计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2025/19(12)-3131-22
doi: 10.3778/j.issn.1673-9418.2502027

大语言模型自动化提示工程技术研究综述

巴泽智[1],张　辉[2+],谢铮涵[1],左晓栋[1,2],侯健玮[3]
1. 中国科学技术大学 网络空间安全学院,合肥 230031
2. 中国科学技术大学 公共事务学院,合肥 230026
3. 中央网信办(国家网信办)数据与技术保障中心,北京 100048
+ 通信作者 E-mail: zhanghui.pas@ustc.edu.cn

摘　要:基于提示学习的提示工程对于提升大语言模型的技术可及性、加速其推广扩散与应用开发至关重要。传统的提示工程过度依赖于提示词设计者的领域知识和使用经验,且不易满足提示空间较大的任务;相比之下,自动化提示工程能够自动化或半自动化地生成或优化提示词,以探索大规模的提示词组合,并通过自动优化技术提升提示词生成的稳定性。然而目前仍缺乏对自动化提示研究的系统性综述,因此,及时跟进该领域的最新研究成果,详细梳理并评述自动化提示工程的实现形式,提出自动化提示工程的未来研究方向。依据自动化提示工程实现形式在逻辑推理和效能导向两个维度的取舍上,将其分为基于思维链的自动化提示工程、基于类机器学习模型的自动化提示工程、基于进化算法的自动化提示工程以及使用预训练包的即插即用系统。全面评估自动化提示工程技术,构建其工作原理的理论解释框架,评估各类实现形式的适用性与局限性。最后,展望多模态大模型、强推理模型以及智能体中自动化提示工程的发展趋势。

关键词:大语言模型;提示工程;自动化提示工程;思维链;机器学习;进化算法;即插即用系统
文献标志码:A　**中图分类号:**TP391

Automatic Prompt Engineering Technology for Large Language Models: a Survey

BA Zezhi[1], ZHANG Hui[2+], XIE Zhenghan[1], ZUO Xiaodong[1,2], HOU Jianwei[3]
1. School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230031, China
2. School of Public Affairs, University of Science and Technology of China, Hefei 230026, China
3. Data and Technology Support Center of the Cyberspace Administration of China, Beijing 100048, China

Abstract: Prompt engineering (PE) based on prompt learning is crucial for improving the technical accessibility of LLMs and accelerating their adoption, diffusion, and application development. Compared with traditional PE, which heavily relies on the domain knowledge and experience of prompt designers and is less adaptable to tasks with large prompt spaces, automatic prompt engineering (APE) can generate or optimize prompts in an automatic or semi-automatic way. This enables the exploration of large-scale prompt combinations and enhances the stability of prompt generation through automated optimization techniques. However, there is currently a lack of systematic reviews on APE, which hinders subsequent researchers from quickly grasping the state of the field. Therefore, this paper keeps up with the latest research developments, systematically reviews the implementation forms of automated prompt engineering, and proposes future research directions. Based on the trade-offs in logical reasoning and performance orientation in the implementation of a

基金项目:国家重点研发计划(2022YFB3103200);国家社会科学基金重点项目(24AGL009);国家社会科学基金重大项目(23&ZD335)。
This work was supported by the National Key Research and Development Program of China (2022YFB3103200), the Key Project of the National Social Science Fund of China (24AGL009), and the Major Project of the National Social Science Fund of China (23&ZD335).
收稿日期:2025-02-17　**修回日期:**2025-06-20

APE, this paper categorizes it into four main types: APE based on chain-of-thought, APE based on machine learning models, APE based on evolutionary algorithms and plug-and-play auto-prompt systems. Subsequently, this paper conducts a comprehensive evaluation of APE techniques, constructing a theoretical explanatory framework for their working principles and assessing the applicability and limitations of each implementation form. Finally, this paper looks ahead to the development trends of APE in multimodal large models, advanced reasoning models and AI-Agents.

Key words: large language models; prompt engineering; automatic prompt engineering; chain of thought; machine learning; evolutionary algorithms; plug-and-play systems

大语言模型(large language model,LLM)已经成为人工智能领域的核心技术之一,以其为中心的技术创新和应用扩散日新月异。随着模型规模的扩大和训练数据量的增加,如GPT-4等模型在语言理解、推理和文本生成等任务上达到了前所未有的水平[1-2]。这些模型不仅能够处理复杂对话、文本生成、翻译、摘要等任务,还能理解深层次的语义、推断隐含的信息、分析上下文联系和内在逻辑。其优势体现在卓越的语言理解与推理能力、强大的文本生成能力和多任务处理能力[3]。通过大规模数据训练,LLM能够精准地理解语言中的语法、语义和上下文,且在多语言处理和翻译方面也具有显著的优势,能够消除语言障碍,提升跨文化沟通的效率[4-5]。随着LLM的便捷应用接口及其对话模式的出现,一般用户获得了友好使用界面,大大降低了LLM使用的技术门槛。基于此,LLM得以加速应用普及,同时与其自身创新形成互补效应。

提示工程(prompt engineering,PE)是LLM对话模式的技术基础。其定义为在用户端设计和编写提示文本,以引导模型生成符合预期要求输出的过程。精心设计的输入提示词能够使得模型在没有额外微调的情况下,通过预先构造的输入指令来执行多种复杂任务。提示工程的作用在于最大化模型的潜力,提升其泛化能力,同时避免在微调过程中带来的庞大计算开销[6-7]。在实际应用中,提示工程技术门槛较低,现已经被广泛应用于文本生成、问答系统等任务中,是提升模型效能和扩展应用范围的重要技术。近期研究表明,LLM中的提示机制具有图灵完备性:即通过合适的提示设计,一个固定大小的Transformer模型理论上可以实现任何可计算函数,这为提示工程提供了坚实的理论基础[8]。

尽管在实践中取得显著成效,但提示工程也存在诸多缺陷,面临着多重挑战。首先,提示工程依赖于提示词设计者的经验和领域知识,细微变化可能导致模型输出出现较大波动,人工设计的效率低下无疑增加了设计者的工作负担。其次,在人机交互过程中,用户模棱两可的输入提示词往往使得LLM的输出结果偏离其预期,尤其是在处理复杂任务时,提示工程往往会叠加偏差。最后,对于任务空间较大的场景,通常需要反复试探多种提示结构以找到最佳的提示词组合,严重影响模型使用的整体效率[9-10]。为了解决这些问题,自动化提示工程(automatic prompt engineering,APE)应运而生。由于其处于初步发展阶段,目前学术界和工业界尚未有明确定义,在本文中将APE定义为:通过全自动化或半自动化的方式实现提示工程的过程。具体而言,其通过优化输入提示,减少人工干预,提高提示词设计效率,并能根据任务需求快速探索大量提示组合,从而找到最优提示词组合。APE的出现标志着提示工程从人工设计向智能化设计方向过渡,为提高LLM的性能提供了新的思路。

当前提示工程领域的研究具有明显的“技术失衡”特征,研究重心长期聚焦于传统的人工提示范式,其核心逻辑依赖于人工经验总结的固定模板(如角色扮演、场景模拟等)而忽视泛化性和动态适应性。此外,当前提示工程领域也缺乏全面的系统性综述,尤其是在自动化提示工程领域几乎处于尚未探索的阶段。因此,本文旨在总结自动化提示工程领域的研究现状、挑战与发展方向,及时跟进该领域的最新研究成果,并提出未来的研究方向。

本文的总体研究结构如图1所示,具体的技术细节将在对应章节介绍。

本文的主要贡献在于:系统化搜集和整理自动提示工程领域现有技术,为该领域研究提供了有价值的理论依据和应用参考;通过总结该领域的研究现状、面临的挑战及发展方向,及时跟进该领域的最新研究成果,并提出了未来可能的研究方向。

1 研究背景

1.1 大语言模型

大语言模型(LLM)是基于深度学习的人工智能模型,旨在处理和理解人类语言。这些模型通常由数十亿到数万亿个参数组成的神经网络构成,通过对大规模文本语料进行训练而得到[11]。大语言模型在自然语言处理任务中表现出色,包括文本生成、机器翻译、情感分析

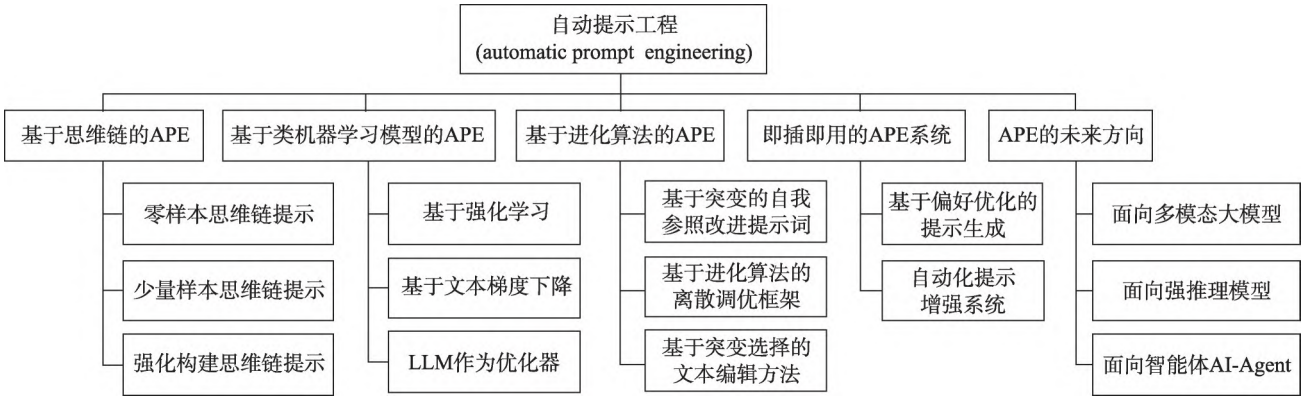

图1 自动提示工程框架图

Fig.1 Framework of automatic prompt engineering

等。无论是像GPT(generative pre-trained transformer)系列的自回归模型，或是类似BERT(bidirectional encoder representations from transformers)的自编码模型还是以T5为代表的序列到序列模型，其训练过程通常包括两个关键步骤：预训练和微调。在预训练阶段，模型在大规模未标注的文本数据上进行训练，学习语言的基本结构和模式；在微调阶段，模型在特定任务的标注数据上进行训练，以适应特定的应用场景。本文将在2.1节详细介绍大语言模型的全流程。

词元(tokens)是语言模型文本处理中的基本概念，是将文本分割成的最小单位，用于后续的语言模型处理。这些单位可以是单词、子词、单个或数个字母、字符，或者其他语言中的基本元素。在LLM中，词元化是预处理步骤的一部分，它影响着模型如何理解和处理输入文本[12]。词元的选择对于模型的性能至关重要，因为它决定了模型能够捕捉到的语言粒度和复杂性。例如，使用子词词元化可以提高模型处理未知词汇的能力，因为它可以将词汇分解成更小的、更常见的部分。在不同的语言和应用场景中，选择合适的词元化策略对于优化模型性能和结果至关重要[13]。

1.2 提示词工程

提示词(prompts)是输入到语言模型中的文本，用于引导模型生成特定的输出。在LLM的应用中，提示词的设计对于激发模型的特定行为和生成相关输出起着关键作用。一个有效的提示词不仅包含必要的信息以引导模型，还能够以一种方式激活模型的知识库，使其能够理解并执行任务[14]。提示词可以是简单的问题、指令、部分句子或更复杂的文本结构，它们在机器翻译、文本摘要、问答系统等多种自然语言处理任务中都有应用。

提示工程(PE)是指设计和编写提示文本，以引导模型生成符合特定要求的语言输出。PE包括选择合适的词汇、语法、上下文、角色扮演和场景等元素，以及使用不同的技巧和策略来影响模型的生成行为和结果等[10]。通过优化提示词，可以使大语言模型更加准确、可控和适应不同的任务和应用场景，因此PE在自然语言处理、文本生成、对话系统、信息检索等领域中具有重要的应用价值[15]。其主要依赖于人工设计，往往依赖于提示词工程师的经验通过静态方式实现。表1为PE与APE的特征对比。

表1 PE与APE的特征对比

Table 1 Characteristic comparison of PE and APE

对比维度	提示工程(PE)	自动化提示工程(APE)
核心技术	依靠人工经验编写	算法驱动的智能优化
优化逻辑	通过试错离散空间调整	通过策略连续空间搜索
泛化性能	低(高度领域依赖)	高(一般能跨任务迁移)
计算成本	低(人工调试)	高(需要算力资源)
应用瓶颈	复杂场景、效率低下等	缺乏可解释性、算力等

自动化提示工程(APE)是指通过全自动化或半自动化的方式优化或生成输入提示，以引导模型产生期望的输出。APE主要依赖各种搜索和优化算法，通过实时感知模型输出进行动态设计。其主要优势在于其高效性、自动化和功能稳定性；能够快速探索大量的提示组合，找到最优配置，并通过自动优化确保生成的提示具有更高的稳定性。APE的兴起标志着人工智能技术在提高模型性能和优化人机交互体验方面迈出了重要一步。

1.3 提示词优化

根据优化提示词的连续性，优化方式可以分为离散优化和连续优化。

离散优化指的是通过对提示或输入进行手动或规则化的调整，以找到更有效的提示形式。这种优化方式是非连续的，通常涉及离散的变化，如词语、句子的替换或重组。离散优化不依赖模型的内部结构和参数更新，

仅对输入的离散元素进行调整。离散优化的方式实现包括手动调整、启发式搜索和提示模板设计等[16]。

连续优化则是基于提示嵌入的优化技术，其中提示或输入被转化为连续的数值向量，并在该连续空间中进行优化。相较于离散优化，连续优化利用模型的嵌入空间来探索提示对输出的影响。连续优化通常用于在更大提示空间中进行更细粒度的搜索和调整，能够捕捉到文本提示中的微小变化，适用于更复杂的模型优化任务[5]。

