Levin, Myunghee Cho Paik, et al. 1981. The measurement of interrater agreement. *Statistical methods for rates and proportions*, 2(212-236):22–23.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically auditing large language models via discrete optimization.

Adyasha Maharana, Amita Kamath, Christopher Clark, Mohit Bansal, and Aniruddha Kembhavi. 2023. Exposing and addressing cross-task inconsistency in unified vision-language models.

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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36.

Yasumasa Onoe, Michael J. Q. Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Philip H. Phenix. 1967. The architectonics of knowledge. *Quest*, 9(1):28–41.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, et al. 2022. Lamda: Language models for dialog applications.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery.

Minghao Wu and Alham Fikri Aji. 2023. Style over substance: Evaluation biases for large language models.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. ALCUNA: Large language models meet new knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1397–1414, Singapore. Association for Computational Linguistics.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. 2023. Kola: Carefully benchmarking world knowledge of large language models.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023. Benchmarking large language models for news summarization.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models.

A Datasets and Baselines

A.1 Datasets

We conduct comparative experiments between our method and the baseline on four datasets. The datasets are described and detailed below:

KAssess KAssess (Dong et al., 2023) is a large-scale assessment suite with 994,123 entities, 600 relations, and their text aliases which are obtained from T-REx knowledge graph (Elsahar et al., 2018). KAssess constructs multiple paraphrased templates for each relation. In total, there are 3,488 templates for 600 relations, with an average of 5.82 paraphrased templates per relation.

PARAREL PARAREL (Elazar et al., 2021) is also a manually curated resource that provides patterns—short textual prompts—that are paraphrases of one another, with 328 paraphrases describing 38 binary relations.

COUNTERFACT COUNTERFACT (Meng et al., 2022) is an evaluation dataset for evaluating counterfactual edits in language models which contains 21,919 records with a diverse set of subjects, relations, and linguistic variations. We use its target knowledge as counterfactual knowledge to query LLMs.

ALCUNA ALCUNA (Yin et al., 2023) is used for evaluating the ability of LLMs in face of new knowledge which consists of a total of 84351 questions about 3554 artificial entities. We only select the cloze-style portion of the questions to be used for the experiments.

A.2 Baselines

We slightly adapted the dataset to the characteristics of the generative model, and examples of the inputs are shown in Table 4.

B Hyperparameter setting for PGDC

Hyperparameter settings are shown in Table 5.

C Pseudocode for our algorithm

We provide pseudocode for ASRA is in Algorithm 1.

D Coverage Analysis

We also examined the knowledge identified by PGDC, as well as the knowledge identified by baseline methods based on KAssess. The result

KAssess	Zero-Shot	1. 10,000 metres record is held by [Kenenisa Bekele]
	Few-Shots	2. Pole vault record is held by Fabiana Murer.\t 800 metres record is held by David Rudisha\t 10,000 metres record is held by [Kenenisa Bekele]
	Discriminator	3. Check whether the following statement is correct: 10,000 metres record is held by Kenenisa Bekele. The statement is (True/False): [True]
PARAREL	Zero-Shot	1. The mother tongue of Go Hyeon-jeong is [Korean]
	Few-Shots	2. The mother tongue of Michel Denisot is French. \t The mother tongue of Thomas Joannes Stieltjes is Dutch. \t The mother tongue of Go Hyeon-jeong is [Korean]
	Discriminator	3. Check whether the following statement is correct: The mother tongue of Go Hyeon-jeong is Korean. The statement is (True/False): [True]
CFACT	Zero-Shot	1. IBM Connections is created by [Adobe]
	Few-Shots	2. Windows Embedded CE 6.0 is created by IBM. \t Sandy Bridge was a product of Apple. \t IBM Connections is created by [Adobe]
	Discriminator	3. Check whether the following statement is correct: IBM Connections is created by Adobe. The statement is (True/False): [True]
ALCUNA	Zero-Shot	1. What's the body length of Leuciaiaivea? [8.1 cm]
	Few-Shots	2. What's the body length of Octopus perralis? 100.0 cm. \t What's the body length of Sepia bidabilis? 17.0 cm. \t What's the body length of Leuciaiaivea? [8.1 cm]
	Discriminator	3. Check whether the following statement is correct: What's the body length of Leuciaiaivea? 8.1 cm. The statement is (True/False): [True]

Table 4: Demonstration of baselines. Answers are in '[]'. Each query has multiple textual expressions and each answer has multiple aliases. The number of examples for few-shots in our experiments is 4. Due to space constraints, we do not show these comprehensively.

