



## A review of prompt engineering methods in large language models

Faeze Eskandari<sup>1</sup>✉<sup>id</sup> and Zahra Roozbahani<sup>2</sup><sup>id</sup>

1. Corresponding author, Assistant Prof., Department of knowledge & information science, Faculty of Management and Economics, Tarbiat Modares University, Tehran, Iran. E-mail: [F.eskandary@modares.ac.ir](mailto:F.eskandary@modares.ac.ir)
2. Assistant Prof., Department of Computer Engineering, National University of Skills (NUS), Tehran, Iran. E-mail: [zroozbahani@nus.ac.ir](mailto:zroozbahani@nus.ac.ir)

Article Info	ABSTRACT
<p><b>Article type:</b> Research Article</p> <p><b>Article history:</b> Received 11 Jan 2025 Received in revised form 8 Feb 2025 Accepted 20 Feb 2025 Published online 20 Mar 2025</p> <p><b>Keywords:</b> prompt engineering, large language models, artificial intelligence tools, content generation.</p>	<p>Prompt engineering is the process of structuring input text for large language models. This process is considered very necessary to optimize the performance of large language models; but at the same time, it is a challenging process. The purpose of this research is to investigate the basics of prompt engineering and the basics and advanced methods of writing prompts, and it also investigates the evaluation methods and applications of prompt engineering in processing natural losses. This research shows how understanding prompt engineering can be effective in improving the results of intelligent content creation tools and reducing the machine Hallucination phenomenon, and also provides valuable insights for researchers to explore in this field. Research in prompt engineering, investigating the basics and techniques and understanding its applications can help manage information and knowledge in terms of improving the accuracy and quality of information, reducing errors caused by machine Hallucination, data analysis, increasing the efficiency of content production, facilitating access. It helps knowledge, decision-making support, data analysis, and generally improves the quality of information and, as a result, facilitates decision-making processes.</p>
<p><b>Cite this article:</b> Eskandari, F. &amp; Roozbahani, Z. (2024)., A review of prompt engineering methods in large language models. <i>Engineering Management and Soft Computing</i>, 10 (2). 246-279. DOI: <a href="https://doi.org/10.22091/jemsc.2025.12095.1247">https://doi.org/10.22091/jemsc.2025.12095.1247</a></p>	
	<p>© The Author(s) DOI: <a href="https://doi.org/10.22091/jemsc.2025.12095.1247">https://doi.org/10.22091/jemsc.2025.12095.1247</a></p>
<p><b>Publisher:</b> University of Qom</p>	

## مروری بر روش‌های مهندسی پرامپت در مدل‌های زبانی بزرگ

فائزه اسکندری<sup>۱</sup> و زهرا روزبهانی<sup>۲</sup>

۱. نویسنده مسئول، استادیار، گروه علم اطلاعات و دانش‌شناسی، دانشکده مدیریت و اقتصاد، دانشگاه تربیت مدرس، تهران، ایران، شناسه. رایانامه:

[F.eskandary@modares.ac.ir](mailto:F.eskandary@modares.ac.ir)

۲. استادیار، گروه مهندسی کامپیوتر، دانشگاه ملی مهارت، تهران، ایران، شناسه رایانامه: [zroozbahani@nus.ac.ir](mailto:zroozbahani@nus.ac.ir)

اطلاعات مقاله	چکیده
<b>نوع مقاله:</b> مقاله پژوهشی	مهندسی پرامپت، فرآیند ساختاربندی متن ورودی برای مدل‌های زبانی بزرگ است. این فرایند جهت بهینه‌سازی عملکرد مدل‌های زبانی بزرگ بسیار ضروری محسوب می‌شود؛ اما در عین حال فرایندی پرچالش است. هدف از انجام این پژوهش، بررسی مبانی مهندسی پرامپت و روش‌های پایه‌ای و پیشرفته نوشتن پرامپت بوده و همچنین به بررسی روش‌های ارزیابی و کاربردهای مهندسی پرامپت در پردازش زبان‌های طبیعی می‌پردازد. این پژوهش نشان می‌دهد که چگونه شناخت مهندسی پرامپت می‌تواند در بهبود نتایج حاصل از ابزارهای هوشمند تولید محتوا و کاهش پدیده توهم ماشینی، مؤثر باشد و همچنین بینش‌های ارزشمندی را برای محققان برای کاوش در این زمینه ارائه می‌دهد. پژوهش در مهندسی پرامپت، بررسی مبانی و تکنیک‌ها و شناخت کاربردهای آن می‌تواند به مدیریت اطلاعات و دانش از نظر بهبود دقت و کیفیت اطلاعات، کاهش خطاهای ناشی از توهم ماشینی، تحلیل داده‌ها، افزایش کارایی تولید محتوا، تسهیل دسترسی به دانش، پشتیبانی از تصمیم‌گیری، تحلیل داده‌ها و در کل بهبود کیفیت اطلاعات کمک کرده و در نتیجه، فرآیندهای تصمیم‌گیری را تسهیل نماید.
<b>تاریخ دریافت:</b> ۱۴۰۳/۱۰/۲۲	
<b>تاریخ بازنگری:</b> ۱۴۰۳/۱۱/۲۰	
<b>تاریخ پذیرش:</b> ۱۴۰۳/۱۱/۳۰	
<b>تاریخ انتشار:</b> ۱۴۰۳/۱۲/۳۰	
<b>کلیدواژه‌ها:</b> مهندسی پرامپت، مدل‌های زبانی بزرگ، ابزارهای هوش مصنوعی، تولید محتوا	

**استناد:** دهقان، ریحان و روزبهانی، زهرا. (۱۴۰۳). «مروری بر روش‌های مهندسی پرامپت در مدل‌های زبانی بزرگ». مدیریت مهندسی و رایانش نرم، دوره

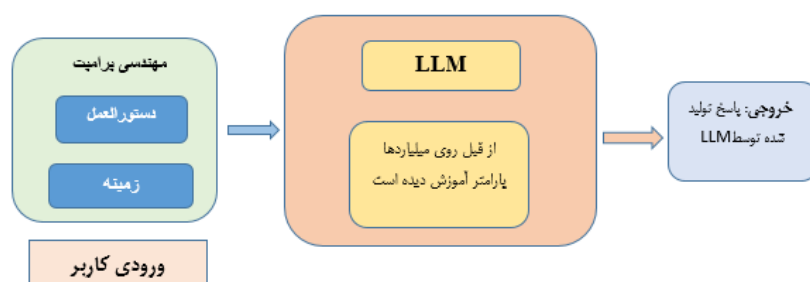
۱۰ (۲). صص: ۲۷۹-۲۴۶. <https://doi.org/10.22091/jemsc.2025.12095.1247>



## ۱) مقدمه

هوش مصنوعی مولد،<sup>۱</sup> بر اساس معماری ترانسفورماتور،<sup>۲</sup> به سرعت در حوزه هوش مصنوعی<sup>۳</sup> در حال پیشرفت است. این مدل‌ها می‌توانند محتوا را در قالب‌های مختلف، مانند متن، تصویر، و موارد دیگر تولید کنند که بسیار شبیه آنچه انسان تولید می‌کند [۱، ۲]. مدل‌های زبانی بزرگ<sup>۴</sup> یک نوع از هوش مصنوعی مولد هستند. با توسعه مدل‌های زبانی بزرگ بسیاری از جنبه‌های زندگی و کار تغییر کرده است. به عنوان مثال، از مدل‌های زبانی بزرگ می‌توان برای پیش‌نویس طرح‌های تحقیقاتی،<sup>۵</sup> نوشتن کدهای کاری در بسیاری از زبان‌های برنامه‌نویسی، ترجمه متن بین زبان‌ها،<sup>۶</sup> کمک به سیاست‌گذاری<sup>۷</sup> و موارد دیگر استفاده کرد [۳]. بسیاری از مدل‌های بزرگ معروف که بر اساس معماری ترانسفورمر [۴] ساخته شده‌اند، بر روی مجموعه‌های داده‌ی<sup>۸</sup> گسترده‌ای که از متون وب استخراج شده‌اند، آموزش دیده‌اند. هدف اصلی در طراحی این مدل‌ها، استفاده از یک هدف یادگیری خود نظارتی است که بر پیش‌بینی واژگان بعدی در جملات ناقص تمرکز دارد.

مدل‌های زبانی بزرگ با رمزگذاری متن ورودی<sup>۹</sup> در فضای برداری با ابعاد بالا<sup>۱۰</sup> عمل می‌کنند، به طوری که روابط معنایی<sup>۱۱</sup> بین کلمات و عبارات حفظ می‌شود. سپس این مدل‌های زبانی متن را رمزگشایی کرده و به کمک الگوهای آماری<sup>۱۲</sup> یاد گرفته شده پاسخ‌هایی را تولید می‌کند [۵]. کیفیت پاسخ تولیدی تحت تأثیر عوامل مختلفی قرار می‌گیرد، که یکی از مهمترین این عوامل نحوه نوشتن جمله دستوری است که به مدل ارائه می‌شود. برای اینکه ارزش واقعی ابزارهای زبانی تولیدکننده محتوا مشخص شود، ابتدا باید نحوه صحیح وارد کردن کلید - یا همان نوشتن دستور - را فرا گرفت. این مهارت ضروری، مهندسی پرامپت<sup>۱۳</sup> نامیده می‌شود. مولفه‌های مهندسی پرامپت<sup>۱۴</sup> به تفکیک در شکل ۱ نشان داده شده است. این شکل، مدل‌های زبانی بزرگ را نشان می‌دهد که بر روی داده‌های گسترده، دستورالعمل‌ها و زمینه به عنوان عناصر محوری شکل‌دهنده پرامپت و یک رابط ورودی کاربر آموزش دیده‌اند.



شکل ۱. تفکیک نموداری مولفه‌های مهندسی پرامپت

۱. Generative Artificial Intelligence
۲. Transformer Architectures
۳. Artificial Intelligence
۴. LLMs: Large Language Models
۵. Draft Research Proposals
۶. Translate Text Between Languages
۷. Policy Making
۸. Dataset
۹. Encoding the Input Text
۱۰. High-Dimensional Vector Space
۱۱. Semantic Relationships
۱۲. Statistical Patterns
۱۳. Prompt Engineering

۱۴. پرامپت در زبان فارسی به معنای «درخواست» یا «دستورالعمل» کاربرد دارد.

مهندسی پرامپت مجموعه‌ای از تکنیک‌ها و روش‌ها برای طراحی، نوشتن و بهینه‌سازی دستورالعمل‌ها برای مدل زبانی بزرگ است که به آنها پرامپت<sup>۱۵</sup> گفته می‌شود، به طوری که پاسخ‌های مدل دقیق، ملموس، قابل تکرار و از نظر واقعی درست باشد [۶،۷،۸]. در کاربردهای واقعی، دستور (پرامپت) داده شده همان ورودی مدل است و مهندسی آن می‌تواند تفاوت قابل توجهی در خروجی ایجاد کند [۹]. تغییر ساختار (مانند طول، ترتیب مثال‌ها) و محتوای (مانند نحوه بیان، انتخاب مثال‌ها) پرامپت می‌تواند تأثیر قابل توجهی بر خروجی تولید شده توسط مدل داشته باشد [۱۰]. مطالعات نشان می‌دهند که هر دو نحوه بیان و ترتیب مثال‌هایی که در پرامپت وارد می‌شوند، می‌توانند تأثیر زیادی بر رفتار مدل داشته باشند [۱۰، ۱۱]. همچنین، یک پرامپت به‌خوبی ساخته شده می‌تواند چالش‌هایی مانند «توهومات ماشینی»<sup>۱۶</sup> را که در مطالعات [۱۲، ۱۳] به آن‌ها اشاره شده است، جبران کند. علاوه بر این، علاقمندان به مدل‌های زبانی بزرگ می‌توانند به طور هوشمند دانش مدل زبانی را از طریق مبادله مکالمه زبان طبیعی اولیه<sup>۱۷</sup> یا مهندسی پرامپت استخراج کنند، و به افراد بیشتری حتی بدون پیشینه‌ی عمیق یادگیری ماشینی ریاضی<sup>۱۸</sup> اجازه می‌دهد با مدل بزرگ زبانی آزمایش کنند.

با این حال، نوشتن اعلان‌های موثر برای کاربران غیر فنی پیچیده به نظر می‌رسد و نیاز به خلاقیت، شهود و تمرین و تکرار دارد [۱۴]. هنگامی که اطلاعات دقیقی برای حل وظایف در زمینه‌های خاص نیاز باشد در صورت استفاده از درخواست‌های نادرست، پاسخ‌ها ممکن است مبهم، نادقیق، نادرست و یا خارج از حوزه مورد نیاز باشد [۱۵]. رشته مهندسی پرامپت همراه با توسعه مدل‌های زبانی بزرگ پیشرفت کرده است. آنچه که به عنوان یک عمل اولیه برای شکل‌دهی پرامپت برای هدایت خروجی مدل آغاز شد، به یک حوزه تحقیقاتی ساختاریافته با روش‌ها و بهترین رویه‌های خاص خود تبدیل شده است. مدل‌های زبانی بزرگ به طراحی پرامپت<sup>۱۹</sup> بسیار حساس هستند [۱۶]، امروزه، شناخت مهندسی پرامپت برای متخصصان یک ضرورت است. متخصصان بدون دانستن مهندسی پرامپت ممکن است برای انتقال درخواست خود به مدل‌های زبانی بزرگ مشکل داشته باشند و نتوانند درخواست خود را به صورت اصولی و ساختارمند<sup>۲۰</sup> ارائه دهند [۱۷]. مهندسی پرامپت شامل طیف گسترده‌ای از روش‌ها است، از رویکردهای پایه‌ای<sup>۲۱</sup> مانند «نقش‌دهی»<sup>۲۲</sup> [۱۸] تا روش‌های پیشرفته‌تر مانند «پرامپت زنجیره‌ای»<sup>۲۳</sup> [۱۹]. این حوزه پویا است و تحقیقات جدید به طور مداوم تکنیک‌ها و کاربردهای جدیدی را در مهندسی پرامپت معرفی می‌کنند. اهمیت مهندسی پرامپت به دلیل توانایی آن در هدایت پاسخ‌های مدل و در نتیجه افزایش انعطاف‌پذیری و مرتبط بودن مدل‌های زبانی بزرگ در بخش‌های مختلف برجسته شده است. کاربردهای آن شامل زمینه‌هایی مانند آموزش پزشکی، رادیولوژی و آموزش علوم است. این سیستم‌ها می‌توانند به عنوان مثال به عنوان دستیار مجازی<sup>۲۴</sup> برای مراقبت از دانش آموز یا گزارش نویسی مورد استفاده قرار گیرند و اطلاعات پیچیده را به یک روایت منسجم<sup>۲۵</sup> تبدیل کنند [۲۰، ۲۱، ۲۲]. تلاش‌هایی برای استانداردسازی اصطلاحات و مفاهیم در مهندسی سریع، با طبقه‌بندی‌های مختلف اعلان‌ها بر اساس معیارهای مختلف در حال انجام است.

15. Prompt

16. Machine Hallucinations

17. Basic natural language conversation

18. Math machine learning

19. Prompt Design

20. Systematically

21. Foundational Approaches

22. Role-Prompting

23. Chain of Thought Prompting

24. Virtual Assistants

25. Coherent Narrative

یکی از حوزه‌های موضوعی پیچیده و در حال پیشرفت در هوش مصنوعی، مهندسی پرامپت است. به همین خاطر در این حوزه چالش‌های مهمی نیز وجود دارد که برخی از آنها ناشی از پیشرفت‌های اخیر در مدل‌های زبانی بزرگ است که تولید متن تعاملی از طریق رابط چت (مانند chatGPT) را ممکن کرده است. این رویکرد، اغلب زمینه نوشتن ضمنی و هدف کاربر را نادیده می‌گیرد، از کنترل و استقلال کاربر پشتیبانی نمی‌کند، و کمک محدودی برای معناسازی و بازنگری برنامه‌های نوشتن ارائه می‌کند (۲۹). برخی از چالش‌های اصلی در این حوزه شامل تفسیرپذیری و شفافیت، پیچیدگی طراحی پرامپت‌ها، محدودیت‌های زبان و فرهنگ، وابستگی به مدل‌های خاص، مقیاس‌پذیری، مسائل اخلاقی و اجتماعی، و چالش‌های ارزیابی و بهینه‌سازی است که در شکل ۲ نیز بدان اشاره شده است. در بحث از «تفسیرپذیری و شفافیت»، معمولاً عدم قطعیت در تاثیر پرامپت‌ها و تعمیم‌پذیری پرامپت‌ها قابل بحث است. در هر حال باید به این نکته توجه داشت پرامپتی که در یک مسئله یا مدل خاص خوب عمل می‌کند، ممکن است در مسائل یا مدل‌های دیگر ناکارآمد باشد. از سوی دیگر، دلیل این امر نیز مشخص نیست که چرا برخی از پرامپت‌ها بهتر از سایر پرامپت‌ها عمل می‌کنند. این مسئله ممکن است به محدودیت در تفسیر عملکرد مدل‌های زبانی بازگردد. یکی از مثال‌های عینی در خصوص «تفسیرپذیری و شفافیت» این است که فرض کنید یک پرامپت مانند "Explain quantum physics in simple terms for a 10-year-old" در مدل GPT-3 خروجی خوبی تولید می‌کند، اما وقتی از همان پرامپت در GPT-4 استفاده می‌شود، نتیجه کاملاً متفاوت و پیچیده‌تر است. دلیل این تغییر رفتار برای کاربران معمولاً مشخص نیست.

در بحث از «پیچیدگی طراحی پرامپت‌ها» می‌توان به مهارت انسانی و همچنین نیاز به آزمایش و خطا اشاره کرد. یکی از مشکلاتی که وجود دارد این است که طراحی پرامپت مناسب به دانش زبانی و درک عمیقی از مدل‌های هوش مصنوعی نیاز دارد، حال آنکه این مهارت در میان کاربران عمومی در دسترس نیست. مشکل دیگر، این است که طراحی پرامپت مناسب اغلب فرآیندی زمان‌بر و بر اساس آزمون و خطاست که این امر هم می‌تواند منجر به اتلاف منابع شود. به عنوان مثال، در یک پروژه آموزش زبان، پرامپتی مانند "Translate the following text into French and explain the grammar rules used in the translation" احتمالاً برای دستیابی به یک نتیجه‌ی دقیق طراح شده اما کاربر در صورتی می‌تواند از پاسخ کاملاً صحیح و روشن نسبت به پرسش مطرح شده اطمینان حاصل کند که پاسخ را بارها آزمایش کرده تا به نسخه نهایی برسد.

