Self-Prompt Tuning: Enable Autonomous Role-Playing in LLMs

Anonymous ACL submission

Abstract

Recent advancements in LLMs have showcased their remarkable role-playing capabilities, able to accurately simulate the dialogue styles and cognitive processes of various roles based on different instructions and contexts. Studies indicate that assigning LLMs the roles of experts, a strategy known as role-play prompting, can enhance their performance in the corresponding domains. However, the prompt needs to be manually designed for the given problem, requiring certain expertise and iterative modifications. To this end, we propose self-prompt tuning, making LLMs themselves generate role-play prompts through fine-tuning. Leveraging the LIMA dataset as our foundational corpus, we employ GPT-4 to annotate role-play prompts for each data points, resulting in the creation of the LIMA-Role dataset. We then fine-tune LLMs like Llama-2-7B and Mistral-7B on LIMA-Role. Consequently, the self-prompt tuned LLMs can automatically generate expert role prompts for any given question. We extensively evaluate self-prompt tuned LLMs on widely used NLP benchmarks and open-ended question test. Our empirical results illustrate that self-prompt tuned LLMs outperform standard instruction tuned baselines across most datasets. This highlights the great potential of utilizing fine-tuning to enable LLMs to self-prompt, thereby automating complex prompting strategies. We release the dataset, models, and code at this url.

1 Introduction

Recent advances in large language models (LLMs) such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), Llama (Touvron et al., 2023), and Mistral (Jiang et al., 2023) have dramatically reshaped the field of natural language processing (NLP). These models exhibit exceptional text understanding and generation capabilities, with performance that critically depends on the quality of the prompts used. To sufficiently unleash the potential of LLMs, a range of innovative prompting strategies have emerged. These include, but are not limited to, chain-of-thought prompting (Wei et al., 2022b), tree-of-thought prompting (Yao et al., 2023), step-back prompting (Zheng et al., 2024), and the increasingly popular role-play prompting (Wu et al., 2023; Salewski et al., 2023; Kong et al., 2023). This paper concentrates on the development of self-prompt tuning to facilitate autonomous role-play prompting, a flexible method that may also be adapted for other prompting strategies.

Modern LLMs can seamlessly embody human characters[1] and non-human entities[2], exhibiting incredible role-playing capabilities. While role-playing brings novel modes of interaction, it can also serve as a prompting strategy, termed role-play prompting, to enhance the performance of LLMs in various downstream NLP tasks. For instance, Wu et al. (2023) have LLMs impersonate judges with distinct personas and backgrounds to improve their summary assessment quality. In multi-domain QA tasks, Salewski et al. (2023) instruct LLMs to act as domain experts, leading to improved performance. Furthermore, Kong et al. (2023) assign diverse expert roles to LLMs more immersively through multi-turn dialogue, boosting their reasoning abilities. Despite its efficacy, role-play prompting faces two significant limitations common to many popular prompting strategies:

(i) It is task-specific. The role selection and prompt design must be tailored to individual tasks, and prompts are often not transferable to different tasks.

(ii) The prompt design is labor-intensive, requiring significant domain expertise and iterative refinement, which can be time-consuming.

[1] Character.AI offers LLMs impersonating celebrities, such as Albert Einstein.

[2] DeepMind researcher requires ChatGPT to act as a Linux terminal in the blog.