2 APE理论框架

APE主要解决两个问题：一是减少人工参与提示设计的过程，通过自动化生成和优化提示来实现这一点，因为手动进行提示工程既费时又需要相当的专业知识；二是通过自动化的动态优化来持续提升提示词质量，形成动态化更新提示词的过程，从而提高大语言模型在特定任务上的性能，以更好地与所需输出对齐[17]。本章结合LLM的实现流程，详细介绍APE的技术流程框架，并提出具体分类，给出分类拓扑。

2.1 APE在LLM全流程中的作用

一般来说，成熟可用的LLM训练过程应当包括数据工程、预训练、微调与对齐、测试与评估四个主要阶段，其中每阶段又包含诸多过程，如图2所示。

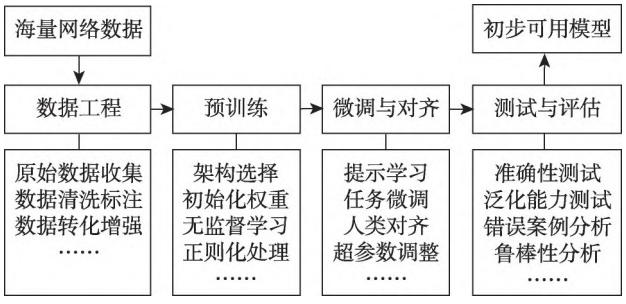

图2 LLM的训练流程

Fig.2 Training process of LLM

在微调与对齐阶段，提示学习（prompt learning）作为一种常用手段，其通过向模型输入增加“提示信息”，使得在不显著改变模型结构和参数的情况下提升模型对特定任务的解决能力[18]。提示学习的本质是在模型训练的过程中提前告知模型的下游任务，从而在较为前期的过程实现模型对提示词的适应能力。因此提示学习也可看作提示工程的一部分。

更为广义的提示工程，是在LLM下游任务中使用的，基于用户意图或是原始提示进行的提示词编写。其通过精心设计的提示来引导模型生成更准确、更符合预期的输出结果。其能够显著提高模型对特定任务的理解和执行效率，尤其是在资源有限的情况下，通过优化提示来实现性能的提升。

2.2 APE实现的流程框架

APE的核心是通过一系列有序的步骤和优化迭代，自动化地提升提示词的质量，优化其在目标大语言模型上的输出效果。这个过程可以表示为：

$$P^{*}=\arg\max_{P}\mathcal{F}(\mathcal{M}(P;\theta),\mathcal{D}_{\text{eval}}) \tag{1}$$

其中，P即提示词Prompt，是APE中的优化变量；P^{*}为最优提示词，通过优化得到输出结果；$\mathcal{M}$为目标大语言模型；θ为$\mathcal{M}$经过预训练和微调后的参数；$\mathcal{D}_{\text{eval}}$为评估数据集，包含标准的“输入提示词-输出结果”对；$\mathcal{F}$为得分函数，用于评估模型输出的质量。

为了更好地理解APE的工作过程，本文提出了如图3所示的流程化框架，为APE的综合研究提供了高效的结构和方法论。此外，框架中的主要组件（①至⑦）的具体工作流程和细节也在文中进行了详细说明。

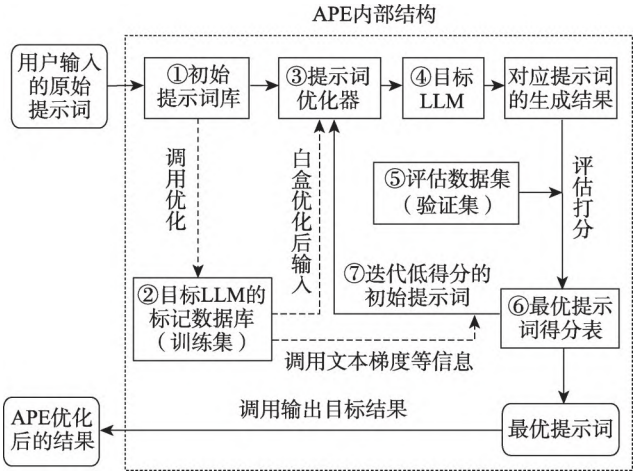

图3 APE工作的基本流程

Fig.3 Basic process of APE

（1）初始提示词生成。首先，根据使用者的基本命令C生成一个初始提示词P_0，作为后续迭代优化的基础。

$$P_0=\text{GenerateInitialPrompt}(C;\phi) \tag{2}$$

其中，C为用户的基本命令或需求；ϕ为初始提示生成规则集，如使用专家系统的规则或思维链的方法等。

（2）标记数据词库。部分APE实现采用白盒方法，能够访问目标大语言模型$\mathcal{M}$的内部结构。在本理论框架中，这一过程通过虚线标记的方式实现，即通过调取模型的词元空间来计算文本梯度等，从而对初始提示词P_0进行初步优化。

$$P_0'=P_0+\nabla_{P_0}\mathcal{LM} \tag{3}$$

其中，P_0'为根据文本梯度等模型内部信息初步优化后的提示词；∇_{P_0}表示对提示词P_0的文本梯度；$\mathcal{LM}$为目

标语言模型的损失函数，用于衡量生成输出与期望结果之间的差距。

（3）提示词优化器。提示词优化器 $\mathcal{O}$ 是APE的核心组件。它采用不同的优化算法，如贪婪算法、遗传算法等。其主要作用是优化每个对应的“输入提示词-输出结果”对，以实现得分函数 $\mathcal{F}$ 的最大化。此优化过程旨在通过迭代调整，提升目标大语言模型 $\mathcal{M}$ 的输出质量。

$$P^{*}=\mathcal{O}(P_{0}',\mathcal{D}_{\text{eval}},\theta,H) \tag{4}$$

其中，P^{*} 为最优提示词；$\mathcal{D}_{\text{eval}}$ 为评估数据集；θ 为目标大语言模型 $\mathcal{M}$ 的参数；H 为优化器的超参数集。

优化器通过多次迭代调整提示词，使得分函数 $\mathcal{F}$ 最大化：

$$P_{k+1}=\mathcal{O}(P_{k},\mathcal{S}_{k};H) \tag{5}$$

$$\mathcal{S}_{k}=\mathcal{F}\mathcal{M}(P_{k};\theta),\mathcal{D}_{\text{eval}}) \tag{6}$$

其中，P_{k+1} 为第 k 次迭代的提示词；$\mathcal{S}_{k}$ 为第 k 次迭代的得分表。

（4）目标大语言模型。目标大语言模型 $\mathcal{M}$ 是APE工具作用的对象，通常包括如GPT-4等主流大语言模型。在APE实现过程中，实验一般通过API接口等方式调用目标模型进行交互。

（5）评估数据集。评估数据集 $\mathcal{D}_{\text{eval}}$ 包含标准的“输入提示词-输出结果”对，并提供相应的评估基准，用于评估目标大语言模型输出结果的质量。这些数据集为模型优化提供了标准化的评分依据。

（6）最优提示词得分表。APE系统维护一个最优提示词得分表，记录每个“输入提示词-输出结果”对的得分情况。该得分表用于追踪提示词优化的进度和效果，确保优化过程朝着正确的方向发展。

$$\mathcal{S}=\{(P_{k},\mathcal{F}(\mathcal{M}(P_{k};\theta),\mathcal{D}_{\text{eval}}))\}_{k=1}^{K} \tag{7}$$

（7）迭代优化过程。根据得分表中的结果，系统保留高分提示词，并将低分提示词重新输入优化器，继续优化其对应的输入提示词。通过不断迭代优化，系统持续记录优化结果，直至输出结果收敛或满足最大迭代次数 K。最终返回最优的提示词 P^{*} 作为APE的输出结果。

$$P^{*}=\lim_{k\to K}P_{k} \tag{8}$$

2.3 APE基于实现方式的APE分类

根据APE实现流程和优化技术，可将其分为下面四类：

（1）基于思维链的自动提示工程（APE based on chain-of-thought）。其在生成初始提示词时，按照思维链技术的模板进行构建（图3中①），并在后续的优化迭代阶段（图3中⑦）重点优化思维链的推理过程。

（2）基于类机器学习模型的自动提示工程（APE based on machine learning models）。其在优化过程中，使用强化学习等模型，或直接采用另一个大语言模型作为优化器（图3中③）。

（3）基于进化算法的自动提示工程（APE based on evolutionary algorithms）。通过迭代优化过程生成新的提示词，从而不断提升最优提示词的得分表（图3中⑦）。

（4）即插即用的预训练系统（plug-and-play auto prompt systems）。将APE的内部结构进行封装，提供便捷的综合应用方案，是对自动提示工程各类方法的有效整合。

2.3.1 基于思维链的APE

基于思维链的APE是一种通过模拟人类思考过程来引导模型进行推理的方法。思维链（chain of thought，CoT）提示方法通过在提示中加入“让我们逐步思考”这样的指令，鼓励模型分步骤进行推理，从而提高模型解决复杂问题的能力[19]。例如，在解决数学问题时，可以提示模型先确定所需的运算步骤，然后逐步进行计算。这种方法在处理需要逻辑推理的问题时特别有效，能够显著提升模型的推理能力[20]。

基于思维链的自动提示工程，即APE优化的优化算法能够通过优化思维链或自动生成思维链的方式来优化提示词。其实现方式主要有两种范式：一种是零样本思维链构造，即通过添加简单的提示词（如“Let’s think step by step”）来促进模型逐步推理；另一种是少量样本提示词构造，每个演示由一个问题和一个推理链组成，从而通过推理链引导至答案。第二种范式的性能优势依赖于手工制作的特定任务演示。

2.3.2 基于类机器学习模型的APE

机器学习是人工智能的一个分支，它使计算机系统能够利用数据和算法自动学习和改进其性能。其本质上是使用特定的算法模型进行基于数据的学习，并做出预测或决策，在一个训练集上进行参数调优，目的是实现在测试集上最优化结果[21]。

机器学习的做法是结果导向的，其不在乎结果的可解释性。由于大语言模型自身就充满着不可解释性，使用者与LLM交互的过程本身就类似于机器学习的过程。因此本文提出“基于类机器学习模型的自动提示工程”，其优化器的优化过程就是一个类似于机器学习的结果调优过程。这些实现方式的共同点都是需要在一个给定的初始提示词集合（初始提示词库）上训练，在评分数据库上最优化得分。其主要特点是只追求结果的最优化而不注重“为什么会得到这个结果”，即不注重其可解释性。

在机器学习研究领域，研究者常以“炼丹”这一比喻来形象地描述模型超参数选择过程所特有的复杂性、耗

时性、对经验的依赖性以及结果的不确定性。基于机器学习的自动提示词生成与优化，可被视为在大语言模型能力尚未完全成熟阶段对其进行辅助性“炼丹”的一种尝试。尽管与传统机器学习中难以解释的参数优化不同，提示词优化的输出是人类可读的自然语言，但其最终生成的提示词本身的有效性逻辑，有时也超出直觉理解的范围。例如，Zero-Shot CoT方法[22]通过添加“Let's think step by step”这一简单指令，成功引导模型生成思维链推理，而谷歌研究人员在OPRO方法[23]中发现了更具效果的提示“Take a deep breath and work on this problem step-by-step”，该提示将模型在GSM8K评分集上的准确率从71.8%显著提升至80.2%。此类案例表明，一个看似微妙的提示词改动便能带来模型性能的显著飞跃，其作用机制虽不易直观理解，但效果引人关注。

2.3.3 基于进化算法的APE

传统的APE主要在初始提示词数据库中做筛选优化，而不会产生新的提示词。而基于进化算法的APE，其主要特点是其优化器部件能够根据进化算法产生突变提示词，从而能够在迭代优化的过程中赋予APE产生新提示词的能力。

进化算法属于生物启发式算法的一类，通过模拟生物进化过程的优化算法，它们通过自遗传算法和差分进化等机制来迭代地改进候选解[24]。这些算法通常从一个随机的候选解种群开始，通过评估适应度函数来确定哪些解更有可能被保留和复制，然后通过交叉和变异操作产生新的后代，最终在多代迭代后找到问题的近似最优解。

将进化算法应用于大语言模型自动化提示工程中是合理的。一方面，有效的提示词设计往往需要大量的实验和调整，进化算法不仅能够自动化这一过程，还具有出色的准确性和快速的收敛性。LLM在这一过程中扮演进化算子的角色，生成新的候选提示词，通过模拟自然选择和遗传变异来迭代优化提示，从而提高LLM在特定任务上的表现和效率。另一方面，目前先进的LLM通常通过API进行交互，其梯度和参数是不可访问的，而进化算法是无梯度算法，可以在模型下游进行黑盒优化。更进一步来说，这种实现方式不仅能够节省人力成本，还能探索到人类可能难以想到的创新提示，增强模型的适应性和灵活性，在处理需要高度创造性和多样性的提示词设计时表现出色。

2.3.4 即插即用的APE系统

上述三种APE在其各自研究方式上表现出了一定潜力，但在实际应用中面临显著挑战：第一，效率欠佳，许多方法依赖大量人工标记的数据集，导致人力和计算资源的浪费；第二，灵活性不足，许多模型缺乏与任务和模型无关的适应能力，限制了其在不同场景中的应用；第三，有效性差，评估指标具有很强的时效性，使用过时的评估条件会使模型不能产生显著改进，并且缺乏来自人类用户的反馈；第四，消耗资源很大，现有模型收敛到高质量提示词需要时间和计算资源开销往往都很大。这些源于大模型系统训练的内生问题限制了APE在实用领域的发展。

