S2LPP: Small-to-Large Prompt Prediction across LLMs

Anonymous ACL submission

Abstract

The performance of pre-trained Large Language Models (LLMs) is often sensitive to nuances in prompt templates, requiring careful prompt engineering, adding costs in terms of computing and human effort. In this study, we present experiments encompassing multiple LLMs variants of varying sizes aimed at probing their preference with different prompts. Through experiments on Question Answering, we show prompt preference consistency across LLMs of different sizes. We also show that this consistency extends to other tasks, such as Natural Language Inference. Utilizing this consistency, we propose a method to use a smaller model to select effective prompt templates for a larger model. We show that our method substantially reduces the cost of prompt engineering while consistently matching performance with optimal prompts among candidates. More importantly, our experiment shows the efficacy of our strategy across fourteen LLMs and its applicability to a broad range of NLP tasks, highlighting its robustness[1].

1 Introduction

Recent research (Wei et al., 2022; Reynolds and McDonell, 2021; Fernando et al., 2023; Nye et al., 2021; Wang et al., 2022; Zhou et al., 2022; Wang et al., 2023; Arora et al., 2022) has demonstrated that prompting is crucial to the downstream performance of foundation LLMs, requiring efficiently prompt engineering for practical applications. While manually crafted prompts (Reynolds and McDonell, 2021) have been widely used, Shin et al. (2020) introduced an automated method for creating prompts for various tasks using a gradient-guided search. However, the method requires iterative refinement for the prompts, which would be prohibitively expensive for current LLMs. Also, their assumption of access to LLM logit outputs is invalid for black-box LLMs. With the advancement of LLMs, Zhou et al. (2022), Kazemi et al. (2022), and White et al. (2023) have leveraged LLMs to generate instruction candidates and have selected prompts by optimizing a chosen score function. These methods require calculating the score across all candidate prompts using large-sized LLMs to reach optimal performance for each task, which is also computationally expensive. What is worse, the rapid evolution of LLMs also might appear to pose challenges in efficiently updating the prompt template selections for new emerging LLMs.

To ascertain whether LLMs of different sizes exhibit similar *preferences* for various prompts, we introduce a series of experiments by generating multiple natural language prompts for Question Answering (QA) and then extends to Natural Language Inference (NLI) tasks. We evaluate these prompts across a range of LLMs of varying sizes. Our studies prove that various LLMs consistently select identical optimal prompts from the pool of candidate prompts.

Based on our findings, we exploit the prompt preferences of smaller models as proxies to that of larger models. With smaller models, it is less computationally expensive to gain knowledge of their prompt preference. We propose a **Small-to-large Prompt Prediction (S2LPP)** approach, leveraging smaller models to identify optimal prompt templates from automatically generated prompt candidates for larger target models. This approach would help to reduce the deployment cost of LLMs, especially when faced with diverse and dynamic sets of open-domain knowledge. We show the effectiveness of the S2LPP approach on open-domain QA and NLI across fourteen LLMs of varying sizes, and further extend it to broader NLP tasks such as retrieval-augmented generation and arithmetic reasoning, showcasing its robustness and generalizability. The main contributions of this paper can be summarized as follows:

[1]Our code and data will be released upon publication.

(a) We provide evidence to present the consistency of prompt preference across LLMs of different sizes.

(b) Utilizing the observed consistency, we propose a lightweight, automatic strategy to leverage small LMs to select optimal prompt templates for larger LLMs.

(c) Through evaluation of QA and NLI tasks, we show that our approach outperforms the baselines and effectively reduces computational costs of prompt engineering while consistently maintaining high performance in larger target models.

2 Background

The performance of contemporary LLMs heavily depends on the forms and nuances present in the natural language prompts they are given (Jiang et al., 2022; Jin et al., 2021; Zhang et al., 2023; Shin et al., 2020; Arora et al., 2022). However, owing to the black-box nature of LLMs, their prompt preference is also underexplained and sometimes dependent on nuanced variations (Webson and Pavlick, 2021; Lin, 2024; Kassner and Schütze, 2020; Shin et al., 2020), requiring extensive prompt engineering to achieve optimal performance for each task.

Prompt Engineering: Research on *manually* designed prompts (Brown et al., 2020; Reynolds and McDonell, 2021; Ouyang et al., 2022) highlights the essential role of expert involvement in manual prompting processes, which is time-consuming and expensive. In addition to manually designed prompts, *automatically* generated prompts for LLMs have also been explored. Shin et al. (2020) introduced AutoPrompt, a method that employs gradient-guided search to automatically generate prompts. Kazemi et al. (2022) propose a backward selection method for optimizing prompts, while Yang et al. (2023) present a framework utilizing LLMs as optimizers for prompt tuning, demonstrating improvements over manually crafted prompts. However, training the optimal prompt using large-sized LLMs across diverse tasks involves extensive computation, making the approaches costly and unstable when generalizing to out-of-domain scenarios (Theophilou et al., 2023; Zhao et al., 2021) .

Prompt Consistency: Prompt consistency has long been an important topic in the NLP research. Si et al. (2022) find that certain prompts maintain consistent performance across different sizes of the GPT-3 model. Wang et al. (2024) discover that some prompts can yield similar performance across models in the biomedical domain. Additionally, Li et al. (2025) reported that different LLMs exhibit consistent preference of templates in code generation. On the other hand, Voronov et al. (2024) argue that rigid and structured prompt templates perform inconsistently across different models in in-context learning. However, their work focused on analyzing consistency among rigid and structured templates. In contrast, our work studies organic natural language prompt templates, addressing a broader and more common scenario in NLP research.

In this work, we set up a series of experiments to demonstrate the consistency of prompt preference across LLMs. We present the findings from our analyses in §3, and propose a lightweight approach to leverage these findings for various tasks in §4.

3 Consistency of Prompt Preferences across Different Model Sizes

In this section, we analyze consistency in prompt preference among LLMs of varying sizes. We set up a series of experiments on two tasks: open-domain QA (§3.1) and NLI (§3.2), respectively, which pose challenges to the current state-of-the-art LLMs. First, we collect multiple natural language prompt templates for QA and NLI. Then, we evaluate these prompts across LLMs of varying sizes, comparing their performance to determine whether models from the same family, despite differences in scale, exhibit similar preferences for the best-performing prompt.

Models: In our experiments, we evaluate multiple prompt templates on **DeepSeek-R1** (DeepSeek-AI et al., 2025), **LLaMA-2-chat** (Touvron et al., 2023), **LLaMA-3-instruct** (AI@Meta, 2024), and **Vicuna** (Zheng et al., 2023) model families, using models of varying sizes within each family.

