Journal of Computer Science & Technology

Indexed in: SCIE, EI, INSPEC, CBST, DBLP, etc.

Sponsored by: ICT, CAS & CCF

JCST Papers

Only for academic and non-commercial use

Thanks for reading!

JCST WeChat Subscription Account

JCST URL: https://jcst.ict.ac.cn

SPRINGER URL: https://www.springer.com/journal/11390

E-mail: jcst@ict.ac.cn

Online Submission: https://mc03.manuscriptcentral.com/jcst

Twitter: JCST_Journal

LinkedIn: Journal of Computer Science and Technology

Song YF, He YQ, Zhao XF *et al.* A communication theory perspective on prompting engineering methods for large language models. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(4): 984-1004 July 2024. DOI: 10.1007/s11390-024-4058-8

A Communication Theory Perspective on Prompting Engineering Methods for Large Language Models

Yuan-Feng Song (宋元峰), Yuan-Qin He (何元钦), Xue-Fang Zhao (赵雪芳), Han-Lin Gu (古瀚林)
Di Jiang (姜　迪), Hai-Jun Yang (杨海军), and Li-Xin Fan (范力欣)

AI Group, WeBank Co., Ltd, Shenzhen 518000, China

E-mail: yfsong@webank.com; yuanqinhe@webank.com; summerzhao@webank.com; allengu@webank.com
dijiang@webank.com; navyyang@webank.com; lixinfan@webank.com

Received December 21, 2023; accepted April 12, 2024.

Abstract The springing up of large language models (LLMs) has shifted the community from single-task-orientated natural language processing (NLP) research to a holistic end-to-end multi-task learning paradigm. Along this line of research endeavors in the area, LLM-based prompting methods have attracted much attention, partially due to the technological advantages brought by prompt engineering (PE) as well as the underlying NLP principles disclosed by various prompting methods. Traditional supervised learning usually requires training a model based on labeled data and then making predictions. In contrast, PE methods directly use the powerful capabilities of existing LLMs (e.g., GPT-3 and GPT-4) via composing appropriate prompts, especially under few-shot or zero-shot scenarios. Facing the abundance of studies related to the prompting and the ever-evolving nature of this field, this article aims to 1) illustrate a novel perspective to review existing PE methods within the well-established communication theory framework, 2) facilitate a better/deeper understanding of developing trends of existing PE methods used in three typical tasks, and 3) shed light on promising research directions for future PE methods.

Keywords prompting method, large language model, communication theory

1 Introduction

Large language models (LLMs) (e.g., GPT-3[1], GPT-4[2], LLaMa[3]) make it possible for machines to understand users' attention accurately, thus revolutionizing the human-computer interaction (HCI) paradigm. Compared with traditional machine systems like databases and search engines, LLMs demonstrate impressive capability in understanding, generating, and processing natural language, facilitating a series of services ranging from personal assistants[4], healthcare[5] to e-commercial tools[6] via a unified natural language interface between users and machines.

The research paradigm around LLM has shifted from single-task-orientated natural language processing (NLP) research to a holistic end-to-end multi-task learning approach. Along this line of research endeavors, LLM-based prompting engineering (PE) methods[1, 7] have attracted much attention, partially because they are the key techniques in making full use of the superior capabilities of LLMs via constructing appropriate prompts. PE refers to the process of carefully constructing instructional prompts to steer and shape the behavior of LLMs, and it greatly helps in bridging the gap between the pre-training tasks used to construct the LLM with the down-streaming tasks queried by the end users. Through careful prompt designing, users can steer LLM's output in the desired direction, shaping its style, tone, and content to align with their goals.

To this end, numerous prompt engineering (PE) methods have been explored with the notable progress of LLM advancement and technologies[7–24]. A common theme of PE development lies in continuously improving accuracy and responsiveness of designed prompts, which often include components like Role, Context, Input, Output Format, and Examples. Specifically, prompt template and answering engineer-

Survey

ing have evolved from solely utilizing discrete prompts to continuous prompts, and even to exploring hybrid prompts that combine continuous and discrete elements, which provides a larger optimization space to achieve better performance. With the emerging capabilities of LLMs, these models can leverage their in-context learning abilities to plan and utilize external tools, significantly enhancing their performance in specialized domains and broadening their applications across diverse fields.

Following these studies, representative PE methods can be categorized as three groups that correspond to three prompting tasks proposed to improve the qualities of LLMs' outputs, namely prompt template engineering, prompt answer engineering, and multi-prompt engineering and multi-turn prompt engineering, respectively. An example of the input and output for the above-mentioned tasks can be found in Table 1.

• First, prompt template engineering methods aim to carefully design a piece of "text" that guides the language models to produce the desired outputs. For example, in Table 1, to finish a classical sentiment detection for an input A="Delicious dining options close to my current location", the prompt template engineering designs a template "$[A]$ In summary, it was a $[Z]$ restaurant" to enforce the LLM to fill the desired comments in the blank (i.e., $[Z]$). Essentially this type of template engineering method induces LLM to focus on word embeddings that are relevant to the questions. A common designing principle of existing prompt template engineering methods is to better align information between users and LLMs. Such a trend is manifested by the evolution from using discrete prompts (i.e., a piece of human-readable text)[9, 11] to continuous ones (i.e., a continuous task-specific vector)[13, 20].

• Second, prompt answer engineering [7] refers to the process of exploring the vast answer space and a map to the desired, intended output, which enhances users' understanding of the information encapsulated within the LLM. For the same example in Table 1, the prompt answer engineering aims to find a mapping from the result "good" obtained from the LLM to the desired answer "positive". The field of prompt answer engineering is currently witnessing a notable development trend characterized by the pursuit of models that excel in decoding model information from simple mapping to complex mapping to enhance human comprehension.

• Third, multi-prompting methods mainly apply ensemble techniques[10] to mitigate the sensitivity of LLM to different formulations and to obtain a more stable output. In Table 1, the multi-prompting methods combine three different templates (i.e., 1) "It was a $[Z]$" place, 2) "A $[Z]$ place to eat", and 3) "In general, it was $[Z]$"), and their inference results (i.e., 1) "good", 2) "fantastic", and 3) "okay") to obtain the final desired one (i.e., "positive"). Later, as LLMs become more capable, multi-turn prompt methods attract more attention that aims to provide more context to LLM by leveraging information either from LLM itself or external tools[25, 26]. In the field of multi-prompting methods, researchers are endeavoring to develop adaptive strategies that enhance LLM's ability to task planning and the utilization of tools.

In this article, we summarize the prompting methods from a communication theory perspective with which the ultimate goal of PE is to reduce the information misunderstanding between the users and the LLMs. Therefore, as delineated in Section 2, the communication theory perspective provides a coherent explanation of different PE methods in terms of their objectives and underlying principles. Moreover, this novel perspective also offers and presents insights into scenarios where existing prompting methods come short.

The remainder of the article is structured as follows: Section 2 details the overview of the prompting methods from the communication theory perspective. Sections 3, 4, and 5 review and summarize the recent

Table 1. Running Examples for PE Methods

Stage	Input	Output
Prompt template engineering	Delicious dining options close to my current location	Delicious dining options close to my current location. In summary, it was a $[Z]$ restaurant
Large language model	Delicious dining options close to my current location. In summary, it was a $[Z]$ restaurant.	Delicious dining options close to my current location. In summary, it was a good restaurant
Prompt answering engineering	Good	Positive
Multi-prompt	1) It was a $[Z]$ place; 2) A $[Z]$ place to eat; 3) In general, it was $[Z]$	1) good, 2) fantastic, and 3) okay

progresses from three PE tasks namely prompt template engineering, prompt answer engineering, and multi-prompt engineering and multi-turn prompt engineering, respectively. Sections 6 discusses other related surveys and potential research directions. Finally, we conclude this article in Section 7 by summarizing significant findings and discussing potential research directions. We summarize the main symbols and abbreviations in Table 2 for the convenience of readers.

2 Communication Theory Perspective of Prompting Methods

The study of modern communication theory, which dates back to the 1940s and the following decades, gave rise to a variety of communication models including both linear transmission models and non-linear models such as interaction, transaction, and convergence models[27–29]. A common theme of these early studies is to analyze how individuals utilize verbal and non-verbal interactions to develop meaning in diverse circumstances. Conceptually, the communication process is often modeled as a chain of information processing steps involving encoding, transmitting, and decoding of messages, between a sender and a receiver.

To give a better illustration, Fig.1(a) depicts the classical Model of Communication in the communication theory, which includes a sender encoding a message and transmitting it to the receiver over a channel. Then, the receiver decodes the message and delivers some type of response. During the transmission process, the message may be distorted due to noise, leading to the necessity of multi-turn interaction.

The original communication theory is widely utilized to examine factors including social[30], cultural[31], and psychological[32] that influence human communication. The overall goal of communication theory is to reveal and clarify the common human experience of interacting with others through information exchange.

Among early studies of various communication models, we are particularly inspired by two influential works, namely, Shannon-Weaver Model of Communication[33] and Schramm Communication Model[34]. Shannon-Weaver's pioneering work, first pub-

Table 2. Summary of Key Symbols and Abbreviations

Symbol	Description
PES	Prompt engineering system, a mathematical formulation for interactive user-LLM communication
X	Input to the LLM, which can be text or other data
P_T	Prompt template, a carefully crafted piece of text designed to guide the LLM to produce desired outputs
P_A	Prompt answer, the output yielded by the LLM following the input P_T
Y	Target output or desired result from the LLM
g_{ω_T}	Function representing the mapping from the input X to P_T
f_θ	Function representing the mapping from the prompt P_T to the answer P_A
h_{ω_A}	Function representing the mapping from the answer P_A to Y
$I(X;Y)$	Mutual information between two random variables X and Y, used in the context of maximizing information flow in PES

