

مهندسی پرامپت در مدل های زبانی بزرگ

Prompt engineering in LLMs

فرزانه الشریف^۱، شهرام محمدی^۲، نگار اغمازی^۳

^۱ گروه معماری کامپیوتر ، دانشکده مهندسی کامپیوتر ، دانشگاه اصفهان ، اصفهان ، ایران

^۲ دانشکده مهندسی کامپیوتر ، واحد اصفهان ، دانشگاه آزاد اسلامی ، یاسوج ، ایران

^۳ دانشکده فنی حرفه ای، واحد اصفهان، دانشگاه ملی مهارت ، اصفهان، ایران

چکیده :

مهندسی پرامپت به عنوان روشی موثر برای بهبود عملکرد مدل های زبانی و بینایی-زبانی معرفی شده است. این تکنیک با طراحی ورودی های هدفمند، بدون تغییر در پارامتر های اصلی، امکان سازگاری و ارتقای بهره وری مدل هارا فراهم می آورد . همچنین به بررسی پیشرفت های اخیر در این حوزه می پردازد و تکنیک های متنوعی همچون زنجیره تفکر و یادگیری پرامپت چندحالتی را معرفی میکند. و چالش ها و فرصت های امنیتی مرتبط با این تکنیک ها را تحلیل کرده و چارچوبی برای تحقیقات آیند ارائه می دهد.

کلمات کلیدی: مدل های زبانی بزرگ , پرامپت دهی تک شات و چند شات , زنجیره تفکر , خودسازگاری ,

۱. مقدمه

تصور کنید مدل‌های زبانی بزرگ مانند مغز انسان هستند که با خواندن تعداد بسیار زیادی کتاب و مقاله، توانایی درک و تولید متن را به دست می‌آورند. این مدل‌ها، که به آن‌ها LLM نیز می‌گویند، با استفاده از معماری‌های پیچیده و یادگیری از حجم عظیمی از داده‌ها، توانایی پردازش و تولید زبان انسان را به دست آورده‌اند. نمونه‌هایی مانند GPT و Gemini نشان‌دهنده توانایی‌های این فناوری در پردازش و تولید متون پیچیده هستند.

مهندسی پرامپت به عنوان یک ابزار قدرتمند برای کنترل و هدایت این مدل‌ها مطرح شده است. با استفاده از این تکنیک، می‌توانیم به مدل‌ها بگوییم که چه کاری را انجام دهند و چه نوع متنی را تولید کنند. به عنوان مثال، می‌توانیم از یک مدل زبانی بزرگ بخواهیم که یک شعر کوتاه بنویسد، یک مقاله علمی خلاصه کند، یا حتی یک مکالمه طبیعی با ما داشته باشد.

در این مقاله، به بررسی نقش مهندسی پرامپت در بهبود عملکرد مدل‌های زبانی بزرگ و کاربردهای این تکنیک در حوزه‌های مختلف می‌پردازیم. ابتدا به معرفی مفاهیم پایه مهندسی پرامپت و انواع مختلف پرامپت‌ها و سپس، به بررسی تکنیک‌های مختلف برای طراحی پرامپت‌های موثر خواهیم پرداخت. در ادامه، به برخی از کاربردهای عملی مهندسی پرامپت در حوزه‌های مختلف مانند آموزش، ترجمه، و تولید محتوا و در نهایت، به چالش‌های موجود و آینده این حوزه خواهیم پرداخت.

۱.۲ ضرورت مهندسی پرامپت

با وجود قدرت بالای مدل‌های زبانی، عملکرد آن‌ها به شدت به کیفیت پرامپت‌ها وابسته است و پرامپت‌های دقیق و ساختاریافته می‌توانند خروجی‌های مرتبط و باکیفیت‌تری تولید کنند.

- دقت در نتایج: با طراحی دقیق پرامپت‌ها، می‌توانیم از مدل‌های زبانی بخواهیم که دقیقاً همان چیزی را تولید کنند که ما می‌خواهیم.
- خلاقیت و نوآوری: مهندسی پرامپت به ما اجازه می‌دهد تا از مدل‌های زبانی برای تولید ایده‌های جدید و خلاقانه استفاده کنیم.
- کنترل بر خروجی: با تغییر دادن پرامپت‌ها، می‌توانیم خروجی مدل را به طور کامل کنترل کنیم.