3.1 Task 1: Open-domain QA

Datasets: For open-domain QA, we experiment with two open-domain QA datasets: **Google-RE** (Petroni et al., 2019) and **T-REX** (Elsahar et al., 2018). The Google-RE dataset is meticulously curated from the Wikipedia knowledge base[2] and comprises 5.5K meticulously extracted facts structured in the form of relation triples ([X], relation, [Y]). This corpus contains three

[2] https://dumps.wikimedia.org/enwiki

Datasets	Task	Samples	Prompt source	Num of relations	Num of prompts	prompt description
Google-RE	QA	5.5k	auto-generated	3	10 per relation	A natural question to describe a relation, like PlaceOfBirth. e.g. “What is the birthplace of [X]?”
T-REX	QA	31k	auto-generated	41	10 per relation	
Levy/Holt	NLI	1.8k	manual-generated	1	5	A binary question to judge if [premise] entails [hypothesis]. e.g. “If Google bought Youtube, then Google owns Youtube”

Table 1: Details of the test sets. For QA, Google-RE includes 3 relations, and T-REX encompasses 41 relations, each with 10 automatically generated prompt templates per relation. For NLI, the Levy/Holt dataset consists of 1 relation with 5 manually crafted prompts.

distinct relations: `PlaceOfBirth`, `PlaceOfDeath`, and `DateOfBirth`. In a similar data format to Google-RE, the T-REX dataset contains knowledge sourced from a subset of Wikidata (Vrandečić and Krötzsch, 2014) with 41 relations, and it subsamples at most 1000 triples per relation.

Prompt Candidates: We *automatically* generate prompt templates for QA. Here, we input each relation from the test set into ChatGPT (OpenAI, 2022) and generate 10 distinct natural question prompts per relation. For instance, the prompt *“What is the birthplace of [X]?”* is employed for the `PlaceOfBirth` relation. These prompts are then filled with the facts to generate relevant questions for analysis and evaluation.

3.2 Task 2: Natutral Language Inference

Dataset: In our NLI experiments, we select the **Levy/Holt** (Levy and Dagan, 2016; Holt, 2019) dataset as our test set. The Levy/Holt dataset comprises premise-hypothesis pairs structured in a specific task format: *<premise, hypothesis, label>*. Each premise and hypothesis is also structured as a relation triple, containing a single predicate with two entity arguments, wherein identical entities are present in both the premise and the hypothesis. A distinctive feature of the Levy/Holt dataset is the inclusion of inverse pairs for all premise-hypothesis-label entailments. Following prior work (Mckenna et al., 2023; Cheng et al., 2023; Chen et al., 2022), we study the challenging *directional* subset, where the entailments hold in one direction but *not* both.

Prompt Candidates: We employ the same prompts utilized in prior work (Mckenna et al., 2023) for evaluation, consisting of five natural question prompts crafted by human experts. We present the manually crafted prompts in Appendix B and the detailed experimental settings in Table 1.

3.3 Metrics

Accuracy: For open-domain QA tasks, we consider a response from an LLM to be correct if it contains the target entities. This approach allows us to calculate accuracy. For NLI tasks, we use the hypothesis-premise pairs from the Levy/Holt dataset as *binary questions* for the LLMs and subsequently calculate the accuracy.

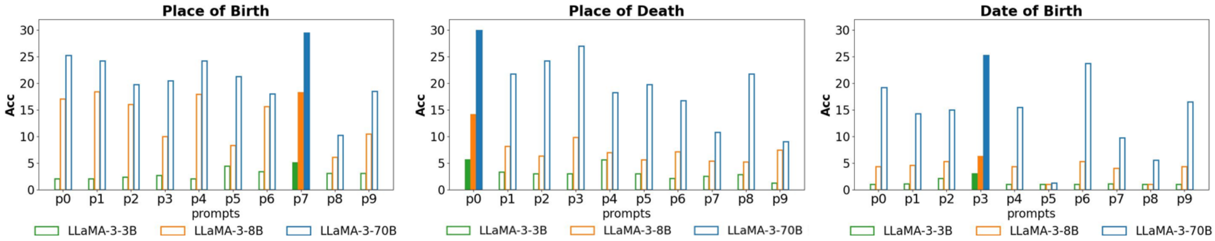
Proportion of Optimal-Prompt Matches: In QA and NLI, we take the prompt that achieves the highest accuracy as the *optimal-prompt*, and we introduce the Proportion of Optimal-Prompt Matches (POPM) as the metric to measure the ratio of optimal-prompt matches between pairs of LLMs `X` and `Y`. For each relation in each dataset, if model `X` and model `Y` share the same optimal prompt template, we count it as 1. The POPM metric is then calculated by dividing the number of matched relations by the total number of relations.

3.4 Findings

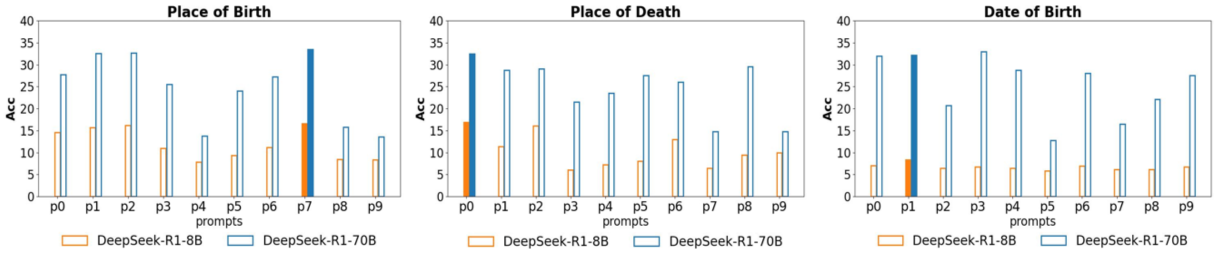
In open-domain QA task, Figure 1 compares the performance of LLMs of different sizes across a spectrum of generated prompts, spanning all the relations present within the Google-RE. The results indicate that, despite differences in model size, LLMs within the same family consistently achieve the highest accuracy with the same prompts (For LLaMA-3, P_7 yields the best performance for `PlaceOfBirth`, P_0 for `PlaceOfDeath` and P_3 for `DateOfBirth`)[3], as depicted by the solid bar in the image. Additionally, as shown in Appendix D, we observe the same consistency in LLaMA-2 and Vicuna-1.5 model families. These findings suggest that models of different sizes within the same LLM family exhibit consistent prompt preferences in QA tasks. Due to presentation constraints, we leave the optimal prompts and their performance for individual relations in T-REX to Appendix C and Appendix E, where results are consistent.