Hyperparameter	
Learning rate	1e-2
Optimizer	Adam
Scheduler	ExponentialLR
Schedule Step	5
Iteration Rounds	25
λ_2	0.01

Table 5: Hyperparameter settings of PGDC.

Algorithm 1 PGDC Algorithm

Input: LLM θ, Embedding Table $E^{|\mathcal{V}|}$, Input Question $q = \{q_1, q_2, ..., q_n\}$, Answer a, Loss Function Φ, Optimization Step T, Learning Rate γ, Projection Ceil c

$p \leftarrow E^{|\mathcal{V}|}[q]$
for $i = 1, 2, ...T$ **do**
  Generate $z = \{z_1, z_2, ..., z_m\}$ with p as an input into θ
  $L \leftarrow \Phi(z)$
  $p \leftarrow p - \gamma \nabla_p L$
  for $p_j \in p$ **do**
    $t \leftarrow \arg\min_{k \leq |\mathcal{V}|} ||p_j - v_k||_2$
    if $||p_j - v_t||_2 \leq c$ **then**
      $p_j \leftarrow v_t$
    end if
  end for
end for
return p

Human Evaluation Guideline
Task Overview
Thank you for participating in this task! We are currently working on a project focused on benchmarking knowledge boundary for Large Language Model (LLM). You will be randomly presented with two texts. Your task is to determine whether the semantics of two texts are consistent, which means judging whether the two texts possess consistent semantics and can both be used to inquire about the same knowledge. Note that the texts do not need to be fluent or grammatically correct, and the presence of non-disruptive gibberish that does not affect the recognition of semantics is allowed. If the semantics of the two texts are consistent, label 1; if not, label 0. Please maintain high quality in your annotations.
Emphasis and Caution
Support and Reference: If you encounter any confusion regarding professional knowledge or context while performing this task, please feel free to reach out to us for clarification.You may also refer to Wikipedia or other reliable sources to gain further understanding. **Feedback Mechanism**: You can directly submit your queries, concerns, or suggestions to us.

Table 6: Human evaluation guideline.

of PGDC and the baseline methods P-few, few, P-zero, zero, P-dis and dis on the strong LLaMA2 model are shown in Figure 6. Our analysis reveals that the knowledge boundaries we derived can effectively encompass those of the baseline methods.

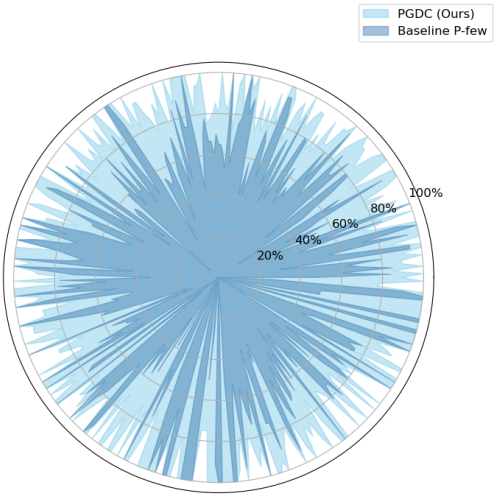
E Autoprompt for Model Evaluation

We implement Autoprompt by extending the question with five trigger tokens initialized with the last token in original prompt. The trigger tokens are updated for three rounds according to the algorithm described in Shin et al. (2020).