- بهبود کارایی مدل‌ها: با طراحی پرامپت‌های بهینه، می‌توانیم کارایی مدل‌های زبانی را بهبود بخشیم و از آن‌ها بهترین بهره را ببریم.

و در نهایت مهندسی پرامپت به ما کمک می‌کند تا با مدل‌های زبانی بزرگ به صورت مؤثرتر و کارآمدتر ارتباط برقرار کنیم و از آن‌ها برای حل مشکلات پیچیده و ایجاد نوآوری استفاده کنیم.

ساختار مقاله در ادامه....

۲. اصول اولیه مهندسی پرامپت

با استفاده از چند عنصر کلیدی، می‌توان یک پرامپت پایه ایجاد کرد که مدل‌های زبانی بزرگ (¹LLMs) را قادر به تولید پاسخ‌های می‌کند که یک پرامپت مؤثر باید شامل موارد زیر باشد

- دستورالعمل شفاف: به صورت واضح مشخص کند که از مدل چه می‌خواهید.
- زمینه مناسب: اطلاعات مرتبط ارائه دهد تا مدل درک درستی از مسئله داشته باشد
- مثال‌های آموزشی: برای راهنمایی مدل، مثال‌های مرتبط ارائه دهد.

جدول ۱

مورد	مزایا	معایب	توضیحات
دستورالعمل شفاف	کاهش ابهام و تولید پاسخ‌های دقیق	نیازمند دقت بالا در طراحی دستورالعمل	ارائه یک دستورالعمل واضح که هدفه درخواست را مشخص کند.
زمینه مناسب	بهبود دقت و کاهش پاسخ‌های نادرست	ممکن است در مسائل گسترده زمان بر باشد	ارائه اطلاعات مرتبط به مدل برای درک بهتر مسئله.
مثال راهنما	بهبود عملکرد مدل در وظایف پیچیده	افزایش طول پرامپت و هزینه محاسباتی	ارائه نمونه‌های از ورودی و خروجی برای هدایت مدل.

¹ Large language model systems

۲.۱ پرامپت‌دهی تک‌شات و چندشات

در پرامپت‌دهی تک‌شات^۲، مدل تنها با یک نمونه هدایت می‌شود، در حالی که پرامپت‌دهی چندشات^۳ از چندین نمونه برای بهبود درک مدل استفاده می‌کند. وظایف ساده معمولاً با پرامپت‌دهی تک‌شات قابل انجام هستند، اما برای وظایف پیچیده‌تر، پرامپت‌دهی چندشات عملکرد بهتری ارائه می‌دهد. (Chen B et al , 2023)

- کاربرد تک شات: وظایف ساده مانند ترجمه .
- کاربرد چندشات: وظایف پیچیده مثل حل مسائل ریاضی

۳. تکنیک‌های پیشرفته مهندسی پرامپت

۳.۱ زنجیره تفکر^۴

این تکنیک با تقسیم وظایف به مراحل منطقی، مدل را به تولید پاسخ‌های دقیق‌تر هدایت می‌کند. برای نمونه در حل یک مسئله ریاضی چندمرحله‌ای، پرامپت باید هر مرحله را مشخص کند تا مدل به نتیجه نهایی برسد. یکی از مزایای اصلی این روش، شفاف‌سازی فرآیند تفکر مدل است که به ما کمک می‌کند بفهمیم مدل‌ها چگونه به نتایج خود می‌رسند. این شفافیت موجب افزایش اعتماد به عملکرد مدل و بهبود دقت و قابلیت اطمینان آن‌ها می‌شود. نکات کلیدی در زنجیره تفکر:

- Few-Shot: مدل می‌تواند تنها با چند مثال نحوه حل مسائل را بیاموزد.
- Chain-of-Thought : مدل مسائل را به صورت زنجیره‌ای و گام‌به‌گام تحلیل می‌کند.
- شفافیت : تصمیم‌گیری مدل واضح‌تر بوده و ارزیابی عملکرد آن آسان‌تر می‌شود.