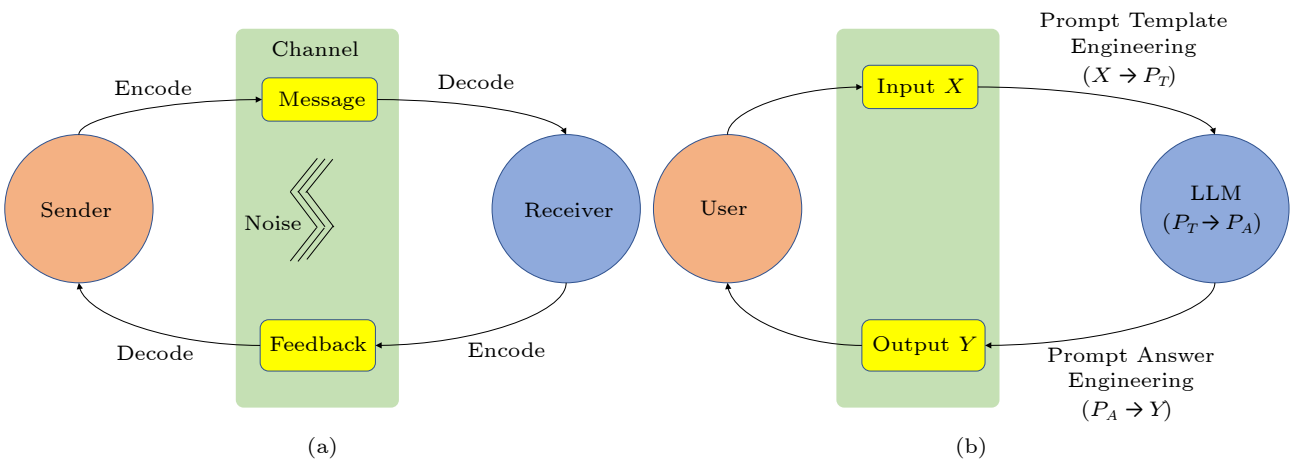


Fig.1. Prompting methods from the communication theory perspective. (a) Classical interaction model of communication. (b) Different aspects of existing prompting methods.

lished in 1948, provides a strong mathematical foundation to analyze information flow between an active sender and a passive receiver. It is however over-simplistic in the sense that it does not take into account of complexities involved in interactive communication between active senders and receivers, who may respond by sending their messages as a form of feedback. The interaction models of communication were first studied by Scharmm and published in his 1954 book[34], which pictorially illustrates the feedback loop as depicted in Fig.1(a). Nevertheless, Scharmm's model falls short of rigorous theoretical and mathematical formulation to accommodate quantitative analysis, e.g., information gain or mutual information between senders and receivers.

Various prompting engineering methods for LLM, in our view, can be understood from Scharmm's model point of view (see Fig.1(b)). In the same vein of Shannon-Weaver's analysis, we, therefore, delineate a mathematical formulation of prompting engineering systems for interactive user-LLM communication as follows.

Definition 1 (PES). *A prompt engineering system (PES) consists of a processing chain:*

$$X \xrightarrow{g_{\omega_T}} P_T \xrightarrow{f_\theta} P_A \xrightarrow{h_{\omega_A}} Y,$$

where g_{ω_T} represents the mapping from the input X to the prompt P_T, f_θ denotes the mapping from the prompt P_T to the answer P_A, and h_{ω_A} denotes the mapping from the answer P_A to the output Y (see Fig.1(b) for an illustration).

Definition 2 (Goal of PES). *PES aims to maximize the mutual information between the inputs X and outputs Y, i.e.,*

$$\max_{\omega_T,\,\omega_A} I(X,Y) = \max_{\omega_T,\,\omega_A} I(X, h_{\omega_A} \circ f_\theta \circ g_{\omega_T}(X)), \quad (1)$$

where $f \circ g(x) = f(g(x))$.

It is worth noting that prompt engineering is consistently divided into two procedures: prompt template engineering and prompt answer engineering. Each procedure has specific goals similar to (1) that align with its intended purpose.

While the capacity in Definition 2 is well-known in information theory[35], how to reach the maximum of (1) for LLMs illustrated in Fig.1(b) remains an unexplored research direction. There exists a large variety of prompting engineering methods, which, in our view, essentially aim to reduce information misunderstanding between users and LLMs. In other words, they aim to reach the capacity of PES as defined. For instance, Sorensen *et al.*[36] demonstrated selecting the prompt with the greater mutual information (MI) enhanced the model performance. This underscores the objective of PE techniques, which is to optimize the mutual information between prompts and answers (see practical examples in Appendix C of [36]). This connection between PES and the communication models has never been explicitly stated before.

Moreover, the existing work can be divided into three categories: prompt template engineering ($X \xrightarrow{g_{\omega_T}} P_T$), prompt answer engineering ($P_A \xrightarrow{h_{\omega_A}} Y$), and multi-prompt engineering and multi-turn prompt engineering as shown in Fig.1(b). Specifically, the prompt template engineering aims to reduce the encoding error or look for the prompt that is easily understood by the machine, while the prompt answering engineering aims to reduce the decoding error or look for the prompt that can be easily understood by the human. The development of LLMs aims to enhance the capability of the receiver that could better handle users' information needs, and most importantly, the multi-turn prompting and multi-prompt engineering aim to constantly reduce the information misunderstanding via multi-turn interactions.

- Prompt template engineering aims to optimize

$$\max_{\omega_T} I(X, P_A) = \max_{\omega_T} I(X, f_\theta \circ g_{\omega_T}(X)), \quad (2)$$

which looks for an additional piece of text, namely a prompt, to steer the LLMs to produce the desired outputs for downstream tasks. From the communication theory perspective, it acts as an "encoder" to bridge the gap between the users and the LLMs by encoding the messages in a way that the model can understand and then elicit knowledge from LLMs (see details in Section 3). In the encoding process, the challenge lies in the accurate understanding of the user's intention by LLM with limited instruction following capability. Template engineering aims to reduce this mismatch by translating the user's request to a format that could be better understood by LLM.

- Prompt answer engineering aims to optimize

$$\max_{\omega_A} I(P_T, Y) = \max_{\omega_A} I(P_T, h_{\omega_A} \circ f_\theta(P_T)), \quad (3)$$

which focuses on developing appropriate inputs for prompting methods. It has two goals: 1) to search for a prompt answer P_A and 2) to look for a map to the target output Y that will result in an accurate predictive model. In the decoding process, LLM-generated output often carries redundant information in addi-

tion to the expected answer due to its unlimited output space. Answer engineering aims to confine the output space and extract the target answer. The field of prompt answer engineering is currently witnessing a notable development trend characterized by the pursuit of effective answer engineering such that ultimate outputs (i.e., Y) are well aligned with that of end users' expectations (see details in Section 4).

• To further reduce the information misunderstanding, the user could conduct multi-interaction according to (2) and (3), called multi-prompt/multi-turn PE. Multi-prompting methods aim to optimize

$$\max_{\omega_{T_1},\ldots,\omega_{T_M}} \sum_{i=1}^{M} I(X, f_\theta \circ g_{\omega_{T_i}}(X)),$$

which mainly applies ensemble techniques[10] to mitigate the sensitivity of LLM to different formulations and to obtain a more stable output. Later, as LLMs become more capable, multi-turn prompt methods focus on providing more context to LLM by leveraging multiple communication procedures between the machine and person[25, 26]. In the field of multi-prompting methods, researchers are endeavoring to develop adaptive strategies that enhance LLM's ability to task planning and the utilization of tools. The adaptive and iterative nature of multi-prompting methods is by the communication theory (see Section 5 for an elaborated explanation).

3 Prompt Template Engineering

Given the information chain $X \to P_T \to P_A$, the answer P_A is determined by the prompt-processed P_T and model M with pre-trained weights θ. Suppose that $\bar{P}_A$ is the targeted prediction, the key problem of prompt template engineering is to find a good prompt that maximizes the probability $p(\bar{P}_A|M, P_T, \theta)$ on diverse downstream tasks with limited data. To obtain the optimal prompt, current work[8–24] can be formulated into three categories: constructing P_T, ranking P_T, and tuning P_T.

3.1 Constructing the Prompt

The basic motivation of constructing P_T is to transform the specific task to make it align with the pre-training objective (i.e., next-word prediction, masked LM) of the LM. Existing prompt construction methods[8–11, 15, 37–39] could be categorized into five different approaches as shown in Table 3, which are discussed in detail as follows.

3.1.1 *Manually-Designed*

Initially, the prompt templates are manually designed in the natural language based on the user's experience, and they have been validated to be able to improve the performances of downstream tasks, especially in a zero-shot setting[1, 8]. The most frequent style is to reformulate the original task as a "fill-in-the-blank" cloze one[9, 10], and the answer is obtained by predicting the words in the given "[mask]" place. For example, as illustrated in Table 3, Petroni *et al.*[9] manually designed prompts to re-structure the relational knowledge, while studies like [10, 37] focused on solving the text classification and language understanding tasks by several self-defining prompt patterns and proposed[10] a new training procedure named PET. Another line of work involves developing prefix prompts for generation tasks, which provide instructions and steer the LLMs to finish the sentence. For example, a summarization task can be handled by adding "TL;DR:"[8], and a translation task can be conducted into "English Translate to Spanish:"[38]. Even though manually designed prompts show some effectiveness[39], they are also criticized for being time-consuming and unstable[15]. A subtle difference in the designed prompts may result in a substantial performance decrease. As such, how to explore the prompt space and construct prompts more thoroughly and more effectively becomes an important and challenging issue.

3.1.2 *Heuristic-Based*

The heuristic-based methods focus on finding

Table 3. Summary of Prompt Construction Methods

Method	Automated	Gradient-Free	Few-Shot	Zero-Shot	Stability	Interpret-Ability
Manually-designed[8–10]	✗	✓	✓	✓	✗	✓
Heuristic-based[11, 19, 40]	✓	✓	✓	✓	✓	✓
Paraphrasing-based[11, 14, 41]	✓	✓	✓	✓	✗	✓
Generation-based[17, 42]	✓	✓	✓	✓	✓	✓
Optimization-based[12, 22]	✓	✗	✓	✗	✓	✗

prompts by some intuitive strategies. For example, to construct more flexible and diverse prompts for different examples (rather than fixed ones), Jiang *et al.*[11] proposed to use the most frequent middle words and the phrase spanning in the shortest dependency path that appeared in the training data as a prompt. This method shows a large performance gain compared with the manually-designed prompts. Han *et al.*[19] tried to form task-specific prompts by combining simple human-picked sub-prompts according to some logic rules. Different from the above methods, Logan *et al.*[40] used an extremely simple uniform rule by null prompts, which only concatenates the inputs and the "[mask]" token, and it is able to gain a comparable accuracy with manually-defined prompts.

3.1.3 Paraphrasing-Based

The paraphrasing-based methods are widely used in data augmentation, aiming at generating augmented data that is semantically related to the original text, and this could be achieved in various ways using machine translation, model-based generation, and rule-based generation[43]. The paraphrasing-based methods could naturally be used to construct prompt candidates based on the original text, and we could further select the best one or integrate them to provide better performance. Representative studies includes [11, 14, 41]. Specifically, Jiang *et al.*[11] used back-translation to enhance the lexical diversity while keeping the semantic meaning. Yuan *et al.*[41] manually created some seeds and found their synonyms to narrow down the search space. Haviv *et al.*[14] used a BERT-based model to act as a rewriter to obtain prompts that LLMs can understand better.