In NLI tasks, as demonstrated in Figure 2, our findings are also consistent in NLI tasks. Various sizes of LLaMA-3 models exhibit identical prompt preferences, achieving the highest accuracy with

[3]A different set of prompt templates is generated as natural questions for each relation, so prompt indices are not comparable across different relations.

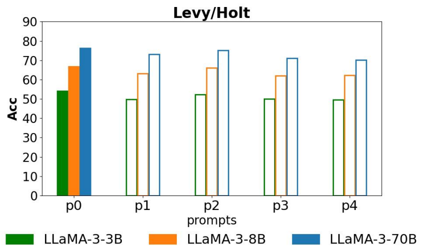


(a) Accuracy of various prompt templates across LLaMA-3 models of different sizes.

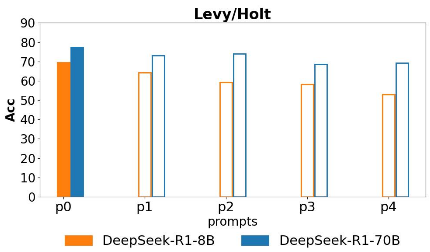


(b) Accuracy of various prompt templates across DeepSeek-R1 models of different sizes.

Figure 1: Accuracy of different prompts across LLaMA-3 and DeepSeek-R1 models on Google-RE. The x-axis represents the various prompts being evaluated. The solid bar indicate the optimal prompt for each respective LLMs.



(a) Accuary of prompts across LLaMA-3 of different sizes.



(b) Accuary of prompts across DeepSeek-R1 of different sizes.

Figure 2: The figure illustrates the accuracy of different prompts across LLaMA-3 and DeepSeek models of varying sizes on the directional Levy/Holt (NLI task). The x-axis represents the various candidate prompts, while the solid bar represents the optimal prompt for each LLM.

Models	Datasets		
	Google-RE	TREx	Levy/Holt
LLaMA-2-7B	100% (3/3)	70.7% (29/41)	100% (1/1)
LLaMA-2-13B	100% (3/3)	75.6% (31/41)	100% (1/1)
Vicuna-7B	100% (3/3)	78.0% (32/41)	0% (0/1)
Vicuna-13B	100% (3/3)	87.8% (36/41)	100% (1/1)
Vicuna-33B	34% (1/3)	68.3% (28/41)	100% (1/1)
LLaMA-3-8B	34% (1/3)	61.0% (25/41)	100% (1/1)
LLaMA-3-70B	34% (1/3)	68.3% (28/41)	100% (1/1)
DeepSeek-R1-8B	67% (2/3)	73.2% (30/41)	100% (1/1)
DeepSeek-R1-70B	67% (2/3)	78.0% (32/41)	100% (1/1)

Table 2: This table presents the POPM scores across various LLMs in comparison to GPT-3.5. The table also presents the number of optimal-prompt-matched relations relative to the total number of relations.

the same prompt, P_0. In the DeepSeek-R1 series models, the P_0 is still the optimal prompt.

Furthermore, we present our findings across different model families with the POPM scores in Table 2, where we observe a consistently high ratio of optimal prompt overlaps between different model families.

These findings demonstrate a consistent preference for prompt template selection across LLMs of varying sizes within the same model family. Notably, the prompts that perform optimally in smaller models demonstrate effectiveness even when applied to larger models. Furthermore, the observed high ratio of overlaps across different LLM families indicate that it is possible to utilize smaller models from different families to approximate the prompt preference of larger models, and prompt the larger models with approximated optimal prompts at inference time, to reach near-optimal performance on unseen tasks at minimal computational cost.

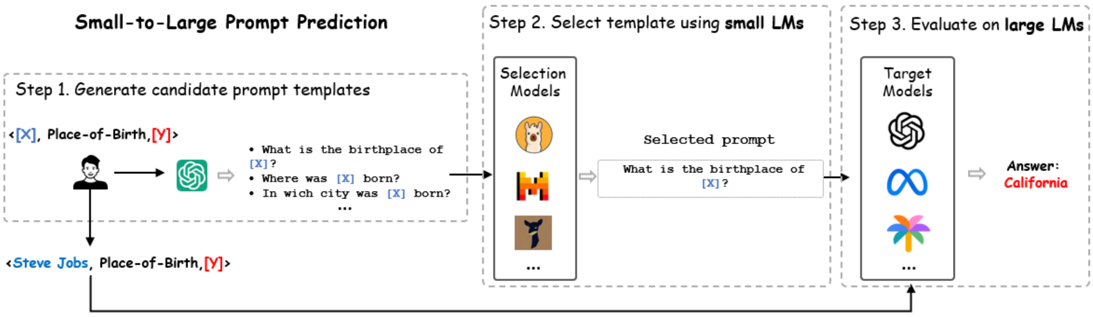


Figure 3: The workflow of S2LPP on open-domain QA: **Step 1**: For each relation, we utilize the prompt-generation model to produce top-k candidate prompts. **Step 2**: We employ the small Selection Model to discern the optimal prompt from candidates. **Step 3**: We use the selected prompt to ask questions. Subsequently, we employ the Target Model to provide responses to these questions.

4 Small-to-large Prompt Prediction

The previous experiments in §3 have shown the existence of consistency in prompt preference among various sizes of LLMs. In this section, we exploit this consistency to reduce the development cost of LLMs in NLP applications.

We propose the **S**mall-to-**L**arge **P**rompt **P**rediction (S2LPP) method, leveraging this consistency to automatically generate and select high-performing prompts for new, unseen open-domain knowledge in a computationally efficient manner. We evaluate S2LPP on open-domain QA and NLI tasks and extend the pipeline to a wider range of NLP applications, including using smaller LLMs for retrieved document selection in open-domain QA and for Chain-of-Thought (CoT) prompt selection in arithmetic reasoning tasks.

4.1 Method

The S2LPP framework primarily comprises three steps: prompt generation, prompt selection, and prediction with large target models. We illustrate an example workflow of S2LPP in Figure 3.

Prompt generation: A prompt-generation model is used to generate a set of candidate natural language prompt templates.

Prompt selection: Prompt selection is the crucial step in the S2LPP pipeline. By leveraging the consistency of prompt preference, we utilize *smaller* LMs as the prompt-selection models to assess each prompt by its performance on a few examples to efficiently select the prompts with the best performance.