جدول ۲_زنجیره تفکر (cot)

مزایا	معایب	توضیحات
افزایش دقت در مسائل چند مرحله ای	نیازمند طراحی پرامپت‌های طولانی و پیچیده	تقسیم وظایف پیچیده به مراحل منطقی برای تولید پاسخ‌های دقیق‌تر

² One-Shot

³ Few-Shot

⁴ Chain-of-Thought

۳.۲ خودسازگاری^۵

این تکنیک پاسخ‌های متناقض مدل را کاهش می‌دهد و با تولید چندین پاسخ مختلف برای یک مسئله و انتخاب دقیق‌ترین پاسخ، دقت مدل را افزایش می‌دهد.

- تولید چندین پاسخ: در این روش مدل، چندین پاسخ مختلف تولید می‌کند. این کار به مدل اجازه می‌دهد تا طیف وسیعی از احتمالات را بررسی کند.
- ارزیابی پاسخ‌ها: هر یک از پاسخ‌های تولید شده توسط مدل، از نظر منطق، سازگاری با اطلاعات موجود و سایر معیارهای ارزیابی، مورد بررسی قرار می‌گیرد.
- انتخاب بهترین پاسخ: از بین تمام پاسخ‌های تولید شده، مدلی که بیشترین امتیاز را کسب کرده است به عنوان پاسخ نهایی انتخاب می‌شود. این امتیازدهی بر اساس معیارهایی مانند منطق، انسجام، ارتباط با پرسش و ... انجام می‌شود.

از دلایل مهم بودن خودسازگاری میتوانیم به موارد زیر اشاره کنیم

افزایش دقت : با کاهش پاسخ‌های متناقض و انتخاب بهترین پاسخ، دقت به طور قابل توجهی افزایش می‌یابد.

افزایش اعتماد : زمانی که یک مدل پاسخ‌های سازگار و منطقی تولید کند، اعتماد کاربران به آن بیشتر می‌شود.

کاهش خطاهای انسانی : با خودکارسازی فرایند ارزیابی پاسخ‌ها، احتمال بروز خطاهای انسانی کاهش می‌یابد.

جدول ۳_ خودسازگاری

مزایا	معایب	توضیحات
کاهش خطا و افزایش دقت	افزایش هزینه محاسباتی و زمان پردازش	تولید چند پاسخ و انتخاب بهترین گزینه.

⁵ Self-Consistency

۳.۳ تولید دانش⁶

مدل در این روش، پیش از ارائه پاسخ نهایی، اطلاعات مرتبط با پرسش را تولید می‌کند و از این دانش برای ایجاد پاسخ‌های دقیق‌تر بهره می‌برد. به‌ویژه برای وظایفی که به استدلال مبتنی بر دانش نیاز دارند.

- خلاقیت ماشینی: تولید داستان، شعر، موسیقی و کدهای برنامه‌نویسی
- حل مسئله: یافتن راه حل برای مسائل پیچیده در حوزه‌های مختلف مانند علم، مهندسی و پزشکی
- آموزش: ایجاد محتوای آموزشی شخصی‌سازی شده
- توسعه محصول: تولید ایده‌های جدید برای محصولات و خدمات

جدول ۴_تولید دانش

مزایا	معایب	توضیحات
مناسب برای وظایف علمی و پژوهشی	خطر تولید اطلاعات نادرست یا نامرتبط	تولید اطلاعات مرتبط قبل از پاسخ دهی نهایی.

۳.۴ تولید تقویت‌شده با بازیابی⁷

در این روش، مدل اطلاعات موردنیاز را از منابع خارجی بازیابی کرده و بهترین پاسخ را بر اساس آن به کاربر ارائه می‌دهد.

- دریافت پرسش: کاربر سوالی را از مدل می‌پرسد.
- جستجوی اطلاعات: مدل به یک پایگاه داده یا اینترنت متصل می‌شود و اطلاعات مرتبط با پرسش را جستجو می‌کند.
- انتخاب اطلاعات مرتبط: از بین اطلاعات بازیابی شده، مدل بخش‌هایی را که بیشترین ارتباط را با پرسش دارند انتخاب می‌کند.
- تولید پاسخ: مدل با استفاده از اطلاعات بازیابی شده و دانش قبلی خود، یک پاسخ جامع و دقیق تولید می‌کند

جدول ۵_تقویت شده با بازیابی

مزایا	معایب	توضیحات
کاهش خطا و ارائه پاسخ‌های واقعی‌تر	نیاز به دسترسی به منابع داده خارجی	استفاده از اطلاعات خارجی برای افزایش دقت پاسخ.