3.1.4 Generation-Based

The generation-based methods treat prompt searching as a generative task that can be carried out by some LMs. For example, Gao *et al.*[17] first leveraged the generative ability of T5[38] to fill in the placeholders as prompts, and then the prompts could be further improved by encoding domain-specific information[42].

3.1.5 Optimization-Based

To alleviate the weakness of insufficient exploration space faced by existing methods, the optimized-based methods try to generate prompts guided by some optimization signals. For example, Shin *et al.*[12] employed gradients as the signals, and then searched for discrete trigger words as prompts to enrich the candidate space. Deng *et al.*[22] generated the prompt using a reinforced-learning approach that is directed with the reward function.

3.2 Ranking the Prompt

After obtaining multiple prompt candidates with the above-mentioned methods, the next step is to rank them to select the most effective one. Existing studies solve this problem by finding prompts that are close to the training samples to reduce the information mismatch between the pre-training and inference phases.

3.2.1 Execution Accuracy

Since the prompts are designed to accomplish specific downstream tasks, it is intuitive and straightforward to evaluate their performance by measuring the execution accuracy on those tasks[11, 17, 44].

3.2.2 Log Probability

The log probability criterion prefers the prompt that delivers the correct output with higher probability, rather than being forced to give the exact answer. For example, a prompt template that can work well for all training examples is given the maximum generated probability in [17]. Furthermore, language models can also be utilized to evaluate the quality of prompts. In [45], the prompt with the highest probability given by an LM is selected, which indicates closer to the general expression that appears in the training dataset.

3.2.3 Others

Other criteria can be used to select the top one or the top-k prompt. For example, Shin *et al.*[12] regarded the words that are estimated to have the largest performance improvement as the most crucial elements.

3.3 Tuning the Prompt

Recent studies turn to optimizing the prompt as continuous embeddings to further improve the performance. The main idea is to learn a few continuous pa-

rameters, referred to as soft prompts, and these continuous parameters can be optionally initialized by the previously obtained discrete prompt. Li *et al.*[13] first introduced a continuous task-specific "prefix-tuning" for generative tasks. Studies like [20] and [15] adopted a similar strategy and proved its effectiveness in various natural language understanding tasks. Following the above-mentioned studies, many improvements have been conducted to find better prompts, such as better optimizing strategies[16], better vector initialization[21, 23], and indicative anchors[15]. Furthermore, studies like [13, 20, 46] further point out that prompt position, length, and initialization all affect the performance of continuous prompts[13, 20, 46] (Table 4). In this subsection, we summarize these factors as follows:

• *Different Positions*. There are three different positions for autoregressive LM that the prompt can be inserted into, that is, the prefix $[PREFIX; X_T; Y]$, the infix $[X_T; INFIX; Y]$, and the hybrid one $[PREFIX; X_T; INFIX; Y]$. There is no significant performance difference between those positions. Li *et al.*[13] showed that prefix prompt sightly outperforms infix prompt, and the hybrid one is much more flexible than the others.

• *Different Lengths*. There is no optimal length for all tasks, but there is always a threshold. The performance will increase before reaching the threshold, then it will either plateau or slightly decrease.

• *Different Initializations*. A proper initialization is essential for the performance of the prompts and the performance of random initialization is usually unsatisfactory. Typical methods include initialized by sampling real words[13, 20], using class labels[20], using discrete prompts[16], and using pre-trained based vectors[21, 23]. Furthermore, the manually designed prompts serve as a good starting point for the following search process.

Besides the above-mentioned methods, PE methods have also been used for tuning and constructing the LLMs. Typical methods in this area include BitFit[47], Partial-k tuning, MLP-k tuning, side-tuning[48], adapter tuning[49], Ladder Side-Tuning[50], and the essential Prompt Tuning[13]. These methods aim to achieve a comparable performance by fine-tuning the whole network by only tuning some parts of the parameters of LLMs.

3.4 Trends for Prompt Template Engineering

There are two trends in prompt template engineering.

• Increased reliance on automated methods over manual design when constructing prompts, reducing the need for human involvement.

• Development of optimization-based techniques. The gradient-based searching method shows better performance than the derivative-free one in hard prompts construction while the soft prompts appear more promising than hard prompts.

From the communication theory perspective, the development history of prompting template engineering reflects the trends of utilizing prompts with stronger expressive ability to better capture the user's intent.

4 Prompt Answering Engineering

As depicted in Fig.1(b), prompt answer engineering (PAE) aims to align LLMs outputs with the intended purpose. The use of PAE is motivated by the need to mitigate the gap between the capabilities of pre-trained LLMs and a large variety of requirements of different downstream tasks (see more discussion in Section 2). Technology-wise, PAE involves a set of methods that control the admissible answer space and optimization mechanisms of LLMs' output (see overview in Table 5).

4.1 Search for an Answer Space

4.1.1 *Pre-Defined Answer Space*

This involves a set of pre-defined answers for the

Table 4. Summary of Prompt Tuning Methods

Work	Position	Length	Initialization
Prefix tuning[13]	Prefix, infix	200 (summarization), 10 (table-to-text)	Random, real words
Prompt tuning[20]	Prefix	1, 5, 20, 100, 150	Random, sampled vocabulary, class label
P-tuning[15]	Hybrid	3 (prefix), 3 (infix)	LSTM-trained
DART[18]	Infix	3	Unused token in vocabulary
OPTIPROMPT[16]	Infix	5, 10	Manual prompt
Dynamic[46]	Hybrid, dynamic	Dynamic	Sampled vocabulary

Table 5. Summary for Prompt Answer Engineering Methods

Answer Space Type	Answer Mapping Method	Work	Task Type
Optimizing the mapping	Discrete answer space	[10, 12, 17, 51]	Classification & regression
	Continuous answer space	[52]	Classification
Broadening the output	Discrete answer space	[11]	Generation
Decomposing the output	Discrete answer space	[53]	Classification
Manually mapping	Pre-defined answer	[9, 54, 55]	Generation

question-answering task, e.g., pre-defined emotions ("happiness", "surprise", "shame", "anger", etc.) for the sentiment classification task. The model can then be trained to select the best answer from this pre-defined space. As an illustration, the answer space P_A can be defined as the set of all tokens[9], fixed-length spans[56], or token sequences[8]. Furthermore, in certain tasks like text classification, question answering, or entity recognition, answers are crafted manually as word lists that pertain to relevant topics[7, 54, 55].

4.1.2 *Discrete Answer Space*

The discrete answer space refers to a set of specific and distinct answer options that a language model can choose from when generating a response to a given prompt.

Specifically, the possible answers are limited to a fixed set of choices, such as a small number of named entities or keyphrases (e.g., the total choice of the planet in the solar system is eight). The model can then be trained to identify whether the correct answer is among this set of possibilities[10–12].

4.1.3 *Continuous Answer Space*

The continuous answer space refers to a scenario where the possible answers or responses are not restricted to a predefined set of discrete options. Instead, the answers can take on a range of continuous values or be any text, number, or value within a broader, unbounded spectrum[52, 57].

The model can then be trained to predict a point in the continuous space that corresponds to the correct answer.

4.1.4 *Hybrid Approach*

This involves combining multiple methods to design the answer space, such as using a pre-defined list of entities for certain types of questions, but allowing for free-form text answers for other types of questions[58].

Remark 1. *Answer shapes summarized as follows are also needed in prompt answer engineering. In practice, the choice of the answer shape depends on the desired outcome of the task.*

- *Tokens: individual tokens within the vocabulary of a pre-trained language model, or a subset of the vocabulary.*
- *Span: short sequences of multiple tokens, often comprising a phrase or segment of text.*
- *Sentence: a longer segment of text that can encompass one or more complete sentences.*

4.2 Search for an Answer Mapping

There are several strategies to search for an answer mapping.

4.2.1 *Manually Mapping*

In many cases, the mapping from potential answers space P_A to output Y is obvious such that this mapping can be done manually. For instance, the answer is output itself for the translation task[9] such that the mapping is identity mapping. Additionally, Yin *et al.*[54] designed related topics ("health", "food", "finance", "sports", etc.), situations ("shelter", "water", "medical assistance", etc.), or other possible labels. Cui *et al.*[55] manually proposed some entity tags, e.g., "organization", "person", and "location", for the named entity recognition problem.

4.2.2 *Broadening Answer P_A*

Broadening P_A ($P'_A = B(P_A)$) is expanding the answer space to obtain a more accurate mapping. Jiang *et al.*[11] proposed a method to paraphrase the answer space P_A by transferring the original prompt into other similar expressions. In their approach, they employed a back-translation technique by first translating prompts into another language and then translating them back, resulting in a set of diverse paraphrased answers. The probability of the final output

can be expressed as $P(Y|x) = \sum_{y \in B(P_A)} P(y|x)$, where $B(Y)$ represents the set of possible paraphrased answers.

4.2.3 *Decomposing the Output*

Decomposing Y ($\mathrm{D}(Y)$) aims to expand the information of Y, which makes it easier to look for a mapping g_θ. For example, Chen *et al.*[53] decomposed the labels into several words and regarded them as the answer. Concretely, they decomposed label/output "per:city_of_death" into three separated words {person, city, death}. The probability of final output can be written as $P(y|x) = \sum_{y \in D(Y)} P(y|x)$.

4.2.4 *Optimizing the Mapping*

There exist two approaches to optimizing the mapping function. The first approach is to generate the pruned space $\tilde{P}_A$ and search for a set of answers within this pruned space. Schick *et al.*[10, 51] introduced a technique for generating a mapping from each label to a singular token that represents its semantic meaning. This mapping, referred to as a verbalizer v, is designed to identify sets of answers. Their approach involves estimating a verbalizer v by maximizing the likelihood w.r.t. the training data conditioned on the verbalizer v. Shin *et al.*[12] proposed an alternative approach for selecting the answer tokens. They employed logistic classifiers to identify the top-k tokens that yield the highest probability score, which together form the selected answer. In addition, Gao *et al.*[17] constructed a pruned set $\tilde{P}_A^c$ containing the top-k vocabulary words based on their conditional likelihood for each class c. As for the second approach, it investigates the potential of utilizing soft answer tokens that can be optimized through gradient descent. Hambardzumyan *et al.*[52] allocated a virtual token to represent each class label and optimized the token embedding for each class along with the prompt token embedding using gradient descent.