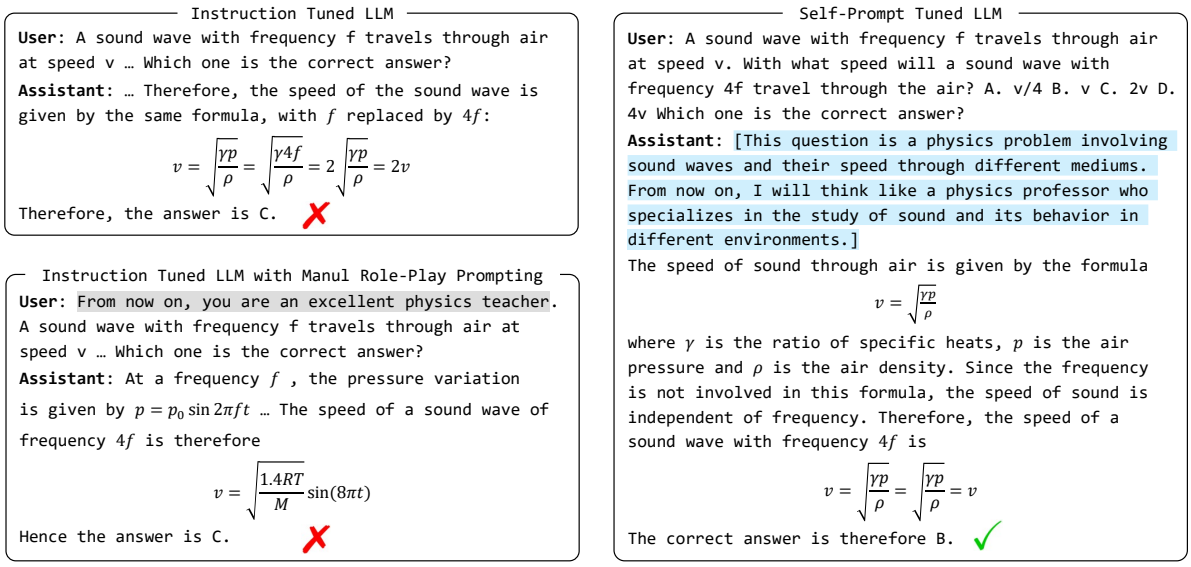

Figure 1: Examples of standard instruction tuned LLM, instruction tuned LLM with manual role-play prompting, and self-prompt tuned LLM on the same physics question. Manual and automatic role-play prompts are highlighted in gray and blue respectively. LLM used here is Mistral-7B.

To address these limitations, could we leverage LLMs themselves to generate prompts, thereby reducing the reliance on human intervention? A natural idea is to utilize prompts to instruct models to generate prompts themselves. The NLP community has attempted to automatically situate LLMs in the appropriate role for the user across multiple rounds of dialogue guided by well-designed prompts[3]. However, this prompt-based automation method tends to complicate the interaction process and introduce an excessive number of additional tokens, leading to diminished practicality.

While prompting strategies have positively modulate the behavior of LLMs in a cost-efficient manner, the pursuit of directly adjusting model parameters has led to the emergence of new methods like instruction tuning (IT) (Wei et al., 2022a; Wang et al., 2023a; Zhou et al., 2023a). Through fine-tuning LLMs on a collection of datasets described via instructions, IT enables LLMs to follow huamn instructions without any additional prompts. Building on this foundation, this paper introduces **self-prompt tuning**, an innovative approach that enables LLMs to autonomously establish an appropriate role (i.e., role-play prompting) and respond accordingly through fine-tuning. Specifically, we leverage GPT-4 with in-context learning to reconstruct LIMA (Zhou et al., 2023a), a small scale IT datasets, by adding corresponding role descriptions to each question. The resulting dataset is termed LIMA-Role. Subsequently, we fine-tune LLMs, such as Mistral-7B and Llama-2-7B, on this augmented dataset. The self-prompt tuned LLMs can automatically generate corresponding role-play prompts for a given question as shown in Figure 1. We compare self-prompt tuned LLMs with instruction tuned baselines using 8 traditional benchmarks and an open-ended question test. Our results demonstrate consistent improvements over standard instruction tuned baselines on the majority of datasets, proving the efficacy of self-prompt tuning.

To the best of our knowledge, self-prompt tuning is the first to make LLMs themselves to generate prompts by fine-tuning. Our method opens a new avenue for automating diverse prompting strategies. We believe our work will catalyze further exploration in automating more advanced prompting techniques, such as least-to-most prompting (Zhou et al., 2023b) and tree-of-thought prompting (Yao et al., 2023).

Our main contributions are as follows:

- We propose self-prompt tuning, a novel approach achieving automation of role-play prompting through fine-tuning LLMs.
- We release LIMA-Role, an enhanced version of the LIMA dataset annotated with role-play

[3]https://github.com/JushBJJ/Mr.-Ranedeer-AI-Tutor

prompts using GPT-4, alongside LLMs fine-tuned on this dataset.

- We thoroughly evaluate self-prompt tuned LLMs using 8 traditional benchmarks and an open-ended question test, demonstrating the efficacy of self-prompt tuning.