«محدودیت‌های زبان و فرهنگ» نیز از جمله چالش‌های اصلی در حوزه‌ی مهندسی پرامپت در هوش مصنوعی است؛ زیرا بحث از چند زبانه بودن و حساسیت فرهنگی پرامپت‌ها چالش برانگیز بوده و نیازمند دقت فراوانی است. پرامپت‌هایی که در یک زبان خوب کار می‌کنند، این احتمال وجود دارد که در زبان‌های دیگر عملکرد ضعیفی داشته باشند. از سوی دیگر، به نظر می‌رسد پرامپت‌ها اگر با دقت طراحی شوند، قادر خواهند بود تا به حساسیت‌های فرهنگی و زبانی احترام بگذارند. پرامپتی مانند "Translate this English proverb into Persian and provide an equivalent proverb in Persian culture" احتمالاً با مشکل مواجه خواهد شد، زیرا مدل ممکن است معادل فرهنگی دقیق را نشناسد. همچنین در زبان‌هایی مانند عربی یا چینی که ساختار زبانی پیچیده‌ای دارند، این احتمال وجود دارد که پرامپت‌ها خروجی‌های غیرطبیعی یا اشتباه تولید کنند.

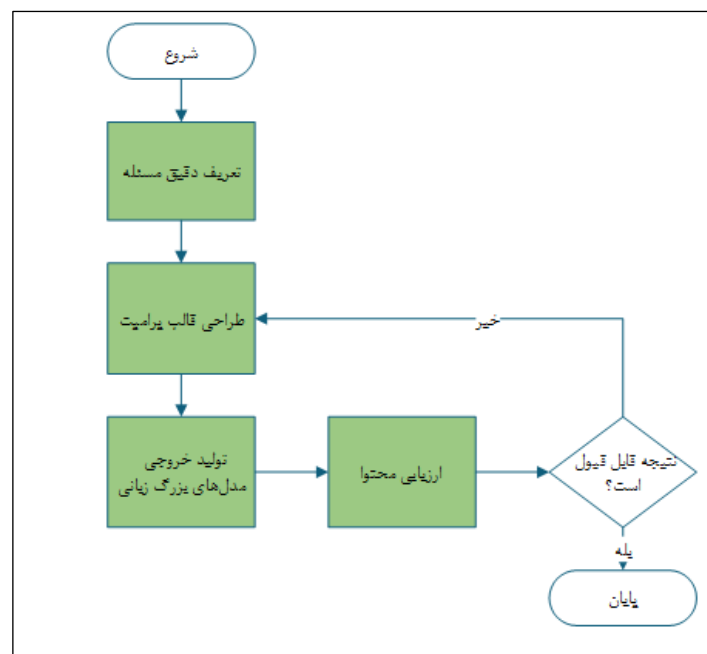
همچنین در بحث از چالش‌های موجود در مهندسی پرامپت در هوش مصنوعی «وابستگی به مدل‌های خاص» نیز محل بحث است. اختلاف در عملکرد مدل‌ها بدین معناست که مدل‌های مختلف، حتی آن دسته از مدل‌هایی که از یک خانواده هستند مثل GPT-3 و GPT-4، ممکن است نسبت به یک پرامپت خاص رفتار متفاوتی نشان دهند. از سوی دیگر، به روزرسانی مداوم مدل‌ها، ممکن است منجر به تغییر در عملکرد پرامپت‌ها شود و از این جهت نیاز به بازطراحی داشته باشد. پس در کنار اختلاف در عملکرد مدل‌ها، به روزرسانی مداوم مدل‌ها نیز حائز توجه است. کاربران Chat GPT متوجه شدند که برخی از پرامپت‌های قدیمی که در نسخه‌های قبلی مدل خوب کار می‌کردند، وقتی مدل ارتقاء پیدا کرد دیگر قادر به تولید پاسخ دقیق نبوده و نیاز به تغییر داشتند. این امر به دلیل این است که ساختار پاسخی مدل تغییر کرده است. نمونه‌ای عینی از توجه به این عامل در طراحی پرامپت‌ها، مثل این است که GPT-4 نسبت به پرامپتی مانند "Summarize" "this medical article in layman's terms" ممکن است بسیار دقیق عمل کند اما در مدل کوچک‌تر GPT-2 به دلیل محدودیت‌های معماری، مدل خروجی ضعیف و مبهمی ارائه می‌دهد.



شکل ۲. چالش‌های مهندسی پرامپت در هوش مصنوعی

چالش‌های مرتبط با «مقیاس پذیری» که مربوط به مدیریت پرامپت‌ها در سیستم‌های پیچیده و هزینه پردازش است، از جمله چالش‌های مهمی است که از یک جهت در سیستم‌هایی که از چندین پرامپت استفاده می‌کنند، مدیریت هماهنگی بین آنها چالش برانگیز خواهد بود. از سوی دیگر، پرامپت‌های طولانی یا پیچیده ممکن است به افزایش زمان و هزینه پردازش منجر شوند. به عنوان مثال، اگر یک سیستم مدیریت مکالمات مشتریان را در نظر بگیرید، در آن ممکن است از پرامپت‌های متعددی برای پردازش سوالات استفاده شود. پرامپتی که مثلاً برای پاسخ به شکایات استفاده می‌شود، ممکن است با پرامپتی که برای پیشنهاد محصولات طراحی شده، تداخل داشته باشد و خروجی‌های متناقض تولید شود که از این جهت چالش برانگیز خواهد شد. همچنین پرامپت‌های طولانی می‌تواند پردازش زیادی از مدل بخواهد و در نتیجه هزینه‌های محاسباتی را افزایش دهد. در ادامه، نمونه‌ی یک پرامپت طولانی که احتمالاً منجر به هزینه پردازش خواهد شد، عبارت است از: "Analyze this 5,000-word document and summarize it into a report with three sections, including a detailed SWOT analysis"

چالش قابل طرح دیگری که می‌تواند منجر به تولید محتوای نادرست یا سوگیرانه شده و در نتیجه منجر به سوءاستفاده شود، «مسائل اخلاقی و اجتماعی» است. برخی از پرامپت‌ها منجر به خروجی‌های غیر اخلاقی، غیر قابل قبول، سوگیرانه یا نادرست می‌شوند یا اینکه طراحی پرامپت‌هایی که مخرب هستند می‌تواند به منظور اهداف غیرقانونی و غیر اخلاقی استفاده شود. اخبارهای جعلی یا محتواهای خطرناک تولید شده نمونه‌ای از این دست پرامپت‌های چالش برانگیز محسوب می‌شوند. پرامپتی که به صورت "List reasons why one race is superior to another" طراحی شود، ممکن است در مدل منجر به تولید اطلاعات حساس یا سوگیرانه شود که می‌تواند برای اهداف غیر اخلاقی مورد استفاده قرار گیرد. در مواردی خاص نیز پرامپت‌ها به صورت ناآگاهانه شامل سوگیری هستند. مثلاً "Why are women less successful in tech?" می‌تواند خروجی‌هایی تولید کند که منجر به تقویت کلیشه‌ها و تبعیض‌ها شود. و در نهایت، چالش‌های مرتبط با «ارزیابی و بهینه‌سازی» قابل طرح است که نبودن معیارهای جامع و استاندارد برای ارزیابی کیفیت پرامپت‌ها و طراحی پرامپت‌هایی که همزمان چندین هدف مثل دقت و سرعت را بهینه می‌کنند، چالشی دشوار خواهد بود. به عنوان مثال پرامپتی همچون "Generate a creative story about space exploration" بشدت وابسته به عامل انسانی و دیدگاه وی است چون ممکن است معیار مشخصی برای خلاقیت یا جذابیت داستان وجود نداشته باشد. یا برخی از پرامپت‌هایی که برای تولید پاسخ سریع و کوتاه طراحی شده‌اند مانند "Answer this question in 50 words or less" ممکن است دقت علمی پاسخ را فدای محدودیت تعداد کلمات کند. بر همین اساس، هر یک از چالش‌های نامبرده می‌تواند مانعی برای کاربرد گسترده‌تر مهندسی پرامپت و افزودن به پیچیدگی فضای کاری این حوزه شود که مطالعات مروری ذیل روش‌های مهندسی پرامپت و کاربردهای آن‌ها با نگاهی بر مرور دقیق منابع معتبر منتشر شده در این حوزه خواهد توانست مسیر دستیابی به راه‌حل‌های ممکن نسبت به رفع اشکالات موجود را هموار نماید.



شکل ۳. فرایند استفاده از یک LLM از تعریف مسئله تا دریافت پاسخ و ارزیابی آن

در این مقاله، یک بررسی جامع در مورد مهندسی پرامپت مدل‌های زبانی بزرگ ارائه می‌دهیم. به طور کلی فرایند استفاده از یک مدل زبانی بزرگ را می‌توان یک فرایند ۴ مرحله‌ای دانست. گام اول تعریف دقیق مسئله و هدف، گام دوم طراحی متن درخواست یا پرامپت، گام سوم ارزیابی پاسخ و گام چهارم تکرار در صورت نیاز است. این ۴ فرایند در شکل ۳ نشان داده شده است.

فرآیند مهندسی پرامپت با تعریف دقیق مسئله و هدفی که باید توسط مدل هوش مصنوعی به دست آید، آغاز می‌شود. در گام دوم مسئله مورد نیاز باید در قالب یک پرامپت طراحی شود. با توجه به ساختار، پرامپت را می‌توان با استفاده از سوالات باز یا بسته<sup>۲۶</sup> طراحی نمود. سوالات باز پاسخ مشخص یا محدودی ندارند و امکان پاسخگویی گسترده‌تر و دقیق‌تر از مدل را فراهم می‌کنند [۲۳]. در مقابل، سوالات بسته معمولاً پاسخ‌های مشخص و محدودی دارند، اغلب بله یا خیر، چند گزینه‌ای یا پاسخی کوتاه و مشخص دارند. به عنوان مثال، به جای اینکه پرسید «پایتخت ایران کجاست؟» (سؤال بسته)، یک سؤال باز ممکن است این باشد: «درباره تاریخ و فرهنگ ایران بنویسید». در [۲۴]، فهرستی از روش‌های پرامپت‌نویسی ارائه و مورد بحث قرار گرفته است. در این پژوهش ۱۲ الگو برای طراحی پرامپت معرفی شده است. با توجه به هدف تعیین شده، کاربر باید از الگویی استفاده کند که به بهترین وجه با هدف او مطابقت دارد. در ادامه به طور مختصر به برخی روش‌های طراحی پرامپت اشاره شده است. درخواست از LLM برای شکستن نتایج به پاسخ‌های کوچک‌تر و اجتناب از عباراتی مانند «لطفاً، به من بدهید...» در درخواست‌های برنامه‌نویسی قابل توجه است [۲۵]. استفاده از استراتژی‌هایی مانند درخت افکار [۲۶] و همچنین افزودن مثال‌هایی برای بیان خروجی مورد نظر [۲۷] نیز از جمله موارد مورد نیاز جهت بهبود پاسخ‌های دریافتی است.

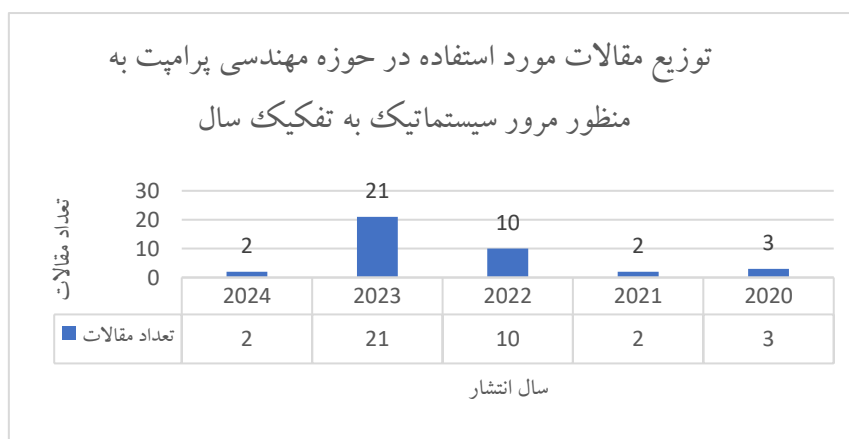
مرحله سوم از فرایند استفاده از یک LLM مرحله ارزیابی محتوای تولید شده است. ارزیابی پاسخ به دست آمده می‌تواند به طور بالقوه توهم سیستم را کاهش دهد. روش‌های مختلفی برای ارزیابی پاسخ‌های تولید شده ارائه شده است، یک رویکرد شامل بازنویسی یک سوال برای ایجاد پاسخ‌های مختلف است که می‌تواند به شناسایی ناسازگاری‌ها در میان پاسخ‌های متعدد کمک کند [۲۳]. در [۱۵]، روش‌های رسمی در طراحی اعلان‌های سیستم‌های بحرانی و مستقل با هدف نظارت بر خود و تشخیص خودکار خطاها و توهمات ادغام شده‌اند. جهت ارزیابی محتوا می‌توان مواردی از جمله «آیا پاسخ به همان اندازه که انتظار می‌رود کامل، دقیق، مرتبط و حقیقی است؟» را در نظر گرفت [۲۸]. اگر پاسخ نتواند معیارهای ارزیابی را برآورده کند، باید درخواست اصلاح شود که مستلزم تنظیم طرح برای به دست آوردن و ارزیابی پاسخ جدید است. این فرآیند پالایش تکراری تا زمانی که پاسخ سیستم تصحیح شود ادامه می‌یابد. هر یک از روش‌های ارائه شده در مهندسی پرامپت یک یا چند جنبه از چهار فرایند طراحی پرامپت را مورد بررسی قرار داده‌اند و ایده‌هایی جهت بهبود آنها ارائه کرده‌اند.

با توجه به اینکه حوزه مهندسی پرامپت بسیار جدید است، مقالات در این حوزه در سال‌های اخیر منتشر شده‌اند. در این پژوهش تلاش شده تا تمام مقالات معتبر این حوزه مورد بررسی قرار بگیرد. بر همین اساس پایگاه‌های Scopus, google scholar, science direct, IEEE و ... با کلیدواژه‌های مهندسی پرامپت جستجو شده و پس از بررسی اولیه، ۱۳۰

<sup>26</sup> . Open-Ended or Closed-Ended Questions



عنوان مقاله مورد بررسی مروری قرار گرفته و از این تعداد ۳۸ عنوان از مقالات شناسایی شده و جهت تجزیه و تحلیل دقیق تر انتخاب شده‌اند. شکل ۴ نسبت تعداد مقالات مورد استفاده در سال‌های اخیر را نشان می‌دهد.



شکل ۴. توزیع مقالات مورد استفاده به تفکیک سال

مقاله‌ی حاضر تاکید دارد که مهندسی پرآمپت به صورت گسترده در حال تحول است و بر همین اساس به دو بخش روش‌های پایه‌ای و روش‌های پیشرفته مهندسی پرآمپت تقسیم می‌شود. همچنین چالش‌های این حوزه همچون توهم ماشینی در هوش مصنوعی، طراحی پرآمپت‌های دقیق برای کاربران غیرمتخصص یا ارزیابی پاسخ‌ها نیز از جمله مسائل مهمی است که در کنار بررسی و دسته‌بندی روش‌های مهندسی پرآمپت بر حسب کاربردهای عملی آنها چارچوبی نوین فراهم کرده که پیش‌تر مورد بحث نبوده و از این جهت یکی از نوآوری‌های اصلی مقاله است. فرآیند چهار مرحله‌ای مهندسی پرآمپت شامل تعریف مسئله، طراحی پرآمپت، ارزیابی پاسخ و پالایش نیز از جمله نقاط برجسته و قابل توجه در این مقاله است. اشاره به تکنیک‌ها و کاربردهای جدید پرآمپت مانند «درخت افکار» به همراه ارائه مثال‌هایی برای طراحی بهتر پرآمپت، و یا بررسی ۱۴ الگوی طراحی پرآمپت همراه با بررسی شاخص‌های ارزیابی آنها از جمله رویکردهای جدید در این حوزه است که در مقاله مورد اشاره قرار گرفته است. همچنین نگاهی کلی بر چارچوب شکل گرفته از چالش مطالب و ساختار محتوایی فراهم آمده در این مقاله که تلاش دارد ضمن استانداردسازی اصطلاحات و مفاهیم در مهندسی پرآمپت به طبقه‌بندی جدیدی از ارائه‌ی مباحث در این حوزه پردازد، یکی دیگر از جنبه‌های نوآورانه‌ی مقاله محسوب می‌شود.

با نگاهی اجمالی به مقالات مروری در حوزه بررسی کاربردهای مهندسی پرآمپت مشخص شد که برخی از این مقالات با وجود اینکه روش‌های مهندسی پرآمپت را بر اساس کاربرد آنها مقایسه و بررسی کرده‌اند اما روش‌های مهندسی پرآمپت و نحوه عملکرد آنها مورد بررسی قرار نگرفته است بلکه صرفاً کاربرد آنها مقایسه و بررسی شده است. مقالات مروری دیگر در این حوزه نیز با وجود اینکه متدهای مهندسی پرآمپت را بررسی کرده‌اند و یا چندین روش مهندسی پرآمپت و نحوه‌ی تعامل با مدل‌های زبانی معرفی شده که به نحوی مباحث مرتبط با این حوزه را در بر می‌گیرد؛ اما مقایسه‌ای میان این متدها انجام نشده و در این حالت نخواهد توانست اطلاعات کامل و جامعی برای خوانندگان ایجاد کند.

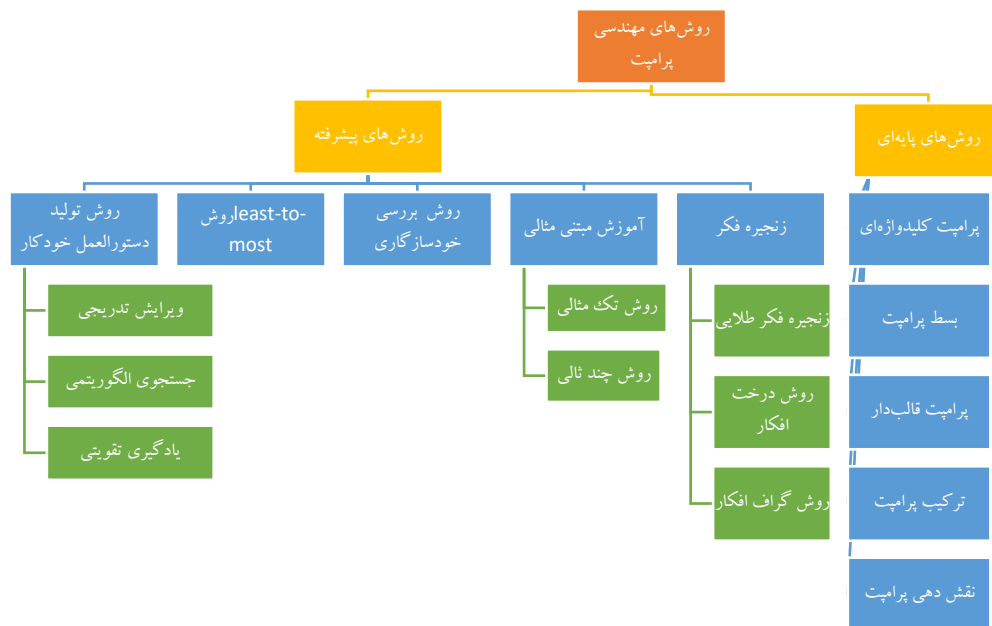
هدف از پژوهش حاضر این است که روش‌های مهندسی پرآمپت که تاکنون ارائه شده است را در گام اول به دو دسته روش‌های پایه‌ای و پیشرفته تقسیم کرده و معرفی شود. بر همین اساس ۱۴ روش مهندسی پرآمپت در دو دسته پایه و

پیشرفته بررسی شده است. در گام دوم، به جهت نشان دادن کاربرد و اهمیت هر یک از کاربردها یک مثال واقعی از نحوه استفاده از آن روش در محیط یکی از ابزارهای چت هوش مصنوعی ارائه شده است. و در نهایت نیز با توجه به رویکرد هر یک از روش‌ها، از نظر کاربرد مقایسه شده‌اند. جهت انتخاب هر یک از روش‌ها در یک کاربرد خاص، لازم است شناخت کافی نسبت به هر یک از روش‌های مهندسی پرامپت وجود داشته باشد که در این مقاله تلاش شده تا هر یک از روش‌های ارائه شده به صورت جزئی مورد بررسی قرار بگیرد.