即插即用系统（plug-and-play systems）因其在不同工作流程中的模块化和易集成性而受到重视，这些系统允许快速灵活地增强功能，轻松添加或替换新的处理模块，而无需重新设计整个算法。由于它们能够无缝增强目标LLM的功能，随着人工智能技术的快速发展，对即插即用系统的需求也在不断增长[25]。

2.4 APE的分类拓扑

当前大语言模型的自动提示词工程的发展可以沿着两个主要维度进行划分。这两个维度分别是：

逻辑推理维度（logical reasoning dimension）：这个维度强调模型的可解释性，通过构建思维链等方法，使模型能够通过逻辑推理来生成提示。这种方法侧重于理解模型的决策过程，提高大模型的可解释性，以及如何通过透明的推理步骤来引导模型的行为。这种方式期望最大程度地理解模型行为以便控制和引导模型的行为。

效能导向维度（effectiveness-oriented dimension）：与逻辑推理维度不同，这个维度不关注模型的可解释性，而是通过类似于“炼丹”的过程，即反复实验和调整，面向最终结果来优化模型的训练过程。这种方法更注重实际效果，通过实验和迭代来寻找最佳的训练策略，以实现预期的输出。事实上，这种做法是依赖于资源密集型的训练和参数优化来实现的，这种训练往往是黑盒空间下，对抽象的提示词的词元空间进行调优的过程。

如图4所示，本文将自动提示工程的四个类别进行了简要区分。在这个框架中，每种实现方法的子类都可以根据其在逻辑解释性和效能导向性两个维度上的特征，分别位于相应的区域内。其中，基于思维链的实现方式通过逻辑链条合理化提示的生成，具有高度的逻辑解释性；即插即用系统则使用预训练的数据集，既不关注逻辑解释也不强调结果导向，因此可以视为框架之外的独立类别。基于类机器学习模型的实现方式侧重于通过计算资源的堆积与算法优化，逐步逼近最佳结果，忽视了可解释性，属于高度结果导向型；而基于进化算法的实现方式则综合运用了传统方法，并创新性地引入了提示词的突变，兼顾了逻辑解释与结果导向。

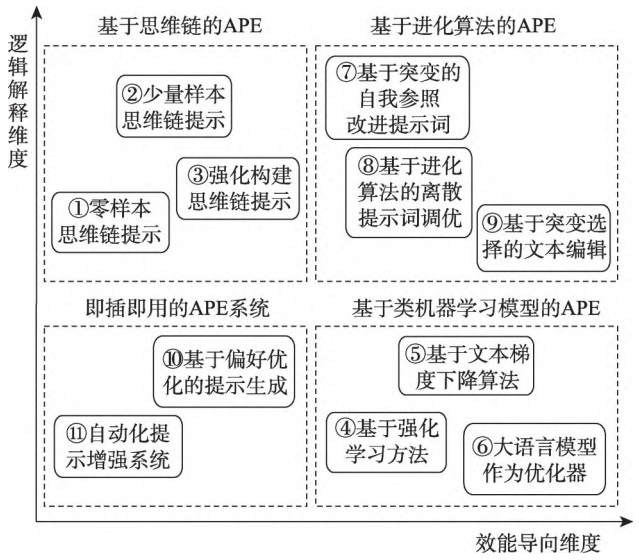

图4 APE的分类拓扑

Fig.4 Classification topology of APE

3 基于实现方式的APE分类

本章将详细介绍4类实现方式的19个算法实例，并给出相关做法的代表性论文研究。需要声明的是，一是，由于现在APE正处于研发阶段，研究者主要关注的还是各种算法的准确性、任务泛化性等基础性能指标，尚未出现成熟的商业量化指标。二是，由于不同种类的实现方式所擅长的任务类型有所不同，如基于思维链的APE善于数学、符号和逻辑推理，基于类机器学习的APE擅长常识推理和文本任务，进化算法类型的APE则以创新类文本任务见长。故此处仅针对19种具体算法给出笼统的实用性对比，具体的性能对比见本章后续小节。表2中的3个维度分别是：是否是黑盒优化方式，即不需要目标模型内部结构，可面向API使用；是否具有任务迁移性，即可面向不同类型的提示词优化或生成任务；是否具有目标模型迁移性，即该算法可以面向不同类别的大语言模型使用。

3.1 基于思维链的APE

本节首先介绍思维链技术及其常用变体等相关技术细节，后续部分具体介绍使用思维链技术实现APE的主要方式，最后对比总结这些实现方式的性能和效果。

3.1.1 思维链与泛思维链

思维链技术由谷歌团队提出[19]，其核心思想是通过精心设计提示词，帮助语言模型模仿人类的思维过程。通过模拟人类的逐步推理，语言模型能够更好地执行复杂的任务，如数学推理、常识推理和符号推理。思维链的关键在于，它将复杂问题拆解为更小、更易处理的子问题，并通过生成中间推理步骤逐步解决这些问题。

表2 19种APE算法可用性对比

Table 2 Usability comparison of 19 APE algorithms

实现方式	黑盒优化	任务迁移性	目标LLM迁移性
Zero-shot-CoT	√	√	√
Auto-CoT	√	×	√
Active-Prompt	√	×	√
Automate-CoT	√	×	√
AutoReason	×	×	×
RLPrompt	√	×	√
TEMPERA	√	×	√
ProTeGi	√	√	×
RePrompt	√	√	×
APEer	√	×	×
OPRO	√	×	×
PE2	√	×	×
APS	√	×	×
PB	×	√	√
EvoPrompt	×	√	√
GrIPS	√	√	√
LongPRO	√	×	√
BPO	√	√	√
PAS	√	√	√

具体而言，思维链的目标是赋予语言模型逐步推理的能力，即通过一系列连贯的中间步骤最终得出答案。其主要特点包括：(1)辅助逻辑推理。将多步骤问题分解为中间环节，帮助模型进行复杂推理。此方法可能需要为更多推理步骤分配额外的计算资源。(2)可解释性。通过观察中间推理步骤，提供对模型决策过程的透明度，便于发现和修正推理错误。(3)上下文利用。通过在少量示例中引入思维链序列，模型能在现有语言模型中自然激发出思维链推理。

随着研究的深入，思维链技术逐渐成为提升大语言模型在复杂推理任务中表现的重要工具[26]。除了数学推理，思维链也被证明在常识推理任务中有效。然而，原始的链式结构虽然适合简单的线性推理，但在某些复杂任务中表现不佳。为此，研究者提出了多种思维链变体，如思维树(tree of thought，ToT)、思维图(graph of thought，GoT)和思维图解(diagram of thought，DoT)等，这些变体统称为“泛思维链(XoT)”。

(1)思维树(ToT)

思维树通过显式构建树形结构并引入树搜索算法，为大语言模型提供了一个强有力的推理框架[27]。其核心思想是将复杂推理过程拆解为多个子问题，并以树状结构组织这些子问题，使模型能够探索不同的推理路径，并执行回溯、自我评估和剪枝等操作。这些特性使

得树结构变体在处理复杂任务和规划任务时具有显著优势,能够有效提升模型性能。

思维树的一个关键优势在于,它能够模拟人类的决策过程,通过树状结构进行规划。与单路径推理方法不同,思维树能够评估并选择多条潜在路径,进行前后向探索。例如,在数学问题求解中,思维树能评估不同解题步骤,并选择最有可能通向正确答案的路径。此外,思维树框架支持多种搜索算法,如广度优先搜索(breadth-first search,BFS)和深度优先搜索(depth-first search,DFS)。BFS在每一步维护一组最有可能的状态,而DFS则沿最有希望的路径深入探索,直到找到答案或评估器判定无解。这种灵活性使得思维树能够根据问题的特性,调整搜索策略,从而适应不同任务和问题域。

(2)思维图(GoT)

相较于树结构,图结构引入了更为复杂的拓扑结构,使得模型在处理复杂问题时能够模拟人类的思维网络,其中不仅包括一系列的想法,也包括这些想法之间的复杂关系。因此思维图结构能够进行自我修复,并根据图的拓扑结构进行信息聚合[28]。

与传统的链式结构或树结构相比,图结构提供了更高的灵活性和表达能力,能够捕捉到更丰富的逻辑依赖和信息流动[29]。在处理复杂任务时能够展现更优秀的性能,因为它能够更好地模拟人类解决复杂问题时的思维过程。

然而,尽管图结构变体在理论上具有显著的优势,但在实际应用中也面临着挑战。如同树结构变体一样,图结构变体在任务选择上存在局限性,需要针对特定任务设计具体的提示指令,这限制了其广泛应用。此外,图结构的复杂性和成环性可能导致多条不同的推理路径,进而带来更高的推理成本,影响模型推理的效率和可扩展性。

(3)思维图解(DoT)

姚期智院士领衔的团队[30]提出了一种名为思维图解的新框架,这是一个新颖的人工智能推理框架,它通过在单一模型内构建有向无环图(directed acyclic graph,DAG)来模拟大语言模型中的迭代推理过程。与传统的将推理表示为线性链或树的方法不同,思维图解将命题、批评、改进和验证组织成一个有凝聚力的无环图结构,允许模型在保持逻辑一致性的同时探索复杂的推理路径。每个节点对应一个已被提出、批评、改进或验证的命题,使LLM能够通过自然语言反馈迭代地改进其推理。

思维图解利用自动回归下一令牌预测和特定角色的令牌,促进了提出想法和批判性评估之间的无缝过渡,提供了比二进制信号更丰富的反馈。此外,思维图解框架通过拓扑斯理论进行了形式化,提供了一个数学基础,确保推理过程的逻辑一致性和健全性。这种方法增强了单个LLM内的训练和推理过程,消除了对多个模型或外部控制机制的需求。

(4)思维缓冲区(buffer of thought,BoT)

思维缓冲区框架由北京大学、加州大学伯克利分校和斯坦福大学的研究团队提出[31],旨在显著提升大语言模型在推理任务中的准确性、效率和鲁棒性。该框架引入了两个核心组件:元缓冲区(meta-buffer)和缓冲区管理器(buffer-manager)。元缓冲区是思维缓冲区框架的核心,存储一系列高级思维模板,这些模板来源于各种任务的解决经验。每当遇到新问题时,系统会检索相关模板,并根据具体情况进行适应性实例化,从而实现高效推理。此过程模拟了人类在面对新问题时,如何借助既有经验快速做出反应。缓冲区管理器则负责动态更新元缓冲区,随着任务的推进不断增强系统能力。这一动态更新机制使得思维缓冲区能够持续学习、适应新问题,确保系统的可扩展性和稳定性。

思维缓冲区框架在多个推理密集型任务中表现出显著的优势,特别是在提升准确性和推理效率方面。与传统的思维树方法相比,思维缓冲区显著降低了计算成本,同时提升了推理速度。这一框架不仅在处理复杂推理任务时展现出更高的准确性,还具备更强的泛化能力,能够适应多样化的复杂场景。此外,思维缓冲区在鲁棒性方面也表现卓越,能够有效应对输入变化和不确定性,从而增强了模型的整体稳定性和可靠性。

3.1.2 零样本思维链提示

Kojima等人[22]提出了一种基于零样本模板生成推理思维链的自动提示方式,即零样本思维链提示(zero-shot-CoT)。这种提示词生成方式本质上是与任务无关的,因此其不需要逐步的小样本示例,并且可以使用单个模板在广泛的任务中促使模型产生适用性推理。

该方法的核心思想非常简单:在每个答案前添加“Let's think step by step”或类似的文本,以促使模型逐步推理。这与之前需要手工操作的少量样本方法形成鲜明对比。结果表明,使用单一提示模板的零样本思维链提示词方法,在没有手工制作的少样本示例的情况下,显著优于传统的零样本LLM的性能。相比于无思维链的提示词,其在推理数据集的表现有显著提升。这一发现揭示了LLM在零样本推理任务中未被充分利用的潜力,表明通过简单的提示可以提取出高级、多任务的广泛认知能力。

总的来说,这种方式将思维链提示词方法应用到

LLM中，实现了基于零样本模板的自动思维链推理提示，这一开创性的想法成为了该领域后续研究的基石。

然而，零样本思维链方法在处理复杂推理任务时仍存在一定局限性。为此，来自土耳其伊兹密尔理工学院的研究者们提出了一种自动生成推理链的系统“Auto-Reason”，旨在增强LLM的多步隐式推理能力[32]。其核心思想是将隐含的复杂查询分解为一系列明确的问题，由较强模型（GPT-4）生成详细的推理轨迹，然后由较弱模型（GPT-3.5）基于这些轨迹得出答案，从而自动生成推理过程。

AutoReason通过以下步骤实现其功能：（1）初始推理生成。针对用户的零样本查询，利用强模型生成初步推理过程。通过提示模型“让我们一步一步地思考”，引导其输出详细的推理链。（2）问题分解。从初始推理中提取关键步骤，形成一系列子问题。此过程将复杂的隐含查询转化为更易处理的明确问题，便于后续模型解答。（3）子问题解答。使用相对较弱的模型逐一回答子问题。此方法不仅提升了模型的推理能力，还减少了对高性能模型的依赖，提高了系统效率。（4）答案整合。汇总各子问题的答案，形成对原始查询的完整解答，确保最终答案的连贯性和准确性。

AutoReason实现了从零样本提示自动生成针对每个查询的特定推理过程，避免了手工制作示例的繁琐工作。这种方法不仅提高了LLM在多步推理任务中的性能，特别是在StrategyQA等数据集上表现出色，还增强了模型的可解释性。