2 Related Work

2.1 Instruction Tuning

Original pre-trained large language models (LLMs) excel as few-shot learners but struggle in zero-shot scenarios. Wei et al. (2022a) propose instruction tuning, a technique that fine-tunes LLMs on a diverse set of NLP datasets described via instructions, significantly improving their zero-shot performance. Following this approach, subsequent works like T0 (Sanh et al., 2022), FLAN-T5 (Chung et al., 2024), and ZeroPrompt (Xu et al., 2022) expand the variety of tasks and the scale of data used for instruction tuning, further enhancing the models' capabilities. However, the data utilized in these works originated from traditional NLP datasets, which still lack diversity and complexity compared with real queries of human users. To solve this problem, researchers have attempted to leverage human annotators or LLMs to construct new datasets that better align with real-world human instructions. OpenAssistant (Köpf et al., 2023) is an open-source assistant-style conversation corpus annotated by worldwide crowd-sourcing. Self-Instruct (Wang et al., 2023a) generates 52k instruction-response pairs based on 175 manually-written prompts using LLMs. Evol-Instruct (Xu et al., 2024) also relies on an initial set of instructions and employs LLMs to iteratively rewrite them into more complex instructions. LIMA (Zhou et al., 2023a) trains a LLM that approaches the capabilities of proprietary models using small-scale but high-quality data collected from wikiHow, Stack Exchange, and Reddit. Orca (Mukherjee et al., 2023) progressively fine-tunes LLMs on a massive corpus generated by GPT-4 to enhance their reasoning abilities. Essentially, instruction tuning alleviates the burden on users to craft prompts. And our proposed self-prompt tuning takes a further step by automating more complex prompting strategy.

2.2 Role-playing Abilities of LLMs

Modern LLMs exhibit remarkable adaptability and interactive capabilities in role-playing tasks. These models can flexibly adjust their output style according to the needs of different roles, providing users with a customized conversation experience. Shanahan et al. (2023) advocates LLMs as role simulators and warns against falling into the trap of anthropomorphism. Wang et al. (2023b) propose RoleLLM, a role-playing framework of data construction and evaluation. Beyond facilitating immersive interactions, role-playing can also enhance the model's performance across downstream NLP tasks. Wu et al. (2023) employ LLMs to emulate judges possessing unique personas and backgrounds, thereby enhancing the quality of their summarization assessments. Salewski et al. (2023) direct Large Language Models (LLMs) to embody domain-specific expertise, leading to enhanced performance in multi-domain QA tasks. Kong et al. (2023) immerse LLMs in diverse expert roles via multi-turn dialogues, thereby augmenting their reasoning capabilities. Role-play is also employed in LLM-based multi-agent frameworks (Park et al., 2023; Xiong et al., 2023; Liang et al., 2023). These studies utilize role-play prompting to facilitate the cooperative interaction among multiple agents. While the efficacy of role-play prompting has been demonstrated, the need of manually crafting prompts for each task hinders its broader application. To alleviate this bottleneck, we propose self-prompt tuning, a novel approach that automates prompt design by LLMs themselves, thereby minimizing human intervention.

2.3 Prompting Strategies

Extensive research and practice have demonstrated that prompts significantly impact the performance of LLMs. To fully unlock the potential of LLMs, various complex prompting strategies, not just role-play prompting, have been developed. Least-to-most prompting (Zhou et al., 2023b) decomposes the original problem into simpler subproblems and solves them in sequence. Self-refine prompting (Madaan et al., 2023) generates an output first and then employs the same LLM to provide feedback and refinement, iteratively improving the initial output. Tree-of-thought (Yao et al., 2023) prompting represents potential reasoning paths as a branching tree structure and utilizes search algorithms like DFS or BFS to explore and identify the correct reasoning path. Step-back prompting (Zheng et al., 2024) involves abstracting information to derive high-level concepts and first principles, which are then utilized to guide the reasoning process. These

Generation of Role-Play Prompts by GPT-4

USER: Next, I will provide you with a question, and you should select a role who is suitable to answer this question. The selection of roles should adhere to the following principles:
1. Try to choose a role that is skilled at answering the given question.
2. Try to choose specific professions.
3. Try to choose the roles that match the difficulty level of the problem.
No need to actually answer the question. Please format your response as follows:
A: This question is a [] problem involving []. To better solve it, I will act as a [] who [].
Here is an example:
A: This question is a mathematical problem involving basic arithmetic and logic. To better solve it, I will act as an excellent math teacher who can always provide clear explanations for math-related problems.
Here is the question I need you to deal with:
Q: Can brain cells move? By movement I mean long distance migration (preferably within the brain only).