در ادامه ساختار مقاله به صورت زیر سازماندهی شده است: در بخش دوم روش‌های پایه‌ای مهندسی پرامپت معرفی و بررسی شده است. روش‌های پیشرفته مهندسی پرامپت در بخش سوم مورد بررسی قرار گرفته است. در بخش چهارم به بررسی چالش‌ها و برخی مسائل مهم در این حوزه از جمله توهم در هوش مصنوعی مولد اشاره شده است. بخش پنجم به بررسی و دسته‌بندی روش‌های مهندسی پرامپت بر اساس کاربردهای عملی آن‌ها تمرکز دارد. در انتهای مقاله نیز نتیجه‌گیری و جمع‌بندی مطالب ارائه شده آمده است. در این پژوهش مثال‌ها با استفاده از مدل زبانی «GPT-4 پیش فرض» تولید شده‌اند.

## ۲) روش‌های پایه‌ای مهندسی پرامپت<sup>۲۷</sup>

روش‌های مختلفی جهت بهبود تعامل با مدل‌های بزرگ زبانی ارائه شده است. هر یک از این روش‌ها دارای نقاط قوت و ضعفی هستند. با اضافه کردن چند عنصر کلیدی، می‌توان یک پرامپت ساده ساخت که مدل‌های زبانی بزرگ را قادر به تولید پاسخ‌های با کیفیت بالا می‌کند. به طور کلی روش‌های مهندسی پرامپت به دو دسته کلی پایه‌ای و پیشرفته تقسیم می‌شوند. شکل ۵ دسته‌بندی این روش‌ها را نشان می‌دهد. در ادامه به معرفی «روش‌های پایه‌ای مهندسی پرامپت» پرداخته شده است.



شکل ۵. دسته‌بندی روش‌های مهندسی پرامپت

## ۲-۱) پرامپت کلیدواژه‌ای<sup>۲۸</sup>

این نوع پرامپت از کلمات کلیدی یا عبارات خاصی برای هدایت مدل به سمت تولید متن مرتبط با یک موضوع یا مفهوم خاص استفاده می‌کند. به عنوان مثال، اگر می‌خواهید متنی در مورد «تأثیر هوش مصنوعی بر آموزش» بنویسید، می‌توانید از پرامپت کلیدواژه‌ای مانند «هوش مصنوعی، آموزش، تأثیر» استفاده کنید. نتایج بدست آمده در شکل ۶ قابل ملاحظه است.

ورودی <sup>۲۹</sup>	می‌تونی یک متن مرتبط با این سه کلیدواژه برای من تولید کنی که ۵۰ کلمه باشه: هوش مصنوعی، آموزش، تأثیر
خروجی <sup>۳۰</sup>	هوش مصنوعی به طور چشمگیری بر آموزش تأثیر گذاشته است. با شخصی‌سازی یادگیری و تحلیل داده‌ها، این فناوری به معلمان کمک می‌کند تا نقاط قوت و ضعف دانش‌آموزان را شناسایی کنند. همچنین، محتوای آموزشی جذاب‌تری ارائه می‌دهد که انگیزه و علاقه به یادگیری را افزایش می‌دهد.

شکل ۶. نمونه‌ی یک پرامپت کلیدواژه‌ای و پاسخ بدست آمده از Chat GPT 4

## ۲-۲) پرامپت قالب‌دار<sup>۳۱</sup>

این نوع پرامپت از قالب‌ها یا ساختارهای از پیش تعریف‌شده‌ای برای هدایت مدل در تولید متن استفاده می‌کند. به عنوان مثال، یک پرامپت قالب‌دار برای تولید توضیحات محصول ممکن است دارای بخش‌هایی برای نام محصول، ویژگی‌ها و مزایای آن باشد. این قالب به مدل کمک می‌کند تا اطلاعات را به صورت سازماندهی شده ارائه دهد.

## ۲-۳) بسط پرامپت<sup>۳۲</sup>

این تکنیک شامل گسترش یک پرامپت کوتاه به یک ورودی طولانی‌تر و جزئی‌تر است که اطلاعات و زمینه بیشتری را در اختیار مدل قرار می‌دهد. برای ایجاد پرامپت‌های ساده، «واضح و دقیق بودن پرامپت‌ها» بسیار مهم است. این روش شامل اعمال پرامپت‌هایی است که بدون ابهام و خاص باشند، تا بتوانند مدل را به سوی تولید خروجی مورد نظر هدایت کنند. به عنوان مثال، به جای پرامپت مبهمی مانند «هوش مصنوعی را در ۱۰۰ کلمه شرح دهید»، یک پرامپت دقیق‌تر می‌تواند این‌گونه باشد: «هوش مصنوعی در زمینه مدل‌های بزرگ زبانی را در ۱۰۰ کلمه شرح دهید».

## ۲-۴) ترکیب پرامپت<sup>۳۳</sup>

این تکنیک شامل ترکیب چندین پرامپت برای ارائه طیف گسترده‌تری از اطلاعات و زمینه به مدل است. به عنوان مثال، می‌توانید یک پرامپت کلیدواژه‌ای در مورد «انرژی خورشیدی» را با یک پرامپت قالب‌دار در مورد «مزایا و معایب» ترکیب کنید تا مدل یک متن جامع در مورد مزایا و معایب انرژی خورشیدی تولید کند.

## ۲-۵) نقش‌دهی پرامپت<sup>۳۴</sup>

روش «نقش‌دهی» یکی دیگر از روش‌های اساسی در مهندسی پرامپت است. این روش شامل اختصاص دادن یک نقش

<sup>28</sup> . Keyword-Based Prompts

<sup>29</sup> . Input

<sup>30</sup> . Output

<sup>31</sup> . Template-Based Prompts

<sup>32</sup> . Prompt Expansion

<sup>33</sup> . Prompt Combination

<sup>34</sup> . Role-Prompting

خاص به مدل، مانند یک دستیار مفید<sup>۳۵</sup> یا یک کارشناس آگاه<sup>۳۶</sup> می‌باشد [۲۹]. این روش می‌تواند در هدایت پاسخ‌های مدل و اطمینان از هم‌سویی آن‌ها با خروجی مورد نظر، مؤثر باشد. به عنوان مثال، اگر مدل به عنوان یک متخصص هوش مصنوعی نقش‌دهی شود، احتمالاً پاسخ‌های دقیق‌تر به سوالات مربوط به حوزه هوش مصنوعی ارائه خواهد کرد [۳۰]. در پژوهش [۳۱] نشان داده شده است که در بسیاری از موارد هنگامی که به یک مدل زبانی بزرگ گفته می‌شود در زمینه خاصی متخصص است، نتایج بهتری حاصل می‌شود. نمونه این روش در شکل ۷ نشان داده شده است.

## ۲-۶) استفاده از نقل قول‌های سه گانه برای جداسازی پرامپت‌ها<sup>۳۷</sup>

در مهندسی پرامپت، استفاده از علامت‌های نقل قول سه گانه یک تکنیک است که برای جدا کردن بخش‌های مختلف یک پرامپت یا به‌منظور محصور کردن رشته‌های چندخطی<sup>۳۸</sup> به کار می‌رود. این تکنیک به‌ویژه زمانی مفید است که با پرامپت‌های پیچیده<sup>۳۹</sup> که شامل چندین جزء هستند، سروکار داریم یا زمانی که پرامپت خود حاوی علامت‌های نقل قول<sup>۴۰</sup> است، که باعث می‌شود مدل، دستورالعمل‌های کاربر را بهتر درک کند [۳۲].

## ۳) روش‌های پیشرفته مهندسی پرامپت<sup>۴۱</sup>

در این بخش به معرفی تکنیک‌های پیشرفته‌تر در مهندسی پرامپت می‌پردازیم. به کمک روش‌های پیشرفته می‌توان مدل را برای تولید محتوای خاص‌تر، دقیق‌تر و با کیفیت بالاتر<sup>۴۲</sup> هدایت نمود [۳۳].

## ۳-۱) زنجیره تفکر<sup>۴۳</sup>

یک توسعه نسبتاً جدید در حوزه هوش مصنوعی مولد، مفهوم پرامپت نویسی به کمک «زنجیره تفکر» است؛ که نشان داده شده است دقت مدل‌ها را در وظایفی که نیاز به استدلال منطقی<sup>۴۴</sup> دارند، به طور قابل توجهی بهبود می‌بخشد [۳۴، ۳۵].

ورودی	مدل‌های بزرگ زبانی را به طور مختصر شرح دهید؟	بدون نقش‌دهی به پرامپت
خروجی	مدل‌های بزرگ زبانی (LLMs) نوعی از هوش مصنوعی هستند که برای پردازش و تولید زبان طبیعی طراحی شده‌اند. این مدل‌ها با استفاده از داده‌های متنی فراوان آموزش می‌بینند و می‌توانند وظایف مختلفی را انجام دهند، از جمله: ۱. تولید متن: تولید متن‌های طبیعی و مرتبط بر اساس ورودی کاربر. ۲. ترجمه زبان: ترجمه متون از یک زبان به زبان دیگر. ۳. خلاصه‌سازی: ایجاد خلاصه‌های مختصر از متون طولانی. ۴. پاسخ به سوالات: پاسخ به سوالات بر اساس اطلاعات موجود در متن.	پرامپت
ورودی	شما به عنوان یک کارشناس هوش مصنوعی مدل‌های بزرگ زبانی را توضیح دهید؟	نقش‌دهی به پرامپت
خروجی	مدل‌های بزرگ زبانی (LLMs) نوعی از سیستم‌های هوش مصنوعی هستند که به‌طور خاص برای پردازش و تولید زبان طبیعی طراحی شده‌اند. این مدل‌ها با استفاده از تکنیک‌های یادگیری عمیق و معماری‌های پیچیده مانند ترنسفورمرها، توانایی درک و تولید متن را بهبود می‌بخشند. در ادامه به توضیح جنبه‌های مختلف این مدل‌ها می‌پردازم. ۱. معماری ترنسفورمر مدل‌های بزرگ زبانی معمولاً بر پایه معماری ترنسفورمر ساخته می‌شوند. این ویژگی موجب می‌شود که مدل بتواند با متن‌های طولانی	پرامپت

<sup>35</sup> . A Helpful Assistant

<sup>36</sup> . A Knowledgeable Expert

<sup>37</sup> . Use of Triple Quotes to Separate

<sup>38</sup> . Encapsulate Multi-Line Strings

<sup>39</sup> . Complex Prompts

<sup>40</sup> . Quotes

<sup>41</sup> . Advanced Methodologies

<sup>42</sup> . Produce High-Quality Answers

<sup>43</sup> . Chain of Thought - CoT

<sup>44</sup> . Logical Reasoning Tasks

و پیچیده بهتر کار کند. ۲. آموزش و داده‌ها مدل‌های بزرگ زبانی معمولاً با استفاده از داده‌های متنی بسیار بزرگ آموزش می‌بینند. این داده‌ها شامل کتاب‌ها، مقالات، وب‌سایت‌ها و سایر منابع متنی هستند. ۳. تولید زبان طبیعی یکی از ویژگی‌های کلیدی این مدل‌ها، توانایی تولید متن طبیعی و مرتبط است. این مدل‌ها می‌توانند بر اساس ورودی کاربر، متن‌هایی ایجاد کنند که از نظر معنایی و زبانی مناسب باشند. ۴. ##۴. کاربردها مدل‌های بزرگ زبانی در کاربردهای مختلفی مورد استفاده قرار می‌گیرند، از جمله - چت‌بات‌ها: برای پاسخ به سوالات کاربران و ارائه مشاوره - ترجمه زبان: ترجمه متون از یک زبان به زبان دیگر - خلاصه‌سازی متن: ایجاد خلاصه‌های مختصر از متون طولانی - تحلیل احساسات: تحلیل نظرات و احساسات کاربران در متون. ۵. ##۵. چالش‌ها با وجود پیشرفت‌های چشمگیر، مدل‌های بزرگ زبانی با چالش‌هایی نیز مواجه هستند - تبعیض و تعصب - عدم دقت: - مصرف منابع. ۶. ##۶. آینده با پیشرفت فناوری و تحقیقات در زمینه هوش مصنوعی، انتظار می‌رود که مدل‌های بزرگ زبانی بهبود یابند و کاربردهای جدیدی پیدا کنند.

### شکل ۷. مقایسه‌ی تطبیقی پاسخ‌های Chat GPT4 در حالت نقش‌دهی به پرامپت و بدون نقش‌دهی

پرامپت‌نویسی به کمک زنجیره تفکر به معنای ارائه مراحل میانی استدلال، برای راهنمایی پاسخ‌های مدل است که می‌تواند از طریق راهنمایی‌های ساده‌ای مانند «بیایید قدم به قدم فکر کنیم»<sup>۴۵</sup> یا از طریق مجموعه‌ای از نمایش‌های دستی<sup>۴۶</sup>، هر یک متشکل از یک سوال و زنجیره استدلال<sup>۴۷</sup> که به یک پاسخ منجر می‌شود، تسهیل شود [۳۶، ۳۷].

شبهه به «گام به گام فکر کنید»، زنجیره فکر، مدل‌های زبانی بزرگ را به تقسیم وظایف پیچیده به چند مرحله میانی راهنمایی می‌کند [۳۸] این روش از چگونگی حل مشکلات پیچیده توسط انسان الهام گرفته است. به عنوان مثال، برای بهتر کردن یک مدل زبانی بزرگ در حل مسائل ریاضی، کاربران یک راه‌حل مثال ارائه می‌دهند که گام به گام<sup>۴۸</sup> انجام می‌شود. به طور کلی، پیشنهاد زنجیره فکر برای حل مسائل پیچیده<sup>۴۹</sup> مفید است، اما در مسائل ساده<sup>۵۰</sup>، خیلی سودمند نیست [۳۹].

یکی از معروف‌ترین روش‌های مهندسی پرامپت این است که عبارت «گام به گام فکر کنید» را به انتهای یک درخواست اضافه کنید. محققان دانشگاه توکیو و گوگل دریافتند که افزودن این عبارت باعث افزایش دقت GPT-3 (مدل text-davinci-002) می‌شود. به عنوان مثال، این روش در ارزیابی Multi Arith دقت را از ۱۷.۷٪ به ۷۸.۷٪ افزایش داده است [۴۰]. در پژوهش [۴۱] توضیح داده شده است که چرا و چگونه اضافه کردن عبارت «گام به گام فکر کنید» تا این حد مؤثر عمل می‌کند. شکل ۸ نمونه‌ای از تاثیر عبارت «گام به گام فکر کنید» را به وضوح نمایش می‌دهد.

ورودی	خروجی
علی ۱۰۰ میلیون تومان دارد و هر رایانه کوچک ۱۵ میلیون تومان است. علی چند رایانه می‌تواند بخرد؟ قدم به قدم فکر کن	برای محاسبه تعداد رایانه‌هایی که علی می‌تواند بخرد، مراحل زیر را دنبال می‌کنیم: ##۱: مرحله ۱: تعیین مبلغ کل و قیمت هر رایانه -مبلغ کل علی: ۱۰۰ میلیون تومان -قیمت هر رایانه کوچک: ۱۵ میلیون تومان ##۲: مرحله ۲: محاسبه تعداد رایانه‌ها برای پیدا کردن تعداد رایانه‌هایی که علی می‌تواند بخرد، باید مبلغ کل را بر قیمت هر رایانه تقسیم کنیم: تعداد رایانه‌ها = مبلغ کل / قیمت هر رایانه ##۳: مرحله ۳: جایگذاری مقادیر حالا مقادیر را در فرمول قرار می‌دهیم: تعداد رایانه‌ها = ۱۰۰ میلیون تومان / ۱۵ میلیون تومان ##۴: مرحله ۴: انجام محاسبه

45. Let's Think Step by Step

46. Series of Manual Demonstrations

47. Reasoning Chain

48. Step by Step

49. Complex Problems

50. Simpler Steps

حال محاسبه را انجام می‌دهیم: تعداد رایانه‌ها =  $6.67 \approx 15/100$   
 ##### مرحله ۵: نتیجه‌گیری  
 چون علی نمی‌تواند یک بخش از رایانه را بخرد، باید تعداد کامل رایانه‌ها را در نظر بگیریم. بنابراین، علی می‌تواند ۶ رایانه کوچک بخرد  
 ##### نتیجه نهایی علی می‌تواند ۶ رایانه کوچک بخرد.

شکل ۸. پرامپت بر اساس روش «گام به گام فکر کنید» و پاسخ آن توسط chat GPT 4

### ۳-۲) زنجیره فکری طلایی<sup>۵۱</sup>

مفهوم «زنجیره فکری طلایی» [۴۲] رویکردی نوآورانه برای تولید پاسخ‌ها به سوالات مبتنی بر دستورالعمل<sup>۵۲</sup>، ارائه می‌دهد. این روش از مجموعه‌ای از راه‌حل‌های «زنجیره تفکر بر پایه حقیقت»<sup>۵۳</sup> استفاده می‌کند که درون پرامپت گنجانده شده است و این کار را برای مدل به‌طور قابل توجهی ساده‌تر می‌کند، زیرا نیاز به تولید مستقل زنجیره تفکر نیست. به‌طور همزمان، معیاری جدید شامل معماهای کارآگاهی<sup>۵۴</sup> طراحی شده است تا ظرفیت‌های استدلال مدل‌های زبانی بزرگ<sup>۵۵</sup> را ارزیابی کند که همچنین به‌عنوان ارزیابی زنجیره فکری طلایی در نظر گرفته می‌شود. در نهایت، طبق آزمایشات<sup>۵۶</sup> [۴۲]، در زمینه زنجیره طلایی تفکر، GPT-4 عملکرد قابل قبولی از خود نشان می‌دهد و با نرخ حل<sup>۵۷</sup> ۸۳٪ در مقایسه با نرخ حل ۳۸٪ زنجیره تفکر استاندارد<sup>۵۸</sup>، موفق عمل می‌کند. منظور از نرخ حل به‌عنوان نسبت تعداد مشکلات یا چالش‌هایی که به درستی حل شده‌اند به کل مشکلات موجود بیان می‌شود. به‌عنوان مثال، اگر از ۱۰۰ مسئله، ۷۰ مورد حل شده باشد، نرخ حل ۷۰ درصد خواهد بود.

