

TEXT AS PARAMETER: INTERACTIVE PROMPT OPTIMISATION FOR LARGE LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Large language models (LLMs) can handle a variety of tasks conditioned on natural language instructions. While fine-tuning improves task-specific performance, adjusting the model weights of LLMs requires a huge amount of computational resources, and it is impractical for real-time updates. Alternatively, prompting allows LLMs to adapt to a broad range of tasks without the need for computationally intensive gradient-based optimisation. However, crafting effective prompts remains a challenge, to the extent that it is even unclear if expert in-domain knowledge is what is needed or experience in writing prompts or something else. Approaches like meta-prompting and self-feedback seek to alleviate this burden, but they rely primarily on a numerical feedback signal, leaving the potential of textual feedback unexplored. These methods also typically require numerous interactions with the environment to gather sufficient context, leading to significant computational overhead.

In this work, we propose a novel framework that takes a prompted large language model as an optimiser and treats the text-based prompt itself as a parameter. By interacting with the environment to collect feedback, our proposed method constructs the updated textual prompt. Our experimental results demonstrate that this method not only achieves superior performance but also automatically incorporates domain-specific knowledge, establishing a scientifically motivated, practical and efficient approach to prompting for future research.

1 INTRODUCTION

Large language models (LLMs) have shown an extraordinary ability to perform a wide range of tasks, from generating images in various styles to writing code in different programming languages for diverse purposes. These tasks, once considered extremely challenging for artificial intelligence and even beyond the average human’s skill set, are now achievable by simply interacting with LLMs. Recently, increasing research efforts have been put into specialisation of LLMs, i.e. improving its capability on specific tasks, in the most effective and efficient manner.

The first family of methods belong to the gradient-based optimisation approaches, such as supervised fine-tuning with low-rank adaptation (Hu et al., 2022), and continuous prompting (Lester et al., 2021; Qin & Eisner, 2021; Li & Liang, 2021; Liu et al., 2023). Although LLMs fine-tuned in this way can excel on the specific target tasks, their in-context learning capabilities may be reduced, and they may also suffer from catastrophic forgetting, losing previously acquired ability when learning the new skill (Zhai et al., 2023; Luo et al., 2023; Wang et al., 2024). In addition, training LLMs via gradient-based optimisation is impractical for real-time updates. The computational requirements for the model size lead to challenges when resources are limited local compute only. More detrimentally, these methods are simply inapplicable to LLMs with only API access.

The second family of methods focus on precise instructions and task descriptions to adapt LLMs via natural language prompting, without updating their model weights. However, creating proper and effective prompts is not straightforward, requiring some tricks, making it at times more similar to art rather than science. Via trial-and-error, various tricks have been found to improve prompting of language models, some of which are widely shared in discussion forums on the internet¹. Consequently,

¹For example, <https://www.reddit.com/r/ChatGPTPromptGenius/>

approaches like self-feedback or meta-prompting have been introduced to enhance the performance of LLMs without gradient-based optimisation. Self-feedback methods, e.g. LLM-augmenter (Peng et al., 2023) or Reflexion (Shinn et al., 2023), gather sufficient experience by interacting with the environment multiple times and refine the model’s response based on verbal summarisation of experiences. In other words, when serving end users, these methods still require numerous queries for self-reflection and refinement of their response, which can cause a huge cost on computation and non-negligible inference time.

On the other hand, meta-prompting methods aim to automatically generate a proper prompt based on a given set of examples or refine a prompt based on scores. Yang et al. (2024) propose Optimisation by PROMpting (OPRO), which employs LLMs to refine an initial prompt based on a number of performance metrics. Tang et al. (2024) introduce the Gradient-inspired LLM-based Prompt Optimizer (GPO), which also uses numerical feedback, controlling edit distance via a cosine decay mechanism. Kong et al. (2024) and Cheng et al. (2024) propose sequence-to-sequence prompt rewriters, trained by reinforcement learning or human preference data, to reduce the difference between human intention and the understanding of LLMs. Although these methods are capable of generating a better prompt, they highly rely on numerical feedback, which is not always available when facing real users.

We propose the *Text-as-Parameter optimisation* (TaP), a meta-prompting approach which leverages natural language textual feedback. Our approach is based on the way humans learn new things: they need instructions, experiences, and feedback. Various studies have shown humans learn better with informative feedback than only simply instruction (Kulhavy, 1977; Mory, 1992; Lyster & Mori, 2006; Hattie & Gan, 2011). To echo this sentiment, in TaP, we first initialise the prompt via corpus-based prompt generation, followed by iterative meta-prompting updates through an interactive optimisation. More specifically, the LLM-based system first interacts with the environment, e.g. real or simulated users trying to complete a particular task such as information seeking or medical question answering. A feedbacker, e.g. human experts or LLMs, provides textual-based feedback based on these interactions, then a separate LLM-based rewriter refines the prompt based on the feedback and the prompt from the previous interaction. The refined prompt is then used by the system on the next user interaction, requiring no additional memory to store previous interactions or past feedback. Textual feedback, which has more information than a scalar or a binary preference, can provide richer information for prompt optimisation.

Our contributions are as follows:

- We propose a Text-as-Parameter optimisation method which leverages textual feedback from external resources for improved prompting.
- Our method can be generalised through different prompting styles, applied with different LLMs, e.g. GPT-4o mini or Gemini-1.5-flash, and is task-agnostic.
- Our method learns domain-specific knowledge through interaction and incorporates it into the optimised prompt. Our experiments show that this reduces the performance gap between different initial prompting methods.