4.3 Trends for Prompt Answer Engineering

There are two trends in prompt answer engineering:

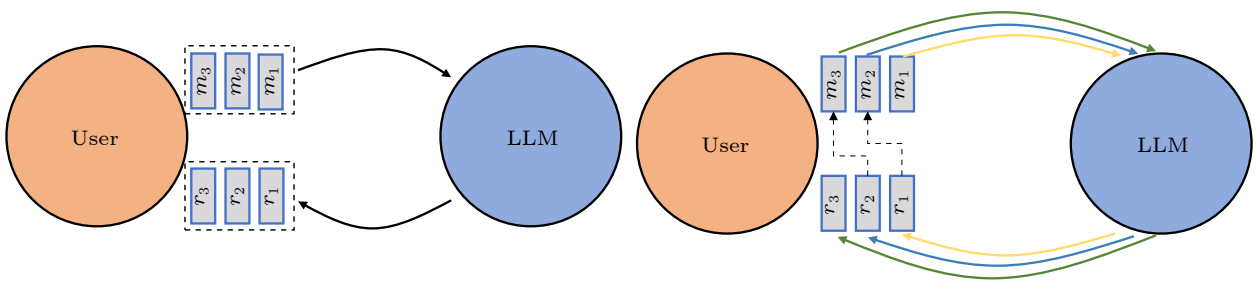
• Developing more robust and generalizable question-answering models that can handle more complex tasks and a broader range of inputs. For example, the answer space is some discrete spans at the beginning (see Section 6) and developed to the complex continuous space (see Subsection 4.1.3).

• There is also a focus on improving the quality and relevance of prompts to improve model performance. Specifically, several techniques have been explored, such as paraphrasing and pruning, after the direct mapping approach. More recently, optimization methods[59, 60] using gradient descent have been proposed to enhance accuracy.

The prompt answering engineering also shows a trend of exploring prompts to decode the machine language with less information loss, i.e., has a better understanding of the machine.

5 Multiple Prompting Methods

Multiple prompts can be utilized to further reduce the information mismatch during the encoding and decoding process. These methods can be categorized into two main types, namely "multi-prompt engineering" and "multi-turn prompt engineering", depending on the interrelationship of prompts (see Fig.2). Multi-prompt engineering is akin to an ensemble system, whereby each response serves as a valid answer, and responses from multiple prompts are aggregated to produce a more stable outcome. This type



(a) (b)

Fig.2. Overview of multiple prompting methods. (a) Multi-prompt methods utilize several similar prompts to produce a more stable result. (b) Multi-turn prompt methods produce the final result by aggregating responses from a sequence of prompts.

of method can be thought to extend the use of prompts in the spatial domain. On the other hand, multi-turn PE entails a sequence of prompts, whereby subsequent prompts depend on the response generated from previous prompts or the obtaining of the final answer relies on multiple responses. Consequently, this type of method can be viewed as an extension in the temporal domain. Table 6 summarizes the main multiple prompting methods.

5.1 Multi-Prompt Engineering Methods

Multi-prompt methods employ multiple prompts with similar patterns during the inference aiming to enhance information preservation. This method is closely associated with assembling techniques[91–93]. Although the primary motivation is to exploit the complementary advantages of different prompts and reduce the expenses associated with PE, it can also be integrated with prompt-engineering techniques to further improve efficacy. From a communication theory perspective, multi-prompt engineering can be considered as sending multiple copies of the message to ensure the authentic delivery of data (see Fig.3(a)).

5.1.1 *Expanding the Prompt*

Expanding the prompt P_T aims to cover a larger semantic space around the sender's true intention, and a more stable approximation of the target output, $\overline{X}_A$, can be obtained by aggregating the responses.

Jiang *et al.*[11], Lester *et al.*[20], and Hambardzumyan *et al.*[52] proposed to combine outputs of different prompts to get the final result for classification tasks. Qin *et al.*[61] incorporated multi-prompt ideas with soft prompts and optimized the weights of each prompt together with prompt parameters. Yuan *et al.*[41] proposed to use text generation probability as the score for text generation evaluation, and aggregated multiple results of different prompts as the final score.

Table 6. Summary of PE Methods Involving Multiple Prompts

	Method	Language Understanding	Language Generation	Reasoning
Multi-prompt	Expanding P_T	[11, 20, 52, 61]	[41]	–
	Diversifying P_A	–	–	[62–67]
	Optimizing θ	[10, 37]	[17, 68]	–
Multi-turn prompt	Decomposing P_T	–	[59, 60, 69]	[70–77]
	Refining P_T	–	[78, 79]	[25, 60, 79–82]
	Augmenting P_T	–	[83, 84]	[85–87]
	Optimizing θ	–	[59, 69]	[87–90]

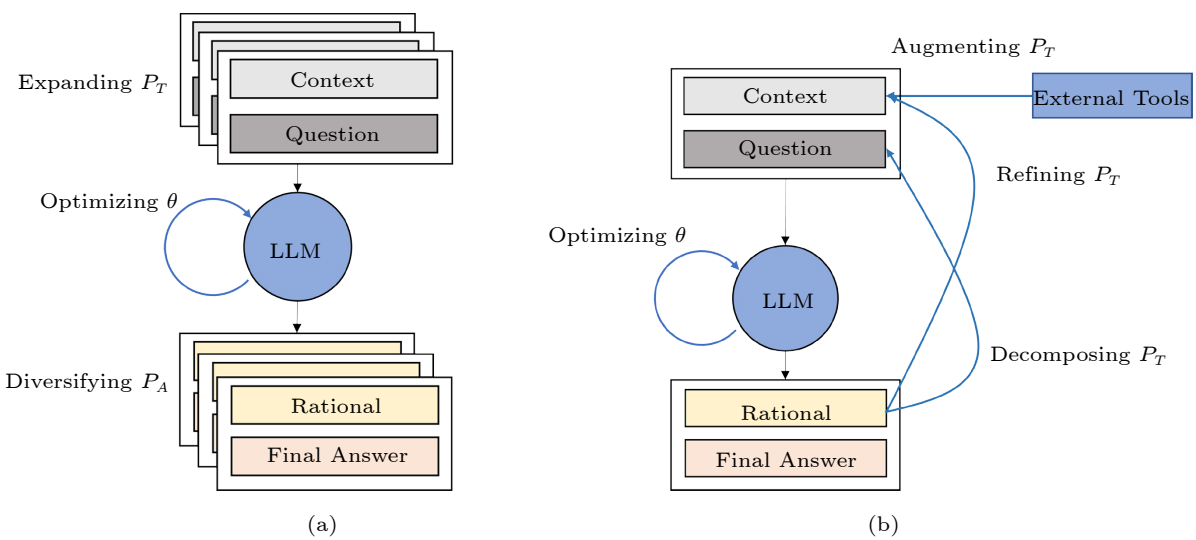


Fig.3. Schematic illustrations of multi-prompting methods. (a) Multi-prompt methods utilize several similar prompts to produce a more stable result. (b) Multi-turn prompt methods mainly leverage LLMs or external tools to provide clearer and more helpful context.

5.1.2 Diversifying the Answer

Different from expanding the prompt P_T whose main goal is to leverage the input space around P_T, diversifying the answer P_A aims to exploit the various "thinking paths" of the LLM through sampling its decoder. This is especially effective for handling complex tasks, such as mathematical and reasoning problems.

Wang *et al.*[62] proposed a self-consistency method based on the Chain-of-Thoughts (CoT) which samples multiple reasoning paths and selects the most consistent answer by majority voting or weighted averaging. Lewkowycz *et al.*[63] applied a similar idea to quantitative problems by combining multiple prompts and output sampling. Wang *et al.*[64] investigated various ensemble variants in reasoning problems and found that rational sampling in the output space is more efficient. These methods solely use the final answer as the selection criterion and do not exploit the generated rationals from various sampling paths. To take advantage of these intermediate results, Li *et al.*[65] proposed to generate more diversified reasoning paths with multiple prompts and used a model-based verifier to select and rank these reasoning paths. Fu *et al.*[66] introduced a complexity-based metric to evaluate reasoning paths and prioritize those with higher complexity in the aggregation. Weng *et al.*[94] employed the LLM to self-verify various reasonings by comparing predicted conditions using the generated reasonings to original conditions. The consistency score is then used to select the final result. Yao *et al.*[95] proposed the "Tree of Thoughts" to explore the intermediate steps across various reasoning paths, and used the LLM to evaluate the quality of each possible path. Besta *et al.*[67] further proposed the "Graph of Thoughts" to treat the various reasoning paths as graphs so that the essence of the thought networks can be extracted.

5.1.3 Optimizing the Model

This line of work treats multiple prompts as a label generator to address the sample deficiency problem. Schick *et al.*[10] first proposed pattern-exploiting training (PET) that employs a knowledge distillation strategy to aggregate results from multiple prompt-verbalizer combinations (PVP). They first utilized PVP pairs to train separate models that generate pseudo-labels for unlabeled datasets. This extended dataset was then used to train the final classification model. Schick *et al.*[96] extended this idea to the text generation task by using the generation probability of decoded text as the score. Gao *et al.*[17] used a similar method for automatic template generation. Schick *et al.*[37] further expanded PET with multiple verbalizers. This was achieved by introducing sample-dependent output space.

5.2 Multi-Turn Prompt Engineering Methods

Multi-turn prompt engineering methods involve decomposing the full prompting task into several sub-tasks, each addressed by a corresponding prompt. This process typically entails a sequence of encoding and decoding operations, where subsequent prompts may depend on the decoded message from previous prompts or each prompt is responsible for a sub-task. The outcome can be obtained either from the result of the last prompt or by aggregating the responses generated by all prompts. This strategy is designed to tackle challenging tasks, such as complex mathematical questions or reasoning tasks. It mainly involves two components: 1) decomposing the prompt P_T into sub-tasks to reduce the difficulty of each sub-task; and 2) modifying the prompt P_T to generate better intermediate results for later steps. These two components can help to bridge the gap between complex X and Y (see Fig.3(b)).

5.2.1 Decomposing the Prompt

Decomposing the prompt P_T is the first step in handling complex tasks, and a proper decomposition requires a good understanding of both the target task and the user's intention. Yang *et al.*[97] decomposed SQL operations using fine-tuned few-shot models and untrained zero-shot models combined with predefined rules. However, ruled-based decomposition heavily relies on human experiences, and thus it is desirable to automate this step with LLMs. Min *et al.*[59] proposed an unsupervised method that utilizes a similarity-based pseudo-decomposition set as a target to train a seq2seq model as a question generator. The decomposed simple question is then answered by an off-the-shelf single-hop QA model. Perez *et al.*[69] treated the decomposition in a multi-hop reading comprehension (RC) task as a span prediction problem which only needs a few hundreds of samples. For each task, various decomposition paths are generated, with each

sub-question answered by a single-hop RC model. Finally, a scorer model is used to select the top-scoring answer based on the solving path. Khot *et al.*[60] proposed a text modular network leveraging existing models to build a next-question generator. The training samples are obtained from sub-task models conditioned on distant supervision hints.