### ۳-۳) روش درخت و گراف افکار<sup>۵۹</sup>

تکنیک «درخت افکار»<sup>۶۰</sup> یک روش پیشرفته است که از رویکردی ساختاریافته<sup>۶۱</sup> برای هدایت مدل‌های زبان بزرگ در فرآیندهای استدلال و تولید پاسخ استفاده می‌کند. برخلاف روش‌های سنتی پرامپت‌نویسی<sup>۶۲</sup> که به توالی خطی دستورات<sup>۶۳</sup> متکی هستند، روش «درخت افکار» پرامپت‌ها را به صورت سلسله‌مراتبی<sup>۶۴</sup>، مشابه با ساختار درخت<sup>۶۵</sup>، سازماندهی می‌کند [۴۳]. به‌عنوان مثال، زمانی که مدل برای حل یک مسئله ریاضی پیچیده مأمور می‌شود، پرامپت نویسی ساده ممکن است مستقیماً از مدل بخواهد که پاسخ را ارائه دهد. در مقابل، با استفاده از روش درخت افکار، پرامپت اولیه<sup>۶۶</sup> از مدل می‌خواهد تا مراحل مورد نیاز برای حل مسئله را ترسیم کند. پرامپت‌های بعدی نیز به تفصیل به بررسی هر مرحله پرداخته و مدل را از طریق یک فرآیند سلسله‌مراتبی حل می‌کنند.

<sup>51</sup> . Golden Chain of Thought

<sup>52</sup> . Responses to Instruction-Based Queries

<sup>53</sup> . Ground-Truth Chain-of-Thought

<sup>54</sup> . Detective Puzzles

<sup>55</sup> . The Abductive Reasoning Capacities of LLMs

<sup>56</sup> . Experiment

<sup>57</sup> . Solve rate

<sup>58</sup> . Standard CoT

<sup>59</sup> . Thought Tree and Graph Method

<sup>60</sup> . ToT: Tree of Thoughts

<sup>61</sup> . Structured Approach

<sup>62</sup> . Traditional Prompting Methods

<sup>63</sup> . Linear Sequence of Instructions

<sup>64</sup> . Hierarchical Manner

<sup>65</sup> . Tree Structure

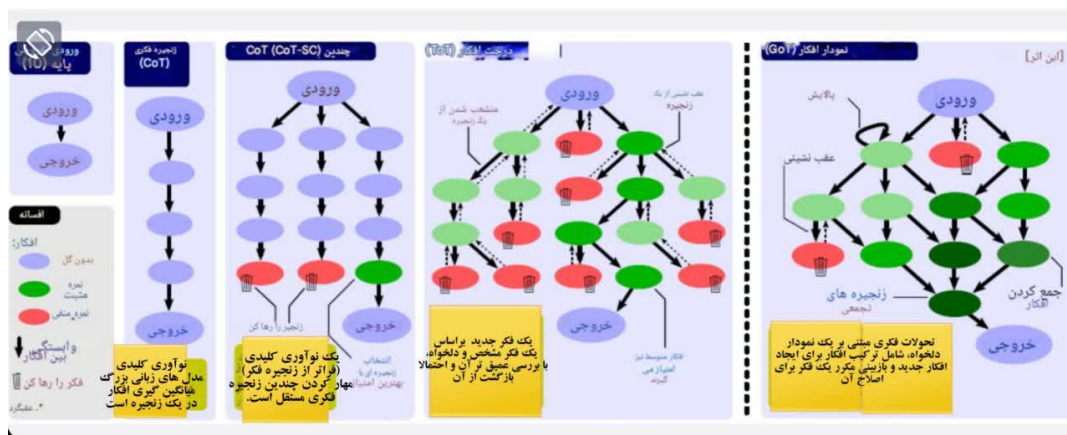
<sup>66</sup> . The Initial Prompt

این تکنیک می‌تواند عملکرد مدل‌های زبانی بزرگ را با ساختاردهی به فرآیندهای فکری<sup>۶۷</sup> آن‌ها بهبود بخشد [۴۴]. پژوهش [۴۵] به معرفی «پرامپت‌نویسی به روش درخت افکار»<sup>۶۸</sup> پرداخته و اصول پایه‌ای چارچوب‌های درخت افکار<sup>۶۹</sup> را در قالب یک روش پرامپت‌نویسی ساده<sup>۷۰</sup> شده، پیاده‌سازی کرده است. شکل ۹ نتایج بدست آمده از پرامپت با تکنیک «درخت افکار» و پاسخ آن با استفاده از Chat GPT را نشان داده است.

ورودی	پرسش: «به یک تجربه اخیر که در آن با چالشی روبرو شدید، فکر کنید. در آن لحظه چه افکاری داشتید؟ این افکار چگونه بر اقدامات شما تأثیر گذاشت؟ حالا که به آن نگاه می‌کنید، چه چیزی را در روند فکرتان تغییر می‌دادید؟»
خروجی	####ساختار پاسخ. ۱. توصیف تجربه - به طور خلاصه چالش‌هایی که با آن روبرو شدید را توضیح دهید. ۲. افکار اولیه - در طول چالش، افکار یا احساسات اولیه شما چه بودند؟ ۳. تأثیر بر اقدامات - این افکار چگونه بر تصمیمات یا اقدامات شما تأثیر گذاشتند؟ ۴. تفکر مجدد - با نگاهی به گذشته، درباره روند فکرتان چه احساسی دارید؟ آیا اکنون به شکل دیگری به آن نزدیک می‌شوید؟

شکل ۹. نمونه‌ی پرامپت «درخت افکار» و پاسخ به آن توسط Chat GPT 4

رویکرد «گراف افکار»<sup>۷۱</sup> (GoT) [۴۶] روش پیچیده‌تری را برای نمایش اطلاعات تولید شده توسط مدل‌های زبانی بزرگ ارائه می‌دهد. در این روش اطلاعات به شکل یک گراف دلخواه مدل‌سازی می‌شوند. در این گراف، واحدهای اطلاعاتی فردی که به‌عنوان «افکار مدل زبانی بزرگ»<sup>۷۲</sup> شناخته می‌شوند، به‌صورت رئوس گراف نمایش داده می‌شوند. در مقابل، لبه‌های گراف<sup>۷۳</sup> وابستگی‌های بین این رئوس را به تصویر می‌کشند. این نمایش منحصر به فرد اجازه می‌دهد تا افکار مدل زبانی بزرگ به‌طور دلخواه ترکیب شوند و به این ترتیب اثر هم‌افزایی در خروجی‌های مدل، ایجاد می‌شود. اثربخشی چارچوب GoT<sup>۷۴</sup> به قابلیت تطبیق‌پذیری آن و درک عمیق موضوع و طراحی دقیق پرامپت‌ها برای دستیابی به نتایج بهینه، وابسته است [۴۷]. روش‌های زنجیره فکر و درخت و گراف فکر در شکل ۱۰ به صورت گرافیکی نشان داده شده است.



شکل ۱۰. نمایش گرافیکی روش‌های زنجیره فکر [۴۸]

67. Thought Processes
68. Tree-of-Thought Prompting
69. The ToT Frameworks and Transforms
70. A Singular Prompt
71. The "Graph of Thoughts" (GoT) Framework
72. LLM Thoughts
73. Edges of The Graph
74. The Efficacy of the GoT Framework



### ۳-۴) آموزش مبتنی بر مثال<sup>۷۵</sup>

تکنیک‌های «تک‌مثالی» و «چندمثالی»<sup>۷۶</sup> دو روش مهم در مهندسی پرامپت هستند. پرامپت تک‌مثالی<sup>۷۷</sup> به روشی اشاره دارد که در آن مدل تنها با یک مثال آموزش می‌بیند، در حالی که در پرامپت چندمثالی<sup>۷۸</sup> [۴۹] مدل با چندین مثال ارائه می‌شود [۵۰]. انتخاب بین این دو روش معمولاً به پیچیدگی درخواست و توانایی مدل بستگی دارد. برای مثال، در وظایف ساده یا مدل‌های بسیار قدرتمند، پرامپت تک‌مثالی ممکن است کافی باشد. نمونه‌ای از این وضعیت در شکل ۹ نشان داده شده است. اما در وظایف پیچیده‌تر یا مدل‌های کمتر توانمند<sup>۷۹</sup>، پرامپت چندمثالی می‌تواند زمینه و راهنمایی بیشتری ارائه دهد و «بهبود عملکرد مدل»<sup>۸۰</sup> را ممکن سازد.

با ارائه نمونه‌هایی از خروجی‌های مورد نظر شما، مدل‌های زبانی بزرگ، توانایی بیشتری برای تولید خروجی‌های مورد نظر دارد [۵۱]. با توجه به این حقیقت که به طور کلی خروجی‌های معتبر بسیاری برای یک پرامپت وجود دارد، ارائه نمونه‌های خاصی از آنچه به دنبال آن هستید، به محدود کردن فضای خروجی کمک می‌کند. وجود مثال‌های متنوع و متعادل در این نوع درخواست‌ها بسیار مهم هستند. برای مثال، تصور کنید که از GPT می‌خواهید تا یک وظیفه طبقه‌بندی احساسات مثبت و منفی را انجام دهد. اگر شش مثال ارائه دهید و پنج مثال مثبت باشد، این کار می‌تواند GPT را به سمت پیش‌بینی احساسات مثبت، هدایت کند. این مسئله در شکل ۹ به وضوح نمایش داده شده است.

با این حال، همان‌طور که در پژوهش [۵۲] بیان شده است، «مثال‌ها همیشه کمک نمی‌کنند»<sup>۸۱</sup>. پیچیدگی‌های واکنش مدل‌های زبان مولد بزرگ<sup>۸۲</sup>، مانند GPT-3، به پرامپت‌ها را مورد بررسی قرار داده است. یکی از یافته‌های مهم این مقاله این است که پرامپت‌های بدون مثال<sup>۸۳</sup> در برخی شرایط می‌توانند عملکرد بهتری نسبت به پرامپت‌های چندمثالی<sup>۸۴</sup> داشته باشند. این یافته نشان می‌دهد که نقش پرامپت‌های چندمثالی ممکن است به جای یادگیری یک وظیفه جدید (یادگیری متا<sup>۸۵</sup>)، بیشتر به هدایت مدل برای به خاطر آوردن وظیفه‌ای که قبلاً آموخته است، مربوط باشد. این در به چالش کشیدن این دیدگاه سنتی که تعداد بیشتر مثال‌ها همیشه به عملکرد بهتر منجر می‌شود [۵۳]، حائز اهمیت می‌باشد. در زمینه پرامپت‌های یک شات یا چند شات، لازم است درک کنیم که در حالی که مثال‌ها می‌توانند مدل را هدایت کنند، اما همیشه عملکرد آن را بهبود نمی‌بخشند. گاهی اوقات یک پرامپت بدون مثال به خوبی طراحی شده می‌تواند مؤثرتر از ارائه پرامپت با چندین مثال باشد [۵۴]. شکل ۱۱ نمونه‌ای از مثال‌های عینی در این زمینه است.

ورودی	مثال ۱: احساس جمله زیر را طبقه‌بندی کنید دانشجویان از شنیدن این خبر شگفت‌زده شدند تحلیل: واژه شگفت‌زده باید با توجه به جملات قبل یا بعد از این جمله مشخص شود که نشان دهنده احساس مثبت یا منفی باشد
-------	--

<sup>75</sup> . Shot-Based Training

<sup>76</sup> . One-Shot or Few-Shot Prompting

<sup>77</sup> . One-Shot Prompting

<sup>78</sup> . Few-Shot Prompting

<sup>79</sup> . More Complex Tasks or Less Capable Models

<sup>80</sup> . Improving the Models Performance

<sup>81</sup> . Examples don't Always Help

<sup>82</sup> . The Intricacies of How Large Generative Language Models

<sup>83</sup> . Zero-Shot Prompts

<sup>84</sup> . Few-Shot Prompts

<sup>85</sup> . Meta-Learning



<p>مثال ۲ آهو توسط شکارچی گرفتار شد.</p> <p>مثال ۳ جملات زیر احساس منفی دارند گرفتار شدن گیر افتادن متحیر شدن با توجه به این مثال‌ها جمله آهو توسط شکارچی گرفتار شد چه احساسی دارد؟</p>	
<p>جمله «آهو توسط شکارچی گرفتار شد» احساس منفی دارد.</p> <p>تحلیل: واژه «گرفتار» در این جمله به معنای عدم آزادی و در معرض خطر بودن است، که به وضوح نشان‌دهنده یک وضعیت ناخوشایند و تهدیدآمیز برای آهو می‌باشد. بنابراین، این جمله احساس منفی را منتقل می‌کند.</p>	<p>خروجی</p>

شکل ۱۱. پرامپت چند مثالی و پاسخ‌های دریافتی از Chat GPT 4

### ۳-۵) درخواست برنامه‌نویسی<sup>۸۶</sup>

در حالی که مدل‌های زبانی بزرگ برای انجام دقیق محاسبات پیچیده تلاش می‌کنند، در نوشتن کدها و برنامه‌های کامپیوتری نیز می‌توانند عالی باشند. یک استراتژی ساده این است که به سادگی از مدل‌های زبانی بزرگ بخواهید برای حل مشکلات کد بنویسند و کد را در یک محیط توسعه مانند Google Colab یا Visual Studio Code اجرا کنند [۵۵] اما همه مدل‌های زبانی بزرگ برای نوشتن کد آموزش ندیده‌اند. همچنین، مدل‌های زبانی بزرگ در زبان‌های برنامه‌نویسی که به طور گسترده استفاده می‌شوند و به طور گسترده در اینترنت مستند شده‌اند، مانند پایتون<sup>۸۷</sup>، بهتر عمل می‌کنند. آنها می‌توانند از نمونه‌های بی‌شمار پایتون در داده‌های آموزشی یاد بگیرند. برعکس، مدل‌های زبانی بزرگ در زبان‌های مبهم‌تر مانند OCaml ضعیف‌تر هستند. با این حال، این امکان وجود دارد که مدل‌های زبانی بزرگ را بر روی زبان‌های برنامه‌نویسی خاص تنظیم کنید و همچنین مجموعه داده‌های آموزشی آن‌ها را برای نمونه‌های بیشتری از زبان‌های برنامه‌نویسی خاص تقویت کنید.

### ۳-۶) خودسازگاری<sup>۸۸</sup>

تکنیک پیشرفته‌ای به نام «خودسازگاری» در مدل‌های زبانی بزرگ وجود دارد که هدف آن اطمینان از «هم‌خوانی پاسخ‌های مدل<sup>۸۹</sup>» با یکدیگر است [۵۶، ۵۷]، که به طور قابل توجهی شانس دستیابی به نتایج بسیار دقیق را افزایش می‌دهد. اصل این تکنیک بر این پایه استوار است که اگر مدل به مجموعه‌ای از سوالات مرتبط پاسخ دهد، این پاسخ‌ها نباید با یکدیگر متناقض باشند. مطالعات نشان داده‌اند که «خودسازگاری نتایج<sup>۹۰</sup>» را در وظایف محاسباتی، استدلال کلی و نمادین بهبود می‌بخشد [۵۸، ۵۹]. علاوه بر این، در عمل، خودسازگاری می‌تواند با سایر تکنیک‌ها ترکیب شود تا عملکرد مدل را بیشتر ارتقا دهد. به عنوان مثال، یک مطالعه نشان داد که ترکیب خودسازگاری با «رویکرد استدلال چندمرحله‌ای هدایت شده توسط متمایزکننده<sup>۹۱</sup>» به طور قابل توجهی قابلیت‌های استدلال مدل را بهبود بخشیده است [۶۰].

### ۳-۷) پرامپت‌نویسی به روش از کمترین تا بیشترین<sup>۹۲</sup>

مفهوم پرامپت‌نویسی از «کمترین تا بیشترین» [۶۱] به عنوان یک روش پیشرفته در نظر گرفته می‌شود که شامل شروع با یک

<sup>86</sup> . Ask for Code

<sup>87</sup> . Python

<sup>88</sup> . Self-Consistency

<sup>89</sup> . Ensure the Models Responses

<sup>90</sup> . Self-Consistency Outcomes

<sup>91</sup> . A Discriminator-Guided Multi-Step Reasoning Approach

<sup>92</sup> . Least-to-Most Prompting

پرامپت حداقلی<sup>۹۳</sup> و به تدریج افزایش پیچیدگی آن به منظور استخراج پاسخ‌های پیچیده‌تر از مدل زبانی است. پایه‌گذار این رویکرد، تجزیه مشکلات پیچیده به یک سری از مسائل ابتدایی‌تر است که به ترتیب حل می‌شوند. حل هر مسئله کوچک‌تر با استفاده از راه‌حل‌های استخراج شده از مسائل پیشین تسریع می‌شود. نتایج آزمایش‌های دقیق در حوزه‌هایی نظیر دستکاری نمادین<sup>۹۴</sup>، تعمیم ترکیبی<sup>۹۵</sup>، و استدلال ریاضی<sup>۹۶</sup>، شواهدی از [۶۱] به دست می‌آید که نشان می‌دهد «پارادایم پرامپت‌نویسی کمترین تا بیشترین<sup>۹۷</sup>» قادر به تعمیم به چالش‌های پیچیده‌تر از آنچه که در ابتدا در پرامپت‌ها ارائه شده است، می‌باشد. این مطالعه نشان می‌دهد که مدل‌های زبانی بزرگ به طور مؤثری به این روش پاسخ می‌دهند و پتانسیل آن برای تقویت قابلیت‌های استدلال این مدل‌ها را نشان می‌دهد.

### ۳-۸) روش تولید دستورالعمل خودکار<sup>۹۸</sup>

روش «اطلاعات تولید شده<sup>۹۹</sup>» [۶۲] در مهندسی پرامپت، تکنیکی است که از توانایی مدل‌های زبان بزرگ برای تولید اطلاعات مفید درباره سوال یا پرامپت داده شده پیش از ارائه پاسخ نهایی استفاده می‌کند. این روش به‌ویژه در وظایفی که نیاز به استدلال رایج دارند، مؤثر است، زیرا به مدل این امکان را می‌دهد که زمینه‌های اضافی را که ممکن است در پرامپت اولیه به‌طور صریح وجود نداشته باشند، تولید و استفاده کند. به‌عنوان مثال، هنگامی که از مدل پرسیده می‌شود: «تصور کنید که ورودی به‌طور نامتناهی عریض است، چه چیزی بیشتر احتمال دارد که از آن عبور کند، یک تانک نظامی یا یک ماشین؟»، پرامپت‌های استاندارد<sup>۱۰۰</sup> غالباً پاسخ‌هایی تولید می‌کنند که ارتفاع ورودی را در نظر نمی‌گیرند. برعکس، هنگامی که از مدل خواسته می‌شود که ابتدا اطلاعات مربوطه را تولید کند و سپس از این اطلاعات تولید شده در پرامپت استفاده کند، خروجی‌ها با انسجام منطقی و جامعیت بیشتری همراه هستند. این روش به‌ویژه مدل را تحریک می‌کند تا عوامل بارز مانند «ارتفاع ورودی<sup>۱۰۱</sup>» را مدنظر قرار دهد.