2 RELATED WORK

Gradient-based optimisation for LLMs For high parameter counts, training or fine-tuning an entire large language model is infeasible since it requires a huge amount of computation resources. As a result, parameter-efficient fine-tuning, such as training only part of the model or freezing the model and training an adapter, is widely used to refine LLMs (Hu et al., 2022; 2023; Lialin et al., 2023). On the other hand, continuous prompting, e.g. prefix-tuning and soft-prompting, is also popular to adapt LLMs to specific tasks or improve their performance (Lester et al., 2021; Qin & Eisner, 2021; Li & Liang, 2021; Liu et al., 2023). By updating inputs of every attention layer (Li & Liang, 2021), or task-related vectors (Lester et al., 2021), these methods can achieve comparable performance to full fine-tuning across various model sizes and tasks (Liu et al., 2022). Although these methods can improve LLMs effectively, they do not apply to API-access-only LLMs and such training processes cannot be carried out in real-time.

Self-feedback To improve the performance of text-based prompts, various prompting styles are proposed, e.g. Chain-of-Thought (Wei et al., 2022) or ReAct (Yao et al., 2023). These prompting methods encourage LLMs to reason before taking action or generating responses, which leads to better performance. However, optimising the prompt for better performance by manual trial and error is inefficient. As a result, self-feedback methods are introduced to refine the LLMs’ response, e.g., LLM-augmenter generates feedback by itself and leverages external knowledge to rewrite its response (Peng et al., 2023) and Reflexion summarises previous interactions with the environment as ‘reflections’ to improve the model’s response (Shinn et al., 2023).

While this demonstrates the ability of LLMs for self-correction, these self-feedback methods rely on frequent API calls since their original prompt is not optimal. As a result, the computation cost and latency during inference are not negligible.

Prompt optimisation Meta-prompting methods are widely used to generate a prompt without human editing. The automatic prompt engineer (APE) method leverages an LLM which is instructed to generate an initial prompt and selects the prompt with the best performance on the target task (Zhou et al., 2023). Automatic prompt optimisation (APO) further employs a self-feedback module to provide textual feedback which gives suggestions on how to edit the old prompt (Pryzant et al., 2023). Ye et al. (2024) propose a meta-prompt LLM to edit the original prompt step-by-step. Kong et al. (2024) and Cheng et al. (2024) train a sequence-to-sequence model for prompt rewriting by reinforcement learning and preference data, respectively. Yang et al. (2024) propose optimisation by prompting (OPRO), which leverages LLMs to rewrite the original prompt based on a corresponding performance score. In addition, to properly include experience for improving performance, Zhang et al. (2023) treats LLMs as semi-parametric reinforcement learning agents with a stored experience memory, including task information, observation, action and corresponding Q value estimation. These experiences are sampled dynamically for few-shot in-context learning. Zhang et al. (2024) propose Agent-Pro, which constructs policy-level reflections according to the numerical feedback from the environment and improves its policy incrementally. Tang et al. (2024) introduce the Gradient-inspired LLM-based Prompt Optimizer (GPO), which updates the prompt iteratively based on numerical feedback and controls the edit distance through a cosine-based decay strategy.

Although these methods demonstrate promising performance in generating or improving a prompt, they do not leverage external textual feedback, which could provide abundant information for optimisation. The comparison between our method and the mentioned related works is listed in Table 1.

Learnability of LLMs with prompting Although transformers are universal approximators (Yun et al., 2020) and in-context learning in LLMs can be viewed as implicit fine-tuning (Dai et al., 2023), the following remain open questions: Can we prompt LLMs for arbitrary tasks, and what are the limitations of in-context learning?

Petrov et al. (2024) highlight the limitations of context-based fine-tuning methods, e.g. in-context learning, prompting and prefix tuning, for new task learning in transformers. Specifically, transformers struggle to acquire new tasks solely through prompting, as prompts cannot change the model’s attention patterns. Instead, they can only bias the output of the attention layers in a fixed direction and elicit skills learned through pre-training. In other words, only models with billions of parameters trained on vast, diverse datasets are capable of in-context learning—adapting to new tasks through examples or instructions without modifying their underlying weights. Therefore, we focus on fundamental models large enough to demonstrate their in-context learning ability, to investigate text-as-parameter prompt optimisation, which is fully composed of in-context learning with LLMs.

Table 1: Comparison of our proposed method and related works

Method	What is optimised	Frequent API-calls	Final feedback
Gradient-based	model weights	-	-
Self-feedback	-	on inference	self-generated
Meta-prompting	system prompt	on optimisation	numerical
Text-as-Parameter (ours)	system prompt	on optimisation	external textual

3 TEXT-AS-PARAMETER

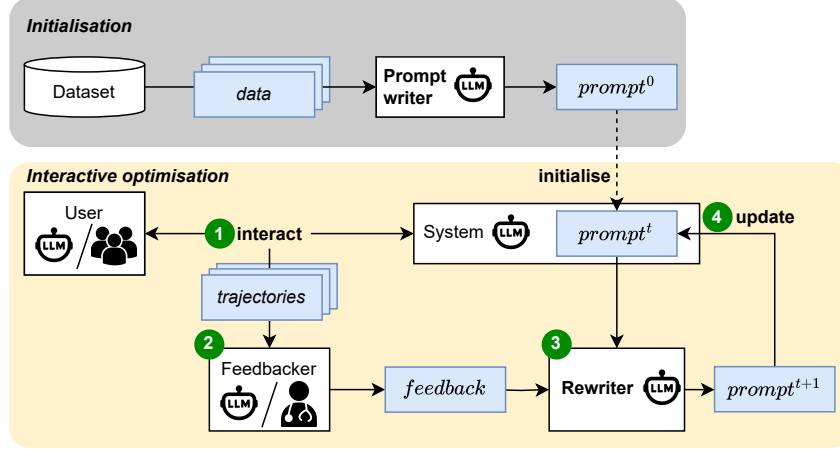


Figure 1: The structure of Text-as-Parameter optimisation. For initialisation, the prompt writer generates an initial $prompt^0$ based on sampled data. In interactive optimisation, the system will first interact with the environment, e.g. simulated or real users. The feedbacker, e.g. human experts or LLMs, will provide textual feedback based on trajectories. The rewriter generates a new prompt based on the original prompt and the textual feedback to update the system’s original prompt. One cycle of interactive optimisation is called an epoch.