With the emergent general ability of LLMs, instead of training a task-specific decomposition model, LLMs are used to fulfill decomposition tasks. Zhou *et al.*[70] proposed the least-to-most prompting method where hard tasks are first reduced to less difficult subtasks by LLMs. Then answers from previous subproblems are combined with the original task to facilitate subsequent question solving. Dua *et al.*[71] employed a similar idea and appended both questions and answers from the previous stage to the subsequent prompt. Creswell *et al.*[72] proposed a selection-inference framework. It uses LLM to alternatively execute selecting relevant information from a given context and inferring new facts based on the selected information. Arora *et al.*[73] proposed to format the intermediate steps as open-ended question-answering tasks using LLMs. It further generates a set of prompt chains and uses weak supervision to aggregate the results. Khot *et al.*[74] proposed a modular approach for task decomposition with LLMs by using specialized decomposition prompts. Drozdov *et al.*[98] introduced a dynamic least-to-most prompting method for semantic parsing tasks by utilizing multiple prompts to build a more flexible tree-based decomposition. Ye *et al.*[75] used LLMs as the decomposer for table-based reasoning tasks. LLMs are used for both sub-table extraction and question decomposition. Press *et al.*[99] proposed Self-Ask which decomposes the original task by repeatedly asking the LLM if follow-up questions are needed. Wu *et al.*[76] proposed to build an interactive chaining framework with several primitive operations of LLMs to provide better transparency and controllability of using LLMs. Wang *et al.*[77] proposed a Plan-and-Solve (PS) method that explicitly prompts LLM to devise a plan before solving the problem to address the missing-steps error in the reasoning.

5.2.2 *Refining the Prompt*

Refining the prompt P_T aims to construct a better representation of P_T based on the feedback from previous prompting results. This is especially important for multi-step reasoning, where the quality of generated intermediate reasonings has a critical impact on the final answer.

Following the success of the few-shot chain-of-thoughts (CoT) prompting method, Kojima *et al.*[25] proposed a zero-shot CoT method that utilizes the fixed prompt "Let's think step by step" to generate reasonings. These intermediate results are then fused with the original question to get the final answer. To select more effective exemplars, various methods were proposed. Li *et al.*[78] used LLMs to first generate a pseudo-QA pool, then a clustering method combined with similarity to the question was adopted to dynamically select QA pairs from the generated QA pool as demonstration exemplars. Shum *et al.*[80] leveraged a high-quality exemplar pool to obtain an exemplar distribution using a variance-reduced policy gradient estimator. Ye *et al.*[79] employed a self-consistency method[62] to generate pseudo-labels of an unlabeled dataset. The accuracy of these silver labels serves as the selection criterion of exemplars. To further reduce the search complexity of various combinations, additional surrogate metrics were introduced to estimate the accuracy. Diao *et al.*[81] addressed this problem by using hard questions with human annotations as exemplars. The hardness is measured by the disagreement of results obtained by multiple sampling of the LLM. Zhang *et al.*[82] proposed automatic CoT methods. They introduced question clustering and demonstration sampling steps to automatically select the best demonstrations for the CoT template.

5.2.3 *Augmenting the Prompt*

Different from refining the prompt P_T which mainly focuses on finding prompts that generate better intermediate results, augmenting the prompt P_T leverages the exploitation of external information, knowledge, tools, etc. in the prompting. We present some examples in this field below, and for more details we refer the reader to the specific survey[100]. Yang *et al.*[83] proposed a recursive reprompting and revision (3R) framework for long story generation leveraging pre-defined outlines. In each step, the context of the current status and the outline of the story are provided to the prompt to ensure better content coherence. Yang *et al.*[84] proposed to use more detailed outlines so that the story generation LLM can focus more on linguistic aspects. Information retrieved from other sources is also often used to aug-

ment P_T. Yao *et al.*[101] gave the LLM access to information from Wikipedia. Thoppilan *et al.*[102] taught the LLM to use search engines for knowledge retrieval. More broadly, Paranjape *et al.*[26] introduced a task library to enable the LLM using external tools. Schick *et al.*[85] trained the LLM to use various external tools via API. Shen *et al.*[86] utilized an LLM as a central controller to coordinate other models to solve tasks.

5.2.4 *Optimizing the Model*

General LMs (language models) are not optimized for producing intermediate rationals or decomposing a complex task or question. Before the era of LLMs, these tasks require specifically trained LMs. Min *et al.*[59, 69] trained an LM model for decomposing the original task into sub-tasks. Nye *et al.*[88] trained the LLM to produce intermediate steps stored in a scratch pad for later usage. Zelikman *et al.*[89] utilized the intermediate outputs that lead to the correct answer as the target to fine-tune the LLM. Wang *et al.*[87] proposed an iterative prompting framework using a context-aware prompter. The prompter consists of a set of soft prompts that are prepared for the encoder and decoder of the LLMs, respectively. Taylor *et al.*[90] employed step-by-step solutions of scientific papers in the training corpus, which enables the LM to output reasoning steps if required.

5.3 Trends for Multiple Prompting Methods

Ensemble-based methods are easy to implement and flexible to incorporate with various strategies, e.g. expanding the input space and aggregating the output space. However, this brings limited advantages for complex problems whose final answers are hard to obtain directly, but rely heavily on the intermediate thinking steps. Therefore, multi-turn PE methods emerged. A multi-turn method essentially adjusts its input dynamically during the interaction based on the knowledge and feedback from the LLM or external tools. In this way, LLMs can leverage more context and understand better the true intention of the user. Initially, specialized LLMs are trained to handle planning and solving specific subtasks, which not only introduces extra training effort but also constrains the generalization capability of LLM. With the increasing understanding ability and larger input length of LLMs, in-context learning becomes the preferred paradigm, which utilizes embedded knowledge and the capability of LLMs to handle various tasks via prompting. This paradigm soon dominated because of its efficiency and flexibility.

There are two trends in multiple prompting engineering.

• Developing an enhanced adaptive prompting strategy for LLM-based task decomposition is imperative. The extensive range and intricacy of tasks render human-based or rule-based task decomposition infeasible. While some studies have explored the use of LLM prompting to generate intermediate questions or actions for specific tasks, a comprehensive strategy is currently lacking.

• Enabling LLMs to leverage tools without the need for fine-tuning is a crucial objective. By incorporating external tools, LLMs can address their limitations in specialized domains or capabilities. Previous studies[85] have employed fine-tuning based approaches to train LLMs in utilizing web search or other tools accessible through APIs.

From the communication theory perspective, multiple prompting methods evolved from the extension in the spatial domain (ensemble-based methods) into the temporal domain (mulit-turn), to better align the user's intention and LLM's capability by decomposing the user's request and leveraging external tools.

6 Discussion

Researchers have proposed several surveys to recapitulate the rapid advancements in the field of PE methods[7, 103–107]. To name a few, Liu *et. al* proposed a comprehensive survey about existing PE methods, which covers common aspects like template engineering, answering engineering, training strategies, applications, and challenges[7]. They revealed the development history of prompting learning and describe a set of mathematical notations that could summarize most of the existing studies. Furthermore, they considered prompt-based learning as a new paradigm that revolves around the way we look at NLP. In another survey[103] that mainly focuses on the reasoning abilities (e.g., arithmetic, commonsense, symbolic reasoning, logical, and multi-modal) of LLMs, Qiao *et al.* summarized the studies that harness these reasoning abilities via advanced PE methods like chain-of-though and generated knowledge prompts. Additionally, some focused surveys cover specific topics like parameter-efficient fine-tuning (PEFT) LLMs using

PE methods[104]. Different from the above-mentioned studies, we try to interpret existing PE methods from a communication theory perspective.

Following this line of research, we also would like to discuss some potential challenges and future directions for PE methods, which could be divided into three categories including finding the optimal prompts, privacy issue and concern, and interactive and multi-turn prompting.

• *Finding the Optimal Prompts.* One of the points of discrete prompts is that it is difficult to design and choose an optimal prompt, causing its instability. Although soft prompts partly address this problem, the discrete prompt is still very important because it has good interpretability and has been proven to be able to help soft prompts search effectively. Looking through the existing methods, we can find that accuracy-based criteria are resource-consuming, while LM-based log probability is not sufficient to evaluate the prompt. Therefore, a well-designed ranking criterion combined with a mass of auto-based generated prompts may be a good direction for the future. Furthermore, even though the prompt has been proven effective in many tasks such as classification and text generation, most of the existing work has to design a specific prompt for a given task, which makes it complex and complicated[108]. Thus, how to generate a task-agnostic prompt or transfer the prompt to other fields quickly may be a challenging problem. Discrete (meta-learning[109]) and continuous (decomposition[110]) prompts are applied to tackle this issue. However, they are not well-optimized and can not serve unseen tasks.

• *Privacy Issue and Concern.* There are two aspects of privacy and security issue in LLMs. First, users' data may be leaked during the training and inference of LLMs. For instance, training LLMs requires vast amounts of data including personal information, private conversations, or copyrighted material. By providing a series of queries or prompts, an attacker might be able to extract personal details or confidential information from the model's responses. Privacy-preserving methods including techniques such as differential privacy, homomorphic encryption, and federated learning[111–113] may preserve the privacy of the data used for training and inference. Second, the LLMs may be stolen by attackers to misuse. LLMs are highly valuable assets, requiring substantial time and financial investment for their training. Protecting them from unauthorized access and misuse is crucial. Developing robust security measures, such as watermarking techniques[114], is essential to prevent theft and ensure the rightful ownership of LLMs.

• *Interactive and Multi-Turn Prompting.* Besides the automation in prompting methods, humans in the loop can bring more controllability, transparency, and explainability over the process, producing more reliable results. The success of the chain-of-thoughts methodology[115] demonstrates the "thinking path" can enhance LLMs' reasoning capability. This property can also be exploited to generate step-by-step task-solving procedures like scratch paper in exams, so that the final answer can be better justified. Following this idea, Wu *et al.*[76] built an interactive framework involving human interaction for better controllability of the process. However, frequent human intervention will diminish the efficiency gained by using LLMs. Therefore, in addition to the granularity of decomposed tasks, it is also required to determine when to involve human feedback. This could be designed manually for each task, but it would be much more efficient if LLMs could plan these stages by themselves.