مهندسی پرامپت به دو شکل دستی و خودکار<sup>۱۰۲</sup> انجام می‌شود. در روش دستی جهت یافتن یک پرامپت خوب از روش آزمون و خطا استفاده می‌شود. روش‌های معرفی شده تاکنون، از نوع دستی بوده است. برای جلوگیری از تکرار آزمون و خطا به صورت دستی، محققان، اخیراً روش‌های مختلفی را برای خودکارسازی این فرآیند توسعه داده‌اند که در ادامه به برخی از آنها اشاره شده است.

– به روز رسانی گرادیان: برخی از روش‌ها از رویکردهای مبتنی بر گرادیان برای بهینه‌سازی دستورات استفاده می‌کنند و امکان تعاملی موثرتر با مدل هوش مصنوعی را فراهم می‌کنند [۶۳].

<sup>93</sup> . A Minimal Prompt

<sup>94</sup> . Symbolic Manipulation

<sup>95</sup> . Compositional generalization

<sup>96</sup> . Mathematical Reasoning

<sup>97</sup> . The Least-to-Most Prompting Paradigm

<sup>98</sup> . Automatic Instruction Generation Method

<sup>99</sup> . Generated Knowledge

<sup>100</sup> . Standard Prompts

<sup>101</sup> . Entrance Height

<sup>102</sup> . Manual and Automatic Prompts

- یادگیری تقویتی<sup>۱۰۲</sup>: تکنیک‌هایی مانند بهینه سازی سریع گسسته مبتنی بر خط مشی متشکل از دیالوگ (DP2O) شامل یادگیری تقویتی برای بهبود تکراری دستورات از طریق تعامل با مدل است [۶۴].

- پرامپت نرم<sup>۱۰۴</sup>: روش‌هایی مانند AutoPrompt از یک مدل زبان منجمد استفاده می‌کنند و «توکن‌های محرک» خاص را در طول آموزش به روز می‌کنند تا اثربخشی سریع را افزایش دهند [۶۳].

همچنین از تکنیک‌هایی مانند ویرایش تدریجی<sup>۱۰۵</sup> [۶۲]، جستجوی الگوریتمی<sup>۱۰۶</sup> [۶۵]، تولید نمونه‌های درون‌متنی به صورت تطبیقی<sup>۱۰۷</sup> [۶۶] در روش‌های مهندسی پرامپت خودکار استفاده شده است. اخیراً نیز یک خط تحقیق بر روی متا-پرامپتینگ<sup>۱۰۸</sup> مدل‌های زبانی بزرگ برای مهندسی پرامپت خودکار ارائه شده است [۶۷].

یکی از مهم‌ترین مزیت‌های مهندسی پرامپت خودکار عبارت است از کارآمدسازی مهندسی پرامپت که این امر باعث صرفه جویی در زمان و منابع می‌شود و امکان تکرار و اصلاح سریعتر را فراهم می‌کند و همچنین به حفظ ثبات در کیفیت کمک می‌کند، که برای دستیابی به خروجی‌های قابل اعتماد از مدل‌های هوش مصنوعی بسیار مهم است. روش مهندسی پرامپت خودکار نشان دهنده پیشرفت قابل توجهی در نحوه تعامل کاربران با ابزارهای هوش مصنوعی، به ویژه در پردازش زبان طبیعی است. با استفاده از تکنیک‌های خودکار برای بهینه سازی دستورات، این روش اثربخشی و کارایی سیستم‌های هوش مصنوعی تولیدی را افزایش می‌دهد و راه را برای برنامه‌های کاربردی پیچیده تر در زمینه‌های مختلف هموار می‌کند.

## ۴) کارایی روش‌های مهندسی پرامپت<sup>۱۰۹</sup>

### ۴-۱) شاخص‌های ارزیابی

جهت سنجش اثربخشی روش‌های پرامپت‌نویسی به کمک ابزارهای «تولید محتوای مبتنی بر هوش مصنوعی<sup>۱۱۰</sup>»، روش‌های ارزیابی به طور کلی به دو دسته‌ی ارزیابی ذهنی و ارزیابی خودکار<sup>۱۱۱</sup> تقسیم می‌شوند. ارزیابی‌های ذهنی توسط انسان انجام می‌شود. یک ارزیاب می‌تواند با خواندن محتوای تولید شده توسط ابزار هوش مصنوعی، کیفیت محتوای تولید شده را ارزیابی کرده و آن را از نظر کیفیت نمره‌گذاری کنند. ارزیابی‌های ذهنی، شامل جنبه‌های دقت، نوآوری و ارتباط می‌باشند [۶۸]. در ارزیابی ذهنی معمولاً از تحلیل احساسات و نظرکاوی<sup>۱۱۲</sup> استفاده می‌شود که در آن کاربران عملکرد مدل را ارزیابی کرده و نظرات خود را بیان می‌کنند. این ارزیابی امکان بهبود مستمر این سیستم‌ها را فراهم می‌کند. با این حال، با توجه به پتانسیل آن برای انتشار داده‌های مغرضانه یا اشتباه و پیامدهای آن برای سرقت ادبی و حق چاپ، پیامدهای اخلاقی گسترده‌ای، نیز برای استفاده از آن وجود دارد [۶۹، ۷۰]. در اصل، مدل‌های زبانی مولد<sup>۱۱۳</sup>، فرصت‌های قابل توجهی

<sup>103</sup> . Reinforcement Learning

<sup>104</sup> Soft-Prompting

<sup>105</sup> . Incremental Editing

<sup>106</sup> . Algorithmic Search

<sup>107</sup> . Generating in-Context Demonstrations Adaptively

<sup>108</sup> . Meta-Prompting LLMs

<sup>109</sup> . Assessing the Efficacy of Prompt Methods

<sup>110</sup> . AIGC: Artificial Intelligence Generation Content

<sup>111</sup> . Subjective and Objective Evaluations

<sup>112</sup> . Sentiment Analysis and Opinion Mining

<sup>113</sup> . Generative Language Models

را در زمینه‌های مختلف به وجود آورده‌اند، اما به اجرای دقیق، نظارت اخلاقی، و سازگاری پاسخ برای اطمینان از تأثیر مثبت و قابلیت اطمینان آنها نیاز است [۷۱، ۷۲]. ارزیابی ذهنی معمولاً قابل اعتمادتر از ارزیابی عینی است، اما همچنین هزینه‌بر و زمان‌بر است.

در مقابل، روش‌های ارزیابی خودکار، از «الگوریتم‌های یادگیری ماشین»<sup>۱۱۴</sup> برای نمره‌گذاری کیفیت متن تولید شده استفاده می‌کنند. هرچند، این معیارهای خودکار اغلب قادر به درک کامل نتایج ارزیابی‌های انسانی نیستند [۷۳] اما در مقابل روش‌های ذهنی از دقت بالاتری برخوردار هستند. این ارزیابی‌ها از معیارهای خودکار مانند BLEU<sup>۱۱۵</sup> [۷۴] بهره می‌برند که به خروجی‌های تولید شده توسط سیستم نمره می‌دهد و راهی راحت و سریع برای مقایسه ابزارهای مختلف و نظارت بر میزان پیشرفت‌های آنها، فراهم می‌آورد. دیگر ارزیابی‌ها مانند ROUGE<sup>۱۱۶</sup> [۷۵] و METEOR<sup>۱۱۷</sup> [۷۶]، مشابهت بین متن تولید شده و متن مرجع را ارزیابی می‌کنند. روش‌های ارزیابی جدیدتر، مانند BERTScore [۷۷]، به ارزیابی در سطح بالاتری از نظر معنایی می‌پردازند. ارزیابی خودکار هزینه کمتری دارد و سریع‌تر از ارزیابی ذهنی است. در نهایت، بهترین روش برای «ارزیابی کیفیت خروجی مدل‌های زبان»<sup>۱۱۸</sup> بستگی به کاربرد خاص آن دارد [۷۸]. اگر کیفیت مهم‌ترین عامل باشد، استفاده از ارزیابان انسانی گزینه بهتری است. اگر هزینه و زمان مهم‌ترین عوامل باشند، استفاده از روش‌های ارزیابی خودکار مناسب‌تر است.

#### ۴-۲) مقایسه روش‌ها

در حوزه مهندسی پرامپت، ارزیابی و مقایسه سیستماتیک رویکردهای مختلف پرامپت‌ها به دلیل برخی محدودیت‌ها انجام نشده است. مطالعات عمدتاً بر روی مدل‌ها یا وظایف خاصی آزمایش شده است و همچنین از معیارهای ارزیابی متفاوتی جهت ارزیابی هر یک از روش‌ها استفاده شده است که مقایسه‌پذیری بین روش‌ها را محدود می‌کند [۷۹، ۸۰]. همان‌طور که اشاره شد، مطالعات پیشین عمدتاً بر «طراحی و بهینه‌سازی روش‌های خاص پرامپت‌ها»<sup>۱۱۹</sup> تمرکز داشته‌اند به جز معدود پژوهش‌هایی مانند مطالعه [۸۱] که یک چارچوب ارزیابی کلی به نام InstructEval [۸۱] را پیشنهاد داده است.

این چارچوب امکان ارزیابی تکنیک‌های پرامپت در میان مدل‌ها و وظایف مختلف را فراهم می‌آورد. نتایج مطالعه InstructEval نشان داده است روش‌های چند مثالی<sup>۱۲۰</sup> با حذف پرامپت‌های کلی و غیر مرتبط معمولاً عملکرد بهتری نسبت به سایر روش‌ها دارد. یکی دیگر از نتایج این پژوهش مربوط به روش بدون مثال است که نشان می‌دهد اگر پرامپت‌ها توسط متخصصان به دقت نوشته شوند می‌توانند عملکرد را به طور قابل توجهی بهبود بخشند. در رابطه با پرامپت‌نویسی خودکار این نتیجه بیان شده است که عملکرد روش‌های تولید پرامپت خودکار ناپایدار است و در مدل‌ها و انواع وظایف مختلف متفاوت است، که نشان‌دهنده عدم تعمیم‌پذیری آنها است. در این پژوهش هرچند برخی روش‌های مهندسی

<sup>114</sup> . Machine Learning Algorithms

<sup>115</sup> . BiLingual Evaluation Understudy

<sup>116</sup> . Recall-Oriented Understudy for Gisting Evaluation

<sup>117</sup> . Metric for Evaluation of Translation with Explicit ORDERing

<sup>118</sup> . Evaluate the Quality of LLM Output

<sup>119</sup> . Design and Optimization of Specific Methods of Prompts

<sup>120</sup> . Few-Shot Methods

پرامپت مورد ارزیابی قرار گرفته‌اند اما همچنین اشاره شده است نیاز به الگوهای ارزیابی جهانی و قابل اعتماد بیشتری برای طراحی بهینه پرامپت‌ها وجود دارد.

### ۳-۴) کاهش توهم<sup>۱۲۱</sup>

یکی از چالش‌های موجود در حوزه هوش مصنوعی مولد، تولید توهمات است. توهمات به وجود اطلاعات غیرواقعی یا نادرست در خروجی مدل‌هایی مانند GPT-4 گفته می‌شود [۸۲]. این خروجی‌ها معمولاً از نظر گرامری صحیح بوده اما با واقعیت‌ها ناسازگار بوده و یا از «داده‌های دنیای واقعی»<sup>۱۲۲</sup> پشتیبانی نمی‌کنند.

اخیراً جهت مبارزه با این چالش روش‌های مختلفی ارائه شده است [۸۳، ۸۴، ۸۵]. به عنوان نمونه در برخی پژوهش‌ها، روش «تقویت بازیابی»<sup>۱۲۳</sup> به کار می‌رود که هدف آن ادغام اطلاعات خارجی به‌روز، در ورودی مدل است [۸۶، ۸۷]. روش‌های زنجیره‌ای از فکر و انواع آن نیز می‌تواند توهمات را با اعمال نفوذ دانش درونی مدل، کاهش دهند [۸۸، ۸۹، ۹۰، ۹۱، ۹۲، ۹۳]. در پژوهش دیگری [۹۴]، نشان داده شده است که GPT-3 می‌تواند با مطالعه اجزای مختلف معماری‌ها مانند تولید تقویت‌شده با بازیابی<sup>۱۲۴</sup> [۹۵]، ادغام در رمزگشا<sup>125</sup> [۹۶]، Seq2seq [۹۷-۹۸]، توهمات را کاهش دهد. همچنین [۹۹] تکنیک زنجیره‌ای تأیید<sup>۱۲۶</sup> را برای کاهش توهمات توسعه داده است که نشان می‌دهد استفاده از ابزارهایی مانند تقویت بازیابی در مرحله اجرای تأیید، احتمالاً منجر به کسب نتایج بهتری خواهد شد.

تولید تقویت‌شده با بازیابی (RAG) یک روش پیچیده است که توانایی‌های مدل‌های تولیدی را با ادغام فرایندهای بازیابی اطلاعات خارجی افزایش می‌دهد. این رویکرد به ویژه برای وظایفی که نیاز به دانش به روز یا اطلاعات دقیق دارند که ممکن است در داده‌های آموزشی مدل وجود نداشته باشد، مفید است.

مکانیسم‌های اصلی RAG از یک فرایند سه مرحله‌ای تکراری استفاده می‌کند: ۱- ایجاد جملات موثر: مدل ابتدا یک جمله موقت ایجاد می‌کند که به عنوان یک طرح محتوا برای خروجی بعدی عمل می‌کند. ۲- بازیابی دانش خارجی: با استفاده از جمله موقت به عنوان یک پرس‌وجو، مدل اطلاعات مربوطه را از منابع خارجی بازیابی می‌کند. ۳- تزریق دانش بازیابی شده: اطلاعات بازیابی شده سپس در جمله موقت ادغام می‌شود تا جمله خروجی بعدی تولید شود. این فرایند تکراری امکان پالایش مداوم و تقویت محتوای تولید شده را فراهم می‌کند [96]. RAG دارای مزایای بسیاری است که یکی از مهم‌ترین آنها بهبود دقت مدل است. در این رویکرد با دسترسی به دانش خارجی، سیستم‌های RAG می‌توانند پاسخ‌های دقیق‌تر به ویژه برای پرسش‌های پیچیده‌ای که نیاز به اطلاعات دقیق دارند، ارائه دهند. یکی دیگر از مزیت‌های این روش دسترسی به دانش پویا است. این رویکرد به مدل‌ها اجازه می‌دهد به آخرین اطلاعات دسترسی یابند و آنها را با تغییرات و تحولات دنیای واقعی سازگار کنند. بنابراین، RAG نشان‌دهنده یک جهش قابل توجهی به جلو در قابلیت‌های

<sup>121</sup> . Reduce Hallucinations

<sup>122</sup> . Real-World Data

<sup>123</sup> . Retrieval Augmentation

<sup>124</sup> . RAG: Retrieval Augmented Generation

<sup>125</sup> . FID: Fusion-in Decoder

<sup>126</sup> . CoVe

هوش مصنوعی مولد است. با ترکیب موثر تکنیک‌های بازیابی با فرآیندهای تولیدی، RAG نه تنها کیفیت محتوای تولید شده توسط هوش مصنوعی را بهبود می‌بخشد بلکه آن را برای کاربردهای مختلف در صنعت و تحقیق آماده می‌کند. این روش نمونه‌ای از پتانسیل ادغام منابع دانش خارجی برای افزایش عملکرد سیستم‌های هوش مصنوعی است.

## (۵) کاربردها<sup>۱۲۷</sup>

این بخش به بررسی و دسته‌بندی روش‌های مهندسی پرامپت بر اساس کاربردهای عملی آن‌ها تمرکز دارد. پیش از این بررسی‌های سیستماتیک زیادی در مورد مهندسی پرامپت وجود نداشته است. پژوهش [۱۰۰] ۲۹ روش مهندسی پرامپت را بر اساس کاربرد آنها بررسی کرده است. نتایج حاصل از این پژوهش یک طبقه‌بندی بسیار گسترده از وظایف متعدد پردازش زبان طبیعی<sup>۱۲۸</sup> است. به عنوان مثال، یکی از کاربردهایی که بررسی شده است، استدلال و منطق است که می‌تواند وظایف پردازش زبان طبیعی، استدلال عرفی<sup>۱۲۹</sup>، استدلال حل مسئله ریاضی<sup>۱۳۰</sup>، استدلال چند هاپ<sup>۱۳۱</sup> را داشته باشد. همچنین در پژوهش دیگری [۱۰۱] ۴۴ مقاله تحقیقاتی تجزیه و تحلیل شده که اکثر آنها در دو سال گذشته منتشر شده‌اند و شامل ۳۹ تکنیک پرامپت هستند که بر روی ۲۹ وظیفه‌ی مختلف پردازش زبان طبیعی اعمال شده‌اند.

### (۱-۵) طبقه‌بندی جفت متن<sup>۱۳۲</sup>

طبقه‌بندی جفت-متن وظیفه تعیین رابطه طبقاتی بین دو جمله را بر عهده دارد [۱۰۲]. در حقیقت مهم‌ترین وظیفه‌ی دسته‌بندی جفت متن که تحقق اهداف آن را به دنبال دارد، پیش‌بینی رابطه<sup>۱۳۳</sup> (مانند شباهت، نتیجه‌گیری) بین دو جمله داده شده است. این وظایف شامل شناسایی عبارات<sup>۱۳۴</sup>، استنتاج زبان طبیعی، پیش‌بینی شباهت متنی و موارد دیگر می‌شوند. مشابه با وظایف دسته‌بندی متن، در وظایف دسته‌بندی جفت متن نیز پرامپت‌های بسته<sup>۱۳۵</sup> به طور گسترده استفاده می‌شود [۱۰۳، ۱۰۴]. در توضیح پرامپت‌های بسته لازم بذکر است که پرامپت‌های باز و بسته<sup>۱۳۶</sup> دو نوع هستند که می‌توانیم برای دریافت پاسخ به کاربر بدهیم. یک پرامپت باز مانند «چگونه می‌توانم کمک کنم؟» که انتخاب نامحدودی از پاسخ‌ها را در اختیار کاربر قرار می‌دهد، در حالی که یک پرامپت بسته مانند «بله یا نه؟» مجموعه بسته‌ای از پاسخ‌های بالقوه را برمی‌انگیزد. در مورد مهندسی پرامپت، محققان عمدتاً بر جستجوی الگوها در محیط یادگیری چندمثالی تمرکز دارند و فضای پاسخ Z معمولاً به صورت دستی از واژگان انتخاب می‌شود.