The structure of the **Text-as-Parameter** (TaP) optimisation method is shown in Figure 1. The initial instruction is generated by a prompt writer LLM_P in the initialisation phase (the upper part of Figure 1) inspired by the automatic prompt engineer (APE) (Zhou et al., 2023), where we do not provide demonstrations to the prompt writer but APE does. LLM_P will generate an initial instruction for the system, $prompt^0$, based on a corpus \mathbb{D} for the target task, i.e.

$$prompt^0 \leftarrow LLM_P(\mathbb{D}'), \quad (1)$$

where \mathbb{D}' is sampled interactions from the dataset to fit into the limited context length of LLMs.

In the interactive optimisation (the lower part of Figure 1), the **system** will interact with the environment, e.g. human users or simulated users, and generate several *trajectories*. These trajectories can be single-turn, e.g. question-answering task, or a multi-turn conversation, e.g. information seeking or recommendation in task-oriented dialogue. Then the **feedbacker**, which can be a language model LLM_F or human experts, will provide textual feedback to guide the optimisation direction for the **rewriter** LLM_R , which will generate a new prompt to improve the system’s performance based on the feedback and original prompt. The improved prompt is then used by the system in the next interaction. That is,

$$feedback = LLM_F(trajectories), \quad (2)$$

$$prompt^{t+1} \leftarrow LLM_R(prompt^t, feedback). \quad (3)$$

An example of textual feedback is shown in Figure 2b. It may include strengths, suggestions, and an overall impression for improvement. In contrast, numerical feedback (Yang et al., 2024; Tang et al., 2024) only includes a number quantifying the overall performances, as shown in Figure 2a.

To alleviate human effort for task-specific manual prompt adjustment, our proposed method creates and updates the prompt with LLMs only, where the feedback signal can come from simulated environments or human users. Note that while the feedbacker and rewriter LLMs also require a prompt, they are task-independent and therefore they only need to be engineered once, unlike for the system LLM which may perform a variety of interactive tasks.

The two phases in our method, initialisation and interactive optimisation, are similar to the supervised fine-tuning and reinforcement learning from human feedback respectively. In gradient-based optimisation, the goal is to find the optimal numerical parameters θ , which are initialised by supervised fine-tuning with a loss function $L(\cdot)$ and a dataset \mathbb{D} . Then they are further refined through

reinforcement learning using a reward model $f_r(\cdot)$ and an expected return function $J(\cdot)$. In contrast, Text-as-Parameter optimisation seeks to generate the optimal instruction, *prompt*, for an LLM-based system. The prompt is initialised by a prompt writer LLM_P and is iteratively improved by a rewriter LLM_R according to feedback from LLM_F . The comparison is summarised in Table 2.

Table 2: Comparison of Gradient-Based Optimisation and Text-as-Parameter Optimisation.

	Gradient-based optimisation	Text-as-Parameter optimisation
Supervised fine-tuning	Learned by gradient descent $\theta^0 \leftarrow \theta - \nabla L(\mathbb{D})$	Generated by a prompt writer LLM_P $\text{prompt}^0 \leftarrow \text{LLM}_P(\mathbb{D})$
Reinforcement learning	Reward model f_r $r = f_r(\text{trajectories})$ Optimised by policy gradient $\theta^{t+1} \leftarrow \theta^t - \nabla J(\theta^t)$	Feedbacker LLM_F $\text{feedback} = \text{LLM}_F(\text{trajectories})$ Optimised by Rewriter LLM_R $\text{prompt}^{t+1} \leftarrow \text{LLM}_R(\text{prompt}^t, \text{feedback})$

Numerical feedback	Textual feedback
<p>I have some prompts along with their corresponding scores. The texts are arranged in ascending order based on their scores, where higher scores indicate better quality.</p> <p>prompt: You are an intelligent assistant. Your goal is to assist the user in finding and booking things that fit their specific needs. You should be polite,...</p> <p>scores: 0.62 ...</p>	<p>The system did a commendable job assisting the user in planning their trip to Cambridge by providing relevant information about hotel accommodations and train options. Here are some areas of strength and suggestions for improvement:</p> <p>Strengths: 1. Comprehensive Responses: The system provided detailed information about hotel options, including amenities like free Wi-Fi and parking, as well as the necessary postcodes. 2. Train Options: ...</p> <p>Areas for Improvement: 1. Direct Booking Function: ... Incorporating a direct booking capability would enhance user experience by reducing the need for manual follow-up actions. 2. Inclusion of More Restaurant Options: ... 3. Clarification of Pricing: ...</p> <p>Overall Impression: The system performed well in addressing the user's needs, ... Implementing the above suggestions could further enhance its effectiveness and user satisfaction for future engagements.</p>

(a) An example of numerical feedback.

(b) An example of textual feedback.