Predict with Target Model: After we compute the performance of each prompt in the above mentioned step, we select the prompt with the highest score and use it in the following evaluation. To be more specific, we integrate test examples into the prompt template to form natural queries. Then, we input these queries into the target *larger* model and employ their responses as answers.

4.2 Experimental Setup

Aligned with the experiments in §3, we apply our method to both open-domain QA and NLI tasks. For open-domain QA, in the *prompt-generation step*, we utilize ChatGPT to generate 10 candidate prompts specific to the relations sourced from the Wikidata knowledge base, with temperature fixed at 0. These prompts are generated as a specific natural prompt template, such as *"What is the birthplace of [X]?"* for the Wikidata relation `PlaceOfBirth`. Subsequently, entities sourced from the knowledge base are filled into the prompts, transforming them into natural questions posed to prompt-selection models. In the *prompt-selection step*, we employ fourteen widely-used LLMs of varying sizes as the prompt selection models. In the *predict with target model step*, we use the GPT-3.5 model as the target model to assess whether the selected prompts enhance their performance.

For the NLI task, we similarly use ChatGPT to automatically generate 10 natural language questions as candidates, as presented in Appendix C and then populate these templates with the corresponding hypotheses and premises in the dataset. Note that we do not use the manual prompt templates from the analysis above (§3.2) to avoid human labor in our proposed approach.

4.2.1 Models

Besides LLaMA-3, DeepSeek-R1, LLaMA-2 and Vicuna series LLMs, we also include additional LLMs such as Mistral (Jiang et al., 2023), Stable-

Beluga (Mahan et al.) and falcon (Almazrouei et al., 2023) series models as prompt selection-models for a more in-depth analysis.

4.2.2 Datasets

For QA tasks, we curate a sample of 41 relations sourced from Wikidata, consistent with those in the Google-RE and T-REX datasets. For NLI tasks, we again utilize the directional Levy/Holt dataset, which consists of premise-hypothesis pairs.

Development Set: In our experiment, the first 100 samples of the QA task datasets (Google-RE and T-REX) are designated as the development set, where the prompt-selection models are utilized to identify the optimal prompt. For NLI tasks, we directly select 100 samples from the Levy/Holt development set.

Test Set: With the exception of the selected 100 samples from the Google-RE and T-REX datasets used as development sets, we utilized the remaining subset as the test set.

4.2.3 Baselines

First-generated Prompts: This baseline uses the first generated prompt from the set of 10 generated candidates since the first prompt also tends to be the most favored prompt.

Average scores among prompts: We compute the mean accuracy across the candidates to measure the overall performance of all generated prompts. This methodology allows us to compare the quality of our selected prompts against the average performance level among all prompts.

Manual Prompts: For each relation in each task, we take the manually-crafted prompt templates from prior work (Cheng et al., 2023; Mckenna et al., 2023; Schmitt and Schütze, 2021).

Oracle Prompts: We conduct prompt selection with the target model itself (GPT-3.5) and identified the optimal prompt from the development set as the oracle prompt, which is also the *upper bound* among all generated candidate prompts. This upper bound serves as a reference point against which to assess the performance gaps between our approaches and the pinnacle of performance.

4.3 Evaluation Metrics

Utilizing the target models to identify the oracle prompt can achieve the upper bound of performance among all candidates, but this process is expensive to train. Our prompt selection strategy aims to match this upper-bound performance while incurring lower costs.

Models	Datatsets		
	Google-RE	T-REX	Levy/Holt
Prompt$_{first-generated}$	19.26	64.61	54.95
Prompt$_{average}$	17.11	61.94	56.98
Prompt$_{manual}$	23.0	61.10	56.76
Prompt-selection Model (ours)	**26.06**	**67.63**	**58.74**
Prompt$_{oracle}$ (upper bound)	*27.81*	71.30	64.0

Table 3: Accuracy scores achieved using LLaMA-2-7B as the prompt-selection model on QA and NLI tasks. We compare with the first-generated prompt (Prompt$_{first-generated}$), average scores among all prompts (Prompt$_{average}$) and the manual prompts (Prompt$_{manual}$). Oracle prompt denotes the best-performing prompt on the target model.

In addition to accuracy, we introduce a metric to measure the efficacy of the selected prompts against the upper bound: ***Recovery Rate of Performance (RRoP)***. This metric demonstrates the proportion that we can recover from the performance of oracle prompts using our selected prompts. The *RRoP* is defined as follows:

$$RRoP(pt_S) = \frac{Acc(pt_S)}{Acc(pt_O)}$$

where pt_S and pt_O denote the selected and oracle prompts, respectively, and $Acc(\cdot)$ represents the accuracy of a prompt.

4.4 Results

Performance of Selection Model: Table 3 compares our small-sized LLM-selected prompts against various baselines. Here, we use the LLaMA-2-7B as the smaller model. Our approach outperforms baselines, demonstrating superior performance even when compared to manually crafted prompts. Furthermore, our methods exhibit minimal deviation from the upper bound, providing evidence that the prompts selected using small-size LMs are also performant with target models. The results highlight the efficacy of employing small-size LMs in open-domain QA and NLI tasks to optimize computational costs. We also observed that the accuracy of open-domain QA is limited across all prompts, which is attributed to the sparsity of exact matches. We conjecture that performance improvements can be achieved by using entailments for this task (Cheng et al., 2023).

Performance across Various Selection-Models: We conducted additional experiments to analyze