3.1.3 少量样本思维链提示

自动思维链（automatic chain of thought，Auto-CoT）技术是一种更高级别的思维链技术，它通过简单的提示促使模型自我思考，从而自动产生完整的思维链。LLM作为零样本推理的结果并不完美，仍可能会在推理链中出错[33-34]。为了减轻推理链错误的影响，Zhang等人[35]通过上下文学习等一系列分析证明了问题的关键是演示示例的多样性。基于此，研究者们提出了一种自动思维链提示方法Auto-CoT。Auto-CoT通过多样性抽样问题并生成推理链来构建演示。

Auto-CoT通过自动生成演示来提高链式推理的性能。研究者提出了两种方法：基于相似性检索和基于随机采样。它们使用Sentence-BERT对问题进行编码，并通过GPT-3生成推理链。实验结果显示，基于相似性检索在没有人类注释的情况下表现较差，但在有人工标注的情况下，表现优于手工设计的演示。这表明基于相似性检索的方法在有人工注释时有效，但在没有标注的情况下，自动生成的推理链表现欠佳，尤其是在问题多样性和采样挑战未得到充分解决时。因此，提升自动链式推理方法的有效性，需要更深入理解其面临的挑战。此外，研究者们还详细讨论了Auto-CoT面临的挑战，包括如何通过聚类分析来增加问题多样性，以及如何使用简单的启发式规则来减少错误推理链的生成。

Auto-CoT在提示词构造示例中使用了多样性选择策略，而多样性的策略带来了信息量的降低。因此能够通过提高样本的质量来提高提示词的质量。Diao等人提出了一种新方法——主动提示词（Active-Prompt）思维链构建，实现了半自动化的基于思维链的提示词构造[36]。这种方法借鉴了基于不确定性的主动学习相关问题的构想，引入了几个指标来描述不确定性，以便选择最不确定的问题进行注释。

主动学习（active learning）是机器学习中的一种策略，它允许模型在训练过程中主动选择数据，从而实现更高效的模型训练。与传统的监督学习相比，主动学习能够显著减少人工标注数据的成本，提高模型的泛化能力。由于不同的提示词演示示例会影响到最终生成提示词的质量[37]，而主动学习通过不确定性方法来找到更有价值的样本示例。在特定的任务中，输入提示词使用人工设计的思维链推理注释，LLM通过提示词中的推理过程学习，来增强输出的质量[38]。

Active-Prompt框架在实现上与主动学习的过程类似，分为四个阶段实现：度量阶段、选择阶段、标注阶段和推理阶段。首先在度量阶段利用零样本思维链和少量样本思维链方法对训练集中的问题进行多次采样，以计算答案的不确定性指标。随后，在选择阶段，基于这些指标，框架筛选出最不确定的问题以供人工标注。在标注阶段，专业人员对这些问题进行细致的标注，为模型提供了宝贵的训练数据。最终，在推理阶段，模型使用这些新标注的示例来改进其推理能力。此外，他们还通过进一步分析不同不确定性度量、池大小等条件，证明该方法的有效性。

总的来说，Active-Prompt将基于不确定性的主动学习相关思想引入思维链提示之中，为思维链演示示例的制作提供了新的方式，为基于思维链的APE技术开创了新的思路。

3.1.4 强化构建思维链提示

上面两种少量样本思维链提示词方式致力于使用高质量的提示词，而这一理论的前提是需要有高质量的可用人工标注数据，其在有限少量数据集上表现出高人工依赖性。因此，Shum等人[39]提出了强化构建提示思维链，其命名为“Automate-CoT”方法，实现了在不依赖人工设计的条件下，通过自动化的方法生成思维逻辑链。

Automate-CoT通过有限的标注数据集自动生成推理链，并利用方差减少策略梯度优化链的选择，从而实现最优提示策略。具体步骤包括：(1)自动生成逻辑链。从少量标注数据中自动生成初始推理链，减少人工标注成本，提升模型泛化能力。(2)逻辑链剪枝。筛选高质量推理链，剔除低质量选项，生成候选链池。(3)强化学习优化。通过方差减少策略梯度方法选择最佳逻辑链组合，优化推理效果并加快收敛速度。

结果表明，Automate-CoT方法在多种推理任务以及非推理任务中均获得显著的性能提升。与主动学习策略相比，这种完全自动化的方法具备更强的适用性和通用性，有助于降低人工干预的需求，并在不同任务上实现推理性能的提升。这一策略展示出在未来生成式AI模型优化方面的广泛应用潜力。

3.1.5 小结

图5为思维链技术以及在APE中应用的思维链技术的时间发展脉络。表3展示了5种基于思维链技术的APE在多个典型测试数据集上的平均准确率。这些数值是评估不同方法性能的核心指标，准确率越高代表该方法在相应任务上的解决问题的能力越强。具体到数据集，GSM8K/AQUA数据集主要是数学/逻辑/推理问题，StQA数据集主要是常识推理类问题，LETTER数据集则主要与自然语言相关。

总体来说，基于思维链的自动化提示工程技术的发展具有明显的时间顺序，从初始阶段的依靠人工示例的提示逐渐扩展到更自动化的零样本提示和自动使用多样提示案例的自动化提示，再到更具适应性的主动提示、强化学习驱动的自动构建提示和使用强模型推理能力的自动推理思维链。这些方法有明显的时间发展顺序和发展逻辑：

(1)自动化程度。早期方法更依赖人工设计和注释，而后续的Automate-CoT到AutoReason逐渐减少了人工干预，提升了自动化水平。

(2)数据需求。随着方法的演进，对标注数据的依赖显著降低。主动提示和自动提示技术通过不确定性采样和多样性策略减少了手工标注需求。

(3)适用范围。最早的手工构造提示示例方法多适用于特定任务，而零样本提示及其后续改进方法在广泛任务中表现优异，展现了更大的普适性。

未来的趋势在于进一步降低对数据的依赖，朝向弱监督和无监督方向发展。随着模型自动化生成和选择提示词的能力提升，基于思维链的自动化提示工程技术有望在更广泛的多任务场景中实现高效的应用。

3.2 基于类机器学习模型的APE

图6展示了类机器学习模型APE实现方式的时间发展脉络，本节将详细探讨这8种APE的技术细节。这些方式的主要区别在于初始提示词库的生成方法和优化器使用的优化算法，对于开源大模型，研究者可以利用模型的内部状态或梯度来训练额外的参数，从而生成更具针对性的初始提示词。基于此，可以借鉴强化学习的方法，将提示词设计视为一个序列决策问题，通过逐步调整来寻找最优的离散提示词。而对于闭源模型，由

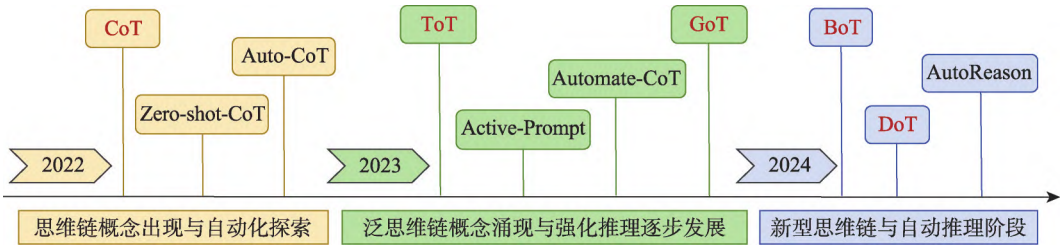

图5 思维链技术及其在APE领域的时间发展

Fig.5 CoT and its timeline in APE field

表3 基于思维链的APE能力对比

Table 3 Ability comparison of CoT based APE

实现方式	是否需要人工数据标注	平均准确率/%			实现原理
		GSM8K/AQUA	StQA	LETTER	
Zero-shot-CoT	否	40.7/33.5	54.8	—	思维链推理
Auto-CoT	是	47.9/36.5	65.4	59.7	多样性选择提示词构造
Active-Prompt	是	73.2/50.3	76.9	67.7	主动学习高质量提示词
Automate-CoT	否	49.7/37.9	—	58.9	强化学习和剪枝策略
AutoReason	否	—	91.6	—	强模型指导弱模型推理

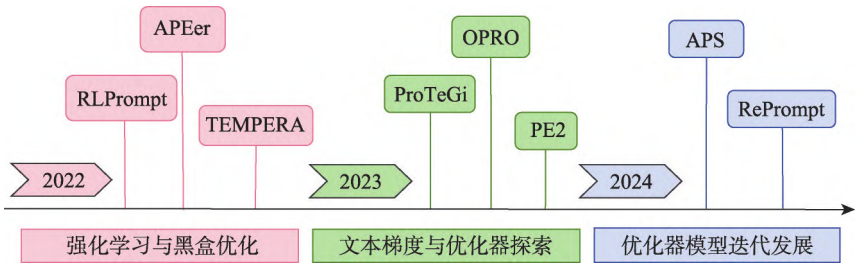

图6 基于类机器学习模型的APE时间发展

Fig.6 Timeline of APE based on machine-learning models

于只能通过应用接口访问，研究者需要采用不同的方法，其中一种可行的方案是使用文本梯度下降来寻找最佳提示。需要特别注意的是，这里的“文本梯度”借用了数值梯度的概念，但与模型内部的梯度不同，它并不直接影响模型参数的更新，而是用于识别当前提示词的“错误”或不适用之处。更进一步，研究者们将目标LLM视作一个黑盒，通过另一个LLM作为优化器，将提示词作为待优化的目标进行黑盒模型的参数调优，通过多轮迭代训练，以生成最优的提示词。

3.2.1 基于强化学习

强化学习（reinforcement learning，RL）通过与环境的交互来学习如何采取行动，以最大化累积的奖励。与监督学习不同，强化学习没有明确的标签数据，而是通过探索和试错的过程，逐步发现哪些行为能够带来更多的奖励[40]。强化学习的目标是通过不断调整策略，选择能够最大化长期回报的行动序列。

在LLM的提示词设计中，强化学习尤其适用。提示词的设计可以视为一个逐步调整的过程，其中每个提示词都可能影响到模型的响应质量。因此，强化学习能够将提示词生成视为一个序列决策问题，智能体通过历史提示词的选择与调整，逐步优化最终提示，从而最大化模型的输出质量。

Deng等人[41]提出了一种基于强化学习的离散文本提示优化方法RLPrompt。该方法的核心是一个参数高效的策略网络，经过奖励训练后能够生成最优的离散提示。为了提高训练效率，RLPrompt引入了有效的奖励稳定化技术，旨在应对大语言模型环境中奖励信号的复杂性和随机性。RLPrompt在少样本分类和无监督文本风格迁移任务中，展现出了优于现有微调或提示方法的性能。一个有趣的发现是，尽管最优提示往往是不符合语法的乱语，但这些提示在不同模型之间具有较好的可转移性，能够保留相当程度的性能，这表明大语言模型的提示不一定遵循人类的语言模式。

总结来说，RLPrompt利用强化学习优化离散文本提示，通过训练一个策略网络生成能够提高预训练语言模型在特定任务上表现的提示。这种方法在不同任务和模型上展现了优越的性能。然而，这种方法的局限性在于，它依赖于静态提示，在测试阶段只能根据输出结果进行反馈调整，无法实时适应输入的变化。

与之不同，测试时编辑（test-time editing）是一种动态调整提示词的方法，它允许根据每个输入的具体情况实时修改提示词，从而提高模型的输出质量。这种方法类似于考生在考试中，根据每道题的特点灵活调整解题策略，从而更有效地解答问题。

来自UC Berkeley的Zhang等人[42]提出了一种基于强化学习的测试时编辑的框架（test-time prompting via reinforcement learning，TEMPERA），结合了强化学习技术，利用测试时编辑构建查询依赖的提示。与以往研究不同，TEMPERA通过在编辑过程中融入查询相关信息，提升了系统的灵活性和性能。该框架允许代理在测试阶段对初始提示进行编辑，从而生成更适合特定查询的最终提示。

在实现方面，TEMPERA将编辑过程建模为一个马尔可夫决策过程，其中状态包括初始提示和查询。在每个时间步，RL代理从动作空间中选择编辑动作，利用预训练语言模型的最后隐藏状态作为状态表示。这种设计确保了编辑过程能够捕捉到上下文信息，进而提高提示的相关性。

动作空间设计包含多种编辑技术，如指令编辑、上下文示例的排列或交换等。这些灵活的编辑方式使得代理能够根据当前查询的需求，选择最适合的编辑方式，从而优化最终输出。奖励机制则通过计算当前提示与前一个提示之间的分数差异，引导代理朝着优化最终提示的方向学习。

TEMPERA采用基于注意力的策略架构，重点关注相关的编辑候选项，使得代理能够根据与查询的语义相似性进行加权调整。通过这些设计，TEMPERA在多个下游任务中展示了优越的数据效率和效果，凸显了动态适应提示的重要性，并强调了查询依赖提示在自然语言

处理中的优势。

3.2.2 基于文本梯度下降

在机器学习领域,数值梯度下降通过计算损失函数的梯度并更新模型的参数,逐步减少模型输出与真实标签之间的误差。提示词优化领域的文本梯度是借鉴数值梯度下降方法,在文本空间通过模拟梯度下降来优化提示词。可以理解为不合理提示词的"哪些地方最不合理",或是改进错误提示词的方式,以最大化优化模型输出结果。微软Azure AI团队提出了一种基于文本梯度的提示优化框架(prompt optimization with textual gradients, ProTeGi),用于自动优化LLM的提示词[43]。