Figure 2: Numerical feedback, used in OPRO (Yang et al., 2024) and GPO (Tang et al., 2024), requires a predefined evaluation function to generate the corresponding score. On the other hand, textual feedback can be generated by LLMs in a simulated environment or by human users.

4 EXPERIMENT SETTINGS

In this study, we focus on iterative meta-prompting by leveraging textual feedback from the environment. Our meta-prompting components are task-agnostic (Section 4.1). They are designed to optimise the prompt of the interactive LLM-based systems (Section 4.2). We conduct our experiments on two challenging human-machine interaction tasks, task-oriented dialogue and medical question-answering (Section 4.3). The environment includes simulated users, where the feedbacker is an LLM-based feedbacker or human expert (Section 4.4). More information on optimisation and evaluation can be found in Section 4.5.

Furthermore, to assess how different prompts affect the system’s performance, all LLMs in this study are prompted in a zero-shot in-context learning fashion², i.e. their prompts solely consist of task descriptions, without any examples or demonstrations included.

4.1 META-PROMPTING COMPONENTS

The initial prompt writer LLM_P in the initialisation phase is built with GPT-4o mini, a cost-efficient model and outperforming GPT-4 (OpenAI et al., 2024) on chat preferences in the LMSYS leaderboard (Zheng et al., 2023). In the interactive optimisation phase, the rewriter LLM_R is built with

²Following the definition by Brown et al. (2020), it is in-context learning since the task description is given to LLMs as context, but is also zero-shot because there is no demonstration.

GPT-4o mini or Gemini-1.5-flash (Team et al., 2024). Across different tasks, we keep the prompts of LLM_P and LLM_R fixed, highlighting the task-independent role of these components.

4.2 INTERACTIVE SYSTEM

Task-oriented dialogue The dialogue system is built with GPT-4o mini and prompted in two ways, standard and ReAct (Yao et al., 2023). The standard prompt instructs the system to generate responses directly for the user’s response (Figure 5a). The ReAct prompt instructs the system to generate its thoughts before responding (Figure 10a). The dialogue system takes the user history as a query and retrieves relevant entities based on Okapi BM25 (Robertson et al., 1995) implemented in langchain³.

Medical question-answering The system is built with GPT-4o mini and prompted only in the standard way. It responds to users’ questions through a single-turn or multiple-turn interaction. In addition, it operates without access to external knowledge bases, relying solely on its knowledge from pre-training for generating answers.

4.3 DATASET

Task-oriented dialogue We conduct our experiments on MultiWOZ 2.1 (Budzianowski et al., 2018; Eric et al., 2020), which includes 10k human-to-human conversations related to information-seeking, recommendation and reservation booking over multiple domains. We focus on the attraction, hotel, restaurant, and train domains, i.e., the *database* is composed of entities from these 4 domains and the *requirements* include user goals related to these domains.

Medical question-answering We conduct our experiments on two Chinese medicine datasets, Huatuo-26M (Li et al., 2023) and ShenNong-TCM (Wei Zhu & Wang, 2023). The medical questions in Huatuo-26M and ShenNong-TCM are collected from the internet, e.g. encyclopedias, books, literature, and web corpus, or generated by an LLM based on a traditional Chinese medicine entity graph in Huatuo-26M and ShenNong-TCM, respectively.

4.4 USER AND FEEDBACKER

Users are all built with GPT-4o mini for both tasks. The instructions are shown in Figure 8. The simulated users will act based on the user goals, which are descriptions in plain text, such as “*You are looking for a place to stay, the hotel should be in the cheap price range and in the city centre. You also need to find a restaurant nearby.*”, or “*我只有咳嗽這一個症狀，請幫我推薦中藥或者方劑。* (*I only have a cough as a symptom. Please recommend Chinese medicine or a prescription.*)”

While the feedbacker in task-oriented dialogue is built with GPT-4o mini, the feedbacker for the medical question-answering task is human experts, i.e., doctors on general medicine and traditional Chinese medicine. The instructions for the LLM feedbacker are shown in Figure 9.

4.5 OPTIMISATION AND EVALUATION

For both tasks, we start by collecting 100 interactions between the user and the system using the initial prompt and goals sampled from the training set. 10 interactions are then sampled randomly and fed to the feedbacker. The sampling is done due to the context length limitation of the LLM-based feedbacker, as well as to leverage the human expert feedbacker wisely. At each epoch, the rewriter generates 5 new prompts based on the previous prompt and feedbacker output. We collect a further 100 interactions for each prompt using user goals sampled from the training set, and the one with the highest score, evaluated by the human expert in medical question-answering or simulated users in task-oriented dialogue, is picked as the new prompt.

Task-oriented dialogue At the end of each turn in the interaction, the user model is instructed to check whether the task is completed, i.e. “*Is the user goal fulfilled? Please check the user goal and the dialogue history carefully. The suggestions from the system must fulfil the user goal, the booked*

³<https://github.com/langchain-ai/langchain>

information should be confirmed, and all requested information should be answered. Please answer ‘fulfilled’ or ‘not finished’.”. The conversation will stop when the goal is fulfilled, i.e. *Complete*, or be forced to quit by the user, i.e. *fail*. For the final evaluation, we collect 100 dialogues with user goals sampled from the test set.

Medical question-answering Different systems will interact with simulated patients. Three experts assess these interactions and give their preference based on three factors: safety, professionalism, and fluency. All experts evaluate 30 pairs of interactions on general medicine and 30 on traditional Chinese medicine, where user goals are randomly sampled from the test set.