ASSISTANT (GPT-4): A: This question is a neuroscience problem involving cell biology and migration. To better solve it, I will act as a neuroscientist who specializes in the study of the brain and its cellular behaviors.

Data Format for Fine-tuning LLMs on LIMA-Role

Below is a chat between a user and an AI assistant.
USER: Can brain cells move? By movement I mean long distance migration (preferably within the brain only).
ASSISTANT: [This question is a neuroscience problem involving cell biology and migration. To better solve it, I will act as a neuroscientist who specializes in the study of the brain and its cellular behaviors.]
The question is relatively broad and one should take into account that the brain not only consists of neurons, but also glial cells (supportive cells) and pre-mitotic neuronal stem cells ... Most notably, post-mitotic neurons destined to fulfill peripheral functions have to migrate over relatively long distances from the neural crest to their target locations (Neuroscience, 2nd ed, Neuronal Migration).

Figure 2: An illustration of LIMA-Role dataset construction process. The upper sub-image displays the prompt used for GPT-4 role-play prompt annotation. The lower sub-image shows how role-play prompts are utilized to construct LIMA-Role. The question to be annotated and the corresponding role-play prompts generated by GPT-4 are highlighted in gray and blue, respectively.

prompting strategies necessitate providing few-shot examples to guide LLMs in following a specific thought pattern. Our proposed self-prompt tuning introduces a novel approach that involves constructing a dataset embodying the desired thought process and then fine-tuning LLMs to inject this thinking pattern into their parameters. Our experiments have demonstrated the success of this method in role-play prompting. And we leave the extension of self-prompt tuning in other prompting strategies to future work.

3 Self-Prompt Tuning

In this section, we introduce our proposed self-prompt tuning in detail. Self-prompt tuning consists of two steps as follows: (1) Modify an existing instruction tuning dataset to include role-play prompts. (2) Fine-tune LLMs on the resulting dataset to enable them automatically generate role-play prompts tailored to the specific questions.

3.1 Construct LIMA-Role Dataset

The small scale yet high-quality instruction tuning dataset, LIMA (Zhou et al., 2023a), comprises 1,000 single-turn dialogues and 30 multi-turn dialogues, making it highly suitable to serve as a foundational dataset. Studies by Salewski et al. (2023); Kong et al. (2023) demonstrate that taking on expert roles for a given task can typically enhance the model's performance. Building on this premise, we employ GPT-4 in one-shot manner to generate expert role-play prompts for each training instance in LIMA (only consider the first question for multi-turns data). These role-play prompts are then prefixed to the corresponding answers, yielding a new dataset, LIMA-Role. Inspired by chain-of-thought prompting (Wei et al., 2022b), the question summarization is also designed into the role-play prompt, aiming to help generate correct role descriptions. We provide prompts utilized for GPT-4 and an example illustrating the process of modifying one data instance in Figure 2. Additionally, GPT-4 declines to generate role prompts to some unsafe, biased or unethical questions in

LIMA, 14 in total. We manually design prompts with the role of "AI assistant" for these questions.

While LLMs have demonstrated remarkable capabilities in data annotation tasks (Wang et al., 2023a; Xu et al., 2024, 2023), it remains necessary to validate the data quality of LIMA-Role. We conduct a random selection of 100 entries from the dataset to undergo manual evaluation, focusing on three key aspects: formatting, question summarization, and role description. The assessment reveals that 100% of the entries maintain a consistent format, 96% correctly summarize the questions, and 97% offer appropriate role descriptions. Therefore, we conclude that the data quality of LIMA-Role meets our criteria.

3.2 Fine-tune LLMs on LIMA-Role

After completing the construction of LIMA-Role, we fine-tune original pre-trained LLMs like Mistral-7B on that dataset with the standard supervised loss. We organize the data in the form of interaction between "AI assistant" and "user", and set a fixed system prompt, as shown in Figure 2.