### (۲-۵) دسته‌بندی متن<sup>۱۳۷</sup>

مدل‌های زبانی بزرگ چنان تغییر گامی در نحوه تعامل مدل‌های یادگیری ماشینی با زبان طبیعی را رقم زده‌اند که منجر به

<sup>127</sup> . Application

<sup>128</sup> . NLP: Natural Language Processing

<sup>129</sup> . Commonsense Reasoning

<sup>130</sup> . Mathematical Problem Solving

<sup>131</sup> . Multi-Hop Reasoning

<sup>132</sup> . Text Pair Classification

<sup>133</sup> . Relationship Prediction

<sup>134</sup> . Identification of Paraphrases

<sup>135</sup> . Cloze Prompts

<sup>136</sup> . Open and Close Prompts

<sup>137</sup> . Text Classification

جرقه‌ی انقلابی در کاربردهای هوش مصنوعی شده‌اند. امروزه میلیون‌ها نفر از مدل‌های زبانی بزرگ برای انجام وظایفی مانند برنامه‌ریزی سفر، نوشتن ایمیل به یک همکار و تهیه نامه برای درخواست شغلی استفاده می‌کنند. از آنجایی که آنها در عملکردهای خاصی از پردازش زبان طبیعی بسیار ماهر هستند، فرض بر این است که ممکن است در سایر عملکردها نیز موثر باشند. در حوزه‌ی دسته‌بندی متن، تحقیقات پیشین عمدتاً از پرامپت‌های از پیش تعیین شده (پرامپت‌های بسته) استفاده کرده‌اند. این تحقیقات هر دو جنبه مهم طراحی ساختار پرامپت (مهندسی قالب پرامپت<sup>۱۳۸</sup>) [۱۰۵، ۱۰۶، ۱۰۷] و بهینه‌سازی نحوه پاسخگویی پرامپت (مهندسی پاسخ پرامپت<sup>۱۳۹</sup>) [۱۰۵، ۱۰۸، ۱۰۹] را به طور گسترده بررسی کرده‌اند. بیشتر این تحقیقات به بررسی کارایی «یادگیری مبتنی بر پرامپت<sup>۱۴۰</sup>» برای دسته‌بندی متن در زمینه‌ای با شرایط چند مثالی و با استفاده از استراتژی‌های «تنظیم مدل زبانی با پرامپت ثابت<sup>۱۴۱</sup>» پرداخته‌اند. دسته‌بندی متن، که به طور گسترده برای تشخیص هرزنامه، تحلیل احساسات و اهداف دیگر استفاده می‌شود، یکی از اساسی‌ترین کاربردهای «پردازش زبان طبیعی» است. توجه به این نکته حائز اهمیت است که مدل‌های زبانی بزرگ برای دسته‌بندی متن ساخته نشده‌اند، اما قدرت آنها در پردازش زبان طبیعی ممکن است آنها را قادر سازد تا کار را در سطح بالایی انجام دهند. اگر چنین باشد، آنها می‌توانند مجموعه جدیدی از برنامه‌های کاربردی برای این فناوری ایجاد کنند.

### ۵-۳) شناسایی موجودیت نام‌دار<sup>۱۴۲</sup>

«شناسایی موجودیت نام‌دار» بخشی از پردازش زبان طبیعی است. در واقع، یکی از اولین گام‌ها در فرآیند استخراج اطلاعات است که منجر به شناسایی و دسته‌بندی موجودیت‌های دارای نام در متن، به دسته‌های از پیش تعریف شده مانند اسامی افراد، سازمان‌ها، مکان‌ها، بیان زمان‌ها، مقادیر، ارزش‌های پولی، درصدها و دیگر موارد می‌شود. «شناسایی موجودیت نام‌دار» وظیفه‌ای است که هدف آن تشخیص موجودیت‌های نام‌دار<sup>۱۴۳</sup> (مانند نام شخص، مکان) در یک جمله مشخص است. یکی از چالش‌های استفاده از یادگیری مبتنی بر پرامپت در وظایف برچسب‌گذاری، مانند شناسایی موجودیت نام‌دار، این است که برخلاف وظایف دسته‌بندی، (۱) هر واحدی که باید پیش‌بینی شود، یک توکن یا بازه از متن است نه کل متن ورودی، و (۲) رابطه پنهانی بین برچسب‌های توکن‌ها در متن وجود دارد. به طور کلی، کاربرد یادگیری مبتنی بر پرامپت در وظایف برچسب‌گذاری هنوز به طور کامل مورد بررسی قرار نگرفته است. اخیراً پژوهشگران [۱۱۰] یک مدل «شناسایی موجودیت نام‌دار» مبتنی بر قالب با استفاده از مدل BERT را پیشنهاد کرده‌اند که بازه‌های متنی را شمارش می‌کند و احتمال تولید هر نوع موجودیت را در قالب‌های دستی ساخته شده مورد بررسی قرار می‌دهد. به عنوان مثال، برای تعیین نوع موجودیت «Mike» در جمله «Mike went to New York yesterday»، آنها از قالب «Mike is a [Z] entity» استفاده می‌کنند و فضای پاسخ Z شامل مقادیری مانند «person» یا «organization» است. همانطور که ذکر شد، «شناسایی

<sup>138</sup> . Prompt Template Engineering

<sup>139</sup> . Prompt Response Engineering

<sup>140</sup> . Prompt-Based Learning

<sup>141</sup> . Fixed-Prompt LM Tuning

<sup>142</sup> . NER: Named Entity Recognition

<sup>143</sup> . Named Entities

موجودیت نام‌دار» در زمینه‌های گوناگون «پردازش زبان طبیعی» کاربرد دارد و می‌تواند در پاسخگویی به بسیاری از پرسش‌های جهان واقعی مانند مواردی که در زیر بیان شده کمک کند.

#### ۴-۵) پیش‌بینی ساختار<sup>۱۴۴</sup>

طی چند سال گذشته، حوزه‌ی مطالعاتی پیش‌بینی ساختاریافته در پردازش زبان طبیعی با مدل‌های گرافیکی احتمالی پیچیده، شبکه‌های مبتنی بر انرژی<sup>۱۴۵</sup> و ترکیب آن با رویکردهای مبتنی بر یادگیری عمیق<sup>۱۴۶</sup>، شاهد پیشرفت عظیمی بوده است. برخی تکنیک‌های اصلی پیش‌بینی ساخت یافته و کاربردهای آن در پردازش زبان طبیعی مانند تجزیه، برجسب گذاری توالی، تولید متن و وظایف توالی قابل طرح کردن است [۱۱۱].

تحلیل معنایی<sup>۱۴۷</sup> به معنای تولید یک نمایش معنایی ساختاریافته بر اساس ورودی زبان طبیعی است. در این بخش، پژوهش [۱۱۲] با رویکرد استفاده از مدل‌های زبانی به صورت چندمثالی، این مسأله را به عنوان وظیفه بازنویسی<sup>۱۴۸</sup> مطرح کرده است. همچنین، فرآیند تولید خروجی را محدود به تولید مقادیر معتبر مطابق با یک دستور زبان مشخص کرده‌اند. آنها در چارچوب یادگیری در متن<sup>۱۴۹</sup> آزمایش‌هایی را انجام داده‌اند و از پرامپت‌های پاسخ داده شده که از نظر معنایی به یک نمونه آزمایشی نزدیک هستند، استفاده کرده‌اند. نتایج این تحقیقات نشان می‌دهد که بازنویسی برای وظایف تحلیل معنایی با استفاده از مدل‌های زبانی پیش‌تمرین شده موثر است.

#### ۵-۵) تولید متن<sup>۱۵۰</sup>

تولید متن یکی از وظایف مرتبط است که کاربردهای متعددی دارد و معمولاً بر اساس اطلاعات دیگری صورت می‌گیرد. روش‌های استفاده از پرامپت به راحتی می‌توانند در این وظایف اعمال شوند، به این صورت که پرامپت‌ها به همراه مدل‌های زبانی خود بازگشتی پیش‌تمرین شده مورد استفاده قرار می‌گیرند. پژوهشگران در [۱۱۳] نشان دادند که این مدل‌ها توانایی چشمگیری در انجام وظایفی مانند خلاصه‌سازی متن و ترجمه ماشینی با استفاده از پرامپت‌هایی نظیر «ترجمه به فرانسه، [X]»، دارند. تولید متن همچنین می‌تواند برای تقویت داده‌های آموزشی برای مدل‌های یادگیری ماشین، بهبود عملکرد و قابلیت‌های تعمیم آنها استفاده شود. در [۱۱۴] پژوهشگران با انجام یادگیری درون‌متنی<sup>۱۵۱</sup> برای تولید متن، پرامپت‌هایی با قالب‌های دستی ایجاد کرده و ورودی را با چندین پرامپت پاسخ داده شده تقویت کردند. همچنین یکی از تکنیک‌های تولید متن این است که می‌تواند توصیه‌ها، تبلیغات و پیام‌های شخصی‌سازی شده را متناسب با تک تک کاربران را ایجاد کنند و تعامل و رضایت کاربران را افزایش دهند. در [۱۱۵] [۱۱۶] نیز تنظیم مدل‌های زبانی با پرامپت ثابت را برای خلاصه‌سازی متن با استفاده از چند نمونه و قالب‌های دستی بررسی کردند. علی‌رغم توانایی مدل‌های زبانی بزرگ در تولید متون دستوری صحیح و روان، برنامه‌های کاربردی دنیای واقعی اغلب نیاز به کنترل خاص‌تری بر ویژگی‌های متن، فراتر از

<sup>144</sup> . Structure Prediction

<sup>145</sup> . Energy-Based Networks

<sup>146</sup> . Deep Learning-Based Approaches

<sup>147</sup> . Semantic Parsing

<sup>148</sup> . Paraphrasing Task

<sup>149</sup> . In-Context Learning

<sup>150</sup> . Text Generation

<sup>151</sup> . In-text Learning



ویژگی روان بودن دارند، که ممکن است با اجازه دادن به مدل‌های زبانی بزرگ برای تولید متن آزادانه به دست نیاید [۱۱۷].

تحقیقاتی بر روی تنظیم پرامپت ثابت در مدل‌های زبانی برای خلاصه‌سازی متن و تولید داده به متن در تنظیمات چندنمونه‌ای انجام دادند، که در آن، توکن‌های پیشوندی<sup>۱۵۲</sup> قابل یادگیری به ورودی اضافه می‌شوند، در حالی که «پارامترهای مدل‌های زبانی پیش‌تمرین شده<sup>۱۵۳</sup>» ثابت نگه داشته می‌شوند. پژوهشگران در [۱۱۸] «استراتژی تنظیم پرامپت + مدل زبانی<sup>۱۵۴</sup>» را در وظیفه خلاصه‌سازی متن بررسی کردند، که در آن پرامپت‌های پیشوندی قابل یادگیری<sup>۱۵۵</sup> استفاده می‌شوند و توسط انواع مختلفی از سیگنال‌های راهنما تنظیم می‌شوند که می‌توانند همراه با پارامترهای مدل‌های زبانی پیش‌تمرین شده به‌روزرسانی شوند.

چالش دیگری که تولید متن بدون محدودیت ایجاد می‌کند، تولید سهوی متن مضر است. با زیاد شدن متن‌های گمراه‌کننده، مضر و افتراآمیز که به صورت آنلاین در دسترس هستند، این سوگیری‌های مضر به داده‌های آموزشی مدل‌های زبانی راه پیدا می‌کنند. بر همین اساس، توانایی کنترل تولید متن، هدایت مدل‌های زبان از تولید خروجی‌های مغرضانه و به نوعی سمی، به طور فزاینده‌ای حائز اهمیت است [۱۱۷].

## ۵-۶) یادگیری چندمدلی<sup>۱۵۶</sup>

پرامپت‌ها نقش مهمی در افزایش کنترل، سازگاری و کاربرد مقیاس‌پذیر مدل‌های زبانی بزرگ دارند. در سال‌های اخیر، استراتژی‌هایی شامل پرامپت‌ها نیز برای مدل‌های بصری اعمال شده‌اند. با این حال، میزانی که ادغام پرامپت‌ها چندوجهی (به عنوان مثال پیام‌های متن یا تصویر) می‌تواند عملکرد پایین دستی کار را در مدل‌های بصری بهبود بخشد، به صورت سیستماتیک پرداخته نشده است [۱۱۹]. از جمله پژوهش‌های انجام شده در این حوزه پژوهش [۱۲۰] است که کاربرد یادگیری پرامپت با دستورالعمل را از پردازش زبان طبیعی متنی به تنظیمات چندمدلی (بینایی و زبان) منتقل می‌کنند. به طور کلی، آن‌ها از استراتژی تنظیم دستورالعمل مدل زبانی ثابت به همراه تکنیک‌های تقویت دستورالعمل استفاده می‌کنند. آن‌ها به طور خاص هر تصویر را به صورت دنباله‌ای از جاسازی‌های پیوسته نمایش می‌دهند و یک مدل زبانی پیش‌آموزش دیده با پارامترهای ثابت را با این پیشوند برای تولید متونی مانند توضیحات تصویر، پرامپت‌نویسی می‌کنند. نتایج تجربی نشان می‌دهد که مدل توانایی یادگیری چند مثالی را دارد: با کمک چندین نمایش (پرامپت‌های داده شده)، سیستم می‌تواند به سرعت کلمات مربوط به اشیاء جدید و دسته‌بندی‌های بصری نو را یاد بگیرد.

## ۵-۷) پاسخ به سوالات<sup>۱۵۷</sup>

پاسخگویی به پرسش، نوعی سیستم است که در آن کاربر می‌تواند با استفاده از زبان طبیعی سوالی را مطرح کند و سیستم پاسخی مختصر و صحیح ارائه می‌دهد [۱۲۱].

<sup>152</sup> . Prefix Tokens

<sup>153</sup> . Parameters in Pre-Trained LMs

<sup>154</sup> . The Prompt+LM Tuning Strategy

<sup>155</sup> . Learnable Prefix Prompts

<sup>156</sup> . Multi-Modal Learning

<sup>157</sup> . QA: Question Answering

روش پاسخ به سوالات (QA) به دنبال پاسخ‌گویی به یک سوال ورودی مشخص است که معمولاً بر اساس یک سند متنی انجام می‌شود. این فرآیند می‌تواند به شکل‌های مختلفی انجام گیرد که چند نمونه از آن را به عنوان مثال ذکر خواهیم کرد. مانند QA استخراجی که محتوای مرتبط با پاسخ را از سند متن پیدا می‌کند [۱۲۲] (SQuAD)؛ QA چندگزینه‌ای که مدل باید از بین چند گزینه یک پاسخ را انتخاب کند [۱۲۳] (RACE)؛ و QA آزاد که در آن مدل می‌تواند یک متن دلخواه را به عنوان پاسخ ارائه دهد (Narrative QA). به طور کلی، این فرمت‌های مختلف به وسیله چارچوب‌های مدل-سازی متفاوتی مدیریت شده‌اند.

یک سیستم QA با موتور جستجو متفاوت است زیرا کاربر یک سوال می‌پرسد و خروجی به جای لیستی از اسناد مربوطه، یک پاسخ دقیق است [۱۲۱]. یکی از مزایای حل مسائل QA با استفاده از مدل‌های زبانی بزرگ، به ویژه با استفاده از روش‌های پرامپت نویسی، این است که فرمت‌های مختلف مسائل QA می‌توانند در یک چارچوب واحد حل شوند. به عنوان مثال، [۱۲۴] بسیاری از مسائل QA را به عنوان یک مشکل تولید متن بازنویسی کرده و مدل‌های پیش‌آموزش دیده مبتنی بر توالی به توالی<sup>۱۵۸</sup> مانند T5 را با استفاده از پرسش‌ها و متون مناسب از زمینه و سوالات، تنظیم دقیق کرده‌اند. پژوهش [۱۲۵] به بررسی دقیق‌تری از چنین سیستم‌های QA مبتنی بر پرامپت با استفاده از مدل‌های پیش‌آموزش دیده توالی به توالی (T5، BART، و GPT2) پرداخته و مشاهده کرده‌اند که احتمال‌هایی که این مدل‌های پیش‌آموزش دیده در مسائل QA تولید می‌کنند، خیلی پیش‌بینی‌کننده این موضوع نیستند که آیا مدل پاسخ درستی ارائه کرده است یا خیر.

## ۵-۸) ساخت مجموعه داده<sup>۱۵۹</sup>

اگر دو نفر را تصور کنید که یکی تنبل و دیگری ورزشکار باشد و همزمان در یک مدرسه فوتبال ثبت نام کرده باشند، نسبت به هر یک از آنها تحلیلی در ذهن شما شکل خواهد گرفت. نکته‌ی مشترک در این دو این است که هر دو تا به حال فوتبال بازی نکرده‌اند و مهارتی در این خصوص ندارند اما تفاوت اصلی این دو این است که ورزشکار تحرک بیشتری نسبت به فرد تنبل داشته است. مهارت‌های ورزشکار مانند استقامت، سرعت و ... برای یادگیری فوتبال بسیار مفید است و در واقع ورزشکار از تمرینات پیشین خود سود می‌برد. این امر به نوعی توضیح علت استفاده از مدل‌های پیش‌آموزش دیده است که در کاری متفاوت از وظیفه مورد نظر آموزش می‌بیند، اما نقطه شروع بسیار مفیدی را ارائه می‌دهد؛ زیرا ویژگی‌هایی که در حین آموزش برای کار قبلی آموخته برای کار جدید نیز مفید خواهد بود. در واقع به همین دلیل است که شیک و شوتز [۱۲۶] پیشنهاد می‌دهند که از مدل‌های زبانی پیش‌آموزش دیده برای تولید مجموعه داده‌ها بر اساس دستورالعمل‌های خاص استفاده شود. به عنوان مثال، فرض کنید که یک مجموعه داده بدون برچسب داریم که هر نمونه‌ی آن یک جمله است. اگر بخواهیم مجموعه داده‌ای حاوی جفت جملات مشابه معنایی بسازیم، می‌توانیم از الگوی زیر برای هر جمله ورودی استفاده کنیم: «دو جمله بنویسید که یک معنی مشابه داشته باشند. [X][Z]» و سعی کنیم جمله‌ای تولید کنیم که معنای مشابهی با جمله ورودی داشته باشد.

<sup>158</sup> . Seq2seq

<sup>159</sup> . Dataset Construction

## ۵-۹) تطبیق دامنه<sup>۱۶۰</sup>

تطبیق دامنه به معنای انطباق یک مدل از یک دامنه (برای مثال، متن‌های خبری) به دامنه‌ای دیگر (مثل متن‌های شبکه‌های اجتماعی) است. در حقیقت چنین تعریف می‌شود که تطبیق دامنه یک زیر شاخه از یادگیری انتقال است که با مشکل انطباق یک مدل آموزش دیده بر روی داده‌های یک دامنه (دامنه منبع<sup>۱۶۱</sup>) برای عملکرد خوب روی داده‌های دامنه دیگر (دامنه هدف<sup>۱۶۲</sup>) سروکار دارد [۱۲۷]. در ادامه پژوهشگران برای تقویت ورودی متنی اصلی از «ویژگی‌های مرتبط با دامنه<sup>۱۶۳</sup>» یا DRF خودساخته استفاده می‌کنند و مشکل برچسب‌گذاری توالی<sup>۱۶۴</sup> را به عنوان یک مسئله توالی به توالی با استفاده از مدل زبانی پیش آموزش دیده<sup>۱۶۵</sup> حل می‌کنند. «ویژگی‌های مرتبط با دامنه» به ویژگی‌ها یا اطلاعاتی اشاره دارد که به طور خاص برای یک «دامنه خاص» مرتبط و مناسب هستند و در تطبیق یا تنظیم مدل‌ها در دامنه‌های خاص استفاده می‌شوند. این روش به مدل کمک می‌کند تا قابلیت تعمیم‌پذیری خود را به دامنه‌های جدید افزایش دهد [۱۲۸].

نتایج بدست آمده از بررسی کاربردهای مهندسی پرامپت در پردازش زبان طبیعی و توسعه‌ی مدل‌های زبانی بزرگ در جدول ۱ نمایش داده شده است.