ProTeGi是一种简洁且高效的自动化提示优化方法。它通过使用"文本梯度"来反向修改提示。在每一步中,自动化提示优化器会为当前错误生成三个原因,并根据这些原因生成新的提示,从而提升LLM在具体任务中的表现。ProTeGi旨在减少人工试错的繁琐过程,自动改进大规模语言模型的提示,从而提高任务的性能。该方法不依赖复杂的模型训练或对LLM内部状态的访问,而是通过LLM生成的自然语言梯度和编辑策略进行优化,尤其适用于通过API交互的场景。具体来说,ProTeGi的优化流程如下:(1)生成文本梯度。首先使用小批量数据生成"文本梯度",即LLM对提示词的批评,指出任务表现不足之处。(2)提示编辑。根据生成的梯度,修改提示词,朝着纠正错误的方向优化,使提示更加精确和清晰。(3)束搜索和臂搜索优化。通过束搜索生成多个候选提示并筛选最优结果,进一步提升算法效率。这个过程视为"最优臂识别"问题,优化选择效率。(4)优化过程的稳定性。反复进行优化步骤,每次使用最新的优化提示生成梯度并调整,直到达到预期优化目标或无法进一步提升。

在多个自然语言处理任务上的实验表明,ProTeGi能够在原始提示基础上提升性能最高达31%。同时,ProTeGi大大减少了对API调用的依赖,提高了效率。总体而言,ProTeGi提供了一种高效且创新的APE解决方案。通过借鉴数值梯度下降的思想,利用文本梯度进行反向优化,ProTeGi能够在无需复杂训练或人工微调的情况下,显著提高语言模型在特定任务中的表现。其核心理念"将提示词视为自然语言文本的'黑盒'进行梯度优化"为后续研究提供了重要启发。该方法不仅减少了人工干预,还提升了任务执行的效率和准确性,尤其在数据稀缺和资源有限的场景中展现出巨大潜力。

然而,ProTeGi方式这种"重复试错"的行为局限性也较为明显,其没有充分利用LLM自身在上下文学习和推理上的能力。因此,由南加州大学团队提出的RePrompt(prompt via reflection with LLM)方法可以看作是对ProTeGi方式的扩展,旨在进一步提升LLM在更多推理任务中的表现。其能基于与LLM代理的交互历史优化提示指令,特别适用于需要复杂推理的任务,如旅行规划和编程辅助等[44]。

RePrompt采用迭代优化方式,逐步细化提示,而非直接调整模型内部权重。在实现上其工作流程包括以下几个步骤:(1)初步交互生成。LLM根据初始提示生成回答,并与外部反馈者互动,形成聊天历史。(2)总结与提炼焦点。分析交互历史数据,总结出任务表现中的关键问题或改进点,例如旅行规划中的预算约束。(3)提示优化。利用提炼出的焦点,使用另一个LLM优化当前提示,重点解决通用性问题,如预算计算问题。(4)迭代优化过程。通过多轮迭代更新提示,优化过程逐步收敛,提升任务表现。(5)最终提示生成与应用。输出优化后的提示,并在新任务中进行验证,确保其有效性。

结果表明,RePrompt能够显著提升LLM在复杂任务中的推理能力,尤其在旅行规划和编程辅助等任务中表现突出。与传统方法相比,RePrompt提高了任务的完成度,使推理过程更为高效。

总体而言,RePrompt作为基于交互历史的自动化提示优化方法,展示了其在提升LLM推理能力和减少人工干预方面的巨大潜力。尽管该方法依赖于数据质量和工具集,但未来的研究可通过扩展LLM工具集及引入智能机制进一步提升其性能,预计在多个领域的应用前景广阔。

3.2.3 LLM作为优化器

如果将目标语言模型视为一个"黑盒",而自然语言指令则作为调节该黑盒的工具。通过得分函数衡量目标语言模型在执行下游任务时的表现(即模型对提示词的响应),借鉴机器学习中的参数调优方法,尝试不同的提示词组合,以最大化得分函数,从而自动选择最优提示词。此方法的理论基础在于,大语言模型通常具有不可解释性,尽管它们能够处理各种自然语言指令,但这一处理过程对人类而言往往不够直观,指令的质量只能通过模型在具体任务中的表现来评估[45-46]。

Zhou等人[9]开创性地使用LLM作为优化器,认为"大语言模型是具有人类水平的提示工程师"。为了与自动提示工程(APE)加以区分,本文将这种做法命名为APEer(automatic prompt engineer)。这种实现方式的本质是用一个语言模型当另一个语言模型的提示词优化器。

初始指令生成通过自动化手段创建潜在的指令集合,以应对庞大的搜索空间问题。具体步骤包括:正向模式生成,给定"输入/输出示例",直接提示LLM推测

可能的指令；反向模式生成，使用具有填充能力的语言模型生成指令，提升生成过程的灵活性；定制化提示，根据具体任务设计提示模板，指导LLM生成符合任务要求的指令[47]。

生成初始候选指令后，通过评分函数筛选最优指令，为进一步优化指令的质量，APEer采用迭代蒙特卡罗搜索进行多轮优化：评估候选指令，根据评分函数筛选低分候选；生成新候选指令，基于得分高的指令，生成相似的新指令，进行局部探索；重复优化，持续迭代，直到达到预定的优化目标。APEer不仅适用于零样本学习，还可通过将生成的指令添加到标准的上下文学习提示中，提升少样本学习的性能。

总体而言，APEer方法提供了一种高效且创新的解决方案，通过将大语言模型作为优化器，减少了人工干预并显著提升了模型在多种任务中的表现。这一方法将提示词视为“黑盒”进行优化，突破了传统提示词设计的局限，为后续研究提供了新的思路和方向。采用大语言模型作为优化器的方法，已成为自动提示工程领域众多后续工作的基准模型。

在此工作基础上，谷歌DeepMind团队提出一种实现方式OPRO（optimization by prompting），其核心想法是通过让LLM本身作为优化器[23]，基于过往的迭代记录、优化目标，自己总结规律，逐步迭代提示词。这一方法与传统的基于机器学习的优化算法（如梯度下降、强化学习等）有所不同，是更加原生的实现方式，且整个过程在文本空间内完成[48]。其最大的特点是它不仅能优化传统的提示词，还提供了对数学问题的优化方法，例如线性回归、旅行商问题等。

OPRO框架的核心部分包括以下几个关键环节：（1）元提示词构建。结合包含历史迭代记录的解决方案-得分对和任务描述（包括基本信息、示例及优化目标），为后续优化提供明确指导。（2）优化器LLM。接收元提示词作为输入，分析当前提示及其得分，理解优化方向，并生成改进后的提示。通过多次迭代，逐步提升提示质量以保证结果稳定。（3）评估与记录。将生成的新提示交由评分LLM评估，根据任务执行情况打分，并将评估结果反馈给元提示词，形成闭环优化过程。优化在提示无法进一步提升或达到预定迭代次数时终止，最终返回得分最高的提示。在这个做法中，每个优化步骤都基于之前的优化结果生成新的提示，这样LLM能够通过不断迭代提升任务的精度。OPRO方法展现出LLM在提示优化上的显著优势，特别是在自然语言处理任务中的应用。

总体来说，通过结合历史迭代的元提示词和多轮优化机制，OPRO实现了高效的提示改进，提升了任务性能，并具有广泛的适用性，可扩展到多种优化任务中。并且，通过简单的提示优化能够快速生成高质量的解，甚至有时能匹配或超越传统启发式算法的表现[49]。然而，尽管OPRO在多种任务中表现出色，仍存在一些局限性。例如，LLM在处理大规模问题时，由于上下文窗口的限制，难以有效表示高维数据或复杂的优化景观。此外，对于复杂目标函数，LLM可能无法准确引导优化过程，使搜索过程收敛于次优的性能水平，限制了其发现更优解的能力。因此，尽管OPRO展示了LLM作为优化器的潜力，但在面对更复杂的优化任务时仍存在挑战。

上面两种做法的核心想法都是使用一个LLM作为提示工程师工作，那么是否可以对这个提示工程师LLM进行优化呢？在这种想法的指导下，PE2（prompt engineering a prompt engineer）方法被提出。其核心思想是通过对用于提示工程的LLM（即优化器LLM）进行提示工程，以不断改进和优化提示的生成过程[50]。这一方法的关键在于，通过精心设计和调整优化器LLM的元提示词，使其能够更好地进行提示生成和任务优化，从而提高整个提示优化流程的效果。

PE2的核心在于通过优化优化器LLM的元提示词来实现迭代优化，具体包括两个方向：提供详细的指令和上下文信息，以及融合常见的优化器概念。为了实现这两个目标，PE2设计了多个模块。首先，提供详细的指令和上下文的模块包括提示工程教程、两步任务指令、逐步推理模板和上下文规格说明。其中，逐步推理模板和两步任务指令在实验中被证明有效，能帮助优化器LLM更好地理解和优化提示。其次，PE2还引入了常见的优化器概念模块，如批处理大小、步长和优化历史。通过这一系列方法和策略，PE2能够在提示优化过程中实现更高效的迭代和更好的优化效果，从而提升LLM在各种任务中的表现。

值得注意的是，利用其他语言模型作为优化器通常需要消耗大量的计算资源，Cinnamon AI提出了一种新的自动提示选择方法（automatic prompt selection，APS）[51]，旨在减少计算开销并提升模型性能。APS通过结合聚类、生成和评估机制，能够自动选择最适合的提示，避免了复杂的优化过程，并通过提示评估器提高了提示的相关性，同时保持计算效率和多样性。

APS的实现分为三个主要步骤：（1）提示数据库生成。首先，利用“K-means”聚类算法对训练数据进行聚类，将相似问题和上下文归为一组，以共享相同提示。通过“Sentence-Transformer”对问题和上下文进行编码，生成候选提示并去重，最终形成一个控制规模的综合提

示数据库。(2)提示评估器训练。训练一个模型来评估提示的相关性。通过收集有效和无效提示,定义决策阈值 λ,使用偏好损失函数来优化评估器,从而高效识别最优提示。(3)提示排名。对于新的输入,提示评估器计算候选提示的相关性得分,并选择得分最高的提示。该方法还支持对提示进行排序,选出前 k 个最佳提示,以进一步提高结果的准确性和稳健性。

总体而言,APS通过提示生成、评估和排名机制的创新结合,显著提升了提示选择的效率和效果。在保证灵活性的同时,降低了计算开销和延迟,具有较强的适应性,适用于多种自然语言处理任务。引入训练好的提示评估器后,APS的全流程自动化策略不仅提升了提示的相关性,还能在多样性与稳健性之间实现良好的平衡。实验结果验证了该方法在零样本问答任务中的优越性,展现了广泛的应用潜力。

综上所述,本节介绍了四种方式,表4展示了它们在多个测试数据集上的平均准确率,准确率越高,代表该方法在相应任务上的相对能力越强。这些实现方式有明显的时间发展线和逻辑发展顺序。APEer开创性地使用LLM作为优化器,自动选择最优提示词;OPRO进一步增强了优化过程,通过LLM自身总结迭代规律,实现更智能的提示优化;PE2则在优化器LLM层面进行优化,通过改进元提示词提升整体提示生成效果;APS通过引入聚类和评估机制,提供了一种高效且计算成本较低的提示选择方法。

表4 LLM为优化器实现方式能力对比

Table 4 Ability comparison of LLM as optimizer

实现方式	平均准确率/%			创新点
	GSM8K	MultiArith	AQUA	
APEer	76.02	98.33	61.81	可自动生成迭代初始词
OPRO	80.20	95.30	54.30	根据历史上下文信息迭代
PE2	64.00	92.30	67.70	可批量高效处理
APS	81.49	100.00	64.57	计算开销低

3.2.4 小结

基于机器学习的自动化提示工程通过在初始提示词集合上进行训练,优化模型参数,从而生成最佳提示词。其优势在于高效性和自动化程度,通过强化学习和自然语言生成等技术,可以自我调整提示词,灵活应对不同的任务需求。此外,这种方法还能够在多轮交互中逐步优化提示词,提升任务性能,同时生成的结果是人类可理解的自然语言,更具可解释性。

然而,自动化提示工程的优化过程也面临一定的挑战。其依赖于初始提示词库的质量,如果初始库不够丰富或覆盖不到任务关键点,优化效果可能会受到限制。此外,优化过程的复杂性和不确定性也可能导致过拟合,影响最终结果的稳定性。虽然相较于传统的机器学习,提示词优化更具可解释性,但仍然需要大量的计算资源,特别是在使用强化学习、蒙特卡罗树搜索等计算密集型算法时,这也限制了其应用的广泛性。

3.3 基于进化算法的APE

基于进化算法的APE主要特点是能够在迭代过程中产生新的提示词,因此其能够在没有人工干预的情况下持续优化提示词,且能适应不同任务的需求,提升模型在特定领域的表现。更进一步来说,LLM本身可使用进化算法优化自身提示词[52]。本节主要介绍四种使用了进化算法的APE。

3.3.1 基于突变的自我参照改进提示词

谷歌的DeepMind团队提出了一种用于LLM自我参照自我改进的新方法提示词边界Prompt-Breeder[53],以下简称为PB。其通过给定一组原始风格任务提示词、待解决问题和突变提示词的问题描述,利用LLM进行变异操作,生成原始任务提示词和突变提示词的变体,并通过多代进化选择出适应特定领域的任务提示,图7给出PB生成原始提示词的一个示例。PB使用了五种不同类别的变异操作符来变异任务提示词和突变提示词。因此其能够逐步适应特定领域,同时变异提示通过自我参照的方式不断演化,变得更加有效。采用二元锦标赛遗传算法对变异后的提示进行选择,从而进化出更优的提示[54]。PB不仅能改进任务提示词,还能改进用于优化任务的突变提示词,且这个过程通过LLM实现,因此其实质上是LLM的自我参照的自我改进,并且在这个实现过程中不需要访问和改变LLM的任何参数。