⁶ Generated Knowledge

⁷ RAG

۳.۵ از آسان به سخت ^۸

این تکنیک وظایف پیچیده را به مراحل ساده‌تر تقسیم می‌کند تا مدل بتواند به تدریج به راه‌حل نهایی برسد. این تکنیک به‌ویژه زمانی مفید است که مسئله اصلی شامل چندین بخش به هم مرتبط باشد و نیاز به استدلال گام‌به‌گام داشته باشد.

(Chen B et al , 2023)

جدول ۶_آسان به سخت

مزایا	معایب	توضیحات
بهبود تدریجی عملکرد مدل	نیازمند زمان و طراحی دقیق	تقسیم وظایف پیچیده به مراحل ساده‌تر

۴. ابزارهای مهندسی پرامپت

۴.۱ LangChain

کتابخانه‌ای برای توسعه برنامه‌هایی است که از مدل‌های زبانی بزرگ استفاده می‌کند که با ارائه ابزارهایی برای زنجیره‌سازی پرامپت‌ها، مدیریت داده‌ها و اتصال به ابزارهای خارجی، امکان طراحی سیستم‌های هوشمند و پیچیده را فراهم می‌کند

۴.۲ OpenPrompt

یک ابزار کاربردی برای مهندسی پرامپت است که به پژوهشگران و توسعه‌دهندگان امکان اینکه پرامپت‌ها را به‌سادگی طراحی و تنظیم و قابل آزمایش کنند تا از مدل‌های زبانی به بهترین شکل استفاده کنند.

۴.۳ Promptify

یک ابزار کاربردی برای مهندسی پرامپت است که به توسعه‌دهندگان کمک می‌کند پرامپت‌های بهینه و مؤثری طراحی کنند تا عملکرد مدل‌های زبانی را بهبود دهند. (Marvin et al , 2024)

جدول ۷

ابزار	مزایا	معایب	توضیحات
LangChain	پشتیبانی از تعاملات پیچیده	نیازمند دانش فنی برای استفاده	کتابخانه‌ای برای ترکیب داده‌ها و مدل‌های زبانی.
OpenPrompt	امکان بهینه‌سازی پرامپت‌ها	محدودیت در برخی کاربردها	چارچوبی استاندارد برای طراحی و ارزیابی پرامپت‌ها.
Promptify	مناسب برای پروژه‌های صنعتی	نیازمند تنظیمات اولیه پیچیده	کتابخانه‌ای برای مدیریت و ارزیابی پرامپت‌ها در محیط عملیاتی.

⁸ Least-to-Most Prompting

۵. کاربردهای مهندسی پرامپت

۵.۱ آموزش و تولید محتوا

مهندسی پرامپت می‌تواند در تولید محتواهای آموزشی، خلاصه‌سازی متون، و پاسخ‌دهی به سؤالات قابل استفاده و بسیار مفید است.

جدول ۸- آموزش و تولید محتوا

مزایا	معایب	توضیحات
صرفه جویی در زمان و منابع	خطر تولید اطلاعات ناقص یا نادرست	تولید پاسخ‌های تعاملی، خلاصه سازی متون و محتوای آموزشی

۵.۲ برنامه‌نویسی و توسعه نرم‌افزار

در این روش مهندسی پرامپت به توسعه‌دهندگان کمک می‌کند تا کدهای بهینه‌تری بنویسند و اشکالات موجود را سریع‌تر رفع کنند.

جدول ۹- برنامه نویسی

مزایا	معایب	توضیحات
کاهش زمان توسعه و رفع خطا	محدودیت در درک مسائل پیچیده	تولید و اشکال زدایی کد های برنامه نویسی.

۵.۳ تحلیل مسائل پیچیده

از این تکنیک در تحلیل داده‌های علمی و همچنین حل مسائل منطقی و ارائه بهتر استدلال‌های دقیق بسیار کاربردی است. (Chen B et al , 2023)

جدول ۱۰- تحلیل مسائل پیچیده

مزایا	معایب	توضیحات
مناسب برای مدیریت داده های حجیم	ممکن است در مسائل خاص دقت کافی نداشته باشد	ارائه راه حل های منطقی برای مسائل علمی و پژوهشی.