4 Experiments

4.1 Tasks and Datasets

Initial investigations into instruction tuning (Zhou et al., 2023a; Xu et al., 2024) involved comparing various LLMs' responses to open-ended questions, utilizing both human and GPT-4 assessments to gauge their quality. Gudibande et al. (2024) highlighted that relying solely on this evaluation method may result in an overestimation of model quality. Therefore, we combine traditional NLP benchmarks and open-ended questions to comprehensively evaluate the efficacy of self-prompt tuning.

NLP Benchmarks We hope that self-prompt tuned LLMs can automatically generate expert role-play prompts for different questions. Therefore, datasets containing multi-domain problems are highly suitable for evaluation. MMLU (Hendrycks et al., 2021) is a multi-domain QA dataset and has been widely used to evaluate LLMs. We sample 2000 questions from MMLU, balanced across 10 categories (35 subcategories). CSQA (Talmor et al., 2019), StrategyQA (Geva et al., 2021), TruthfulQA (Lin et al., 2022), and OpenBookQA (Mihaylov et al., 2018) are also muti-domain datasets and included. We additionally add GSM8K (math) (Cobbe et al., 2021), HumanEval (code) (Chen et al., 2021), Date Understanding (reasoning) (Srivastava et al., 2023) to enrich the form and content of the evaluation. More details can be found in Table 2.

Open-ended Questions We leverage the LIMA test set, comprising 300 challenging questions authored by real users, to assess the capabilities of LLMs. See more details in Table 2.

4.2 Experimental Setup

Models We self-prompt tune original Mistral-7B and Llama-2-7B, which are the leading open-source LLMs at the time of writing.

Baselines In addition to comparing self-prompt tuned LLMs on LIMA-Role and instruction tuned LLMs on original LIMA, we also present the experimental results of ChatGPT (gpt-3.5-turbo-0125), Llama-2-chat (the official version), and Mistral-instruct (the official version) to enhance our comprehension of the models' capabilities.

Training Details In line with prior research (Zhou et al., 2023a), we respectively conduct fine-tuning of Mistral-7B on LIMA and LIMA-Role datasets for 4 epochs, employing AdamW optimization with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.1. We initialize the learning rate to $1e-5$ without warmup, implementing a cosine decay schedule that decreases to 0 by the end of training. The batch size is set to 64, with a maximum token limit of 4096. To mitigate overfitting, dropout is applied to attention calculations, starting at p_d = 0.0 at the bottom layer and linearly raising the rate to p_d = 0.25 at the last layer. We utilize FlashAttention-2 (Dao, 2024) to optimize memory usage and expedite training. The method and parameter settings for fine-tuning Llama-2-7B mirror those of Mistral-7B, differing only in the number of training epochs, which is set to 8. Training is performed on 4 A100-80G. Due to the small data scale of LIMA dataset, model performance exhibits variability; hence, we fine-tune four models for the same dataset using different seeds and average their performance across NLP benchmarks.

Evaluation Details For both NLP benchmarks and the LIMA test set, evaluations are conducted in a zero-shot manner, without any few-shot exemplars. Consistent with prior studies (Kojima et al., 2022; Kong et al., 2023), we employ greedy decoding with a temperature of 0 to ensure deterministic results. While averaging the performance of four models fine-tuned on the same dataset across NLP

Model	MMLU	CSQA	Strategy	Truthful	OpenBook	HumanEval	GSM8K	Date	AVG
ChatGPT	67.3	76.9	61.7	60.2	81.6	68.3	80.8	67.8	70.6
Llama-2-7B									
Llama-Chat	44.0	58.6	59.0	40.4	63.6	13.7	29.3	49.3	44.7
Llama-LIMA	40.4	48.6	55.5	39.7	48.2	9.4	13.5	43.1	37.3
Llama-Role	**42.9**	**57.3**	**59.5**	**47.8**	**52.1**	8.7	**13.6**	**43.1**	**40.6**
Llama-LIMA†	41.8	49.5	57.2	38.9	**50.6**	**9.4**	14.0	**44.2**	38.2
Llama-Role†	**44.1**	**58.0**	**59.6**	**48.0**	50.2	8.5	**14.5**	42.8	**40.7**
Mistral-7B									
Mistral-Instruct	51.1	66.4	60.2	51.8	72.2	33.2	35.2	56.4	53.3
Mistral-LIMA	53.2	52.6	58.5	43.9	63.1	25.9	22.4	40.6	45.0
Mistral-Role	**56.0**	**59.8**	**61.9**	**46.1**	**68.2**	**26.6**	**25.8**	**42.7**	**48.4**
Mistral-LIMA†	53.4	54.8	59.3	42.7	63.4	**27.9**	20.4	**42.5**	45.6
Mistral-Role†	**57.1**	**61.3**	**62.8**	**45.3**	**69.6**	27.8	**27.1**	42.0	**49.1**

