آزمایشگاه فناوری وب
Web Technology Lab
WT laboratory

مهندسی پرامپت در مدل‌های زبانی بزرگ: تکنیک‌ها و کاربردها

ارائه دهنده: فاطمه محسن نیا

مقدمه

- مهندسی پرامپت با طراحی هوشمندانه دستورالعمل‌های خاص وظیفه، خروجی مدل را هدایت می‌کند بدون نیاز به تغییر پارامترها.
- این تکنیک تطبیق‌پذیری مدل‌های بزرگ زبانی و زبان-تصویر را افزایش می‌دهد و در حوزه‌های مختلف موفقیت‌هایی را به ارمغان آورده است.

Zero-shot Prompting (پرامپت بدون نمونه) :

در این روش، مدل بدون نیاز به داده‌های آموزشی قبلی، بادستورالعمل‌های کلی برای انجام وظایف هدایت می‌شود. این تکنیک برای مواقعی که داده‌های آموزشی محدود است، بسیار مفید است.

فرض کنید می‌خواهیم یک مدل زبان بزرگ (مانند GPT-4) داستانی درباره یک ربات که به فضا سفر می‌کند بنویسد.

بدون ارائه هیچ نمونه‌ای از داستان‌های قبلی، تنها با دادن پرامپت زیر:

"Write a short story about a robot traveling to space".

مدل می‌تواند یک داستان کوتاه درباره این موضوع تولید کند.

Few-shot Prompting (پرامپت با چند نمونه):

این روش شامل ارائه چند مثال ورودی-خروجی به مدل است تا مدل بتواند الگوی پاسخ‌دهی را بیاموزد.
این تکنیک کمک می‌کند مدل بهتر بتواند وظایف را با داده‌های کم یاد بگیرد.

- فرض کنید می‌خواهیم یک مدل زبانی به سوالات ریاضی پاسخ دهد.
- ابتدا چند مثال ورودی-خروجی به مدل ارائه می‌دهیم:

```
""Solve the following problem: 2 + 2 = ? Answer: 4

Solve the following problem: 5 * 3 = ? Answer: 15

Solve the following problem: 10 - 7 = ? Answer: 3

Now, solve the following problem: 8 / 2 = ?"
```

- مدل با مشاهده این نمونه‌ها، الگویی را می‌آموزد و پاسخ صحیح (۴) را تولید می‌کند.

Chain-of-Thought Prompting (پرامپت زنجیره‌ای):

در این تکنیک، فرآیندهای استدلال به صورت مرحله به مرحله و با دقت بیشتری هدایت می‌شوند. این تکنیک به مدل کمک می‌کند تا استدلال‌های پیچیده‌تر و چند مرحله‌ای را بهتر انجام دهد.

- برای حل یک مسئله منطقی که نیاز به استدلال چند مرحله‌ای دارد، می‌توانیم پرامپت زیر را به مدل بدهیم:

"If John has twice as many apples as Mary, and Mary has three apples, how many apples does John have? Let's break it down step by step.

.۱First, find out how many apples Mary has. Mary has ۳ apples.

.۲John has twice as many apples as Mary. So, ۳ ×۲ =۶.

Therefore, John has ۶ apples".

- این روش مدل را تشویق می‌کند تا فرآیند استدلال را به صورت مرحله به مرحله انجام دهد و به پاسخ صحیح برسد.

Auto-CoT (پرامپت زنجیره‌ای خودکار):

این تکنیک به صورت خودکار زنجیره‌های استدلالی را ایجاد می‌کند تا فرآیند تصمیم‌گیری مدل بهبود یابد. این روش از مدل می‌خواهد که به طور خودکار مراحل استدلال را طی کند و پاسخ نهایی را ارائه دهد.

مثال

Auto-CoT (پرامپت زنجیره‌ای خودکار):

فرض کنید می‌خواهیم مدل یک مسئله پیچیده ریاضی را حل کند و از آن بخواهیم که خود مراحل استدلال را طی کند:

"Solve the following problem step by step :

A car travels ۶۰ miles per hour for ۲ hours, then ۴۰ miles per hour for ۳ hours. What is the total distance traveled?