جدول ۱. دسته‌بندی روش‌های مهندسی پرامپت در کاربردهای پردازش زبان‌های طبیعی

ردیف	کاربردهای پرامپت	روش‌های کلی مهندسی پرامپت	روش‌های فرعی مهندسی پرامپت
۱	دسته‌بندی متن	روش‌های پیشرفته مهندسی پرامپت	آموزش مبتنی بر مثال
۲	دسته‌بندی جفت متن	روش‌های پیشرفته مهندسی پرامپت	آموزش مبتنی بر مثال
۳	شناسایی موجودیت نام‌دار	روش‌های پایه‌ای مهندسی پرامپت	پرامپت کلیدواژه‌ای پرامپت قالب‌دار
۴	پیش‌بینی ساختار	روش‌های پایه‌ای مهندسی پرامپت	پرامپت قالب‌دار
۵	پاسخ به سوالات	روش پایه‌ای مهندسی پرامپت	نقش‌دهی به پرامپت
		روش‌های پیشرفته مهندسی پرامپت	درخت و گراف افکار زنجیره فکر آموزش مبتنی بر مثال
۶	تولید متن	روش‌های پایه‌ای مهندسی پرامپت	پرامپت کلیدواژه‌ای پرامپت قالب‌دار
۷	تطبیق دامنه	روش‌های پیشرفته مهندسی پرامپت	روش درخت و گراف افکار (TOT)
۸	ساخت مجموعه داده	روش‌های پیشرفته مهندسی پرامپت	آموزش مبتنی بر مثال
۹	یادگیری چند مدلی	روش‌های پیشرفته مهندسی پرامپت	آموزش مبتنی بر مثال

<sup>160</sup>. Domain Adaptation

<sup>161</sup>. Source Domain

<sup>162</sup>. Target Domain

<sup>163</sup>. Domain Relevance Features

<sup>164</sup>. Sequence Tagging

<sup>165</sup>. Seq2seq-Based Pre-Trained LMs

## ۶ نتیجه‌گیری و جمع‌بندی

در این پژوهش به بررسی مهندسی پرامپت به عنوان تکنیکی مهم در زمینه‌ی مدل‌های زبانی بزرگ و پردازش زبان طبیعی پرداخته شد. بررسی روش‌های مختلف مهندسی پرامپت در این پژوهش نشان داده است طراحی دقیق پرامپت‌ها می‌تواند به بهبود نتایج حاصل از ابزارهای هوشمند تولید محتوا و کاهش پدیده توهم ماشینی کمک کند. وقتی پرامپت‌ها به صورت بهینه طراحی شوند می‌توانند به سرعت تولید پاسخ کمک کرده و زمان پردازش را نیز کاهش دهند. همچنین در بخش کاربردهای مهندسی پرامپت نشان داده شده است که چگونه توسعه‌دهندگان می‌توانند با فهم بهتر از نیازهای کاربر، پرامپت‌هایی بسازند که بتواند بر اساس روش‌های مختلف مهندسی پرامپت از جمله پرسش و پاسخ، زنجیره فکر، تکرار و دیگر روش‌های ذکر شده، به سوالات و نیازهای کاربران پاسخ دهند. در برخی از حوزه‌های تخصصی و علمی لازم است پاسخ‌ها و محاسبات دقیقی به دست بیاید تا بتوان از آنها در حوزه‌های کاربردی حساس‌تر استفاده کرد؛ بنابراین پرامپت‌ها از این توانایی برخوردارند که با تنظیم مدل‌ها برای وظایف خاصی مانند ترجمه، خلاصه‌سازی، یا تولید متن مورد استفاده قرار بگیرند.

زمینه مهندسی پرامپت به سرعت در حال تکامل است و چندین روش و زمینه امیدوار کننده برای تحقیقات آینده وجود دارد که می‌تواند اثربخشی آن را افزایش دهد. بر اساس بررسی‌های انجام شده در ادامه چند راهکار برای پژوهش‌های آتی در این حوزه ارائه شده است.

- رویکردهای مشارکتی: در آینده می‌توان روش‌های مشارکتی را توسعه داد. یک روش مشارکتی می‌تواند روش‌های مهندسی پرامپت خودکار را با تخصص انسانی ترکیب کند. مشارکت کارشناسان حوزه در فرآیند مهندسی پرامپت می‌تواند منجر به دستورات موثرتر و مرتبط با زمینه مورد نظر شوند.
- طراحی کاربرمحور: کار آینده می‌تواند بر طراحی دستورالعمل‌هایی متمرکز شود که متناسب با نیازها و زمینه‌های خاص کاربر باشد، و تعامل بین کاربران و مدل‌های تولیدی را افزایش دهد.
- رسیدگی به ایمنی و امنیت: همانطور که روش‌های مهندسی پرامپت گسترده‌تر می‌شوند، توسعه پروتکل‌های ایمنی برای کاهش خطرات ناشی از سوء استفاده یا پیامدهای ناخواسته محتوای تولید شده باید به طور خاص مورد توجه قرار گیرد تا بتوان اطمینان حاصل کرد که دستورالعمل‌ها منجر به خروجی‌های مضر یا جانب‌دار نشوند.
- معیارهای ارزیابی پیشرفته: نیاز به توسعه معیارهای ارزیابی قوی‌تری وجود دارد که فراتر از معیارهای سنتی باشد. این معیارها باید اثربخشی دستورات را در زمینه‌های مختلف و تأثیر آنها بر کیفیت خروجی‌های تولید شده را ارزیابی نمایند.

با توجه به مطالب بیان شده در نهایت باید گفت آینده‌ی مهندسی پرامپت، با روندهای نوظهور مانند فرایادگیری<sup>۱۶۶</sup> و معماری ترکیبی<sup>۱۶۷</sup> که قابلیت‌های تقویت شده را نوید می‌دهد، پتانسیل بسیار زیادی دارد. مهندسی پرامپت می‌تواند به

<sup>166</sup>. Meta-Learning

<sup>167</sup>. Hybrid Prompting Architectures

ایجاد تعاملات طبیعی بین انسان و ماشین کمک کنند و در نهایت افزایش رضایت کاربران را به همراه خواهند داشت. با این حال، ملاحظات اخلاقی بسیار مهم هستند و بر توسعه و استقرار مسئولانه برای اطمینان از ادغام مثبت در زندگی ما تأکید دارند. در نهایت پیشنهاد می‌شود با توجه به نقش کلیدی مهندسی پرامپت در بهینه‌سازی و افزایش کارایی مدل‌های زبانی بزرگ، توسعه‌دهندگان و محققان از پتانسیل کامل این فناوری بهره‌برداری کنند.

## منابع

- [1] DK. Dwivedi, et al., (2023). So what if ChatGPT wrote it? Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy, *Int. J. Inf. Manag.*, 71, <https://doi.org/10.1016/j.ijinfomgt.2023.102642>.
- [2] Harrer, S., (2023). Attention is not all you need: the complicated case of ethical and using large language models in healthcare and medicine, *eBioMedicine*, 90, <https://doi.org/10.1016/j.ebiom.2023.104512>.
- [3] Andrew, G., (2023). Implications of ChatGPT and Large Language Models for Environmental Policymaking. *Social Science Research Network*, <https://doi.org/10.2139/ssrn.4499643>.
- [4] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez AN., et al., (2017). Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17*; p. 6000–6010. <https://doi.org/10.48550/arXiv.1706.03762>
- [5] Bender, EM., Gebru, T., McMillan-Major, A., Shmitchell, S., (2021). On the dangers of stochastic parrots: can language models be too big? In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*; p. 610–623. <https://doi.org/10.1145/3442188.3445922>
- [6] Giray, L., (2023). Prompt engineering with ChatGPT: a guide for academic writers, *Ann. Biomed. Eng.*, DOI: [10.1007/s10439-023-03272-4](https://doi.org/10.1007/s10439-023-03272-4)
- [7] White, J. et al., (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt, *ArXiv Prepr. ArXiv230211382*, <https://doi.org/10.48550/arXiv.2302.11382>
- [8] Eager, B., and Brunton, R., (2023). Prompting higher education towards AI-Augmented teaching and learning practice, *J. Univ. Teach. Learn. Pract.*, 20(5), <https://doi.org/10.53761/1.20.5.02>.
- [9] Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., McHardy, R. (2023). Challenges and applications of large language models; *ArXiv:2307.10169*. <https://doi.org/10.48550/arXiv.2307.10169>
- [10] Lu Y, Bartolo M, Moore A, Riedel S, (2022). Stenertorp P. Fantastically ordered prompts and where to find them: overcoming few-shot prompt order sensitivity. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*; p. 8086–8098. <https://doi.org/10.48550/arXiv.2104.08786>
- [11] Webson. A, Pavlick E., (2022). Do prompt-based models really understand the meaning of their prompts? In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. p. 2300–2344. <https://doi.org/10.48550/arXiv.2109.01247>
- [12] Maynez, J, Narayan, S, Bohnet, B, McDonald, R., (2020). On faithfulness and factuality in abstractive summarization. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. p. 1906–1919. <https://doi.org/10.48550/arXiv.2005.00661>
- [13] Bubeck, S, Chandrasekaran, V, Eldan, R, Gehrke, J, Horvitz, E, Kamar E, et al., (2023). Sparks of artificial general intelligence: early experiments with GPT-4. *ArXiv:2303.12712*. <https://doi.org/10.48550/arXiv.2303.12712>
- [14] Lo, L.S., (2023). The art and science of prompt engineering: a new literacy in the information age, *Internet Ref. Serv. Q.*, <https://doi.org/10.1080/10875301.2023.2227621>.
- [15] Jha, S., Jha, S.K., Lincoln, P., Bastian, N.D., Velasquez, A., and Neema, S., (2023). Dehallucinating large language models using formal methods guided iterative prompting, in: *2023 IEEE International Conference on Assured Autonomy (ICAA)*, IEEE. pp. 149-152. DOI: [10.1109/ICAA58325.2023.00029](https://doi.org/10.1109/ICAA58325.2023.00029)
- [16] Zhengbao, Jiang, Frank. F. Xu, Jun Araki, and Graham Neubig. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438. <https://doi.org/10.48550/arXiv.2102.09690>
- [17] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. (2023). Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23*, New York, NY, USA. Association for Computing Machinery. DOI: [10.1145/3544548.3581388](https://doi.org/10.1145/3544548.3581388)
- [18] Shanahan, M, McDonell K, Reynolds. L. (2023). Role-play with large language models; *ArXiv:2305.16367*. <https://doi.org/10.48550/arXiv.2305.16367>
- [19] Wei, J, Wang X, Schuurmans. D, Bosma, M, Ichter, B, Xia F, et al., (2022). Chain-of-thought prompting elicits reasoning in large language models. In: *Advances in Neural Information Processing Systems*. vol. 35; p. 24824–24837. <https://doi.org/10.48550/arXiv.2201.11903>
- [20] Lecler, A, Duron, L., and Soander, P., (2023). Revolutionizing radiology with GPT-based models: current applications, future possibilities and limitations of ChatGPT, *Diagn. Interv. Imaging*, 104(6), pp. 269-274. <https://doi.org/10.1016/j.diii.2023.02.003>.

- [21] Epstein, R.H. and Dexter, F., (2023). Variability in large language Models' responses to medical licensing and certification examinations. comment on How Does ChatGPT Perform on the United States Medical Licensing Examination? The Implications of Large Language Models for Medical Education and Knowledge Assessment, *JMIR Med. Educ.*, 9, <https://doi.org/10.2196/48305>.
- [22] Cooper, G., (2023). Examining science education in ChatGPT: An exploratory study of generative artificial intelligence, *J. Sci. Educ. Technol.*, 32(3), pp. 444-452, <https://doi.org/10.1007/s10956-023-10039-and>.
- [23] Chang, E.Y., (2023). Prompting large language models with the Socratic method, in: 2023 IEEE 13th Annual Computing and Communication Workshop and Conference, CCWC, pp. 351-360. <https://doi.org/10.1109/CCWC57344.2023.10099179>.
- [24] White, J. et al., (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt, *ArXiv Prepr. ArXiv230211382*, <https://doi.org/10.48550/arXiv.2302.11382>
- [25] Shieh, J., (2023). Best practices for prompt engineering with OpenAI API, OpenAI, [online]. Accessed: October 3rd of 2023. <https://doi.org/10.15446/dyna.v90n230.111700>
- [26] Yao, S. et al., (2023). Tree of thoughts: deliberate problem solving with large language models, *ArXiv Prepr. ArXiv230510601*, <https://doi.org/10.48550/arXiv.2305.10601>
- [27] Spasic, A.J., and Jankovic, D.S., (2023). Using ChatGPT standard prompt engineering techniques in lesson preparation: role, instructions and seed-word prompts, in: 2023 58th International Scientific Conference on Information, Communication and Energy and Systems and Technologies, ICEST 2023 - Proceedings, pp. 47-50. <https://doi.org/10.1109/ICEST58410.2023.10187269>.
- [28] Lo, L.S., (2023). The CLEAR path: a framework for enhancing information literacy through prompt engineering, *J. Acad. Librariansh.*, 49(4), <https://doi.org/10.1016/j.acalib.2023.102720>.
- [29] Zhang Z, Gao J, Dhaliwal RS, Jia-Jun. Li T. (2023). VISAR: a human-AI argumentative writing assistant with visual programming and rapid draft prototyping; *ArXiv:2304.07810*. <https://doi.org/10.48550/arXiv.2304.07810>
- [30] Buren, DV., (2023). Guided scenarios with simulated expert personas: a remarkable strategy to perform cognitive work; *ArXiv:2306.03104*. <https://doi.org/10.48550/arXiv.2306.03104>
- [31] Learn, Prompting., (2023). Learn Prompting: Your Guide to Communicating with AI. [learnprompting.org/docs/basics/roles](https://learnprompting.org/docs/basics/roles). <https://doi.org/10.3390/ime2030019>
- [32] Open, AI., (2023). Tactic: use delimiters to clearly indicate distinct parts of the input. Accessed: 2023-09-01. <https://platform.openai.com/docs/guides/gpt-best-practices/tactic-use-delimiters-to-clearly-indicate-distinct-parts-of-the-input>.
- [33] Chen, B. Zhaofeng, Z. Langrené, N, Zhu S., (2024) Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review. (Submitted on 23 Oct 2023 (v1), last revised 18 (this version, v4)). *arXiv:2310.14735 [cs.CL]*. <https://doi.org/10.48550/arXiv.2310.14735>
- [34] Wu S, Shen EM, Badrinath C, Ma J, Lakkaraju H., (2023). Analyzing chain-of-thought prompting in large language models via gradient-based feature Attributions; *ArXiv:2307.13339*. <https://doi.org/10.48550/arXiv.2307.13339>
- [35] Lewkowycz, A, Andreassen, A, Dohan D, Dyer E, Michalewski H, Ramasesh V, et al., (2022). Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*. 35:3843–3857. <https://doi.org/10.48550/arXiv.2206.14858>
- [36] Zhou H, Nova A, Larochelle H, Courville A, Neyshabur B, Sedghi H. (2022). Teaching Algorithmic Reasoning via In-context Learning; *ArXiv:2211.09066*. <https://doi.org/10.48550/arXiv.2211.09066>
- [37] Lee N, Sreenivasan K, Lee JD, Lee K, Papailiopoulos D. (2023). Teaching arithmetic to small transformers; *ArXiv:2307.03381*. <https://doi.org/10.48550/arXiv.2307.03381>
- [38] Wang, Boshi, et al., (2022). Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. *ArXiv:2212.10001 [Cs]*, Dec. [arxiv.org/abs/2212.10001](https://arxiv.org/abs/2212.10001). <https://doi.org/10.48550/arXiv.2212.10001>
- [39] Gao, Andrew., (2023). Prompt Engineering for Large Language Models: A brief guide with examples for non-technical readers. Available at SSRN: <https://ssrn.com/abstract=4504303> or <http://dx.doi.org/10.2139/ssrn.4504303>
- [40] Kojima. Takeshi, et al., (2022). Large Language Models Are Zero-Shot Reasoners. *ArXiv:2205.11916 [Cs]*, [arxiv.org/abs/2205.11916](https://arxiv.org/abs/2205.11916). <https://doi.org/10.48550/arXiv.2205.11916>
- [41] Prystawski, Ben, et al., (2023). Why Think Step by Step? Reasoning Emerges from the Locality of Experience. *ArXiv.org*, <https://doi.org/10.48550/arXiv.2304.03843>.
- [42] Del M, Fishel M., (2023). True detective: a deep abductive reasoning benchmark undoable for GPT-3 and challenging for GPT-4. In: Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (\*SEM 2023); <https://doi.org/10.48550/arXiv.2212.10114>
- [43] Yao S, Yu D, Zhao J, Shafran I, Griffiths TL, Cao Y, et al., (2023). Tree of thoughts: deliberate problem solving with large language models; *ArXiv:2305.10601*. <https://doi.org/10.48550/arXiv.2305.10601>
- [44] Long, J., (2023). Large language model guided tree-of-thought; *ArXiv:2305.08291*. <https://doi.org/10.48550/arXiv.2305.08291>
- [45] Hulbert, D., (2023). Tree of knowledge: ToK aka Tree of Knowledge dataset for Large Language Models LLM. Accessed: 2023-8-15. [figshare https://github.com/dave1010/tree-of-thought-prompting](https://github.com/dave1010/tree-of-thought-prompting).
- [46] Besta, M, Blach N, Kubicek A, Gerstenberger R, Gianinazzi L, Gajda J, et al., (2023). Graph of thoughts: solving elaborate problems with large language models; *ArXiv:2308.09687*. <https://doi.org/10.1609/aaai.v38i16.29720>
- [47] Wang, L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, et al., (2023). A survey on large language model based autonomous agents; *ArXiv:2308.11432*. <https://doi.org/10.1007/s11704-024-40231-1>