Selection Models \ Target Models	StableBeluga-7B	LLama2-7B	Vicuna-7B	LLama3-8B	Deepseek-8B	Falcon3-10B	StableBeluga-13B	LLama2-13B	Vicuna-13B	Vicuna-33B	LLama2-70B	Deepseek-70B	LLama3-70B
StableBeluga-7B	100%	47.97%	81.11%	66.66%	91.26%	81.60%	92.19%	40.28%	85.27%	37.27%	61.33%	89.69%	72.90%
LLama2-7B	94.48%	100%	97.82%	88.16%	82.97%	83.29%	94.42%	84.48%	98.48%	70.55%	85.47%	100%	80.34%
Vicuna-7B	97.05%	88.46%	100%	87.31%	77.41%	74.96%	92.91%	81.06%	96.59%	72.05%	71.05%	94.84%	88.59%
LLama3-8B	92.16%	58.45%	84.95%	100%	72.62%	80.98%	100%	84.61%	64.42%	85.53%	74.43%	85.15%	79.77%
Deepseek-llama-8B	96.15%	67.20%	82.56%	68.79%	100%	77.43%	93.70%	51.39%	93.61%	57.27%	55.92%	90.60%	69.80%
Falcon3-10B	89.21%	65.86%	86.26%	90.25%	72.36%	100%	89.41%	70.58%	87.59%	70.41%	63.04%	88.18%	83.00%
StableBeluga-13B	92.16%	58.45%	84.95%	100%	72.62%	80.98%	100%	84.61%	64.42%	85.53%	74.43%	85.15%	79.77%
LLama2-13B	93.44%	90.50%	90.03%	92.73%	81.35%	95.56%	95.45%	100%	74.27%	87.20%	97.86%	99.39%	85.61%
Vicuna-13B	94.48%	92.30%	97.10%	90.30%	86.14%	81.21%	94.42%	70.80%	100%	65.39%	59.34%	94.84%	73.47%
Vicuna-33B	77.78%	72.97%	91.64%	91.29%	60.64%	78.64%	89.20%	83.04%	87.39%	100%	85.58%	91.21%	100%
LLama2-70B	75.21%	84.51%	89.47%	92.14%	66.20%	86.97%	90.72%	86.46%	89.29%	97.5%	100%	96.36%	91.75%
Deepseek-llama-70B	94.48%	100%	97.82%	88.16%	82.97%	83.29%	94.42%	84.48%	98.48%	70.55%	85.47%	100%	80.34%
LLama3-70B	77.78%	72.97%	91.64%	91.29%	60.64%	78.64%	89.20%	83.04%	87.39%	100%	85.58%	91.21%	100%

Figure 4: The Recovery Rate of Performance (RRoP) across various LLMs on QA tasks. RRoP scores exceeding 70% are highlighted in red.

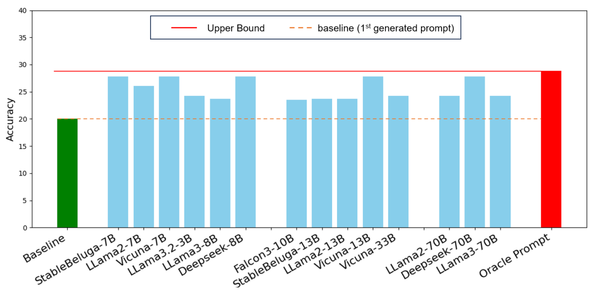


Figure 5: Accuracy of different models in the prompt selection step for QA. The green column represents the *baseline* using the first-generated prompt, while the red column illustrates the accuracy with the oracle prompt, which is the *upper bound* of the target model (GPT-3.5).

the effect of various sizes and families of smaller models in the prompt-selection process, shown in Figure 5. As depicted all LLMs utilized in the prompt-selection stage outperform the baselines. Interestingly, some smaller models outperform their medium and larger versions in the selection process, possibly because larger LLMs from different families are trained on more additional diverse corpora, leading to discrepancies with the target large model.

Recovery Rate of Performance across Various LLMs: Figure 4 demonstrates the RRoP scores. The results show that most selection models can recover a high proportion of the performance achieved by using oracle prompts, approaching the upper bound with lower computing costs. This suggests that, in addition to GPT models, other language models can also be effectively utilized as target models. It highlights the RRoP scores achieved when using different selection and target models separately, demonstrating the efficacy of applying these approaches to new LLM families.

	Google-RE
$\text{Context}_{first-paragraph}$	45.21
$\text{Context}_{DeepSeek-8B}$ $(ours)$	61.90
$\text{Context}_{whole-documents}$	66.82

Table 4: Accuracy across different context settings on the Google-RE dataset. We use DeepSeek-R1-8B to select the most relevant paragraph as context and compare its performance against using the first paragraph of the retrieved documents ($_{first-paragraph}$) and using the whole document ($_{whole-document}$) as context.

4.5 Extend to Broader NLP Applications

The core of the S2LPP approach is leveraging the consistency of prompt preference to enable efficient prompt selection using smaller LLMs, opening up the possibility to extend the pipeline to a broader range of NLP tasks. We further utilize this consistency in more applications, including using smaller LLMs to select relevant contexts for Retrieval-Augmented Generation (RAG) and to select Chain-of-Thought (CoT) prompts for arithmetic reasoning tasks.

Context Selection with Small LLMs for RAG: We evaluate the efficiency of using small-sized LLMs to select relevant contexts from retrieved documents for RAG. For each question in the Google-RE dataset, we retrieve 10 candidate documents using the Google Search API and then employ small-sized LLMs, DeepSeek-R1-8B to select the most relevant paragraphs as context from these candidates. The selected paragraph is then concatenated with the question and passed to GPT-3.5 to generate the final answer.

As shown in Table 4, using DeepSeek-R1-8B to select context from retrieved documents yields accuracy that is slightly lower than using the whole retrieved documents (long context) when evalu-

		GSM8K
	AutomateCoT$_{GPT}$	79.81
	AutomateCoT$_{mistral-7B}$	77.61
ours	AutomateCoT$_{deepseek-8B}$	79.37
	AutomateCoT$_{llama3-8B}$	78.75

Table 5: Accuracy scores of AutomateCoT using different generation and selection models. AutomateCoT$_{GPT}$ refers to the CoT prompts from Shum et al. (2023), where GPT-2 is used for both prompt generation and selection. Our approach uses DeepSeek-8B for prompt generation and small-sized LLMs (Mistral-7B, DeepSeek-8B, LLaMA3-8B) for prompt selection.

ated with GPT-3.5, while saving computing costs[4]. This demonstrates that different LLMs exhibit consistency in their preference for retrieved contexts, aligning with our findings on prompt preference consistency, and further supports the effectiveness of applying this approach to RAG.

CoT Prompts Selection with Small LLMs for Arithmetic Reasoning: Shum et al. (2023) propose a two-step pipeline, *AutomateCoT*, for generating CoT prompts: (1) using the GPT-2 (*davinci-002*) model to automatically generate a pool of CoT examples, and (2) selecting the optimal combination from this pool using a selection model trained on devlopment set via reinforcement learning, guided by performance from GPT-2. The selected CoT examples combination are then used as few-shot examples during evaluation.

In our experiments, we follow the same experimental setup but substitute the GPT-2 model with smaller LLMs. For the CoT examples generation step, we use DeepSeek-R1-8B to automatically create a pool of candidate examples. In the selection step, we randomly generate 100 candidate combinations and employ small LLMs, including *DeepSeek-R1-8B*, *LLaMA-3-8B-Instruct*, and *Mistral-7B*, to select the optimal combination by their performance. Evaluation is performed on GPT-3.5 using the **GSM8K** (Cobbe et al., 2021) arithmetic reasoning dataset, following the same test set as used in Shum et al. (2023).