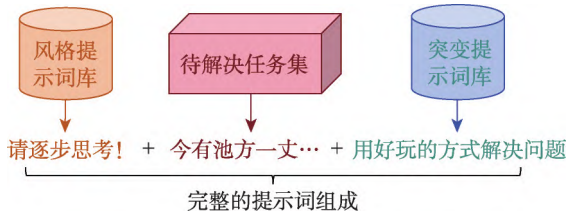

图7 PB生成提示词示例

Fig.7 Example of prompts made by PB

具体实现上,有以下主要步骤:(1)初始化。PB从一个初始种群开始,这个种群由原始任务提示词和突变提示词组成。(2)变异操作。PB使用多种变异操作符来探索提示策略的空间。这些操作符分为几个类别,包括直接变异、变异提示引导、超变异、拉马克变异以及提示交叉和上下文洗牌。(3)二元锦标赛遗传算法。PB运行

一个标准的二元锦标赛遗传算法框架。在这个框架中，从种群中随机抽取两个个体，比较它们的适应度，选择适应度更高的个体进行变异，并用变异后的个体替换适应度较低的个体。(4)适应度评估。为了评估任务提示的适应度，PB会在训练集的一个随机批次上评估其性能。(5)多代进化。PB在多代中重复上述过程，不断进化任务提示和变异提示，观察提示适应目标领域问题的能力，不断迭代以得到最优的结果。

结果显示PB在包括算术推理、常识推理以及仇恨言论分类等多个领域显示出了显著的性能提升，显示了其在自动化提示设计和自我改进方面的巨大潜力。然而，PB的实现涉及多种变异操作和自我参照机制，这增加了系统的复杂性和调试难度。尽管PB在自我改进方面展现了强大的能力，但仍然受限于固定的提示拓扑结构，尚未完全实现更开放的思维过程。未来的研究可进一步探索PB在更复杂任务中的应用及其优化空间。

3.3.2 基于进化算法的离散提示词调优框架

清华大学、微软研究院和东北大学提出了一种自动化提示词优化框架EvoPrompt[55]，其创新之处在于将LLM作为进化算子生成新的提示词，并通过迭代优化直到获得最优提示词。传统的进化算子通常独立改变标记生成候选解，忽视了标记之间的联系，而这种联系对离散提示词的连贯性和可读性至关重要。EvoPrompt通过结合LLM在自然语言处理中的专业知识和进化算法的优化能力，成功解决了这一挑战。

在进化算法的选择上，EvoPrompt使用了遗传算法(genetic algorithm，GA)和差分进化算法(differential evolution，DE)两种广泛应用的算法。遗传算法模拟生物进化过程，通过自然选择、交叉(重组)和变异等机制，从随机生成的候选解种群中逐代优化，最终收敛到近似最优解。差分进化算法是一种基于种群的优化方法，通过引入个体差异探索解空间，并不依赖于概率分布。DE通过变异、交叉和选择操作，利用个体间的差异生成新解，并保留适应度较高的个体[24]。GA因其模拟自然选择的能力在多种优化问题中表现优越，而DE则在连续空间的全局优化问题中表现出色[56]。结合这两种算法的优势，EvoPrompt在离散提示词优化中能够实现更好的性能和更快的收敛。具体实现过程如下：(1)初始群体。EvoPrompt结合人类设计的提示词和随机生成的多样化提示词群体，避免局部最优，提升初始群体质量。同时LLM生成的提示词也被加入其中。(2)进化过程。EvoPrompt利用LLM作为进化算子，结合特定的变异和交叉操作，生成新的候选提示词，通过遗传算法或差分进化算法进行优化。(3)更新。候选提示词在开发集上评估，优秀者被保留，实现类似自然界的适者生存。

结果表明，EvoPrompt在自然语言处理任务中展示了巨大的潜力。作者使用该框架对开源的Alpaca-7B和闭源的GPT-3.5模型进行了提示词优化，开发集上表现最佳的提示词在测试集上也取得了显著的提升。在语言生成任务中，EvoPrompt同样表现突出，特别是在对话总结和文本简化方面，均取得了显著进展。

3.3.3 基于突变选择的文本编辑

来自UNC的研究团队提出了一种创新的提示优化方法，基于编辑的无梯度指令搜索(gradient-free，edit-based instruction search for prompting，GrIPS)。该方法采用了进化算法的思想，以通过指令编辑和评估逐步优化大语言模型的提示[57]。具体而言，GrIPS的优化过程包括以下几个关键步骤：首先，从初始提示出发，通过一系列编辑操作(如替换、删除、重组词语等)生成多个候选提示。每个候选提示都会被输入到预训练的大语言模型中进行评估，评估的标准通常是模型在特定任务上的表现，如准确性或生成质量。然后，根据评估结果，选择那些表现最好的提示进行下一轮编辑和优化。整个过程类似于自然选择，通过多代的“演化”逐渐逼近最优提示，而这一切不依赖于对模型参数的微调或梯度计算。

结果表明，GrIPS方法能够在不需要梯度信息的情况下，通过一系列编辑和选择操作优化提示，显著提高了LLM的任务性能。该方法不仅提高了提示优化的效率和灵活性，还大大降低了计算资源的消耗，展示了进化算法在自动化提示工程中的巨大潜力和应用价值。这种实现方式为APE提供了新的思路，未来可能成为提示优化领域的重要研究方向之一，尤其是在需要高效、低成本优化提示的实际应用中具有重要意义。

现有APE研究的往往是交互文本等较短的提示词，自动化提示工程的侧重方向也是优化一个或几个句子的短优化指令。长提示词往往具有高度的上下文相关性，其搜索空间非常庞大，因此现有的APE实现方式在面对长提示词时具有很大的挑战性。而有些复杂开放的领域任务通常需要长篇的提示，这些提示往往包含数百行和数千个词元，并且设计这些领域相关的提示需要大量的专业认识。

为解决这一问题，来自谷歌的研究团队提出一种面向长篇提示词的自动化提示工程(long prompt，Long PRO)。其目标是生成一个在语义上原始提示相似，同时实现性能提升的新提示[58]。同时，作者避免引入无法解释的标记，以保证提示词的可解释性。在具体实现上，文章将其分为了三个步骤：(1)建立搜索空间。通过限制搜索空间和分解长提示为 m 个句子，利用LLM在保持语义

不变的前提下改写句子，减少过拟合并提高变异效果。(2)选择搜索算法。采用贪婪算法，每轮随机优化一个句子，保留最佳提示。使用束搜索维护“top-k”池，避免局部最优，并结合遗传算法通过变异和交叉生成新解，提升收敛速度。(3)历史引导搜索。通过历史变异数据引导句子变异，提升有效性。利用Lin-UCB算法优化句子选择，同时引入随机性保持多样性。

尽管GrIPS和LongPRO解决的问题有所不同，但它们的优化逻辑有一定的共性，都强调迭代优化和变异选择。GrIPS通过进化算法对短提示进行优化，通过小范围的编辑变更来提升性能，而LongPRO则将这一优化思想扩展到长提示词的领域，采用更复杂的结构化方法来保证长提示的语义完整性和任务表现。从实践应用角度来看，GrIPS和LongPRO具有互补性：GrIPS更适用于短小且简单的提示优化，而LongPRO则是解决长文本提示优化中的难题，特别是在专业领域任务中，需要处理大量上下文信息时表现优异。

3.3.4 小结

作为将进化算法应用到大语言模型提示工程领域的代表，PB、EvoPrompt、GrIPS和LongPRO都展示了巨大的潜力。PB利用自我参照机制和多种变异操作实现了更加复杂的提示词优化过程，能够灵活适应更加挑战性的任务，尤其是在需要高度创造性和多样性的应用场景中展现出其优势。相比之下，EvoPrompt结合经典的进化算法，特别是遗传算法和差分进化算法，提供了一种高效且精确的优化方法，更适用于大多数常规任务。GrIPS方法则通过无梯度的编辑优化方式，引入进化算法中的“变异”和“选择”步骤，通过反复编辑和筛选候选提示，在无需计算梯度的情况下优化提示。它具有较高的灵活性，能够根据任务需求生成个性化的提示，并且避免了计算梯度和模型微调带来的高计算成本。LongPRO专注于长提示词的自动优化，通过分解句子和多轮迭代优化，解决了长提示词的庞大搜索空间问题。

对比来说，PB更适合处理任务复杂度较高、需要动态调整的场景，而EvoPrompt和GrIPS则在标准化流程和收敛性方面表现得更为稳健，尤其是GrIPS在优化效率和资源消耗方面展现了其独特的优势。LongPRO适用于特定任务的长提示优化。

将进化算法应用于APE，充分发挥了进化算法在无需梯度信息的优化能力，克服了传统优化方法的局限性。这种方法不仅能够在没有人工干预的情况下持续优化提示词，还能适应不同任务的需求，提升模型在特定领域的表现。随着LLM技术的快速发展，进化算法在提示词优化中的应用前景广阔，未来有望进一步推动智能化和个性化任务处理的进步，成为大规模自然语言处理系统不可或缺的一部分。

3.4 基于即插即用系统的APE

即插即用系统是综合使用了多种提示优化算法后，将其训练结果封装为可直接使用的模块化APE。从用途角度分析，其往往有较为成熟使用接口，属于APE领域商用化的代表。这种方式的主要优势在于在后续使用时无需再次训练，并且一般可以适配不同类别的目标LLM，在面对不同类别的提示词时，只需更换特定类别提示词的数据集即可使用。本节主要介绍两种较为成熟的基于即插即用系统的APE。

3.4.1 基于偏好优化的提示生成

来自清华大学和智谱AI的研究者们提出一种新颖的提示优化方法，基于偏好优化的提示生成提示词(black-box prompt optimization，BPO)，旨在通过一次训练实现反复使用[59]。与传统的提示工程和优化方法(如APE、APO、OPRO等)需要在训练集上进行优化、在验证集上选择最佳提示后再进行推理不同，BPO的核心是训练一个序列到序列模型，该模型能够直接从原始提示生成优化后的提示，便于即插即用，无需进一步训练。

在具体的实现上，其主要包含：(1)收集人类偏好数据。使用开源数据集，如OASST1和Chatbot Arena等，这些数据集的结构通常为(提示，选择的响应，被拒绝的响应)。(2)生成优化后的提示。使用LLM生成优化后的提示，通过两个主要步骤构建对。首先，LLM对比选择的响应和被拒绝的响应，识别其区别；其次，基于批判结果和原始提示，生成改进后的提示。(3)训练序列到序列模型。论文中选择llama2-7b-chat进行微调，以得到一个预训练的提示优化器。

BPO的优化结果不仅仅是无意义的语义改写，而且是对原始提示的有意义补充，类似于人类专家使用提示词工程指南的方式。这种优化行为之所以有效，是因为其训练数据基于人类的偏好数据，比较正例和负例，促使LLM倾向于补充必要信息，使得提示更加具体、准确，从而更符合人类的偏好。此外，BPO强调了对齐的概念，通过利用人类偏好数据，优化后的提示理论上可以生成更符合人类期望的响应，这也加强了其优化的有效性。

3.4.2 自动化提示增强系统

来自北京大学和百川AI的研究团队提出了一种即插即用式的自动化提示增强系统(plug-and-play prompt augmentation system，PAS)[25]。PAS具有高效性、高度模块化和易迁移等特性，这使得其能够跨不同平台和系统进行广泛的应用和改进。该系统的主要优势在于，它能

够无缝地增强现有LLM的能力，无需进行大规模的预训练。

PAS实现方法的核心是，先用广泛类别数据进行高效提示词数据库的构建，将其作为一个补充数据包，在使用目标LLM时直接调用，使用少量样本提示的方式，实现自动化提示词增强。本质上是一种预先优化知识的迁移与应用，因此在目标数据集上取得高得分的关键是其包含了相关类别的提示词。具体到实现过程中：(1)数据收集与选择。从LMSYS-1M和WildChat数据集中挑选高质量的提示数据，使用SimCSE模型去重，并通过HNSW聚类算法进行分组，确保数据多样性与质量。随后，利用模型对数据进行筛选和分类，最终获得约9 000个高质量提示词。(2)自动补充提示数据生成。设计基于少样本学习的自动化数据生成管道，利用精选的golden数据对每个类别中的提示进行学习，生成相应的补充提示。最终得到约9 000个“提示词-补充提示词”对。(3)数据选择与再生。评估并筛选每个“提示词-补充提示词”对，移除不符合标准的内容，并将其重新进入上一阶段进行优化。该过程反复迭代，直到数据集收敛，最终形成包含14个类别、约9 000个高质量“提示词-补充提示词”对的数据集。(4)生成PAS系统。使用数据集微调选定的LLM，使其具备自动补充提示词的能力。

这两种实现方式都不需要修改目标LLM的参数或权重，因此可以集成到开源白盒模型中，或通过公共API接入黑盒模型(如GPT系列)。表5是对两种实现方式在一些基础大模型上应用的平均准确率，准确率越高代表相对能力越强。其中Arena-Hard测试集是大模型研究组织Lmsys Org开源高质量大模型评估基准；Alpaca-Eval2.0测试集适合工业界快速验证模型在开放域任务中的实用性。综合来说，PAS在二者上的表现略优于BPO。

表5 BPO和PAS能力对比

Table 5 Ability comparison of BPO and PAS 单位：%

基础模型	Arena-Hard测试集上平均准确率		Alpaca-Eval2.0测试集上平均准确率	
	BPO	PAS	BPO	PAS
GPT-4-turbo	76.60	73.54	54.65	62.58
GPT-3.5-turbo	15.90	18.02	10.25	33.11
Qwen2-72B	44.40	47.91	31.25	40.59
LLaMA-3-70B	45.20	46.30	38.92	43.17
平均得分	**45.53**	**46.44**	**33.77**	**44.86**

3.4.3 小结

总体来说，即插即用系统BPO和PAS通过增强输入提示，提供了一种提升LLM性能的机制。这种方法不仅成本效益高，而且效率出众，能够更有效地利用计算资源，加速增强型语言模型在多样化应用中的部署。表6给出了两种方式的特征对比。