• *Bias and Fairness.* LLMs often tend to internalize biases presented in their training datasets, making the mitigation of such biases and the pursuit of fairness a key aspect of existing PE methods. For example, Zhao *et al.*[116] revealed that factors such as the structure of prompts, demonstrations contained in the prompt, and even the order of these demonstrations can lead to diverse performance in in-context learning prompts, and they further proposed calibration to alleviate the bias. Schick *et al.*[117] designed biased or debiased instructions to guide the LLMs to conduct self-diagnosis and self-debiasing. Furthermore, the social biases exhibited by LLMs can potentially lead to discriminatory actions or content targeted at specific groups or demographics. Such problems are often caused from stereotypes that perpetuate harmful generalizations related to gender, race, and religion. For instance, Liu *et al.*[118] introduced a novel prompting approach that reveals how existing LLMs exhibit social biases during text-to-SQL prediction tasks. Despite this progress, the challenge of mitigating biases in LLMs using PE methods remains an area that requires further investigation.

7 Conclusions

This article summarizes the prompting methods from a perspective of communication theory which

provides a coherent explanation of different prompt engineering (PE) methods in terms of their objectives and underlying principles. Theoretical analysis reveals that the ultimate goal of PE is to reduce the information misunderstanding between the users and the LLMs. This novel view facilities a unified review of three PE methods and offers and insights into scenarios where existing prompting methods come short. We hope this survey will inspire researchers with a new understanding of the related issues in prompting methods, therefore stimulating progress in this promising area.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D M, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. Language models are few-shot learners. In *Proc. the 34th International Conference on Neural Information Processing Systems*, Dec. 2020, Article No. 159.

[2] OpenAI. GPT-4 technical report. arXiv: 2303.08774, 2023. https://arxiv.org/abs/2303.08774, Jul. 2024.

[3] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M A, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, Rodriguez A, Joulin A, Grave E, Lample G. LLaMA: Open and efficient foundation language models. arXiv: 2302.13971, 2023. https://arxiv.org/abs/2302.13971, Jul. 2024.

[4] Cheng K M, Li Z Y, Li C, Xie R J, Guo Q, He Y B, Wu H Y. The potential of GPT-4 as an AI-powered virtual assistant for surgeons specialized in joint arthroplasty. *Annals of Biomedical Engineering*, 2023, 51(7): 1366–1370. DOI: 10.1007/s10439-023-03207-z.

[5] Cascella M, Montomoli J, Bellini V, Bignami E. Evaluating the feasibility of ChatGPT in healthcare: An analysis of multiple clinical and research scenarios. *Journal of Medical Systems*, 2023, 47(1): Article No. 33. DOI: 10.1007/s10916-023-01925-4.

[6] George A S, George A S H. A review of ChatGPT AI's impact on several business sectors. *Partners Universal International Innovation Journal*, 2023, 1(1): 9–23. DOI: 10.5281/zenodo.7644359.

[7] Liu P F, Yuan W Z, Fu J L, Jiang Z B, Hayashi H, Neubig G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 2023, 55(9): 195. DOI: 10.1145/3560815.

[8] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. *OpenAI blog*, 2019, 1(8): Article No. 9.

[9] Petroni F, Rocktäschel T, Riedel S, Lewis P, Bakhtin A, Wu Y X, Miller A. Language models as knowledge bases? In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Nov. 2019, pp.2463–2473. DOI: 10.18653/v1/D19-1250.

[10] Schick T, Schütze H. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proc. the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Apr. 2021, pp.255–269. DOI: 10.18653/v1/2021.eacl-main.20.

[11] Jiang Z B, Xu F F, Araki J, Neubig G. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 2020, 8: 423–438. DOI: 10.1162/tacl_a_00324.

[12] Shin T, Razeghi Y, Logan IV R L, Wallace E, Singh S. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Nov. 2020, pp.4222–4235. DOI: 10.18653/v1/2020.emnlp-main.346.

[13] Li X L, Liang P. Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (*Volume 1: Long Papers*), Aug. 2021, pp.4582–4597. DOI: 10.18653/v1/2021.acl-long.353.

[14] Haviv A, Berant J, Globerson A. BERTese: Learning to speak to BERT. In *Proc. the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Apr. 2021, pp.3618–3623. DOI: 10.18653/v1/2021.eacl-main.316.

[15] Liu X, Zheng Y N, Du Z X, Ding M, Qian Y J, Yang Z L, Tang J. GPT understands, too. *AI Open*, 2023. DOI: 10.1016/j.aiopen.2023.08.012.

[16] Zhong Z X, Friedman D, Chen D Q. Factual probing is [MASK]: Learning vs. learning to recall. In *Proc. the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2021, pp.5017–5033. DOI: 10.18653/v1/2021.naacl-main.398.

[17] Gao T Y, Fisch A, Chen D Q. Making pre-trained language models better few-shot learners. In *Proc. the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Aug. 2021, pp.3816–3830. DOI: 10.18653/v1/2021.acl-long.295.

[18] Zhang N Y, Li L Q, Chen X, Deng S M, Bi Z, Tan C Q, Huang F, Chen H J. Differentiable prompt makes pre-trained language models better few-shot learners. In

Proc. the 10th International Conference on Learning Representations, Apr. 2022.

[19] Han X, Zhao W L, Ding N, Liu Z Y, Sun M S. PTR: Prompt tuning with rules for text classification. *AI Open*, 2022, 3: 182–192. DOI: 10.1016/j.aiopen.2022.11.003.

[20] Lester B, Al-Rfou R, Constant N. The power of scale for parameter-efficient prompt tuning. In *Proc. the 2021 Conference on Empirical Methods in Natural Language Processing*, Nov. 2021, pp.3045–3059. DOI: 10.18653/v1/2021.emnlp-main.243.

[21] Gu Y X, Han X, Liu Z Y, Huang M L. PPT: Pre-trained prompt tuning for few-shot learning. In *Proc. the 60th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), May 2022, pp.8410–8423. DOI: 10.18653/v1/2022.acl-long.576.

[22] Deng M K, Wang J Y, Hsieh C P, Wang Y H, Guo H, Shu T M, Song M, Xing E, Hu Z T. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proc. the 2022 Conference on Empirical Methods in Natural Language Processing*, Dec. 2022, pp.3369–3391. DOI: 10.18653/v1/2022.emnlp-main.222.

[23] Hou Y T, Dong H Y, Wang X H, Li B H, Che W X. MetaPrompting: Learning to learn better prompts. In *Proc. the 29th International Conference on Computational Linguistics*, Oct. 2022, pp.3251–3262.

[24] Wang Z, Panda R, Karlinsky L, Feris R, Sun H, Kim Y. Multitask prompt tuning enables parameter-efficient transfer learning. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[25] Kojima T, Gu S S, Reid M, Matsuo Y, Iwasawa Y. Large language models are zero-shot reasoners. In *Proc. the 36th International Conference on Neural Information Processing Systems*, Nov. 28-Dec. 9, 2022, Article No. 1613.

[26] Paranjape B, Lundberg S, Singh S, Hajishirzi H, Zettlemoyer L, Ribeiro M T. ART: Automatic multi-step reasoning and tool-use for large language models. arXiv: 2303.09014, 2023. https://arxiv.org/abs/2303.09014, Jul. 2024.

[27] Narula U. Handbook of Communication: Models, Perspectives, Strategies. Atlantic Publishers & Distributors (P) Ltd, 2006.

[28] Chandler D, Munday R. A Dictionary of Media and Communication. Oxford University Press, 2011.

[29] Cobley P, Schulz P J. Theories and Models of Communication. De Gruyter Mouton, 2013.

[30] Latané B. Dynamic social impact: The creation of culture by communication. *Journal of Communication*, 1996, 46(4): 13–25. DOI: 10.1111/j.1460-2466.1996.tb01501.x.

[31] Orbe M P. From the standpoint(s) of traditionally muted groups: Explicating a co-cultural communication theoretical model. *Communication Theory*, 1998, 8(1): 1–26. DOI: 10.1111/j.1468-2885.1998.tb00209.x.

[32] Segrin C, Abramson L Y. Negative reactions to depressive behaviors: A communication theories analysis. *Journal of Abnormal Psychology*, 1994, 103(4): 655–668. DOI: 10.1037/0021-843X.103.4.655.

[33] Shannon C E. A mathematical theory of communication. *The Bell System Technical Journal*, 1948, 27(3): 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[34] Schramm W. The Process and Effects of Mass Communication. University of Illinois Press, 1954.

[35] Cover T M, Thomas J A. Elements of Information Theory. John Wiley & Sons, 1991.

[36] Sorensen T, Robinson J, Rytting C, Shaw A, Rogers K, Delorey A, Khalil M, Fulda N, Wingate D. An information-theoretic approach to prompt engineering without ground truth labels. In *Proc. the 60th Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), May 2022, pp.819–862. DOI: 10.18653/v1/2022.acl-long.60.

[37] Schick T, Schütze H. It's not just size that matters: Small language models are also few-shot learners. In *Proc. the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2021, pp.2339–2352. DOI: 10.18653/v1/2021.naacl-main.185.

[38] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y Q, Li W, Liu P J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020, 21(1): 140.

[39] Zhou Y L, Zhao Y R, Shumailov I, Mullins R, Gal Y. Revisiting automated prompting: Are we actually doing better? In *Proc. the 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), Jul. 2023, pp.1822–1832. DOI: 10.18653/v1/2023.acl-short.155.

[40] Logan IV R, Balažević I, Wallace E, Petroni F, Singh S, Riedel S. Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Proc. the 2022 Findings of the Association for Computational Linguistics*, May 2022, pp.2824–2835. DOI: 10.18653/v1/2022.findings-acl.222.

[41] Yuan W Z, Neubig G, Liu P F. BARTSCORE: Evaluating generated text as text generation. In *Proc. the 35th International Conference on Neural Information Processing Systems*, Dec. 2021, Article No. 2088.

[42] Ben-David E, Oved N, Reichart R. PADA: Example-based prompt learning for on-the-fly adaptation to unseen domains. *Transactions of the Association for Computational Linguistics*, 2022, 10: 414–433. DOI: 10.1162/tacl_a_00468.

[43] Li B H, Hou Y T, Che W X. Data augmentation approaches in natural language processing: A survey. *AI Open*, 2022, 3: 71–90. DOI: 10.1016/j.aiopen.2022.03.001.

[44] Zhou Y C, Muresanu A I, Han Z W, Paster K, Pitis S, Chan H, Ba J. Large language models are human-level prompt engineers. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[45] Davison J, Feldman J, Rush A M. Commonsense knowledge mining from pretrained models. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Nov. 2019, pp.1173–1178. DOI: 10.18653/v1/D19-1109.

[46] Yang X J, Cheng W, Zhao X J, Yu W C, Petzold L, Chen H F. Dynamic prompting: A unified framework for prompt tuning. arXiv: 2303.02909, 2023. https://arxiv.org/abs/2303.02909, Jul. 2024.