As shown in Table 5, small-sized LLMs used for CoT prompt generation and selection achieve accuracy comparable to GPT-2, while our method reduces the computational cost of prompt selection by 60% compared to the baseline. The comparable performance further suggests that prompt preference consistency can be effectively leveraged not only for prompt selection but also for generation.

5 General Discussion

The common factor across the set of models is the similarity in the distributions of their pre-training corpora, so we conjecture that this prompt-preference consistency originates from the pre-training and that the prompt templates best aligned with the pre-training distribution would prevail. This also explains the differences between the finding in Voronov et al. (2024) and us, where they used rigid templates, and we used organic, natural language prompts, which more closely resemble the pre-training conditions of various LLMs.

The S2LPP approach demonstrates the efficacy of exploiting the consistency of prompt preference and offers an efficient method for prompt selection using small-sized models, which can complement SOTA prompt generation methods. Additionally, the prompt-selection models can be seamlessly updated with newly released LLMs. With the assumption that this prompt preference consistency originates from pre-training, the prompts selected by previous prompt-selection models could be performant with newly released target LLMs as well.

6 Conclusion

Across several major LLM families and experimental settings, we have demonstrated the consistency of prompt preference across LLMs on the QA and NLI tasks, providing significant potential for applications. Our work represent a finding that LLMs from the same model family, regardless of size, exhibit similar preferences across different prompts.

Based on this finding, we further propose a lightweight approach to utilize the consistency of prompt preference for open-domain questions involving new, unseen knowledge, by exploiting smaller models to select highly performant prompts at minimal cost in computation. We validate the efficacy of the approach in QA and NLI. Experiments demonstrate that the prompt templates selected with our strategy outperform baselines. Our methods also possess a strong capability to recover the performance of oracle prompts with significantly lower costs in the prompt selection steps. We further present the generalizability of our method to a broader range of NLP tasks. Deeper investigations into the source of this consistency will be important directions for our future work.

[4]In our experiments, the average length of the selected context is 82 tokens, compared to 1000 tokens for the full documents.

Limitations

In this work, we demonstrate the consistency of prompt preferences across LLMs and their exploitation in natural language tasks. However, our approach still has some limitations. In S2LPP, although we leverage this consistency by using small models in the prompt selection step, we still rely on powerful LLMs to generate candidates. Further research is required in order to explore the potential of using smaller models to generate these prompts for QA. Additionally, due to the limited computational resources and the high cost for evaluation on a wide range of models, we only utilize GPT-3.5 as the target model in the QA, NLI, RAG and arithmetic reasoning tasks. We plan to experiment with more open-sourced large target LLMs.

References

AI@Meta. 2024. Llama 3 model card.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. 2022. Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

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

Zhibin Chen, Yansong Feng, and Dongyan Zhao. 2022. Entailment graph learning with textual entailment and soft transitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5899–5910, Dublin, Ireland. Association for Computational Linguistics.

Liang Cheng, Mohammad Javad Hosseini, and Mark Steedman. 2023. Complementary Roles of Inference and Language Models in QA. In *Proceedings of the 2nd Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 75–91.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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

Xavier Holt. 2019. Probabilistic Models of Relational Implication. *arXiv:1907.12048 [cs, stat]*. ArXiv: 1907.12048.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Ellen Jiang, Kristen Olson, Edwin Toh, Alejandra Molina, Aaron Donsbach, Michael Terry, and Carrie J Cai. 2022. Promptmaker: Prompt-based prototyping with large language models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–8.

Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2021. A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models. *arXiv preprint arXiv:2110.08484*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818.

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2022. Lambada: Backward chaining for automated reasoning in natural language. *arXiv preprint arXiv:2212.13894*.

Omer Levy and Ido Dagan. 2016. Annotating Relation Inference in Context via Question Answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Yixuan Li, Lewis Frampton, Federico Mora, and Elizabeth Polgreen. 2025. Online prompt selection for program synthesis.

Zhicheng Lin. 2024. How to write effective prompts for large language models. *Nature Human Behaviour*, pages 1–5.

Dakota Mahan, Ryan Carlow, Louis Castricato, Nathan Cooper, and Christian Laforte. Stable beluga models.

Nick Mckenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of Hallucination by Large Language Models on Inference Tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Martin Schmitt and Hinrich Schütze. 2021. Language Models for Lexical Inference in Context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1267–1280, Online. Association for Computational Linguistics.

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.

Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, Singapore. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

Emily Theophilou, Cansu Koyutürk, Mona Yavari, Sathya Bursic, Gregor Donabauer, Alessia Telari, Alessia Testa, Raffaele Boiano, Davinia Hernandez-Leo, Martin Ruskov, et al. 2023. Learning to prompt in the classroom to understand ai limits: A pilot study. In *International Conference of the Italian Association for Artificial Intelligence*, pages 481–496. Springer.

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

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. *arXiv preprint arXiv:2401.06766*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Li Wang, Xi Chen, XiangWen Deng, Hao Wen, MingKe You, WeiZhi Liu, Qi Li, and Jian Li. 2024. Prompt engineering in consistency and reliability with the evidence-based guideline for llms. *npj Digital Medicine*, 7(1):41.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

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

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

A Computational Cost

In our experiments, we allocate resources equivalent to 4 GPUs (NVIDIA V100) for prompt-selection steps. For each relation sourced from Wikipedia, the process of selecting the optimal prompt among 10 candidates using small-size LLMs (LLaMA-2-7B, Vicuna-7B, StableBeluga-7B, Mistral-7B, and Falcon-7B) requires approximately 10 minutes, and it will cost about 30 minutes with medium-size LLMs (LLaMA-2-13B, StableBeluga-13B, Vicuna-13B). In contrast to utilizing large-size LLMs to achieve the upper bound prompt, our approaches facilitate significant savings in computational resources while maintaining performance levels with minimal gaps.

B Manually Crafted Prompt in NLI

As discussed in §3, to determine the consistency of prompt preferences in NLI, we utilize five manually crafted prompt templates used in prior works (Mckenna et al., 2023). These prompts are meticulously chosen for their clarity and conciseness, which also consider the prompt templates used in bias-related research on LMs (Schmitt and Schütze, 2021) for textual entailments. We present the manually crafted prompt templates below and highlight the best-performed prompt template on the target model, GPT-3.5, in bold.