Table 1: The performance of self-prompt tuned LLMs, standard instruction tuned LLMs (LIMA version and official version), and ChatGPT on each dataset. Without †: average performance of the four models. With †: results from the model with the best average performance among the four models.

Dataset	N_q	L_q	Format
MMLU	2000	79.4	option (A-D)
CSQA	1221	27.8	option (A-E)
StrategyQA	2290	9.6	yes or no
TruthfulQA	817	47.3	option (A-D)
OpenbookQA	500	26.5	option (A-D)
HunamEval	164	67.7	code
GSM8K	1319	46.9	arabic number
Date	369	35.0	Option (A-F)
LIMA-Test	300	21.3	free

Table 2: Relevant information of benchmarks and LIMA test set. N_q denotes the number of questionsin each dataset. L_q denotes the average words of questions in each dataset. Format denotes the answer format of each dataset.

benchmarks, we select the model with the best average performance from the four and evaluate it on the LIMA test set. The quality of their responses is assessed using GPT-4 (gpt-4-1106-preview, we adopt the prompt proposed by Zhou et al. (2023a)). Role-play prompts generated by self-prompt tuned LLMs are invisible to GPT-4 to ensure fairness.

4.3 Results on NLP Benchmarks

Detailed experimental results on NLP benchmarks are presented in Table 1. We report both the average performance and peak performance of LLMs simultaneously. For HumanEval, the evaluation metric utilized is pass@1, whereas accuracy serves as the metric for the remaining datasets.

Average Performance Comparison As shown in Table 1, self-prompt tuned LLMs consistently outperform those instruction-tuned on LIMA across the majority of benchmarks, demonstrating the efficacy of our approach. Delving deeper, we compare the performance of Mistral-Role and Mistral-LIMA on domain-specific subsets within the MMLU. According to the results in Figure 3, Mistral-Role outperforms Mistral-LIMA in 9 out of 10 domains (28 out of 34 subcategories) revealing that self-prompt tuning is beneficial across a diverse range of fields. Moreover, to assess the capability of self-prompt tuned LLMs to automate role-play prompting, we extract roles automatically generated by Mistral-Role for questions in each domain-specific subset in MMLU. By identifying and visualizing the most frequent roles through word clouds in Figure 4, we observe that Mistral-Role assigns appropriate expert roles to questions across different domains. This highlights that self-prompt tuning successfully enables LLMs to autonomously generate role-play prompts. We also observe that self-prompt tuned LLMs exhibit unstable performance improvement on single-domain tasks compared to multi-domain QA tasks (Llama-Role on HumanEval, GSM8K, and Date). Kong et al. (2023) reveal that while expert roles generally brings performance gains, this improvement is not guaranteed. In single-domain