مدل به صورت خودکار مراحل زیر را طی می‌کند:

.۱Calculate the distance traveled at ۶۰ mph for ۲ hours: ۱۲۰ =۶۰ 2 miles.

.۲Calculate the distance traveled at ۴۰ mph for ۳ hours: ۱۲۰ =۴۰ 3 miles.

.۳Add the two distances together: ۲۴۰ =۱۲۰ +۱۲۰ miles.

Total distance traveled is ۲۴۰ miles".

Self-Consistency Prompting

- توضیح: این روش به مدل اجازه می‌دهد تا چندین پاسخ مختلف به یک سوال تولید کند و سپس پاسخ‌ها را با یکدیگر مقایسه کند تا پاسخ نهایی را تعیین کند.
- مثال: فرض کنید مدل باید به سوالی مانند "در چه سالی جنگ جهانی دوم آغاز شد؟" پاسخ دهد. مدل چندین پاسخ تولید می‌کند (مثلاً ۱۹۳۹، ۱۹۴۰، ۱۹۳۸) و با مقایسه و ارزیابی آنها، پاسخ درست (۱۹۳۹) را انتخاب می‌کند.
- مزیت: این روش دقت پاسخ‌دهی را افزایش می‌دهد و از تولید پاسخ‌های نادرست جلوگیری می‌کند.

Logical Chain-of-Thought (LogiCoT) Prompting

- **توضیح: این روش از مدل می‌خواهد که استدلال‌های منطقی خود را به صورت مرحله به مرحله ارائه دهد. این روش به مدل کمک می‌کند تا فرآیندهای استدلال پیچیده را به دقت بیشتری طی کند.**
- **مثال: برای پاسخ به سوال "اگر همه پرندگان می‌توانند پرواز کنند و پنگوئن‌ها پرندگان هستند، آیا پنگوئن‌ها می‌توانند پرواز کنند؟" مدل باید به ترتیب مراحل زیر را طی کند:**
  - همه پرندگان می‌توانند پرواز کنند.
  - پنگوئن‌ها پرندگان هستند.
  - پنگوئن‌ها نمی‌توانند پرواز کنند. نتیجه‌گیری: "نه، پنگوئن‌ها نمی‌توانند پرواز کنند."
- **مزیت: این روش دقت و صحت استدلال‌های مدل را بهبود می‌بخشد.**

Tree-of-Thoughts (ToT) Prompting

- توضیح: این روش به مدل اجازه می‌دهد تا چندین شاخه از استدلال‌ها و پاسخ‌ها را به صورت همزمان تولید کند و سپس بهترین مسیر را انتخاب کند. این روش شبیه به ایجاد یک درخت تصمیم‌گیری است.
- مثال: برای حل یک مسئله پیچیده ریاضی، مدل می‌تواند چندین مسیر مختلف را برای رسیدن به پاسخ بررسی کند و بهترین مسیر را انتخاب کند.
- مزیت: این روش به مدل کمک می‌کند تا بهترین پاسخ را از میان چندین گزینه ممکن انتخاب کند.

Graph-of-Thoughts (GoT) Prompting:

- توضیح: این روش مشابه ToTاست، اما به جای درخت، از ساختار گراف استفاده می‌کند. مدل می‌تواند استدلال‌ها و پاسخ‌های خود را در یک گراف مرتب کند و از مسیرهای مختلف به پاسخ نهایی برسد.
- مثال: برای تحلیل یک مسئله پیچیده، مدل می‌تواند گرافی از مفاهیم مرتبط ایجاد کند و با استفاده از این گراف، به نتیجه‌گیری دقیق‌تری برسد.
- مزیت: این روش به مدل کمک می‌کند تا استدلال‌های پیچیده‌تر و چندگانه را به صورت موثرتری پردازش کند.

Thread of Thought (ThoT) Prompting

- توضیح: در این روش، مدل چندین رشته از استدلال‌ها را به صورت همزمان دنبال می‌کند و نتایج هر رشته را برای رسیدن به پاسخ نهایی مقایسه می‌کند.
- مثال: برای پاسخ به سوالی مانند "بهترین روش برای کاهش مصرف انرژی در یک خانه چیست؟" مدل می‌تواند چندین رشته از استدلال‌ها (مانند استفاده از انرژی خورشیدی، بهبود عایق‌کاری، استفاده از لامپ‌های LED) را بررسی کند و بهترین راه‌حل را پیشنهاد دهد.
- مزیت: این روش به مدل کمک می‌کند تا بهترین پاسخ را از میان چندین رشته استدلال مختلف انتخاب کند.