1. **prompt_0: "If [premise], then [hypothesis]."**
2. prompt_1: "[P], so [H]."
3. prompt_2: "[P] entails [H]"
4. prompt_3: "[P], which means that [H]."
5. prompt_3: "[H], because [P]."

The prompt_0 outperforms another prompt template in GPT-3.5 and LLaMA-7B, LLaMA-13B, and Vicuna-13B models. The prompt_0 achieves the second highest accuracy among other templates on Vicuna-7B, where the optimal prompt is prompt_3.

C Automatically Generated Prompt Templates from ChatGPT

As discussed in §4, we introduce the S2LPP approach, which selects the automatically generated prompt templates using small LMs. Our method uses ChatGPT to generate 10 candidates for open-domain QA and NLI separately. The ten generated prompt templates used in our experiments for NLI tasks are presented below:

1. prompt_0: "Can [H] be inferred from [P]?"
2. **prompt_1: "Does [P] entail [H]?"**
3. prompt_2: "Is it true that [P] leads to [H]?"
4. prompt_3: "Is [H] a necessary consequence of [P]?"
5. prompt_4: "Do we conclude [H] from [P]?"
6. prompt_5: "If [P] is true, must [H] also be true?"
7. prompt_6: "Does the truth of [P] guarantee the truth of [H]?"
8. prompt_7: "Is [H] a logical consequence of [P]?"
9. prompt_8: "Can we derive [H] from [P]?"
10. prompt_9: "Is [H] implied by [P]?"

We also present the generated prompt templates for open-domain QA in Table 6. In this table, the optimal prompt templates for the target model, GPT-3.5, are highlighted in bold.

D Consistency across Different Models

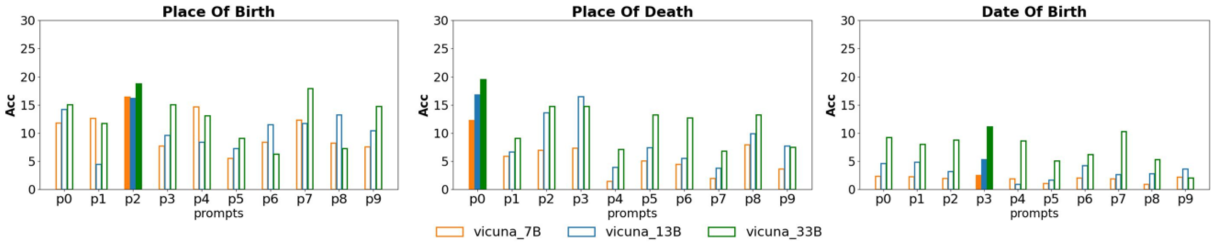
Besides the LLaMA-3 and DeepSeek-R1 models, we compare the performance of more LLMs across a spectrum of generated prompts in Figure 6, spanning all the relations present within the Google-RE. The results indicate that, with the exception of LLaMA-2 70B on `PlaceOfBirth`, LLMs within the same family consistently achieve the highest accuracy with the same prompts, regardless of differences in model size. As demonstrated in Figure 7, our findings are also consistent in NLI tasks. Various sizes of LLaMA-2 models exhibit identical prompt preferences to GPT-3.5, achieving the highest accuracy with the same prompt, P_0. In the Vicuna series models, the P_0 is still the optimal prompt evaluated in Vicuna-13B. Although P_0 achieves the second-highest score in Vicuna-7B, it closely approaches the performance of the prompt with the highest accuracy in Vicuna-7B model.

E Consistency on T-REX

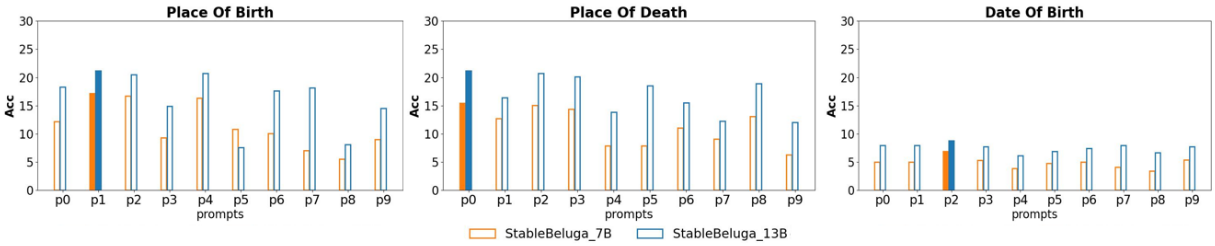
We present our consistency analysis experiments on the T-REX dataset, discussed in §3, in table 7. In this experiment, we use the best-performing prompt on GPT-3.5 as the reference label to determine if other models share the same optimal prompt. In the

Relations	prompt id	Prompt Templates
Pace-of-Birth	p_0	"What is the birthplace of [X]?",
	p_1	"Where was [X] born?",
	$\mathbf{p}_2$	**"In which city or town was [X] born?"**,
	p_3	"What is the native place of [X]?",
	p_4	"Could you provide the birth location of [X]?",
	p_5	"From where does [X] originate?",
	p_6	"What is the hometown of [X]?",
	p_7	"Where did [X] come into the world?",
	p_8	"What is the birth country of [X]?",
	p_9	"Can you tell me the exact location where [X] was born?"
Pace-of-Death	p_0	"Where did [X] pass away?",
	p_1	"What was the location of [X]'s death?",
	$\mathbf{p}_2$	**"In which city or town did [X] breathe their last?"**,
	p_3	"Can you provide the place where [X] died?",
	p_4	"What is the final resting place of [X]?",
	p_5	"Where was [X] when they passed away?",
	p_6	"What was the location of [X]'s demise?",
	p_7	"Could you tell me where [X] met their end?",
	p_8	"Where did [X] take their last breath?",
	p_9	"What was the place of departure for [X]?"
DateOfBirth	p_0	"When was [X] born?",
	p_1	"What is the birth date of [X]?",
	p_2	"Can you provide the date of birth for [X]?",
	p_3	"When did [X] come into the world?",
	p_4	"What day and month was [X] born?",
	p_5	"When did [X] celebrate their birthday?",
	p_6	"What is [X]'s birth year?",
	p_7	"Can you tell me the exact date when [X] was born?",
	$\mathbf{p}_8$	**"When did [X] first open their eyes to the world?"**,
	p_9	"What is [X]'s date of birth according to records?"