5 RESULTS AND DISCUSSION

5.1 TEXTUAL FEEDBACK PROVIDES RICHER INFORMATION FOR OPTIMISATION

Table 3 shows the complete rates of systems optimised by different methods, i.e. without prompt optimisation, optimised with numerical feedback (GPO (Tang et al., 2024)), or our proposed method, optimised with textual feedback (TaP), interacting with an LLM-based user simulator for task-oriented dialogue.

When the rewriter is built with GPT-4o mini, our TaP optimisation pipeline improves the system with a standard prompt by 21% in absolute terms, and the system with a ReAct prompt by 9%. In contrast, the GPO optimisation improves the system with a standard prompt by only 13.1%, with no significant improvement for the system using the ReAct prompt. We observe consistent result on the experiments using Gemini-1.5-flash, as shown in Figure 3. In addition, Figure 3 shows that GPO is barely improved after the first epoch. On the other hand, TaP can still improve the system with more interaction, suggesting a more optimal performance.

These results show that our proposed method is able to improve the model on different prompting styles, standard or ReAct, without limitation between the choice of LLMs, GPT-4o mini or Gemini-1.5-flash. In addition, our proposed method also reduces the gap between the difference between prompting styles, i.e., the performance difference between the standard and ReAct prompt is reduced from 21.6% to 0.32% after optimised by our proposed method with a Gemini-1.5-flash rewrite for 5 epochs. This result suggests our method can provide a more stable, systematic, and scientifically grounded prompt optimisation.

Table 3: Complete rates on 100 MultiWOZ interactions with different optimisation methods. The system is instructed by a standard prompt, i.e. generating responses directly, or by a ReAct prompt (Yao et al., 2023). **F** represents the optimisation algorithm, i.e. without prompt optimisation (-), optimised with numerical feedback (GPO (Tang et al., 2024)), or our proposed method, optimised with textual feedback (TaP). **R** stands for which LLM used by rewriter, i.e. GPT-4o mini (GPT) or Gemini-1.5-flash (Gemini).

System	F	R	Complete rate	System	F	R	Complete rate
Standard	-	-	0.628 \pm 0.049	ReAct	-	-	0.844 \pm 0.037
Standard	GPO	GPT	0.759 \pm 0.043	ReAct	GPO	GPT	0.881 \pm 0.033
Standard	TaP	GPT	0.846 \pm 0.036	ReAct	TaP	GPT	0.934 \pm 0.025
Standard	GPO	Gemini	0.829 \pm 0.038	ReAct	GPO	Gemini	0.887 \pm 0.032
Standard	TaP	Gemini	0.974 \pm 0.016	ReAct	TaP	Gemini	0.942 \pm 0.024

5.2 WHY THE MODEL’S PERFORMANCE IS IMPROVED THROUGH TAP

Token counts Figure 4b shows that no matter which prompting style the system instruction is or which LLM is used for the rewriter, the number of tokens in the prompt is increasing through the optimisation. However, the complete rate is not monotonically increasing, as shown in Figure 4a. This result suggests that simply increasing the length of the prompt does not guarantee an improvement on performance, and that finding specific prompt tokens is important.

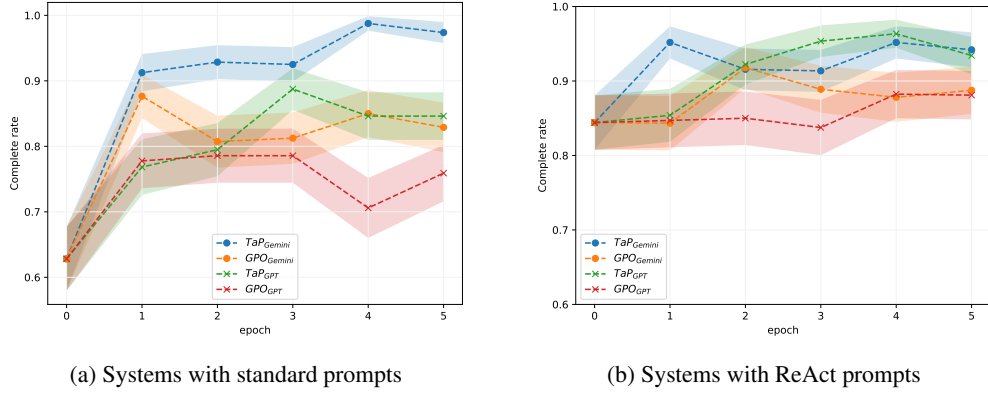


Figure 3: Result on the Task-oriented dialogue task during optimisation. Systems are updated by GPO and TaP where rewriters are built with different LLMs, i.e. GPT-4o-mini (GPT) or Gemini-1.5-flash (Gemini)

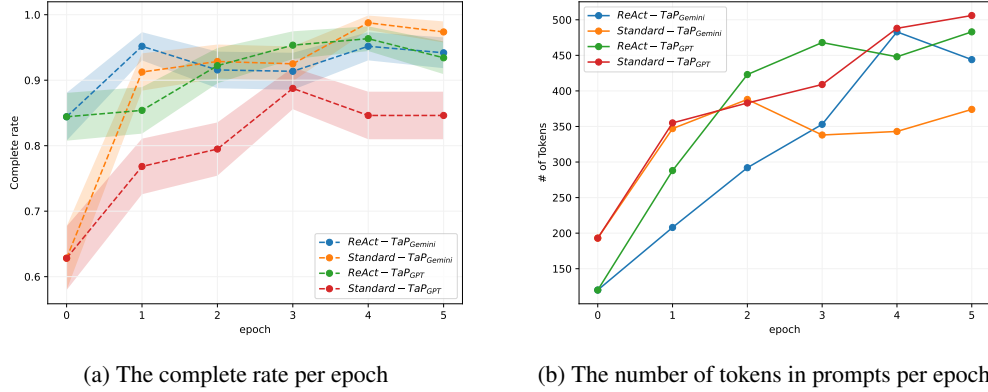


Figure 4: The rewriter tends to increase the number of tokens in the instructions, but the complete rate is not always improved.