Chain-of-Table Prompting

- توضیح: این روش از مدل می‌خواهد تا اطلاعات و استدلال‌ها را به صورت جدول مرتب کند و سپس بر اساس این جداول، پاسخ نهایی را ارائه دهد.
- مثال: برای تحلیل داده‌های پیچیده، مدل می‌تواند داده‌ها را در جداول مختلف مرتب کند و سپس از این جداول برای نتیجه‌گیری استفاده کند.
- مزیت: این روش به مدل کمک می‌کند تا اطلاعات را به صورت سازمان‌یافته‌تری پردازش کند و به نتیجه‌گیری دقیق‌تری برسد.

◀ ۱. تولید زبان:

- توضیح: مدل‌های زبانی بزرگ می‌توانند متون مختلفی مانند مقالات، داستان‌ها، و حتی شعر تولید کنند.
- مثال: با دادن پرامپتی مانند "یک داستان کوتاه درباره یک ربات که به فضا سفر می‌کند بنویس"، مدل می‌تواند داستانی جذاب و خلاقانه تولید کند.
- مزیت: این کاربرد به نویسندگان و تولیدکنندگان محتوا کمک می‌کند تا به سرعت ایده‌ها و متون جدید ایجاد کنند.

◀ ۲. پاسخ به سوالات

- توضیح: مدل‌های زبانی می‌توانند به سوالات کاربران پاسخ دهند، اطلاعات دقیق ارائه کنند و به حل مشکلات کمک کنند.
- مثال: با دادن پرامپتی مانند "اولین رئیس‌جمهور ایالات متحده چه کسی بود؟" مدل می‌تواند پاسخ دهد "جورج واشنگتن".
- مزیت: این کاربرد در موتورهای جستجو، چت‌بات‌ها و سیستم‌های پشتیبانی مشتری مورد استفاده قرار می‌گیرد.

◀ ۳. استدلال منطقی:

- توضیح: مدل‌های زبانی می‌توانند مسائل منطقی و استدلالی را حل کنند و مراحل استدلال را توضیح دهند.
- مثال: با دادن پرامپتی مانند "اگر همه انسان‌ها فانی هستند و سقراط یک انسان است، آیا سقراط فانی است؟ مراحل استدلال را توضیح بده"، مدل می‌تواند مراحل استدلال را طی کند و نتیجه‌گیری کند که "بله، سقراط فانی است".
- مزیت: این کاربرد در آموزش و پژوهش، تحلیل داده‌ها و تصمیم‌گیری‌های پیچیده مفید است.

◀ ۴. ترجمه و خلاصه‌سازی:

توضیح: مدل‌های زبانی می‌توانند متون را از یک زبان به زبان دیگر ترجمه کنند و یا متون طولانی را خلاصه‌سازی کنند.

مثال: با دادن پرامپتی مانند "این متن را به زبان اسپانیایی ترجمه کن" یا "این مقاله طولانی را خلاصه کن"، مدل می‌تواند ترجمه یا خلاصه‌ای مفید ارائه دهد.

مزیت: این کاربرد در ارتباطات بین‌المللی، ترجمه متون علمی و تجاری و صرفه‌جویی در زمان مطالعه متون طولانی مؤثر است.

◀ ۵. تولید و اجرای کد:

توضیح: مدل‌های زبانی می‌توانند کدهای برنامه‌نویسی تولید کنند و حتی اشکالات کد را پیدا و رفع کنند.

مثال: با دادن پرامپتی مانند "یک تابع پایتون برای محاسبه فاکتوریل بنویس"

مزیت: این کاربرد به برنامه‌نویسان کمک می‌کند تا سریع‌تر کد بنویسند و اشکالات را رفع کنند، و در آموزش برنامه‌نویسی نیز مفید است.