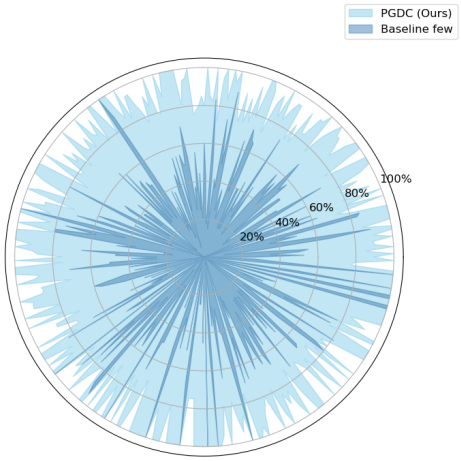
F Human Evaluation

We provide our human evaluation guideline furnished to participants for manually evaluating the semantic preservation task, as presented in Table 6.

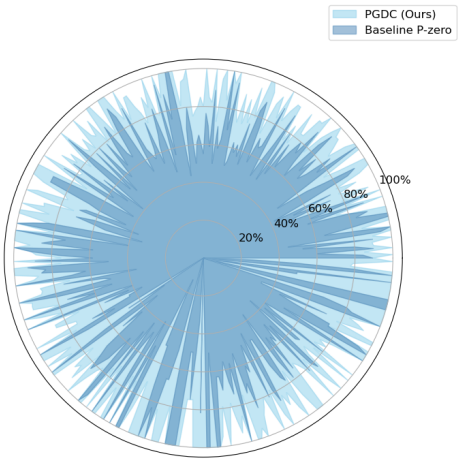
We recruited three college students, all possessing College English Test-6 certificates, demonstrating fluency in English. We first distribute the evaluation guidelines to the evaluators. Subsequently, we conduct training sessions for the evaluators, explaining the evaluation guidelines to ensure a better understanding of the task requirements and addressing any questions or concerns they may have. Before commencing formal annotation tasks, we administered a qualification test. Ten samples were randomly selected. These samples were evaluated by the participants, and subsequently, we assessed



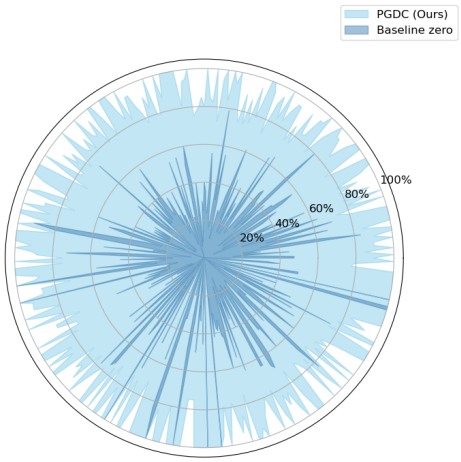
(a) Knowledge boundaries of PGDC and baseline P-few.



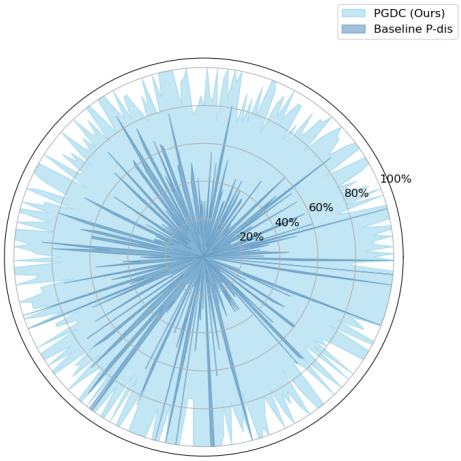
(b) Knowledge boundaries of PGDC and baseline few.



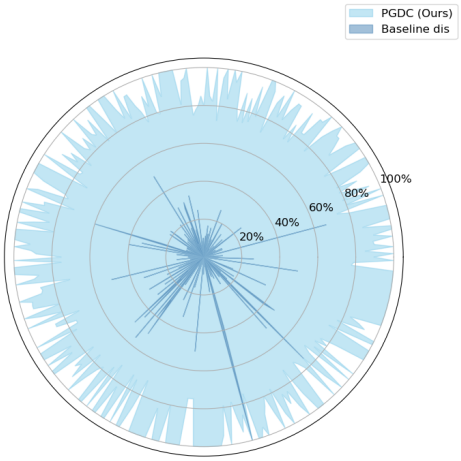
(c) Knowledge boundaries of PGDC and baseline P-zero.