Domain-specific information is learned through interaction The initial standard prompt (Figure 5a) only includes vague task definitions such as “*finding and booking things*”, and the initial ReAct prompt (Figure 10a) includes limited task-related information, such as “*book train ticket, hotel room, restaurant reservation, and attraction tickets*” and general booking related slots *dates* and *times*.

Figure 5 and Figure 10 show that the domain-specific information is learnt through interaction. After 5 epochs, more slots related to information seeking, recommendation, and reservation booking are included in the prompt automatically, e.g. *cuisine* and *dietary preferences* for the restaurant domain in Figure 5b, and *number of guests* and *check-in/check-out duration* for the hotel domain in Figure 10b. More booking-related slots, such as *contact information* and *reference numbers* are also included. It is worth mentioning that neither the simulated user, feedbacker nor rewriter have access to any d task-oriented dialogue information since the user goal, dialogues, and system prompts are all presented as plain texts.

According to Table 3, we can find out the system has better performance when more domain-specific information is included in the instruction. This observation is further supported by the fact that human experts also integrate domain-specific knowledge, such as available domains or slots within the context (Heck et al., 2023; Hudeček & Dusek, 2023). In other words, this result suggests our proposed optimisation method can generate a better prompt through interaction because more important domain-specific information is added to the instruction.

You are an intelligent assistant. Your goal is to assist the user in **finding** and **booking** things that fit their specific needs. You should be polite,...

1. Identify User's Needs: ...
2. Provide Options: ...
3. Booking Process:...
4. Confirmation:...
5. Polite Closure: ...

(a) The initial standard prompt.

You are an advanced intelligent assistant designed to help users **find** and **book** **accommodations**, **transportation**, and **dining** options tailored to their specific needs. ...

1. Identify User's Needs: ...
- Ask targeted questions about requirements, such as **parking**, **pet policies**, **accessibility** **needs**, **dietary** preferences, and **transportation** logistics ...
2. Provide Diverse Options:
- Search for and present a range of options that meet the user's criteria, ensuring diversity in **price** points, **amenities**, and types of **cuisine**. ...
3. Booking Process Guidance:
- Confirm booking details, including the **number of guests**, **duration of stay**, and **start date**. ...
4. Availability Check and Amenities:
- Provide the **availability status** of hotels, along with their **addresses**, **amenities**, and relevant **contact information**, to support informed decisions. ...

(b) The standard prompt updated by TaP after 5 epochs.

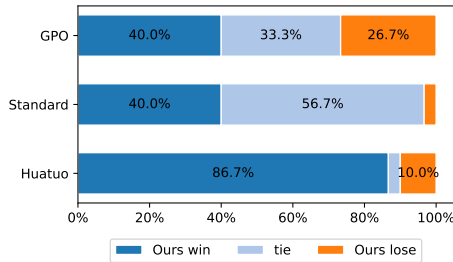
Figure 5: Examples of domain-specific information being learned through interactive prompt optimisation for standard prompts. For illustration, ontology terms, e.g. intent (orange), domain (blue), and slot (green), are manually highlighted based on the ontology in MultiWOZ.

5.3 THE LEARNABILITY OF PROMPTING

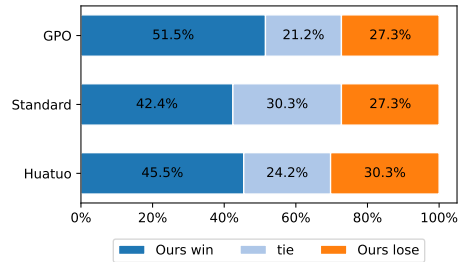
We compare our method against three systems: a standard system, built with GPT-4o mini with the initial prompt, a standard system updated via GPO, and HuatuoGPT-II (Chen et al., 2024), a large language model which is fully fine-tuned on medical data and demonstrates the state-of-the-art performance on Chinese medicine benchmarks. In other words, except HuatuoGPT-II, a fully fine-tuned 7B model, all systems are built with GPT-4o mini by prompting.

In general medicine, our method consistently outperforms the fully fine-tuned HuatuoGPT-II with an 86.7% win rate and is preferred over other prompting-based baselines. On the other hand, traditional Chinese medicine is more challenging. For example, our system’s preference rate drops by 41% compared to Huatuo when transitioning from general medicine to traditional Chinese medicine. However, despite this drop in preference, our proposed method is still favoured in general.

This observation is aligned with the findings by Petrov et al. (2024). Our method performs better in general medicine because the skills present in the pre-training data of LLMs can be elicited by prompting. However, tasks which are unseen or under-represented in pre-training data are hard to learn through prompting. How to properly leverage external knowledge to improve the performance on unseen or under-represented tasks is important in the future.



(a) Result on general medicine.



(b) Results on traditional Chinese medicine.

Figure 6: Overall preference between our method and a standard system (Standard), GPO, and HuatuoGPT-II (Huatuo) on the medical question-answering task. The overall recommendation by human experts is based on safety, professionalism and fluency.

6 CONCLUSIONS

Our proposed method treats text-based instructions for large language models (LLMs) as parameters. These textual prompts are initialised and optimised by LLMs, without the need for domain-specific manual prompt editing. As shown in Table 3, the method is robust to the choice of LLM used

for rewriting, as it works effectively with both GPT-4o mini and Gemini-1.5-flash. Additionally, it supports various prompting styles, such as standard and ReAct, and reduces the performance gap between these strategies after optimisation over several epochs. By using the optimised prompt, the system can minimise the need for extensive self-feedback loops, reducing computational overhead and API call frequency during inference.