◀ ۶. تولید محتوای خلاقانه:

توضیح: مدل‌های زبانی می‌توانند ایده‌های خلاقانه برای داستان‌ها، فیلم‌نامه‌ها، شعرها و حتی موسیقی ارائه دهند.
مثال: با دادن پرامپتی مانند "یک شعر عاشقانه بنویس"، مدل می‌تواند شعری زیبا تولید کند.
مزیت: این کاربرد به هنرمندان و نویسندگان کمک می‌کند تا به سرعت ایده‌های جدید خلق کنند و خلاقیت خود را افزایش دهند.

◀ ۷. تعامل با کاربران:

توضیح: مدل‌های زبانی می‌توانند به عنوان چت‌بات‌ها با کاربران تعامل داشته باشند و خدمات مختلفی ارائه دهند.
مثال: با دادن پرامپتی مانند "به مشتری کمک کن تا سفارش خود را پیگیری کند"، مدل می‌تواند به مشتری کمک کند تا اطلاعات مربوط به سفارش خود را پیدا کند.
مزیت: این کاربرد در خدمات مشتری، پشتیبانی فنی و تعامل با کاربران آنلاین بسیار مفید است.

کاربرد های جزئی تر(۱)

◀ پشتیبانی مشتریان :

استفاده از مدل‌های زبانی برای پاسخگویی به سوالات مشتریان و ارائه پشتیبانی فوری. با طراحی پرامپت ها مناسب، این مدل‌ها می‌توانند به طور خودکار به سوالات متداول پاسخ دهند و زمان پاسخگویی را کاهش دهند.

◀ تولید محتوا

ایجاد مقالات، پست‌های وبلاگ، تبلیغات و سایر محتواهای متنی به صورت خودکار. مهندسی پرامپت به تولید متونی با کیفیت و خوانا کمک می‌کند که نیاز به ویرایش کمتری دارند.

◀ آموزش و آموزش الکترونیک:

طراحی پرامپت ها برای تولید محتوای آموزشی و تمرینات تعاملی. این می‌تواند به معلمان و مدرسان کمک کند تا مواد آموزشی متنوع و جذاب‌تری ایجاد کنند.

◀ تجزیه و تحلیل داده‌ها :

استفاده از پرامپت ها بهینه برای تجزیه و تحلیل داده‌های بزرگ و پیچیده. این پرامپت ها می‌توانند به شناسایی الگوها و استخراج اطلاعات ارزشمند از داده‌ها کمک کنند.

کاربرد های جزئی تر(۲)

◀ مدیریت پروژه :
کمک به مدیریت پروژه‌ها از طریق ایجاد پرامپت ها برای برنامه‌ریزی، زمان‌بندی و پیگیری پیشرفت. این مدل‌ها می‌توانند به مدیران پروژه در تعیین وظایف و نظارت بر پیشرفت کمک کنند.

◀ ترجمه زبان :
طراحی پرامپت ها بهینه برای ترجمه دقیق و روان متون بین زبان‌های مختلف. مهندسی پرامپت می‌تواند به بهبود دقت و روانی ترجمه‌ها کمک کند.

◀ سیستم‌های توصیه‌گر :
استفاده از مدل‌های زبانی برای ارائه توصیه‌های شخصی‌سازی شده بر اساس ترجیحات کاربران. این می‌تواند شامل توصیه کتاب‌ها، فیلم‌ها، محصولات و سایر موارد باشد.

◀ تشخیص گفتار :
بهبود دقت سیستم‌های تشخیص گفتار از طریق طراحی پرامپت ها که به مدل‌ها کمک می‌کند گفتار را به متن تبدیل کنند. این کاربرد در دستیارهای صوتی و سیستم‌های کنترل صوتی بسیار مفید است.

مراحل ایجاد پرامپت مناسب برای پشتیبانی مشتریان:

۱. تعریف هدف پرامپت:

- هدف اصلی پرامپت را مشخص کنید. در اینجا هدف پاسخگویی به سوالات متداول مشتریان است.

۲. شناسایی نیازهای مشتری:

- باید بدانید مشتریان چه سوالاتی می‌پرسند و چه اطلاعاتی برای آن‌ها مهم است.

۳. ساختاردهی پرامپت:

- پرامپت باید به گونه‌ای طراحی شود که مدل زبانی اطلاعات دقیق و مرتبطی ارائه دهد.