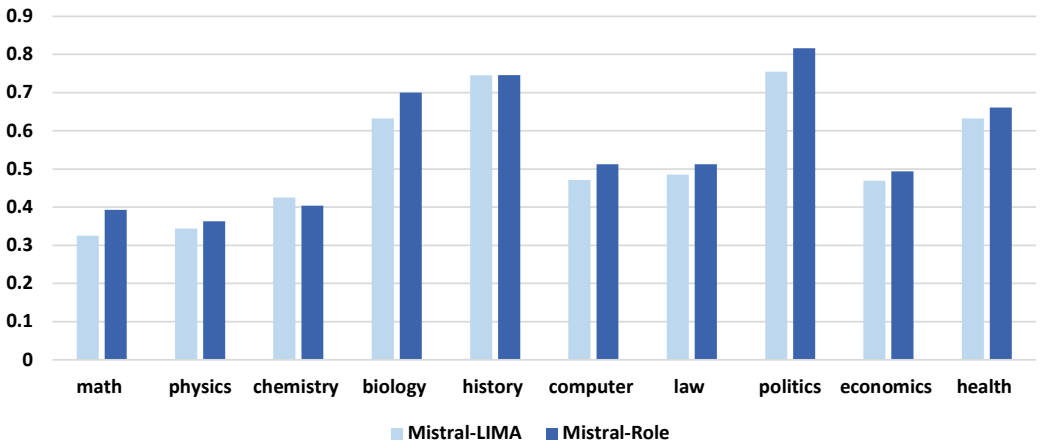

Figure 3: The performance comparison between Mistral-LIMA and Mistral-Role across various domain-specific subsets in MMLU. Mistral-Role outperforms Mistral-LIMA in 9 out of 10 domains and underperforms in chemistry.

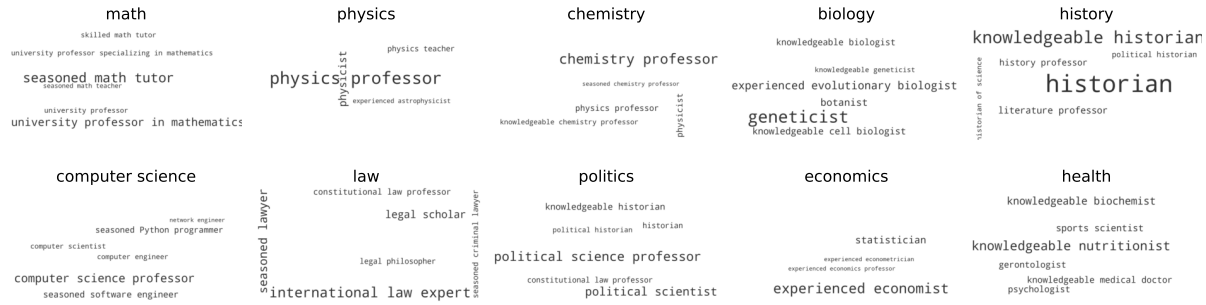

Figure 4: Word clouds based on roles generated by Mistral-Role across domain-specific subsets in MMLU. Words characterized by larger font sizes and deeper color correspond to higher frequencies.

tasks, where the format of questions tends to be highly consistent, the role-play prompts generated by self-prompt tuned LLMs are quite similar. This lack of diversity in the prompts likely contributes to the observed instability in performance improvements. Conversely, for multi-domain QA tasks, the diversity in the generated role-play prompts is notably higher, leading to stable improvement. Thus, the limited improvement of Llama-Role in single-domain tasks can be attributed to this factor.

Peak Performance Comparison Self-prompt tuned LLMs with the best average performance still surpass standard instruction tuned baselines as indicated in Table 1. However, when comparing with official instruction-tuned versions, the self-prompt tuned LLMs tend to underperform. It's crucial to emphasize that both Llama-Role and Mistral-Role are fine-tuned on only 1030 data points, whereas the official versions are fine-tuned on datasets exceeding 10,000 data points and undergo complex RLHF (Ouyang et al., 2022). This discrepancy in training dataset scale and methodology accounts for the performance differences observed.

4.4 Results on Open-ended Questions

We select self-prompt tuned and standard instruction tuned Mistral-7B with the best average performance to conduct open-ended question test. Results annotated by GPT-4 are depicted in Figure 5. Despite only inserting non-substantive role-play prompts into the LIMA dataset, Mistral-Role still generate better responses than Mistral-LIMA 5% of the time, further underscoring the widespread effectiveness of self-prompt tuning. Nonetheless, Mistral-Role exhibits subpar performance compared to the official version and ChatGPT, indicating that merely 1,030 high-quality data points are insufficient for effectively fine-tuning a 7B-parameter model.

No.	Prompt	MDQA	SDTask
0	None	54.3	29.6
1	[Question Description].	53.8	29.5
2	[Question Description]. As a result, I will solve it like [Role Description].	57.3	31.0
3	[Question Description]. Therefore, I will answer it as [Role Description].	57.4	31.8
4	[Question Description]. To solve this problem, I will act as [Role Description].	57.9	24.7
5	[Question Description]. So I will become [Role Description].	58.6	31.3
6	[Question Description]. Fortunately, I am [Role Description].	58.4	32.9
7	[Question Description]. For this reason, I will be [Role Description].	57.4	30.6
8	[Question Description]. From now on, I will think like [Role Description].	58.4	31.7

Table 3: The performance of Mistral-Role adopting different prompt designs. Similarly, we train four models for each prompt design with different random seeds and report the average performance here.