[47] Zaken E B, Goldberg Y, Ravfogel S. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proc. the 60th Annual Meeting of the Association for Computational Linguistics* (*Volume 2: Short Papers*), May 2022. DOI: 10.18653/v1/2022.acl-short.1.

[48] Zhang J O, Sax A, Zamir A, Guibas L, Malik J. Side-tuning: A baseline for network adaptation via additive side networks. In *Proc. the 16th European Conference on Computer Vision*, Aug. 2020, pp.698–714. DOI: 10.1007/978-3-030-58580-8_41.

[49] Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, de Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S. Parameter-efficient transfer learning for NLP. In *Proc. the 36th International Conference on Machine Learning*, Jun. 2019, pp.2790–2799.

[50] Sung Y L, Cho J, Bansal M. LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, Nov. 28-Dec. 9, 2022, Article No. 944.

[51] Schick T, Schmid H, Schütze H. Automatically identifying words that can serve as labels for few-shot text classification. In *Proc. the 28th International Conference on Computational Linguistics*, Dec. 2020, pp.5569–5578. DOI: 10.18653/v1/2020.coling-main.488.

[52] Hambardzumyan K, Khachatrian H, May J. WARP: Word-level adversarial reprogramming. In *Proc. the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (*Volume 1: Long Papers*), Aug. 2021, pp.4921–4933. DOI: 10.18653/v1/2021.acl-long.381.

[53] Chen Y L, Liu Y, Dong L, Wang S H, Zhu C G, Zeng M, Zhang Y. AdaPrompt: Adaptive model training for prompt-based NLP. In *Proc. the 2022 Findings of the Association for Computational Linguistics*, Dec. 2022, pp.6057–6068. DOI: 10.18653/v1/2022.findings-emnlp.448.

[54] Yin W P, Hay J, Roth D. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Nov. 2019, pp.3914–3923. DOI: 10.18653/v1/D19-1404.

[55] Cui L Y, Wu Y, Liu J, Yang S, Zhang Y. Template-based named entity recognition using BART. In *Proc. the 2021 Findings of the Association for Computational Linguistics*, Aug. 2021, pp.1835–1845. DOI: 10.18653/v1/2021.findings-acl.161.

[56] Jiang Z B, Anastasopoulos A, Araki J, Ding H B, Neubig G. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Nov. 2020, pp.5943–5959. DOI: 10.18653/v1/2020.emnlp-main.479.

[57] Nickel M, Kiela D. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *Proc. the 35th International Conference on Machine Learning*, Jul. 2018, pp.3776–3785.

[58] Hou Y T, Che W X, Lai Y K, Zhou Z H, Liu Y J, Liu H, Liu T. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proc. the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp.1381–1393. DOI: 10.18653/v1/2020.acl-main.128.

[59] Min S, Zhong V, Zettlemoyer L, Hajishirzi H. Multi-hop reading comprehension through question decomposition and rescoring. In *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, Jul. 2019, pp.6097–6109. DOI: 10.18653/v1/P19-1613.

[60] Khot T, Khashabi D, Richardson K, Clark P, Sabharwal A. Text modular networks: Learning to decompose tasks in the language of existing models. In *Proc. the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2021, pp.1264–1279. DOI: 10.18653/v1/2021.naacl-main.99.

[61] Qin G H, Eisner J. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proc. the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2021, pp.5203–5212. DOI: 10.18653/v1/2021.naacl-main.410.

[62] Wang X Z, Wei J, Schuurmans D, Le Q V, Chi E H, Narang S, Chowdhery A, Zhou D. Self-consistency improves chain of thought reasoning in language models. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[63] Lewkowycz A, Andreassen A, Dohan D, Dyer E, Michalewski H, Ramasesh V, Slone A, Anil C, Schlag I, Gutman-Solo T, Wu T H, Neyshabur B, Gur-Ari G, Misra V. Solving quantitative reasoning problems with language models. In *Proc. the 36th International Conference on Neural Information Processing Systems*, Nov. 28-Dec. 9, 2022, Article No. 278.

[64] Wang X Z, Wei J, Schuurmans D, Le Q, Chi E, Zhou D. Rationale-augmented ensembles in language models. arXiv: 2207.00747, 2022. https://arxiv.org/abs/2207.

00747, Jul. 2024.

[65] Li Y F, Lin Z Q, Zhang S Z, Fu Q, Chen B, Lou J G, Chen W Z. On the advance of making language models better reasoners. arXiv: 2206.02336, 2022. https://arxiv.org/abs/2206.02336v1, Jul. 2024.

[66] Fu Y, Peng H, Sabharwal A, Clark P, Khot T. Complexity-based prompting for multi-step reasoning. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[67] Besta M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, Gianinazzi L, Gajda J, Lehmann T, Niewiadomski H, Nyczyk P, Hoefler T. Graph of thoughts: Solving elaborate problems with large language models. In *Proc. the 38th AAAI Conference on Artificial Intelligence*, Feb. 2024, pp.17682–17690. DOI: 10.1609/aaai.v38i16.29720.

[68] Schick T, Schütze H. Few-shot text generation with pattern-exploiting training. arXiv: 2012.11926, 2020. https://arxiv.org/abs/2012.11926, Jul. 2024.

[69] Perez E, Lewis P, Yih W T, Cho K, Kiela D. Unsupervised question decomposition for question answering. In *Proc. the 2020 Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Nov. 2020, pp.8864–8880. DOI: 10.18653/v1/2020.emnlp-main.713.

[70] Zhou D, Schärli N, Hou L, Wei J, Scales N, Wang X Z, Schuurmans D, Cui C, Bousquet O, Le Q V, Chi E H. Least-to-most prompting enables complex reasoning in large language models. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[71] Dua D, Gupta S, Singh S, Gardner M. Successive prompting for decomposing complex questions. In *Proc. the 2022 Conference on Empirical Methods in Natural Language Processing*, Dec. 2022, pp.1251–1265. DOI: 10.18653/v1/2022.emnlp-main.81.

[72] Creswell A, Shanahan M, Higgins I. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[73] Arora S, Narayan A, Chen M F, Orr L J, Guha N, Bhatia K, Chami I, Ré C. Ask me anything: A simple strategy for prompting language models. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[74] Khot T, Trivedi H, Finlayson M, Fu Y, Richardson K, Clark P, Sabharwal A. Decomposed prompting: A modular approach for solving complex tasks. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[75] Ye Y H, Hui B Y, Yang M, Li B H, Huang F, Li Y B. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning. arXiv: 2301.13808, 2023. https://arxiv.org/abs/2301.13808, Jul. 2024.

[76] Wu T S, Terry M, Cai C J. AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts. In *Proc. the 2022 CHI Conference on Human Factors in Computing Systems*, Apr. 29-May 5, 2022, Article No. 385. DOI: 10.1145/3491102.3517582.

[77] Wang L, Xu W Y, Lan Y H, Hu Z Q, Lan Y S, Lee R K W, Lim E P. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proc. the 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Jul. 2023, pp.2609–2634. DOI: 10.18653/v1/2023.acl-long.147.

[78] Li J L, Wang J Y, Zhang Z S, Zhao H. Self-prompting large language models for zero-shot open-domain QA. In *Proc. the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (*Volume 1: Long Papers*), Jun. 2024, pp.296–310. DOI: 10.18653/v1/2024.naacl-long.17.

[79] Ye X, Durrett G. Explanation selection using unlabeled data for chain-of-thought prompting. In *Proc. the 2023 Conference on Empirical Methods in Natural Language Processing*, Dec. 2023, pp.619–637. DOI: 10.18653/v1/2023.emnlp-main.41.

[80] Shum K, Diao S Z, Zhang T. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Proc. the 2023 Findings of the Association for Computational Linguistics*, Dec. 2023, pp.12113–12139. DOI: 10.18653/v1/2023.findings-emnlp.811.

[81] Diao S Z, Wang P C, Lin Y, Pan R, Liu X, Zhang T. Active prompting with chain-of-thought for large language models. arXiv: 2302.12246, 2023. https://arxiv.org/abs/2302.12246, Jul. 2024.

[82] Zhang Z S, Zhang A, Li M, Smola A. Automatic chain of thought prompting in large language models. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[83] Yang K, Tian Y D, Peng N Y, Klein D. Re^3: Generating longer stories with recursive reprompting and revision. In *Proc. the 2022 Conference on Empirical Methods in Natural Language Processing*, Dec. 2022, pp.4393–4479. DOI: 10.18653/v1/2022.emnlp-main.296.

[84] Yang K, Klein D, Peng N Y, Tian Y D. Doc: Improving long story coherence with detailed outline control. In *Proc. the 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Jul. 2023, pp.3378–3465. DOI: 10.18653/v1/2023.acl-long.190.

[85] Schick T, Dwivedi-Yu J, Dessí R, Raileanu R, Lomeli M, Hambro E, Zettlemoyer L, Cancedda N, Scialom T. Toolformer: Language models can teach themselves to use tools. In *Proc. the 37th International Conference on Neural Information Processing Systems*, Dec. 2023, Article No. 2997.

[86] Shen Y L, Song K T, Tan X, Li D S, Lu W M, Zhuang Y T. HuggingGPT: Solving AI tasks with ChatGPT and

its friends in hugging face. In *Proc. the 37th International Conference on Neural Information Processing Systems*, Dec. 2023, Article No. 1657.

[87] Wang B S, Deng X, Sun H. Iteratively prompt pre-trained language models for chain of thought. In *Proc. the 2022 Conference on Empirical Methods in Natural Language Processing*, Dec. 2022, pp.2714–2730. DOI: 10.18653/v1/2022.emnlp-main.174.

[88] Nye M, Andreassen A J, Gur-Ari G, Michalewski H, Austin J, Bieber D, Dohan D, Lewkowycz A, Bosma M, Luan D, Sutton C, Odena A. Show your work: Scratchpads for intermediate computation with language models. In *Proc. the 2022 Deep Learning for Code Workshop*, May 2022.

[89] Zelikman E, Wu Y H, Mu J, Goodman N D. STaR: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, Nov. 28-Dec. 9, 2022, Article No. 1126.

[90] Taylor R, Kardas M, Cucurull G, Scialom T, Hartshorn A, Saravia E, Poulton A, Kerkez V, Stojnic R. Galactica: A large language model for science. arXiv: 2211.09085, 2022. https://arxiv.org/abs/2211.09085, Jul. 2024.

[91] Ting K M, Witten I H. Stacked generalization: When does it work? In *Proc. the 15th International Joint Conference on Artificial Intelligence*, Aug. 1997, pp.866–871.

[92] Zhou Z H, Wu J X, Tang W. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 2002, 137(1/2): 239–263. DOI: 10.1016/S0004-3702(02)00190-X.