Domain-specific information is learned and encoded in the prompt throughout the interaction with textual feedback (as shown in Figure 5 and Figure 10), enhancing both model performance and explainability. Furthermore, our method can incorporate feedback from simulated environments as well as human experts, which enables a more feasible way to incorporate human expert knowledge. Our method offers a stable, practical, and efficient approach for automatic prompt optimisation, which could be valuable for future LLM research.

ETHIC STATEMENT

This work uses open-source datasets, such as MultiWOZ, Huatuo-26M, and ShenNong-TCM. The MultiWOZ dataset is widely used in research on task-oriented dialogue. The Huatuo-26M dataset is collected from publicly accessible data without personal information and is available to academic researchers. The ShenNong-TCM dataset is generated by GPT-3.5 based on a traditional Chinese medicine knowledge graph. As a result, these datasets should not be regarded as controversial. All interactions are generated by LLMs, which may inevitably include hallucinations or in-correct information. Human evaluators are also fully aware they are reading interactions generated by LLMs.

REPRODUCIBILITY STATEMENT

The datasets used in this work are all open-sourced. The GPT-4o mini model is available through OpenAI’s platform <https://openai.com/api/>. The Gemini-1.5-flash model is available through Google Cloud Platform Vertex AI. In this work, we use the versions of the model with identifiers gpt-4o-mini-2024-07-18 and Gemini-1.5-flash-001.

REFERENCES

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5016–5026, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1547. URL <https://aclanthology.org/D18-1547>.
- Junying Chen, Xidong Wang, Ke Ji, Anningzhe Gao, Feng Jiang, Shunian Chen, Hongbo Zhang, Song Dingjie, Wenya Xie, Chuyi Kong, Jianquan Li, Xiang Wan, Haizhou Li, and Benyou Wang. HuatuoGPT-II, one-stage training for medical adaption of LLMs. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=eJ3cHNu7ss>.
- Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. Black-box prompt optimization: Aligning large language models without model training. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3201–3219, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.176. URL <https://aclanthology.org/2024.acl-long.176>.

- Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.247. URL <https://aclanthology.org/2023.findings-acl.247>.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In Nicoletta Calzolari, Frédéric B  chet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, H  l  ne Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 422–428, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL <https://aclanthology.org/2020.lrec-1.53>.
- John Hattie and Mark Gan. Instruction based on feedback. In *Handbook of research on learning and instruction*, pp. 263–285. Routledge, 2011.
- Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, and Milica Gasic. ChatGPT for zero-shot dialogue state tracking: A solution or an opportunity? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 936–950, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.81. URL <https://aclanthology.org/2023.acl-short.81>.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5254–5276, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.319. URL <https://aclanthology.org/2023.emnlp-main.319>.
- Vojt  ch Hude  ek and Ondrej Dusek. Are large language models all you need for task-oriented dialogue? In Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani (eds.), *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 216–228, Prague, Czechia, September 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.sigdial-1.21. URL <https://aclanthology.org/2023.sigdial-1.21>.
- Weize Kong, Spurthi Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. PRewrite: Prompt rewriting with reinforcement learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 594–601, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-short.54. URL <https://aclanthology.org/2024.acl-short.54>.
- Raymond W Kulhavy. Feedback in written instruction. *Review of educational research*, 47(2): 211–232, 1977.
- Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November

2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL <https://aclanthology.org/2021.emnlp-main.243>.
- Jianquan Li, Xidong Wang, Xiangbo Wu, Zhiyi Zhang, Xiaolong Xu, Jie Fu, Prayag Tiwari, Xiang Wan, and Benyou Wang. Huatuo-26m, a large-scale chinese medical qa dataset. *arXiv preprint arXiv:2305.01526*, 2023.
- Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL <https://aclanthology.org/2021.acl-long.353>.
- Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL <https://aclanthology.org/2022.acl-short.8>.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 2023. ISSN 2666-6510. doi: <https://doi.org/10.1016/j.aiopen.2023.08.012>. URL <https://www.sciencedirect.com/science/article/pii/S2666651023000141>.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.
- Roy Lyster and Hirohide Mori. Interactional feedback and instructional counterbalance. *Studies in second language acquisition*, 28(2):269–300, 2006.
- Edna H Mory. The use of informational feedback in instruction: Implications for future research. *Educational Technology Research and Development*, 40(3):5–20, 1992.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kopic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez,

- Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeesh Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorný, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.
- Aleksandar Petrov, Philip Torr, and Adel Bibi. When Do Prompting and Prefix-Tuning Work? A Theory of Capabilities and Limitations. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=JewzobRhay>.
- Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with “gradient descent” and beam search. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7957–7968, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.494. URL <https://aclanthology.org/2023.emnlp-main.494>.
- Guanghui Qin and Jason Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5203–5212, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.410. URL <https://aclanthology.org/2021.naacl-main.410>.
- Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at TREC-3. *Nist Special Publication Sp*, 109:109, 1995.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

- Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, Siyuan Lu, Yaliang Li, and Ji-Rong Wen. Unleashing the potential of large language models as prompt optimizers: An analogical analysis with gradient-based model optimizers. *arXiv preprint arXiv:2402.17564*, 2024.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqi, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vondrahal, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurusurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, Hyunjong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayanan Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit

Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihla, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Inuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodgkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiuja Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel

Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak
 Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su,
 Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong,
 Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi
 Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei
 Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C.
 Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven
 Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez,
 Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali
 Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen
 Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan,
 Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith
 Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan,
 Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard,
 Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson
 Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li,
 Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Di-
 ane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-
 Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen
 Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya
 Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hi-
 lal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li,
 Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin
 Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy
 Basu, Li Lao, Adnan Ozturk, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna
 Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Mi-
 lad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy,
 Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Vilella, Haibin Zhang,
 Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mah-
 moud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici,
 Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jo-
 vana Mitrovic, Alex Grills, Joseph Pagadora, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang,
 Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek
 Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu,
 Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias
 Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish
 Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa,
 Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre
 Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto,
 Solomon Kim, William Zeng, Ken Durden, Priya Ponnappalli, Tiberiu Sosea, Christopher A.
 Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam,
 Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn
 Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei
 Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Wooyeol Kim,
 Nandita Dukkhipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi,
 Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim
 Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dan-
 gyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer,
 Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy,
 Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan
 Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong
 Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li,
 Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Braman-
 dia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah,
 Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru,
 Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma
 Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf
 Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai,
 Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadisy,

- Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Pettrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kepa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Mery, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL <https://arxiv.org/abs/2403.05530>.
- Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. Two-stage LLM fine-tuning with less specialization and more generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=pCEgna6Qco>.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.
- Wenjing Yue Wei Zhu and Xiaoling Wang. Shennong-tcm: A traditional chinese medicine large language model. <https://github.com/michael-wzhu/ShenNong-TCM-LLM>, 2023.
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=Bb4VGOWELI>.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.
- Qinyuan Ye, Mohamed Ahmed, Reid Pryzant, and Fereshte Khani. Prompt engineering a prompt engineer. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 355–385, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.21. URL <https://aclanthology.org/2024.findings-acl.21>.
- Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=ByxRM0Ntvr>.
- Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. URL <https://openreview.net/forum?id=RJyfnSoyDC>.
- Danyang Zhang, Lu Chen, Situo Zhang, Hongshen Xu, Zihan Zhao, and Kai Yu. Large language models are semi-parametric reinforcement learning agents. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 78227–78239. Curran Associates, Inc.,

2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/f6b22ac37beb5da61efd4882082c9ecd-Paper-Conference.pdf.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. Agent-Pro: Learning to Evolve via Policy-Level Reflection and Optimization. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. URL <https://openreview.net/forum?id=Ryjxb6f1l9>.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena, 2023.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=92gvk82DE->.

A PROMPTS FOR LLMs IN TAP

Here are the prompts for components in our proposed method.

You are an assistant tasked with improving the prompt instruction of another large language model assistant. You will be given the previous instruction prompt and its feedback. Please generate a new instruction prompt for the next iteration, **with performance improvement**. Please output the new instruction prompt directly without any extra description, since the result would be fed back into the assistant directly. The new prompt should not be longer than 512 tokens.

Here is the previous instruction [OLD PROMPT] and the feedback [FEEDBACK]

Figure 7: The prompt for the rewriter.

You are a user and try to use a task-oriented dialogue system to accomplish your goal. Here are the guidelines for the conversation:

- 1. Provide Information:**
Answer the assistant's questions clearly and provide all necessary information.
- 2. Ask for Clarification:**
If you are unsure about something, ask the assistant to clarify.
- 3. Confirm Booking:**
Check the booking details provided by the assistant and confirm if they are correct and align with the user goal.
- 4. End the Conversation:**
Say 'goodbye' after the system fulfils your user goal, e.g. providing all information in the user goal, answering all your requests, or making a reservation correctly.

If you are unsatisfied, e.g. the system keeps providing recommendations which do not fulfil your goal or the system is not helpful, you should say 'quit' at any time.

Here is the user goal: [USER GOAL]

Figure 8: The prompt for user simulators in task-oriented dialogue systems

Based on the user goal and the dialog history, please provide feedback to the system. The feedback should be constructive and helpful for the system to improve.

Here are the user goals [USER GOALS] and the dialogs [DIALOG]

Figure 9: The prompt for feedback in task-oriented dialogue systems

You are an intelligent assistant tasked with helping users **book train tickets, hotel rooms, restaurant reservations, and attraction tickets**, as well as providing information ...

1. Break down the user's request into specific subtasks.
2. Reason through each step before taking any action.
3. Ask the user for necessary information to complete the task, such as **dates, times**, preferences, or other relevant details.
4. Provide available options ...
5. Respond clearly, concisely, and politely.

Please generate your reasoning first, starting with [Think], and then your response, starting with [Response].

(a) The initial ReAct prompt.

You are an intelligent assistant designed to help users **book train tickets, hotel rooms, restaurant reservations, and attraction entries** ...

1. Strictly Follow User Specifications: ...
2. Decompose Requests: ...
3. Clarify User Preferences: Actively prompt users to share detailed preferences, such as **cuisine, ambience**, and specific needs, to provide personalized recommendations.
4. Gather Comprehensive Information: Collect all necessary details for bookings, including **travel dates, number of guests**, and specific requirements. Suggest standard options like typical **check-in/check-out durations** to facilitate quicker decisions.
5. Present Varied Choices: Offer a range of relevant options, summarizing key details like **pricing and location**.
7. Deliver Detailed Confirmations: Provide thorough confirmations for all bookings, including **dates, times, reference numbers**.

Please generate your reasoning first, starting with [Think], and then your response, starting with [Response].

(b) The ReAct prompt updated by TaP after 5 epochs.

Figure 10: The domain-specific information is learned through the interactive prompt optimisation for ReAct prompts. The ontology terms, e.g. intent (orange), domain (blue), and slot (green), are manually highlighted based on the ontology in MultiWOZ.