4.5 Ablation Study

While the performance of LLMs is highly sensitive to the prompt in various prompting strategies, the influence of prompt design on fine-tuning models remains unexplored. Given the high cost of accessing GPT-4, we maintain the question description and role description, only modifying the left sections of the prompt. The prompts we design and their practical results on Mistral are summarized in Table 3. Prompt 1, containing only the question description, achieves the lowest performance, thereby eliminating interference from question descriptions. Prompts 2-8, which add role descriptions with variations at the junctions, consistently show improvements in both multi-domain QA tasks and single-domain tasks. Among these, Prompts 6 and 8 exhibit relatively optimal performance. We ultimately select Prompt 8, which demonstrates the most balanced performance improvement across each dataset, as the final design. The results indicate that prompt design also impacts the performance of fine-tuning LLMs, but not as sensitively as in non-fine-tuning scenarios.

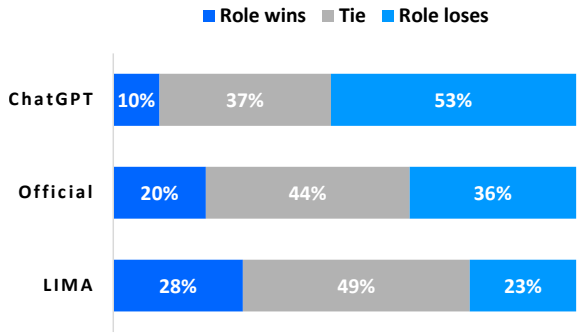

Figure 5: Preference evaluation on LIMA test set using GPT-4 as the annotator. In this context, LIMA refers to Mistral-LIMA, while Role denotes Mistral-Role.

5 Conclusion

In this paper, we propose self-prompt tuning, a novel approach that enables large language models (LLMs) to autonomously generate role-play prompts through fine-tuning. By first constructing the LIMA-Role dataset, which augments the LIMA dataset with expert role-play prompts generated by GPT-4, and then fine-tuning LLMs on this dataset, self-prompt tuned LLMs gained the ability to automatically generate relevant expert role-play prompts tailored to any given question. Comprehensive evaluations on 8 traditional NLP benchmarks and an open-ended question test reveal that self-prompt tuned LLMs consistently outperform standard instruction tuned baselines across the majority of datasets. The results highlight the efficacy of self-prompt tuning in automating role-play prompting. Overall, this work paves a promising new path for automating diverse complex prompting strategies.

Limitations

Due to its small scale and ease of modification, we select the LIMA dataset as the foundational dataset. However, the data scale of 1,030 samples is insufficient to fully fine-tune a 7B parameter model, rendering our models unable to make a meaningful performance comparison with ChatGPT and the official versions. Moreover, we only manually make limited attempts at designing role-play prompts for the LIMA-Role dataset, and cannot guarantee that the optimal effects of self-prompt tuning were achieved. Last, owing to limited computational resources, we are unable to apply our method on LLMs with larger parameter scales. Consequently,

we could not obtain conclusions about how the effects of self-prompt tuning vary as the scale of model parameters increases.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Arnav Gudibande, Eric Wallace, Charlie Victor Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2024. The false promise of imitating proprietary language models. In *The Twelfth International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou. 2023. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Minh Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Alexandrovich Glushkov, Arnav Varma Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Julian Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. 2023. In-context impersonation reveals large language models' strengths and biases. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

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, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan,

Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhu Chen, Jie Fu, and Junran Peng. 2023b. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M.

Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Ning Wu, Ming Gong, Linjun Shou, Shining Liang, and Daxin Jiang. 2023. Large language models are diverse role-players for summarization evaluation. *arXiv preprint arXiv:2303.15078*.

Kai Xiong, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7572–7590, Singapore. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6268–6278, Singapore. Association for Computational Linguistics.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Wang Yanggang, Haiyu Li, and Zhilin Yang. 2022. ZeroPrompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4235–4252, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023a. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023b. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.