表6 BPO和PAS特征对比

Table 6 Characteristic comparison of BPO and PAS

方法	代表公司	目标模型	训练数据对	优势特征
BPO	智谱AI	黑盒或白盒	14 000	可根据偏好修改
PAS	百川AI	黑盒或白盒	9 000	通用且不需要人类劳动

然而，虽然作为较为成熟的商用系统，但不可忽视的是BPO和PAS二者本质上是封装好的预训练数据集，因此保证其数据集的高质量和时效性至关重要。此外，可以通过增加基于人类反馈的学习来优化系统的数据集，也需要适当的数据量来防止过拟合现象。

4 挑战与展望

尽管APE为LLM的优化提供了有效工具，现有的研究也已经充分说明了该领域的巨大潜力，但其也展现了诸多待解决的问题。目前的方法普遍存在以下问题：缺乏对任务的深层理解、依赖固定模板、上下文理解不足、缺乏人类反馈和校正等。为克服这些问题，需要在自动化提示生成过程中引入更加智能、灵活和个性化的技术，同时加强与人类反馈的结合，以推动自动提示工程的进一步发展和广泛应用[60-61]。在此背景下，针对每一类APE，提出以下挑战。

(1)新兴技术的深度融合与探索。尽管思维链和泛思维链等新兴技术已初步展现其在复杂推理与任务分解中的应用潜力，相关理论也在不断创新，但其在自动提示工程领域的整合与优化仍处于起步阶段。随着并行化思维链[62]和思维链压缩[63]等优化技术的不断深入发展，这些技术将显著提升提示生成的逻辑性与适应性，为解决跨领域复杂任务提供更加通用的解决方案。

(2)传统算法的创新应用与再生机。经典机器学习算法在新的技术背景下焕发了生机，如贝叶斯优化在提示选择中的引入，已证明其在搜索效率与性能稳定性上的优势[64-65]。未来，深度学习与传统算法的融合将进一步拓展自动提示工程的边界。例如，通过结合强化学习优化提示搜索空间，可以显著提高复杂任务场景下的提示生成质量。同时，传统统计方法与解释性分析工具的结合有望揭示模型行为的内在机制，为提示工程提供理论支持。

(3)基于进化算法的创新探索。进化算法及其他启发式搜索方法为自动提示工程提供了更大的探索空间，带来了诸多超出预期的创新成果。未来的研究将不仅

关注这些方法在提升性能上的直接应用,更会聚焦于其对提示生成过程的启发性贡献,例如通过多目标优化方法平衡提示的复杂性与可解释性,进而拓宽提示工程的应用边界。

(4)即插即用技术的普及化与商业化前景。即插即用式提示生成技术因其易用性、灵活性和高效性,已成为最具商业化潜力的研究方向之一。尽管现有技术仍面临性能稳定性与应用广泛性方面的挑战,但其直观的用户体验使其成为面向非专业用户的理想选择。未来,通过结合模块化设计与人机协作优化,即插即用技术有望发展为普适性强、可轻松部署的成熟系统,为商业化应用奠定坚实基础,并可能成为自动提示工程商业化的标杆案例。

在此基础上,展望未来。当前LLM技术已从语言文本领域向多模态、强推理、专业化应用方向深度拓展,其发展底座的成熟为APE的全面渗透提供了关键支撑。结合技术趋势与行业需求,APE的未来发展将呈现以下核心方向。

4.1 多模态大模型APE

在当前多模态大语言模型(multimodal large language models)飞速发展的背景下,APE迎来了前所未有的技术需求增长。随着LLM在多个领域的应用不断深入,尤其是在自然语言处理、计算机视觉、语音识别等技术的融合下,APE不仅在文本领域的重要性愈加突出,还逐步扩展到多模态任务的优化[66]。自动提示工程正日益成为下游应用领域的核心环节。这种趋势催生了对更高效、更智能的提示词优化技术的需求,以推动多模态大语言模型的性能提升与应用场景拓展。

多模态大语言模型凭借其能够同时处理文本、图像、语音等多种模态数据的能力,迅速成为人工智能领域的研究热点。在多模态场景下,尤其是在多模态对话系统中,系统需要同时处理来自用户的文本、图像或语音等信息,并生成相应的输出。因此,针对多模态环境的APE技术需求逐渐从传统的单一模态扩展至多模态环境[67-69]。

与传统自动提示工程聚焦于文本生成、通过静态提示词和优化算法改善模型输出的做法不同,在多模态环境下,提示词生成不仅需要考虑文本本身的特性,还必须综合考虑不同模态之间的关系与相互作用。例如,在图像描述和视频场景分析等多模态输入的情况下,提示词的生成不仅要优化文本提示,还需要结合视觉或音频特征,以增强模型对多模态内容的理解与生成能力。

因此,面向多模态模型的自动化提示工程仍存在很多亟待解决的问题。首先是模态对齐的偏差,由于视觉模型和文本模型的语义空间存在不对称性,可能会导致文本空间的“幻觉”问题扩展到视觉信息的逻辑割裂;其次是面向视频流等时序数据的语义分析策略不足,可能会导致语义漂移等现象;最后是计算效率上的瓶颈,多模态模型的提示词优化需要消耗更大的算力,这势必会影响实时场景的APE使用。

相关研究提出了基于图像描述、视频场景分析等多模态输入的提示生成方法。这些方法通过引入视觉和听觉特征,改进了提示词的设计,并提高了模型在处理复杂多模态信息时的表现。例如,Zhuge等人提出了结合视觉信息的提示生成方法,通过结合视觉特征来增强文本生成能力,从而提高了在图像理解和生成任务中的表现[70];Zhou等人[71]利用基于视频的情境分析改进了提示生成方式,使得多模态对话系统能够更好地理解和响应用户的输入。这些研究表明,随着多模态对话系统的兴起,自动提示工程的技术也在不断拓展,变得更加灵活和多样化,以应对更复杂的输入和输出场景。在未来的多模态场景中,跨模态对比学习和动态融合策略等方式或许将成为提示词优化的主要手段,提示词压缩等轻量优化技术或许能够成为突破算力瓶颈的关键。

4.2 推理模型的APE

推理模型(reasoning model)通过引入思维链、多步推理等技术,通过显式逻辑链生成与过程透明化机制,显著提升了模型对数学、代码、思维推理等复杂任务的解决能力[72]。现阶段,结合了基于人类反馈的强化学习、过程奖励函数和监督微调等混合优化策略的推理模型在数学证明、代码调试等场景中展现出超越传统方法的逻辑连贯性与自我纠错能力。

随着模型架构的持续优化与边缘计算技术的突破,强推理模型正加速向实时动态工业流程优化、多模态自动驾驶决策等场景渗透。以DeepSeek为代表的专业化模型,通过思维链实现决策过程可解释性,结合MoE(mixture of experts)架构融合行业专家规则,在工业AI领域形成驱动范式[73]。在自动驾驶领域,AlphaDrive等方案通过强化学习与规划推理技术,首次实现端到端决策规划性能突破,有效应对千亿公里级长尾问题。可以预见,这些强推理模型在未来将会满足工业自动化、汽车自动驾驶等更多应用场景的需求。因此,合理运用这些强推理模型的自身能力,设计出更满足多应用场景的自动化提示词工具也将会是未来发展的一大方向。

然而,面向强推理模型的APE还远远不能和模型能力匹配,目前还存在很多问题。首先是过于精细的提示词可能会导致模型陷入“过度推理”,这是一种类似模型幻觉的冗余循环推理过程,导致相应延迟和能耗激增;其次是依赖于规则优化的APE在面向金融、医疗和

自动驾驶等垂类模型时,往往表现出很差的推理框架适配性;最后是APE优化的复杂提示词可能导致推理模型响应时间过长,无法适配需要实时应用模型的场景。

面向未来,优化推理路径剪枝仍然是激发APE在推理模型中应用的重心,具有领域内自适应能力的APE工具链将成为释放强推理模型潜力的关键。基于垂类行业知识的模型知识蒸馏框架将为APE提供更具价值的原始提示词信息,基于推理模型的可解释路径将为APE在各领域生成自动化工具链提供可行的解决方案。在强推理模型的各垂类应用端,采用NPU(neural-network processing unit)推理加速芯片等硬件协同优化将成为突破计算瓶颈的有效方案。

4.3 面向智能体的APE

基于LLM的智能体(AI-Agent)通过构建"感知-规划-执行-反思"的闭环架构,已突破传统被动响应模式,展现出动态环境适应与复杂任务求解能力[74]。AI-Agent技术以LLM为技术驱动,具备核心驱动,具备自主感知、规划、记忆和使用工具能力的智能系统,能够自动化执行复杂任务。其核心特征在于从被动响应(如传统大模型依赖提示词交互)转向主动规划与执行,例如通过分解任务目标、调用外部工具(如API)并动态调整策略来完成目标[75]。

APE技术正成为推动智能体自主决策能力跃升的核心引擎。在AI-Agent领域,提示词是诱使其完成任务的核心。用户的原始指令可以在APE的优化下变成结构化、完整化的提示词模组,从而作为结构化的指令引导智能体完成意图理解、工具调用和环境交互等复杂步骤。更进一步来说,APE还可以作为输入端的约束化规则,引导模型进行分布推理、产生完整行动链条、降低幻觉风险等安全行为。APE能够使智能体的被动响应升级为主动策略,从而成为连接人类意图和智能体智能行为的纽带。因此,面向AI-Agent的APE模组也将会是未来发展的一大方向。

然而因其强安全相关性和不可解释性等内生问题,面向智能体的APE发展可谓是难上加难。一方面是AI-Agent具有高度的长程提示词依赖性,尤其是在执行多步骤任务时,可能会因为提示词缺乏连贯性而导致其效力大打折扣。另一方面,在开放环境中交互导致的越权操作风险,这要求APE在产生提示词时要同时兼顾安全性和可用性,避免引发安全问题。

面向未来,增强提示的有效记忆和构建多粒度的意图解析机制是面向智能体的APE技术路径中的重要一环。通过构建思维树等分析策略,保留有效的上下文信息,同时将用户需求拆解为元提示以进行深层次的意图分析,这些传统APE的关键技术可以很好地适配这些应用场景。同时,采用基于安全过滤和对抗样本防御分析等安全技术的约束嵌入,能够保证智能体仅调用白名单内的工具和进行有效的越狱防御,从而达到安全性和应用性的平衡。

5 总结

APE作为大语言模型技术落地的重要推手,正通过智能化的提示词生成与优化机制,突破传统人工设计的效率瓶颈。相较于依赖领域专家经验的传统方法,APE通过动态提示空间探索与多目标优化策略,显著提升了模型在复杂任务中的适应性与稳定性。其技术实现可归纳为四大范式,每种方法都有其独特的优势,适用于不同的任务和场景。

第一,基于思维链的APE特别适用于需要逻辑推理的任务。这种方法通过将推理过程分解为多个步骤,帮助模型系统地进行推理,减少了错误回答的可能性。思维链方法能够处理复杂问题,特别是在需要逐步推理和逻辑分析的任务中表现出色,适合处理诸如推理、决策等需要结构化思维的应用。

第二,基于类机器学习的APE适用于需要大量数据支持和模型训练的场景。此类方法通过大量的训练数据来优化模型,依赖于高质量的数据集和强大的计算资源。其优势在于能够根据数据的分布和规律进行自动调整,并在大规模数据处理任务中表现出色,尤其在需要高效处理海量信息的场景中具有广泛的适用性。

第三,基于进化算法的APE则适合探索创新的提示词空间,尤其是在传统方法难以找到有效方案的情况下。进化算法通过模拟自然选择的过程,能够在广阔的搜索空间中发现新的解决方案。这种方法在生成创新的提示词和优化现有提示词时具有显著优势,尤其适合那些需要突破传统思维局限、寻找全新方向的任务。

第四,即插即用APE系统适用于对灵活性和低成本有较高需求的应用场景。其无需复杂的调整,能够快速部署,并且能够在多种环境中高效运行。尽管这种方式在一些高复杂度的任务中可能无法提供最佳的性能,但其易用性和灵活性使其成为很多商业应用场景中的首选。

面向未来应用场景,APE的演进正面临多模态融合、推理逻辑优化与智能体协作三大核心挑战。在多模态领域,跨模态语义对齐偏差与动态场景适应性不足是当前主要瓶颈,需通过跨模态学习和轻量化压缩等技术实现高效协同。推理模型的过度推理与领域迁移难题,则需结合知识蒸馏框架和硬件加速方案提升实时性与

泛化性。面向智能体的长程规划断裂与安全风险问题，需通过记忆增强型提示和形式化验证的约束嵌入实现可靠交互。未来，技术的突破将依赖于多模态统一表示、“推理-决策”闭环优化以及智能体自我进化机制，同时需推动标准化协议和伦理框架的构建，以实现APE在工业、医疗、自动驾驶等场景的规模化落地。

总的来说，APE正处于技术融合与创新加速的关键时期。随着LLM的不断发展，APE的需求也在迅速增长。本综述全面探讨了APE在自然语言处理任务中的重要性，分析了自动化方法的优势与挑战，系统梳理了主流的APE实现方式，并将其分类为基于思维链的APE、基于类机器学习模型的APE、基于进化算法的APE以及即插即用系统。通过对当前主流APE的全面评估，构建了一个理论框架，以更好地理解APE的工作原理，探究该领域研究的发展趋势，为领域内研究者提供一个综合的参考体系，为相关领域的研究提供重要的理论依据和应用参考。展望未来，提出了诸多APE的发展方向，期待APE不断迭代出新的技术范式，逐步满足更多的行业需求。

参考文献：

[1] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019: 4171-4186.