(d) Knowledge boundaries of PGDC and baseline zero.



(e) Knowledge boundaries of PGDC and baseline P-dis.



(f) Knowledge boundaries of PGDC and baseline dis.

Figure 6: Knowledge boundaries of the proposed PGDC and baseline methods on KAssess using LLaMA2 model.

the accuracy of each annotator's evaluations. A higher accuracy score reflects a more consistent understanding of our guidelines. Evaluators who achieved at least 90 % accuracy were deemed qualified to proceed with the evaluation task. We employed Fleiss's Kappa statistic (Fleiss et al., 1981) to assess the agreement between the three annotators, yielding a score of 0.64.

G Model Evaluation on MMLU

G.1 Experimental Settings

Based on the conclusion of our earlier experiments that cloze-style questions are more reliable, we converted the choice-style questions in MMLU to a cloze format. We remove the other options and only keep the contents of the correct option as the answer to the cloze question.

Since the topic of our paper is about knowledge and some of the questions in MMLU are about computation and reasoning, we filter them out The remaining 30 subjects are grouped into six larger subjects, as shown in Table 7.

The PGDC method in this experiment uses the same hyperparameters as in Appendix B.

G.2 Detailed Results

In the main article we report the results in the broad categories, and the results in each subcategory are shown in Table 7.

Broader Subject	Subject in MMLU	GPT-2	GPT-J	LLaMA2	Vicuna	Mistral
Nature Science	astronomy_test	5.08	6.78	6.78	10.17	15.25
	college_biology_test	2.56	10.26	14.10	10.26	16.67
	college_chemistry_test	1.89	1.89	11.32	9.43	5.67
	conceptual_physics_test	14.47	19.30	23.25	23.25	23.25
	high_school_physics_test	0.00	0.00	1.72	1.72	1.72
	high_school_biology_test	1.41	8.45	14.79	11.97	15.49
	high_school_chemistry_test	3.06	4.08	7.14	6.12	8.16
Social Science	high_school_government_politics	10.34	12.64	18.39	16.09	14.94
	high_school_macroeconomics_test	0.93	1.87	3.74	4.67	5,61
	high_school_microeconomics_test	2.50	2.50	5.00	4.17	5.00
	management_test	1.08	0.00	5.38	5.38	2.15
	professional_accounting_test	1.67	0.00	1.67	1.67	0.00
	sociology_test	0.00	3.49	6.98	3.49	5.81
	us_foreign_policy_test	5.08	10.17	10.17	6.78	10.17
	world_religions_test	3.61	4.21	19.88	15.06	18.07
	high_school_psychology_test	10.08	14.47	19.90	17.05	22.22
Engineering	electrical_engineering_test	6.02	5.26	10.53	6.02	6.77
	college_computer_science_test	0.00	4.76	4.76	0.00	14.29
Medicine	clinical_knowledge_test	0.00	2.68	4.70	2.68	6.71
	college_medicine_test	2.86	2.86	10.00	2.86	8.57
	medical_genetics_test	0.00	4.11	9.59	13.70	9.59
	nutrition_test	2.53	3.16	10.13	7.59	8,87
	virology_test	0.79	2.38	3.97	3.97	6.35
	anatomy_test	1.23	9.88	20.99	19.75	25.93
Humanities	global_facts_test	1.14	3.41	7.95	9.09	12.50
	moral_disputes_test	3.52	2.01	7.04	4.52	6.03
	miscellaneous_test	10.30	11.61	23.37	22.79	28.45
Others	high_school_geography_test	4.85	8.48	16.36	12.73	13.94
	logical_fallacies_test	0.00	0.98	2.94	3.92	3.92
	human_aging_test	29.27	3.90	5.37	9.76	8.29

Table 7: Our categorization of subjects in MMLU and detailed scores.