[93] Duh K, Sudoh K, Wu X C, Tsukada H, Nagata M. Generalized minimum Bayes risk system combination. In *Proc. the 5th International Joint Conference on Natural Language Processing*, Nov. 2011, pp.1356–1360.

[94] Weng Y X, Zhu M J, Xia F, Li B, He S Z, Liu S P, Sun B, Liu K, Zhao J. Large language models are better reasoners with self-verification. In *Proc. the 2023 Findings of the Association for Computational Linguistics*, Dec. 2023, pp.2550–2575. DOI: 10.18653/v1/2023.findings-emnlp.167.

[95] Yao S Y, Yu D, Zhao J, Shafran I, Griffiths T L, Cao Y, Narasimhan K. Tree of thoughts: Deliberate problem solving with large language models. In *Proc. the 37th International Conference on Neural Information Processing Systems*, Dec. 2023, Article No. 517.

[96] Schick T, Schütze H. Few-shot text generation with natural language instructions. In *Proc. the 2021 Conference on Empirical Methods in Natural Language Processing*, Nov. 2021, pp.390–402. DOI: 10.18653/v1/2021.emnlp-main.32.

[97] Yang J F, Jiang H M, Yin Q Y, Zhang D Q, Yin B, Yang D Y. SEQZERO: Few-shot compositional semantic parsing with sequential prompts and zero-shot models. In *Proc. the 2022 Findings of the Association for Computational Linguistics*, Jul. 2022, pp.49–60. DOI: 10.18653/v1/2022.findings-naacl.5.

[98] Drozdov A, Schärli N, Akyürek E, Scales N, Song X Y, Chen X Y, Bousquet O, Zhou D. Compositional semantic parsing with large language models. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[99] Press O, Zhang M R, Min S, Schmidt L, Smith N A, Lewis M. Measuring and narrowing the compositionality gap in language models. In *Proc. the 2023 Findings of the Association for Computational Linguistics*, Dec. 2023, pp.5687–5711. DOI: 10.18653/v1/2023.findings-emnlp.378.

[100] Mialon G, Dessi R, Lomeli M, Nalmpantis C, Pasunuru R, Raileanu R, Rozière B, Schick T, Dwivedi-Yu J, Celikyilmaz A, Grave E, LeCun T, Scialom T. Augmented language models: A survey. arXiv: 2302.07842, 2023. https://arxiv.org/abs/2302.07842, Jul. 2024.

[101] Yao S Y, Zhao J, Yu D, Du N, Shafran I, Narasimhan K R, Cao Y. ReAct: Synergizing reasoning and acting in language models. In *Proc. the 11th International Conference on Learning Representations*, May 2023.

[102] Thoppilan R, De Freitas D, Hall J, Shazeer N, Kulshreshtha A, Cheng H T, Jin A, Bos T, Baker L, Du Y, Li Y, Lee H, Zheng H S, Ghafouri A, Menegali M, Huang Y P, Krikun M, Lepikhin D, Qin J, Chen D H, Xu Y Z, Chen Z F, Roberts A, Bosma M, Zhao V, Zhou Y Q, Chang C C, Krivokon I, Rusch W, Pickett M, Srinivasan P, Man L, Meier-Hellstern K, Morris M R, Doshi T, Santos R D, Duke T, Soraker J, Zevenbergen B, Prabhakaran V, Diaz M, Hutchinson B, Olson K, Molina A, Hoffman-John E, Lee J, Aroyo L, Rajakumar R, Butryna A, Lamm M, Kuzmina V, Fenton J, Cohen A, Bernstein R, Kurzweil R, Aguera-Arcas B, Cui C, Croak M, Chi E, Le Q. LaMDA: Language models for dialog applications. arXiv: 2201.08239, 2022. https://arxiv.org/abs/2201.08239, Jul. 2024.

[103] Qiao S F, Ou Y X, Zhang N Y, Chen X, Yao Y Z, Deng S M, Tan C Q, Huang F, Chen H J. Reasoning with language model prompting: A survey. In *Proc. the 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Jul. 2023, pp.5368–5393. DOI: 10.18653/v1/2023.acl-long.294.

[104] Lialin V, Deshpande V, Rumshisky A. Scaling down to scale up: A guide to parameter-efficient fine-tuning. arXiv: 2303.15647, 2023. https://arxiv.org/abs/2303.15647, Jul. 2024.

[105] Zhao W X, Zhou K, Li J Y, Tang T Y, Wang X L, Hou Y P, Min Y Q, Zhang B C, Zhang J J, Dong Z C, Du Y F, Yang C, Chen Y S, Chen Z P, Jiang J H, Ren R Y, Li Y F, Tang X Y, Liu Z K, Liu P Y, Nie J Y, Wen J R. A survey of large language models. arXiv: 2303.18223, 2023. https://arxiv.org/abs/2303.18223, Jul. 2024.

[106] Dong Q X, Li L, Dai D M, Zheng C, Wu Z Y, Chang B B, Sun X, Xu J J, Li L, Sui Z F. A survey for in-con-

text learning. arXiv: 2301.00234, 2022. https://arxiv.org/abs/2301.00234v1, Jul. 2024.

[107] Lou R Z, Zhang K, Yin W P. Is prompt all you need? No. A comprehensive and broader view of instruction learning. arXiv: 2303.10475, 2023. https://arxiv.org/abs/2303.10475v1, Jul. 2024.

[108] Zhong R Q, Lee K, Zhang Z, Klein D. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Proc. the 2021 Findings of the Association for Computational Linguistics*, Nov. 2021, pp.2856–2878. DOI: 10.18653/v1/2021.findings-emnlp.244.

[109] Reynolds L, McDonell K. Prompt programming for large language models: Beyond the few-shot paradigm. In *Proc. the 2021 CHI Conference on Human Factors in Computing Systems*, May 2021, Article No. 314. DOI: 10.1145/3411763.3451760.

[110] Gu Z H, Fan J, Tang N, Cao L, Jia B W, Madden S, Du X Y. Few-shot text-to-SQL translation using structure and content prompt learning. *Proceedings of the ACM on Management of Data*, 2023, 1(2): 147. DOI: 10.1145/3589292.

[111] Abadi M, Chu A, Goodfellow I, McMahan H B, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In *Proc. the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2016, pp.308–318. DOI: 10.1145/2976749.2978318.

[112] Gentry C. A fully homomorphic encryption scheme [Ph. D. Thesis]. Stanford University, Palo Alto, 2009.

[113] Yang Q, Liu Y, Chen T J, Tong Y X. Federated machine learning: Concept and applications. *ACM Trans. Intelligent Systems and Technology*, 2019, 10(2): 12. DOI: 10.1145/3298981.

[114] Kirchenbauer J, Geiping J, Wen Y X, Katz J, Miers I, Goldstein T. A watermark for large language models. In *Proc. the 40th International Conference on Machine Learning*, Jul. 2023, pp.17061–17084.

[115] Wei J, Wang X Z, Schuurmans D, Bosma M, Ichter B, Xia F, Chi E H, Le Q V, Zhou D. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. the 36th International Conference on Neural Information Processing Systems*, Nov. 28-Dec. 9, 2022, Article No. 1800.

[116] Zhao Z H, Wallace E, Feng S, Klein D, Singh S. Calibrate before use: Improving few-shot performance of language models. In *Proc. the 38th International Conference on Machine Learning*, Jul. 2021, pp.12697–12706.

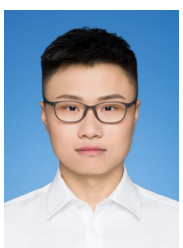
[117] Schick T, Udupa S, Schütze H. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 2021, 9: 1408–1424. DOI: 10.1162/tacl_a_00434.

[118] Liu Y, Gao Y, Su Z, Chen X K, Ash E, Lou J G. Uncovering and categorizing social biases in text-to-SQL. In *Proc. the 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), Jul. 2023, pp.13573–13584. DOI: 10.18653/v1/2023.acl-long.759.

Yuan-Feng Song is a researcher in WeBank AI Group, WeBank, Shenzhen. His research interests include learning to rank, data visualization, and speech-driven applications. In his career, he has published several papers in venues such as KDD, ICDM, EMNLP, MM, TIST, TKDE, and SIGMOD.

Yuan-Qin He is currently a researcher with WeBank AI Group, WeBank, Shenzhen. He received his B.S. degree in Physics from Shanghai Jiao Tong University, and his Ph.D. degree in physics from the Technical University of Munich, Munich, in 2017. His research interests include machine learning and federated learning.

Xue-Fang Zhao received her Master degree in computer science from the Tsinghua University, Beijing, in 2020. She is currently a research engineer at WeBank AI Group, WeBank, Shenzhen. Her research interests include natural language processing and speech recognition.



Han-Lin Gu received his B.S. degree in mathematics from University of Science and Technology of China, Hefei, in 2017. He received his Ph.D. degree in mathematics from Hong Kong University of Science and Technology, Hong Kong, in 2022. He now works as a senior researcher at WeBank AI Group, WeBank, Shenzhen. His research interests include federated learning and privacy-preserving methodology. He has published a series of papers in TPAMI, TDSC, IJCAI, PAKDD, and so on.

Di Jiang received his Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 2014. He is currently the principal scientist at WeBank AI Group, WeBank, Shenzhen. His research interests include information retrieval, natural language processing, and massive data management.

Hai-Jun Yang received his B.E. degree in 2008 and his M.S. degree in 2011, both from Harbin Institute of Technology, Harbin. He is currently the Senior Manager of the AI Group at WeBank, Shenzhen, mainly responsible for promoting the integration and implementation of AI technology with WeBank in customer service, risk control, marketing, and other business scenarios.

Li-Xin Fan is the Principal Scientist of Artificial Intelligence at WeBank, Shenzhen, and the Chairman of the Federal Learning Industry Ecological Development Alliance. His research fields include machine learning and deep learning, computer vision and pattern recognition, image and video processing, 3D big data processing, data visualization and rendering, augmented and virtual reality, mobile computing and ubiquitous computing, and intelligent man-machine interface. He is the author of more than 70 international journals and conference articles. He has worked at Nokia Research Center and Xerox Research Center Europe. His research includes the well-known Bag of Keypoints image classification method. He has participated in NIPS/NeurIPS, ICML, CVPR, ICCV, ECCV, IJCAI and other top artificial intelligence conferences for a long time, served as area chair of AAAI, and organized workshops in various technical fields. He is also the inventor of more than one hundred patents filed in the United States, Europe, and China, and the chairman of the IEEE P2894 Explainable Artificial Intelligence (XAI) Standard Working Group.