[2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems 30, 2017: 5998-6008.

[3] NAVEED H, KHAN A U, QIU S, et al. A comprehensive overview of large language models[EB/OL]. [2024-12-06]. https://arxiv.org/abs/2307.06435.

[4] 王耀祖, 李擎, 戴张杰, 等. 大语言模型研究现状与趋势[J]. 工程科学学报, 2024, 46(8): 1411-1425.
WANG Y Z, LI Q, DAI Z J, et al. Current status and trends in large language modeling research[J]. Chinese Journal of Engineering, 2024, 46(8): 1411-1425.

[5] ZHAO W X, ZHOU K, LI J, et al. A survey of large language models[EB/OL]. [2024-12-06]. https://arxiv.org/abs/2303.18223.

[6] 孙斐. 大模型提示词工程的进展、综述及展望[J]. 计算机应用文摘, 2024, 40(18): 179-182.
SUN F. Progress, review, and prospects of prompt engineering for large models[J]. Chinese Journal of Computer Application, 2024, 40(18): 179-182.

[7] 王东清, 芦飞, 张炳会, 等. 大语言模型中提示词工程综述[J]. 计算机系统应用, 2025, 34(1): 1-10.
WANG D Q, LU F, ZHANG B H, et al. Survey on prompt engineering in large language model[J]. Computer Systems and Applications, 2025, 34(1): 1-10.

[8] QIU R Z, XU Z, BAO W X, et al. Ask, and it shall be given: on the turing completeness of prompting[EB/OL]. [2024-12-06]. https://arxiv.org/abs/2411.01992.

[9] ZHOU Y C, MURESANU A I, HAN Z W, et al. Large language models are human-level prompt engineers[EB/OL]. [2024-12-06]. https://arxiv.org/abs/2211.01910.

[10] SAHOO P, SINGH A K, SAHA S, et al. A systematic survey of prompt engineering in large language models: techniques and applications[EB/OL]. [2024-12-06]. https://arxiv.org/abs/2402.07927.

[11] RAE J W, BORGEAUD S, CAI T, et al. Scaling language models: methods, analysis & insights from training gopher [EB/OL]. [2024-12-06]. https://arxiv.org/abs/2112.11446.

[12] ROBERTS J. How powerful are decoder-only transformer neural models?[C]//Proceedings of the 2024 International Joint Conference on Neural Networks. Piscataway: IEEE, 2024: 1-8.

[13] ARORA S, NARAYAN A, CHEN M F, et al. Ask me anything: a simple strategy for prompting language models[EB/OL]. [2024-12-06]. https://arxiv.org/abs/2210.02441.

[14] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[C]//Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020: 1877-1901.

[15] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[EB/OL]. [2024-12-07]. https://openai.com/research/language-models-are-unsupervised-multitask-learners.

[16] REYNOLDS L, MCDONELL K. Prompt programming for large language models: beyond the few-shot paradigm[C]//Proceedings of the Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. New York: ACM, 2021: 1-7.

[17] KEPEL D, VALOGIANNI K. Autonomous prompt engineering in large language models[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2407.11000.

[18] LESTER B, AL-RFOU R, CONSTANT N. The power of scale for parameter-efficient prompt tuning[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2104.08691.

[19] WEI J, WANG X Z, SCHUURMANS D, et al. Chain-of-thought prompting elicits reasoning in large language models [EB/OL]. [2024-12-07]. https://arxiv.org/abs/2201.11903.

[20] LIU P F, YUAN W Z, FU J L, et al. Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing[J]. ACM Computing Surveys, 2023, 55(9): 1-35.

[21] BARBIERATO E, GATTI A. The challenges of machine learning: a critical review[J]. Electronics, 2024, 13(2): 416.

[22] KOJIMA T, GU S S, REID M, et al. Large language models are zero-shot reasoners[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2205.11916.

[23] LI X L, LIANG P. Prefix-tuning: optimizing continuous prompts for generation[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2101.00190.

[24] KATOCH S, CHAUHAN S S, KUMAR V. A review on genetic algorithm: past, present, and future[J]. Multimedia Tools and Applications, 2021, 80: 8091-8126.
[25] ZHENG M, LIANG H, YANG F, et al. PAS: data-efficient plug-and-play prompt augmentation system[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2407.06027.
[26] WANG X Z, WEI J, SCHUURMANS D, et al. Self-consistency improves chain of thought reasoning in language models [EB/OL]. [2024-12-07]. https://arxiv.org/abs/2203.11171.
[27] YAO S Y, YU D, ZHAO J, et al. Tree of thoughts: deliberate problem solving with large language models[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2305.10601.
[28] BESTA M, BLACH N, KUBICEK A, et al. Graph of thoughts: solving elaborate problems with large language models[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2024, 38(16): 17682-17690.
[29] WEN Y L, WANG Z F, SUN J M. MindMap: knowledge graph prompting Sparks graph of thoughts in large language models[EB/OL]. [2024-12-07]. https://arxiv.org/abs/2308.09729.
[30] ZHANG Y F, YUAN Y, YAO A C. On the diagram of thought [EB/OL]. [2024-12-07]. https://arxiv.org/abs/2409.10038.
[31] YANG L, YU Z C, ZHANG T J, et al. Buffer of thoughts: thought-augmented reasoning with large language models [EB/OL]. [2024-12-07]. https://arxiv.org/abs/2406.04271.
[32] SEVINC A, GUMUS A. AutoReason: automatic few-shot reasoning decomposition[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2412.06975.
[33] CHU Z, CHEN J C, CHEN Q L, et al. Navigate through Enigmatic Labyrinth a survey of chain of thought reasoning: advances, frontiers and future[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2309.15402.
[34] COBBE K, KOSARAJU V, BAVARIAN M, et al. Training verifiers to solve math word problems[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2110.14168.
[35] ZHANG Z S, ZHANG A, LI M, et al. Automatic chain of thought prompting in large language models[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2210.03493.
[36] DIAO S Z, WANG P C, LIN Y, et al. Active prompting with chain-of-thought for large language models[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2302.12246.
[37] AMINI A, GABRIEL S, LIN P, et al. MathQA: towards interpretable math word problem solving with operation-based formalisms[EB/OL]. [2024-12-11]. https://arxiv.org/abs/1905.13319.
[38] YASUNAGA M, CHEN X Y, LI Y J, et al. Large language models as analogical reasoners[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2310.01714.
[39] SHUM K, DIAO S Z, ZHANG T. Automatic prompt augmentation and selection with chain-of-thought from labeled data[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2302.12822.
[40] GAO Y, CHEN S F, LU X. Research on reinforcement learning technology: a review[J]. Acta Automatica Sinica, 2004, 30(1): 86-100.
[41] DENG M K, WANG J Y, HSIEH C P, et al. RLPrompt: optimizing discrete text prompts with reinforcement learning [EB/OL]. [2024-12-11]. https://arxiv.org/abs/2205.12548.
[42] ZHANG T J, WANG X Z, ZHOU D, et al. TEMPERA: test-time prompting via reinforcement learning[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2211.11890.
[43] PRYZANT R, ITER D, LI J, et al. Automatic prompt optimization with "gradient descent" and beam search[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2305.03495.
[44] CHEN W Z, KOENIG S, DILKINA B. RePrompt: planning by automatic prompt engineering for large language models agents[EB/OL]. [2024-12-11]. https://arxiv.org/abs/2406.11132.
[45] WALLACE E, XIAO K, LEIKE R, et al. The instruction hierarchy: training LLMs to prioritize privileged instructions [EB/OL]. [2024-12-13]. https://arxiv.org/abs/2404.13208.
[46] GAO T Y, FISCH A, CHEN D Q. Making pre-trained language models better few-shot learners[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2012.15723.
[47] PEREZ E, KIELA D, CHO K. True few-shot learning with language models[C]//Advances in Neural Information Processing Systems 34, 2021:11054-11070.
[48] MA R T, WANG X L, ZHOU X, et al. Are large language models good prompt optimizers?[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2402.02101.
[49] LIU S C, CHEN C S, QU X H, et al. Large language models as evolutionary optimizers[C]//Proceedings of the 2024 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2024: 1-8.
[50] YE Q Y, AXMED M, PRYZANT R, et al. Prompt engineering a prompt engineer[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2311.05661.
[51] DO V T, HOANG V K, NGUYEN D H, et al. Automatic prompt selection for large language models[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2404.02717.
[52] LANGE R, TIAN Y T, TANG Y J. Large language models as evolution strategies[C]//Proceedings of the Genetic and Evolutionary Computation Conference Companion. New York: ACM, 2024: 579-582.
[53] FERNANDO C, BANARSE D, MICHALEWSKI H, et al. Promptbreeder: self-referential self-improvement via prompt evolution[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2309.16797.
[54] HUSSAIN A, RIAZ S, AMJAD M S, et al. Genetic algorithm with a new round-robin based tournament selection: statistical properties analysis[J]. PLoS One, 2022, 17(9): e0274456.
[55] GUO Q Y, WANG R, GUO J L, et al. EvoPrompt: connecting LLMs with evolutionary algorithms yields powerful prompt optimizers[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2309.08532.
[56] ELTAEIB T, MAHMOOD A. Differential evolution: a survey and analysis[J]. Applied Sciences, 2018, 8(10): 1945.
[57] PRASAD A, HASE P, ZHOU X, et al. GrIPS: gradient-free, edit-based instruction search for prompting large language models[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2203.

07281.
[58] HSIEH C J, SI S, YU F X, et al. Automatic engineering of long prompts[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2311.10117.
[59] CHENG J L, LIU X, ZHENG K H, et al. Black-box prompt optimization: aligning large language models without model training[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2311.04155.
[60] KEPEL D, VALOGIANNI K. Autonomous prompt engineering in large language models[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2407.11000.
[61] HE J, RUNGTA M, KOLECZEK D, et al. Does prompt formatting have any impact on LLM performance?[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2411.10541.
[62] DONG H, SU Q, GAO Y, et al. APPL: a prompt programming language for harmonious integration of programs and large language model prompts[EB/OL]. [2024-12-13]. https://arxiv.org/abs/2406.13161.
[63] CHENG J, VAN DURME B. Compressed chain of thought: efficient reasoning through dense representations[EB/OL]. [2024-12-19]. https://arxiv.org/abs/2412.13171.
[64] LU Y, BARTOLO M, MOORE A, et al. Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity[EB/OL]. [2024-12-19]. https://arxiv.org/abs/2104.08786.
[65] SU H J, KASAI J, WU C H, et al. Selective annotation makes language models better few-shot learners[EB/OL]. [2024-12-19]. https://arxiv.org/abs/2209.01975.
[66] CAO T F, WANG C Y, LIU B Y, et al. BeautifulPrompt: towards automatic prompt engineering for text-to-image synthesis[EB/OL]. [2024-12-19]. https://arxiv.org/abs/2311.06752.
[67] ZHANG Z S, ZHANG A, LI M, et al. Multimodal chain-of-thought reasoning in language models[EB/OL]. [2024-12-19]. https://arxiv.org/abs/2302.00923.
[68] RADFORD A, KIM J W, HALLACY C, et al. Learning transferable visual models from natural language supervision [C]//Proceedings of the 38th International Conference on Machine Learning, 2021: 8748-8763.
[69] LEE Y L, TSAI Y H, CHIU W C, et al. Multimodal prompting with missing modalities for visual recognition[C]//Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2023: 14943-14952.
[70] ZHUGE M C, GAO D H, FAN D P, et al. Kaleido-BERT: vision-language pre-training on fashion domain[C]//Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2021: 12647-12657.
[71] ZHOU L W, PALANGI H, ZHANG L, et al. Unified vision-language pre-training for image captioning and VQA[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 13041-13049.
[72] XU F L, HAO Q Y, ZONG Z F, et al. Towards large reasoning models: a survey of reinforced reasoning with large language models[EB/OL]. [2025-01-25]. https://arxiv.org/abs/2501.09686.
[73] JOSHI S. A comprehensive review of DeepSeek: performance, architecture and capabilities[EB/OL]. [2025-03-29]. https://www.preprints.org/manuscript/202503.1887.
[74] SANWAL M. Layered chain-of-thought prompting for multi-agent LLM systems: a comprehensive approach to explainable large language models[EB/OL]. [2025-01-31]. https://arxiv.org/abs/2501.18645.
[75] WANG L, ZHANG J S, YANG H, et al. User behavior simulation with large language model-based agents[J]. ACM Transactions on Information Systems, 2025, 43(2): 1-37.

巴泽智(2000—),男,河北石家庄人,硕士,主要研究方向为机器学习、人工智能安全、大模型安全与风险。

BA Zezhi, born in 2000, M.S. His research interests include machine learning, artificial intelligence security, large language model safety and risk.

张辉(1985—),男,湖北随州人,博士,副教授,主要研究方向为人工智能治理、网络空间治理等。

ZHANG Hui, born in 1985, Ph.D., associate professor. His research interests include governance of artificial intelligence, governance of cyberspace, etc.

谢铮涵(2001—),男,山东泰安人,硕士,主要研究方向为机器学习、人工智能安全、大模型安全与风险。

XIE Zhenghan, born in 2001, M.S. His research interests include machine learning, artificial intelligence security, large language model safety and risk.

左晓栋(1975—),男,河北石家庄人,博士,教授,主要研究方向为网络空间治理、数据安全等。

ZUO Xiaodong, born in 1975, Ph.D., professor. His research interests include governance of cyberspace, data safety, etc.

侯健玮(1994—),女,安徽六安人,博士,主要研究方向为大语言模型安全、物联网安全、SDN安全。

HOU Jianwei, born in 1994, Ph.D. Her research interests include LLM security, IoT security and SDN security.