- [48] Besta L. M, Blach N, Kubicek A, Gerstenberger R, Podstawski M, Gianinazzi L, & et al., (2024). Graph of Thoughts: Solving Elaborate Problems with Large Language Models. arXiv:2308.09687v4 [cs.CL]. <https://github.com/spcl/graph-of-thoughts>. <https://doi.org/10.1609/aaai.v38i16.29720>
- [49] Logan. IV R, Balažević I, Wallace E, Petroni F, Singh S, Riedel S., (2022). Cutting down on prompts and parameters: simple few-shot learning with language models. In: Findings of the Association for Computational Linguistics: ACL 2022; p. 2824–2835. <https://doi.org/10.48550/arXiv.2106.13353>
- [50] Shyr. C, Hu Y, Harris PA, Xu H., (2023). Identifying and extracting rare disease phenotypes with large language models; ArXiv:2306.12656. <https://doi.org/10.1007/s41666-023-00155-0>
- [51] Brown, Tom, et al., (2020). Language Models Are Few-Shot Learners. Advances in Neural Information Processing Systems, vol. 33, pp. 1877–901, [proceedings.neurips.cc/paper\\_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html?utm\\_medium=email&utm\\_source=transaction](https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html?utm_medium=email&utm_source=transaction). <https://doi.org/10.48550/arXiv.2005.14165>
- [52] Reynolds. L, McDonell K., (2021). Prompt programming for large language models: beyond the few-shot paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems; p. 1–7. <https://doi.org/10.48550/arXiv.2102.07350>
- [53] Brown. TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al., (2020). language models Are Few-Shot Learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20; <https://doi.org/10.48550/arXiv.2005.14165>
- [54] Liu. J, Gardner M, Cohen SB, Lapata M., (2020). Multi-step inference for reasoning over paragraphs; ArXiv:2004.02995. <https://doi.org/10.18653/v1/2020.emnlp-main.245>
- [55] Wang. Boshi, et al., (2022). Towards Understanding Chain-of-Thought Prompting: An Empirical Study of What Matters. ArXiv:2212.10001 [Cs], Dec. [arxiv.org/abs/2212.10001](https://arxiv.org/abs/2212.10001). <https://doi.org/10.48550/arXiv.2212.10001>
- [56] Wei. J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al., (2022). Chain-of-thought prompting elicits reasoning in large language models. In: Advances in Neural Information Processing Systems. vol. 35; p. 24824–24837. <https://doi.org/10.48550/arXiv.2201.11903>
- [57] Wang X, Wei J, Schuurmans D, Le QV, Chi EH, Narang S, et al. (2023). Self-consistency improves chain of thought reasoning in language models. In: Eleventh International Conference on Learning Representations; <https://doi.org/10.48550/arXiv.2203.11171>
- [58] Bender EM, Gebru T, McMillan-Major A, Shmitchell S., (2021). On the dangers of stochastic parrots: can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency; p. 610–623. <https://doi.org/10.1145/3442188.3445922>
- [59] Shum. K, Diao S, Zhang. T., (2023). Automatic prompt augmentation and selection with chain-of-thought from labeled data; ArXiv:2302.12822. <https://doi.org/10.48550/arXiv.2302.12822>
- [60] Khalifa. M, Logeswaran L, Lee M, Lee H, Wang. L., (2023). Discriminator-guided multi-step reasoning with language models; ArXiv:2305.14934. <https://doi.org/10.48550/arXiv.2305.14934>
- [61] Zhou D, Schärli N, Hou L, Wei J, Scales N, Wang X, et al., (2023). Least-to-most prompting enables complex reasoning in large language models. In: Eleventh International Conference on Learning Representations; <https://doi.org/10.48550/arXiv.2205.10625>
- [62] Liu. J, Liu A, Lu X, Welleck S, West P, Le Bras R, & et al., (2022). Generated knowledge prompting for commonsense reasoning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); p. 3154–3169. <https://doi.org/10.48550/arXiv.2110.08387>
- [63] Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., ... & Resnik, P. (2024). The Prompt Report: A Systematic Survey of Prompting Techniques. arXiv preprint arXiv:2406.06608. <https://arxiv.org/abs/2406.06608>
- [64] Deng M, Wang J, Hsieh C-P, Wang Y, Guo H, Shu T, & et al., (2022). RLPrompt: Optimizing discrete text prompts with reinforcement learning. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. <https://doi.org/10.48550/arXiv.2205.12548>
- [65] Xu H, Chen Y, Du Y, Shao N, Wang Y, Li H, and Yang Z., (2022). Gps: Genetic prompt search for efficient few-shot learning. 2022. arXiv preprint arXiv:2210.17041. <https://doi.org/10.48550/arXiv.2210.17041>
- [66] Wan. X, Sun R, Nakhost. H, Dai H, Eisenschlos. J, Arik. S, and Pfister. T., (2023b). Universal self-adaptive prompting. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 7437–7462, Singapore. Association for Computational Linguistics. <https://doi.org/10.48550/arXiv.2305.14926>
- [67] Ye. Q, Axmed. M, Pryzant. R, Khani. F., (2024). Prompt Engineering a Prompt Engineer, Submitted on 9 Nov 2023 (v1), last revised 3 Jul 2024 (this version, v3) arXiv:2311.05661 [cs.CL]. <https://doi.org/10.48550/arXiv.2311.05661>
- [68] Holtzman. A, Buys J, Du L, Forbes. M, Choi Y., (2020). The curious case of neural text degeneration. In: International Conference on Learning Representations; <https://doi.org/10.48550/arXiv.1904.09751>
- [69] Raand P.P., and Majumder, P., (2023). Assessing the Accuracy of responses from the language model ChatGPT to questions regarding bariatric surgery: a critical appraisal, *Obes. Surg.*, 33(8), pp. 2588–2589, <https://doi.org/10.1007/s11695-023-06664-6>.
- [70] Gupta, R., Herzog, I., Weisberger, J., Chao, J., Chaiandasate, K., and Lee, E.S., (2023). Utilization of ChatGPT for plastic surgery research: friend or foe, *J. Plast. Reconstr. Aesthet. Surg.*, 80, pp. 145–147, <https://doi.org/10.1016/j.bjps.2023.03.004>.

- [71] Deiana, G., Dettori, M., Arghittu, A., Azara, A., Gabutti, G., and Castiglia, P., (2023). Artificial intelligence and public health: evaluating ChatGPT responses to vaccination mandths and misconceptions, *Vaccines*, 11(7), art. 11071217, <https://doi.org/10.3390/vaccines11071217>.
- [72] Lo, L.S., (2023). The CLEAR path: a framework for enhancing information literacand through prompt engineering, *J. Acad. Librariansh.*, 49(4), <https://doi.org/10.1016/j.acalib.2023.102720>.
- [73] Sai AB, Mohankumar AK, Khapra MM. (2022). A survey of evaluation metrics used for NLG systems. *ACM Computing Surveys (CSUR)*. 55(2):1–39. <https://doi.org/10.48550/arXiv.2008.12009>
- [74] Papineni K, Roukos S, Ward T, Zhu WJ. BLEU: (2002). a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*; p. 311–318. <https://doi.org/10.3115/1073083.10731>
- [75] Lin, C.Y., (2004) Rouge: A Package for Automatic Evaluation of Summaries. *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL, Barcelona, 25 July 2004.*; p. 74–8.
- [76] Banerjee. S, Lavie. A., (2005). METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: *Proceedings of the ACL Work-shop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*; p. 65–72.
- [77] Zhang, T, Kishore V, Wu F, Weinberger. KQ, Artzi Y., (2020). BERTScore: evalu-ating text generation with BERT. In: *International Conference on Learning Representations*; <https://doi.org/10.48550/arXiv.1904.09675>
- [78] Stent. A, Marge. M, Singhai. M., (2005). Evaluating evaluation methods for generation in the presence of variation. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer; p. 341–351. DOI:10.1007/978-3-540-30586-6\_38
- [79] Deng M, Wang, J, Hsieh CP, Wang Y, Guo, H, Shu, T, et al., (2022). RLPrompt: optimiz-ing discrete text prompts with reinforcement learning. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*; p. 3369–3391. <https://doi.org/10.48550/arXiv.2205.12548>
- [80] Zhou. Y, Muresanu AI, Han Z, Paster. K, Pitis S, Chan H, et al. (2022). Large language models are human-level prompt engineers. In: *Eleventh International Conference on Learning Representations*; <https://doi.org/10.48550/arXiv.2211.01910>
- [81] Ajith. A, Pan C, Xia. M, Deshpande A, Narasimhan. K., (2023). InstructEval: systematic evaluation of instruction selection methods; ArXiv:2307.00259. <https://doi.org/10.48550/arXiv.2307.00259>
- [82] Maynez J, Narayan. S, Bohnet B, McDonald. R., (2020). On faithfulness and factuality in abstractive summarization. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; p. 1. <https://doi.org/10.48550/arXiv.2005.00661>
- [83] Lee. K, Firat. O, Agarwal A, Fannjiang. C, Sussillo. D., (2018). Hallucinations in neural machine translation; <https://doi.org/10.48550/arXiv.2301.07779>
- [84] Ji. Z, Lee. N, Frieske. R, Yu T, Su D, Xu Y, et al., (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*. 55(12):1–38. <https://doi.org/10.1145/3571730>
- [85] Ram. O, Levine Y, Dalmedigos. I, Muhlgay. D, Shashua. A, Leyton-Brown K, et al., (2023). In-context retrieval-augmented language models; <https://doi.org/10.48550/arXiv.2302.00083>
- [86] Lazaridou. A, Gribovskaya. E, Stokowiec. W, Grigorev N., (2022). Internet-augmented lan-guage models through few-shot prompting for open-domain question answering; ArXiv:2203.05115. <https://doi.org/10.48550/arXiv.2203.05115>
- [87] Jiang. Z, Xu FF, Gao. L, Sun Z, Liu Q, Dwivedi-Yu J, et al., (2023). Active retrieval augmented generation; ArXiv:2305.06983. <https://doi.org/10.48550/arXiv.2305.06983>
- [88] Shehzaad. Dhuliawala, Mojtaba. Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston., (2023). Chain-of-Verification Reduces Hallucination in Large Language Models. arXiv preprint arXiv:2309.11495 <https://doi.org/10.48550/arXiv.2309.11495>
- [89] Ziwei Ji, Tiezheng Yu, Yan Xu., (2023). Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards Mitigating Hallucination in Large Language Models via Self-Reflection. *EMNLP Findings* <https://doi.org/10.48550/arXiv.2310.06271>
- [90] Takeshi. Kojima, Shixiang Shane. Gu., (2023). Machel Reid Google Research, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. URL <https://arxiv.org/abs/2205.11916> <https://doi.org/10.48550/arXiv.2205.11916>
- [91] Deren. Lei, Yaxi. Li, Mingyu Wang, Vincent Yun, Emily Ching, Eslam Kamal, et al., (2023). Chain of natural language inference for reducing large language model ungrounded hallucinations. arXiv preprint arXiv:2310.03951 <https://doi.org/10.48550/arXiv.2310.03951>
- [92] Noah. Shinn, Beck Labash, and Ashwin. Gopinath., (2023). Reflexion: an autonomous agent with dynamic memory and self-reflection. arXiv preprint arXiv:2303.11366 <https://doi.org/10.48550/arXiv.2303.11366>
- [93] Zhenhailong. Wang, Shaoguang. Mao, Wenshan. Wu, Tao Ge, Furu. Wei, and Heng. Ji., (2023). Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona selfcollaboration. arXiv preprint arXiv:2307.05300 1, 2. 3. <https://doi.org/10.48550/arXiv.2307.05300>
- [94] Shuster. K, Poff S, Chen M, Kiela. D, Weston. J., (2021). Retrieval augmentation reduces hallucination in conversation; ArXiv:2104.07567. <https://doi.org/10.48550/arXiv.2104.07567>
- [95] Lewis. P, Perez. E, Piktus. A, Petroni F, Karpukhin. V, Goyal. N, et al., (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*.33:9459–9474. <https://doi.org/10.48550/arXiv.2005.11401>



- [96] Izacard. G, Grave. E., (2020). Leveraging passage retrieval with generative models for open domain question answering; ArXiv:2007.01282. <https://doi.org/10.48550/arXiv.2007.01282>
- [97] Lewis. M, Liu Y, Goyal. N, Ghazvininejad. M, Mohamed A, Levy O, et al., (2020). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; p. 7871–7880. <https://doi.org/10.48550/arXiv.1910.13461>
- [98] Roller. S, Dinan. E, Goyal. N, Ju D, Williamson. M, Liu Y, et al., (2021). Recipes for building an open-domain chatbot. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume; p. 300–325. <https://doi.org/10.1609/aaai.v38i16.29720>
- [99] Dhuliawala. S, Komeili M, Xu J, Raileanu. R, Li X, Celikyilmaz. A, et al., (2023). Chain-of-verification reduces hallucination in large language models; ArXiv:2309.11495. <https://doi.org/10.48550/arXiv.2309.11495>
- [100] Sahoo. P, Kumar Singh. A, Saha S, Jain V, Mondal. S, and Chadha. A., (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927, <https://doi.org/10.48550/arXiv.2402.07927>
- [101] Shubham. V, Harsh D., (2024). A Survey of prompt Engineering Methods in Large Language Models for different NLP. <https://doi.org/10.48550/arXiv.2407.12994>. arXiv:2407.12994 [cs.CL] [Submitted on 17 Jul 2024 (v1), last revised 24 Jul 2024 (this version, v2)]
- [102] Besta. M, Blach. N, Kubicek. A, Gerstenberger. R, Podstawski. M, Gianinazzi. L, Gajda J, Lehmann. T, Niewiadomski. H, Nyczyk. P, Hoefler. T., (2024). Graph of Thoughts: Solving Elaborate Problems with Large Language Models. <https://doi.org/10.1609/aaai.v38i16.29720>
- [103] Petroni. F, Rocktäschel. T, Riedel. S, Lewis, P, Bakhtin, A, Wu, Yuxiang, & Miller, A., (2019). Language models as knowledge bases? In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19). Association for Computational Linguistics, 2463–2473. <https://doi.org/10.18653/v1/D19-1250>
- [104] Schick. T, & Schütze. H., (2021). It's not just size that matters: small language models are also few-shot learners. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2339–2352. <https://doi.org/10.18653/v1/2021.naacl-main.185>
- [105] Gao. T, Fisch. A, and Chen. D., (2021). Making pre-trained language models better few-shot learners. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'21). <https://doi.org/10.48550/arXiv.2012.15723>
- [106] Hambardzumyan. K, Khachatrian. H, and May. J., (2021). WARP: Word-level adversarial reprogramming. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 4921–4933. <https://doi.org/10.18653/v1/2021.acl-long.381>
- [107] Lester. B, Al-Rfou. R, and Constant. N., (2021). The power of scale for parameter-efficient prompt tuning. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'21), Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3045–3059. <https://doi.org/10.18653/v1/2021.emnlp-main.243>
- [108] Schick, T, Schmid, H, & Schütze, H., (2020). Automatically identifying words that can serve as la-bels for few-shot text classification. In Proceedings of the 28th International Conference on Computational Linguistics (COLING'20), Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 5569–5578. <https://doi.org/10.18653/v1/2020.coling-main.488>
- [109] Schick, T, & Schütze, H. (2021). Exploiting Cloze-questions for few-shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL'21), Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, 255–269. <https://doi.org/10.18653/v1/2021.eacl-main.20>
- [110] Cui. L, Wu. Yu, Liu. J, Sen Yang. and Zhang. Y., (2021). Template-based named entity recognition using BART. arXiv:2106.01760 [cs.CL]. Retrieved from <https://doi.org/10.48550/arXiv.2106.01760>
- [111] Dev, C, Biyani, N, Suthar, N, Kumar, P, Agarwal, P., (2021) .Structured Prediction in NLP - A survey. arXiv:2110.02057 [cs.CL] (or [arXiv:2110.02057v1](https://doi.org/10.48550/arXiv.2110.02057) [cs.CL] for this version) <https://doi.org/10.48550/arXiv.2110.02057>
- [112] Shin. R, H. Lin. C, Thomson. S, Chen. C, Roy. S, Antonios. Platanios E, & et.al., (2021). Constrained language models yield few-shot semantic parsers. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'21), Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 7699–7715. <https://doi.org/10.18653/v1/2021.emnlp-main.608>
- [113] Radford. A, Wu J, R. Child, Luan. D, Amodei. D, and Sutskever. I., (2019). Language models are unsupervised multitask learners.
- [114] Brown. T, Mann. B, Ryder. N, Subbiah. M, D. Kaplan. J, Dhariwal. P, & et.al., (2020). Language models are few-shot learners. In Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. <https://doi.org/10.48550/arXiv.2005.14165>

- [115] Schick. T. and Schütze. H., (2021). Few-shot text generation with natural language instructions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 390–402. <https://doi.org/10.18653/v1/2021.emnlp-main.32>
- [116] Lisa. Li X. and Liang. P., (2021). Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP'21), Volume 1: Long Papers, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4582–4597. <https://doi.org/10.18653/v1/2021.acl-long.353>
- [117] Deepak Ajwani. R, Zhu. Z, J Rose, Rudzicz. F., (2024) Plug and Play with Prompts: A Prompt Tuning Approach for Controlling Text Generation. arXiv:2404.05143 [cs.CL] <https://doi.org/10.48550/arXiv.2404.05143>
- [118] Dou. Z-Y, Liu. P, Hayashi. H, Jiang. H, and Neubig. G., (2021). GSum: A general framework for guided neural abstractive summarization. Association for Computational Linguistics, Online, 4830–4842. arXiv:2010.08014 [cs.CL]. <https://doi.org/10.48550/arXiv.2010.08014>.
- [119] Xiao. Z, Chen. Y, Yao. J, Zhang. L, Liu. Z, Wu. Z, & et.al., (2024). Instruction-ViT: Multi-modal prompts for instruction learning in vision transformer, Information Fusion, Volume 104. 2024 -102, ISSN 1566-2535, <https://doi.org/10.1016/j.inffus.2023.102204>. )<https://www.sciencedirect.com/science/article/pii/S1566253523005201>
- [120] Tsimpoukelli. M, Menick. J, Cabi. S, S. M. Ali Eslami, Vinyals. O, and Hill. F., (2021). Multimodal few-shot learning with frozen language models. In Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems (NeurIPS'21), Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 200–212. <https://doi.org/10.48550/arXiv.2106.13884>
- [121] Alqifari. R., (2019). Question Answering Systems Approaches and Challenges. Conference Paper. DOI:10.26615/issn.2603-2821.2019\_011
- [122] Rajpurkar. P, Zhang. J, Lopyrev. K, and Liang. P., (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2383–2392. <https://doi.org/10.18653/v1/D16-1264>
- [123] Lai. G, Xie. Q, Liu. H, Yang. Y, and Hovy. E., (2017). RACE: Large-scale reading comprehension dataset from examinations. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 785–794. <https://doi.org/10.18653/v1/D17-1082>
- [124] Khashabi. D, Min. S, Khot. T, Sabharwal. A, Tafjord. O, Clark. P, and Hajishirzi. H., (2020). UNIFIEDQA: Crossing format boundaries with a single QA system. In Findings of the Association for Computational Linguistics: EMNLP. Association for Computational Linguistics, 1896–1907. <https://doi.org/10.18653/v1/2020.findings-emnlp.171>.
- [125] Jiang. Z, Araki. J, Ding. H, and Neubig. G., (2021). How can we know when language models know? On the calibration of language models for question answering. Trans. Assoc. Comput. Ling. 9 (09 2021), 962–977. [https://doi.org/10.1162/tacl\\_a\\_00407](https://doi.org/10.1162/tacl_a_00407)
- [126] Schick. T., and Schütze. H., (2021). Generating datasets with pretrained language models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'21), Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Ed) DOI: [10.5282/ubm/epub.92195](https://doi.org/10.5282/ubm/epub.92195)
- [127] Çalli. E, Sogancioglu. E, Ginneken. B.v, G. van Leeuwen. K, Murphy. K., (2021). Deep learning for chest X-ray analysis: A survey, Medical Image Analysis. Volume 72, p102-125. ISSN 1361-8415. <https://www.sciencedirect.com/science/article/pii/S1361841521001717> <https://doi.org/10.1016/j.media.2021.102125>
- [128] Ben-David. E, Oved. N, and Reichart. R., (2022). PADA: Example-based prompt learning for on-the-fly adaptation to unseen domains. Trans. Assoc. Comput. Linguist. 10 (4 2022), 414–433. [https://doi.org/10.1162/tacl\\_a\\_00468](https://doi.org/10.1162/tacl_a_00468)