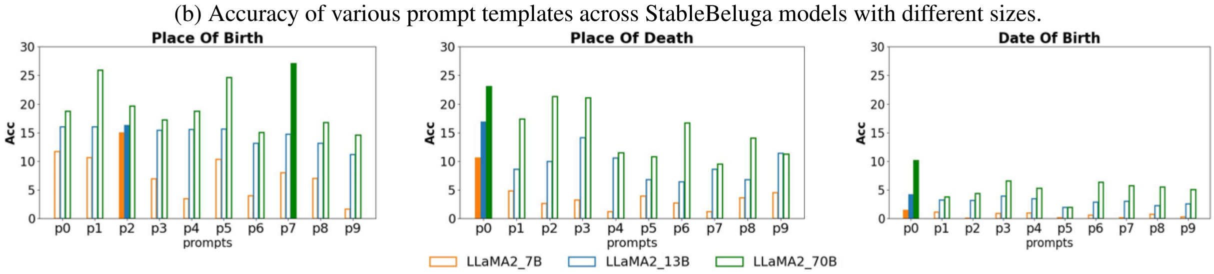
Table 6: The table presents the generated prompts for various relations in the Google-RE dataset. The optimal prompt templates for the target model, GPT-3.5, are highlighted in bold.



(a) Accuracy of various prompt templates across Vicuna models with different sizes.



(b) Accuracy of various prompt templates across StableBeluga models with different sizes.



(c) Accuracy of various prompt templates across LLaMA-2 models with different sizes.

Figure 6: The figure illustrates the accuracy of different prompts across Vicuna, StableBeluga and LLaMA-2-chat on Google-RE. The x-axis represents the various prompts being evaluated. The solid bar indicate the optimal prompt for each respective LLMs.

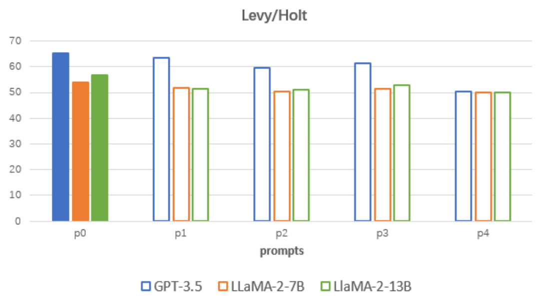
table 7, we highlight the `matches` and `mismatches` in blue and red color, respectively.

F Metrics on open-domain QA

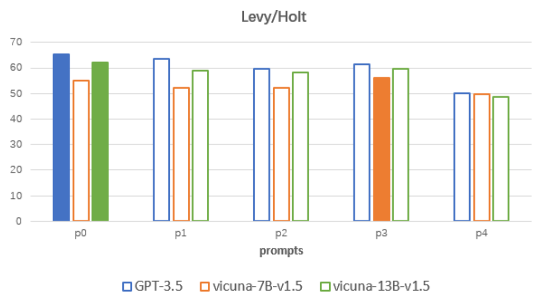
In our experiment settings, discussed in §3.3, we utilize the accuracy in our experimental metrics. Note that previous works (Petroni et al., 2019) on **Google-RE** and **T-REX** use Precision@1 as the metric, which is equivalent to the accuracy used in our work. In this task, the LLMs provide a single response as the answer for each question. Consequently, the score is the same, which is determined by the ratio of correct answers to the total number of questions.

Relations		The optimal prompts across models				
Relation Name	Relation ID	LLaMA-2-7B	LLaMA-2-13B	Vicuna-7B	Vicuna-13B	GPT-3.5
place of birth	P19	p2	p2	p2	p2	p2
place of death	P20	p2	p2	p2	p2	p2
subclass of	P279	p3	p3	p8	p8	p8
official language	P37	p1	p1	p1	p1	p1
position played on team	P413	p0	p0	p0	p0	p0
original network	P449	p0	p0	p0	p0	p0
shares border with	P47	p8	p8	p8	p8	p3
named after	P138	p0	p6	p6	p6	p6
original language of film or TV show	P364	p1	p1	p1	p1	p1
member of sports team	P54	p0	p0	p0	p0	p0
member of	P463	p1	p1	p1	p1	p1
field of work	P101	p6	p2	p2	p2	p0
occupation	P106	p3	p4	p2	p2	p2
has part	P527	p1	p0	p3	p0	p0
diplomatic relation	P530	p0	p0	p0	p0	p0
manufacturer	P176	p3	p3	p1	p1	p0
country of citizenship	P27	p3	p3	p3	p3	p3
language of work or name	P407	p0	p0	p0	p0	p0
is located in continent	P30	p0	p0	p0	p0	p0
developed by	P178	p0	p0	p1	p1	p1
capital of	P1376	p1	p0	p0	p0	p2
located in	P131	p6	p6	p6	p6	p6
used to communicate in	P1412	p0	p0	p0	p0	p0
work for	P108	p1	p1	p1	p1	p1
play	P136	p6	p5	p1	p3	p3
position held	P39	p2	p2	p2	p2	p2
record label	P264	p2	p2	p2	p2	p2
location	P276	p0	p2	p0	p0	p0
work location	P937	p3	p3	p3	p3	p3
religion	P140	p0	p0	p0	p0	p0
play music type	P1303	p1	p1	p1	p1	p1
owned by	P127	p0	p0	p0	p0	p0
native language	P103	p2	p2	p2	p2	p2
twinned administrative body	P190	p2	p2	p2	p2	p2
legal term in	P1001	p2	p2	p0	p0	p4
instance of	P31	p0	p0	p0	p0	p0
country of origin	P495	p5	p5	p5	p5	p5
headquarters location	P159	p0	p2	p0	p2	p2
capital	P36	p0	p0	p2	p0	p0
location of formation	P740	p2	p2	p2	p2	p2
part of	P361	p0	p0	p0	p0	p0
Counts of Matches		29	31	32	36	-

Table 7: This table presents the optimal prompt template matches in the T-REX dataset. We use the best-performing prompt on GPT-3.5 as the reference label. If other models select the same prompt as their optimal prompt, it is counted as a `match`, indicated in blue. Conversely, `mismatches` are indicated in red.



(a) Accuarcy of prompts across LLaMA-2-chat and GPT-3.5.



(b) Accuracy of prompts across Vicuna and GPT-3.5.

Figure 7: The figure illustrates the accuracy of different prompts across LLaMA-2, Vicuna, and GPT-3.5 models on the directional Levy/Holt (NLI task). The x-axis represents the various evaluated prompts, while the solid bar represents the optimal prompt for each LLM.