۶. چالش‌ها و محدودیت‌ها

۶.۱ سوگیری‌های مدل

در این تکنیک، مدل‌های زبانی ممکن است پاسخ‌های متعصبانه یا نادرستی ارائه دهند که با طراحی دقیق و هدفمند می‌توان این سوگیری‌ها را برطرف کرد و دقت نتایج را افزایش داد.

جدول ۱۱_ سوگیری مدل

مزایا	معایب	توضیحات
می‌توان با طراحی مناسب این مشکل را کاهش داد	نیازمند نظارت مدام	احتمال تولید پاسخ‌های متعصبانه یا نادرست

۶.۲ دقت در وظایف پیچیده

در این تکنیک، مدل‌های زبانی در انجام وظایف پیچیده و استراتژیک با چالش‌هایی روبه‌رو هستند، که با طراحی دقیق پرامپت‌ها می‌توان عملکرد آن‌ها را بهبود بخشید و نتایج بهتری بدست آورد.

جدول ۱۲_ دقت در وظایف پیچیده

مزایا	معایب	توضیحات
امکان بهبود با استفاده از تکنیک‌های پیشرفته	محدودیت‌های تکنیکی در وظایف خاص	کاهش دقت در مسائل استراتژیک و چند مرحله‌ای.

۶.۳ هزینه‌های محاسباتی

هزینه محاسباتی در مهندسی پرامپت به استفاده از مدل‌های بزرگ و منابع محاسباتی زیادی وابسته است، که می‌تواند در موارد پیچیده تر یا داده‌های حجیم بسیار هزینه‌بر باشد. (Sahoo et al, 2024)

جدول ۱۳_ هزینه‌های محاسباتی

مزایا	معایب	توضیحات
امکان بهینه‌سازی منابع در طول زمان	هزینه بر بودن برای پروژ‌های کوچک	نیاز به منابع پردازشی بالا برای مدل‌های بزرگ

۷. روندهای آینده

۷.۱ تکنیک‌های پیشرفته

تکنیک‌های پیشرفته در مهندسی پرامپت شامل روش‌هایی مانند یادگیری متا و پرامپت‌های ترکیبی است که به بهبود عملکرد مدل‌ها در وظایف پیچیده و متنوع کمک می‌کنند.

۷.۲ کاهش هزینه‌ها

کاهش هزینه در مهندسی پرامپت از طریق استفاده از ابزارهایی امکان‌پذیر است که مصرف منابع محاسباتی مدل‌ها را بهینه کرده و هزینه‌های محاسباتی را کاهش می‌دهند.

۷.۳ گسترش دسترسی پذیری

گسترش دسترسی پذیری در مهندسی پرامپت از طریق توسعه ابزارهای کاربرپسند صورت می‌گیرد که استفاده از این فناوری را برای کاربران غیرمتخصص آسان‌تر می‌سازد. (Marvin et al , 2024)

۸. نتیجه گیری

مهندسی پرامپت به عنوان یک مهارت کلیدی برای استفاده بهینه از مدل‌های زبانی بزرگ شناخته می‌شود. با استفاده از تکنیک‌ها و ابزارهای پیشرفته، می‌توان چالش‌های موجود را کاهش داده و از ظرفیت‌های این مدل‌ها بهره بیشتری برد. آینده این حوزه با پیشرفت‌های بیشتر در زمینه‌های تکنیکی و کاربردی بسیار روشن است.

¹ Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024, February 5). A Systematic survey of prompt engineering in large language Models: Techniques and applications. arXiv.org. <https://arxiv.org/abs/2402.07927>

² Chen, B., Zhang, Z., Langrené, N., & Zhu, S. (2023, October 23). Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review. arXiv.org. <https://arxiv.org/abs/2310.14735>

³ Kepel, D., & Valogianni, K. (2024, June 25). Autonomous prompt engineering in large language models. arXiv.org. <https://arxiv.org/abs/2407.11000>

⁴ Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. (2024). Prompt engineering in large language models. In Algorithms for intelligent systems (pp. 387–402). https://doi.org/10.1007/978-981-99-7962-2_30