۴. افزودن جزئیات

- جزئیات بیشتری برای دقت بیشتر به پرامپت اضافه کنید، مثلاً محصولات یا خدمات خاص.

۵. ارائه مثال:

- ارائه مثال‌های کاربردی به مدل کمک می‌کند تا الگوهای مناسب پاسخ را شناسایی کند.

مثال

ایجاد پرامپت برای پاسخگویی به سوالات متداول درباره یک محصول

پاسخگویی به سوالات مشتریان درباره محصول خاصی مانند یک گوشی هوشمند.

پرامپت مناسب:

شما یک دستیار هوش مصنوعی هستید که برای پاسخگویی به سوالات مشتریان درباره‌ی گوشی هوشمند X طراحی شده‌اید.

لطفاً به سوالات زیر پاسخ دهید:

۱. ویژگی‌های اصلی گوشی هوشمند X چیست؟

۲. قیمت گوشی هوشمند X در بازار فعلی چقدر است؟

۳. چگونه می‌توانم گوشی هوشمند X را از طریق فروشگاه آنلاین شما خریداری کنم؟

۴. آیا گوشی هوشمند X از تکنولوژی 5G پشتیبانی می‌کند؟

۵. مدت زمان گارانتی گوشی هوشمند X چقدر است و شامل چه مواردی می‌شود؟

۶. نحوه‌ی انتقال داده‌ها از گوشی قدیمی به گوشی هوشمند X چگونه است؟

نکات پرامپت

۱. شفافیت و وضوح: سوالات باید واضح و بدون ابهام باشند.

۲. جزئیات دقیق: مشخص کردن مدل محصول و سوالات مشخص در مورد آن.

۳. تقسیم‌بندی موضوعات: سوالات را بر اساس موضوعات مختلف تقسیم کنید تا مدل بتواند به هر کدام به طور جداگانه پاسخ دهد.

تست و بهینه‌سازی پرامپت

تست و بهینه‌سازی پرامپت:

۱. آزمایش پرامپت: چندین بار پرامپت را امتحان کنید و نتایج را بررسی کنید.

۲. بازخورد و اصلاح: بر اساس بازخوردها و نتایج، پرامپت را اصلاح و بهینه کنید.

۳. تنظیم دقیق: جزئیات و مثال‌های بیشتری اضافه کنید تا مدل بهتر پاسخ دهد.

نقاط قوت تکنیک‌های مهندسی پرامپت

افزایش دقت و کارایی مدل‌ها:

تکنیک‌های مهندسی پرامپت به مدل‌ها کمک می‌کنند تا با دقت بیشتری وظایف را انجام دهند و عملکرد بهتری داشته باشند. این تکنیک‌ها با هدایت مدل از طریق پرامپت ها خاص، کارایی مدل را بهبود می‌بخشند.

کاهش نیاز به داده‌های آموزشی:

یکی از مزایای مهم مهندسی پرامپت این است که مدل‌ها می‌توانند با تعداد کمی از نمونه‌ها آموزش ببینند. این امر در مواقعی که داده‌های آموزشی کمیاب یا دسترسی به آنها دشوار است، بسیار مفید است.

انعطاف‌پذیری بالا:

مهندسی پرامپت امکان تطبیق مدل‌ها با وظایف مختلف را فراهم می‌کند، بدون نیاز به تغییر پارامترهای اصلی مدل. این انعطاف‌پذیری به مدل‌ها اجازه می‌دهد تا در محیط‌های مختلف و با نیازهای متفاوت به خوبی عمل کنند.

نقاط ضعف تکنیک‌های مهندسی پرامپت

پیچیدگی در طراحی پرامپت ها:

طراحی پرامپت ها مناسب و مؤثر برای هدایت مدل‌ها می‌تواند پیچیده باشد و نیاز به دانش تخصصی دارد. یافتن پرامپت ها بهینه ممکن است زمان‌بر و چالش‌برانگیز باشد.

احتمال بروز خطا در زنجیره‌های استدلالی:

تکنیک‌های پیشرفته مانند Chain-of-Thought Prompting ممکن است باعث ایجاد زنجیره‌های استدلالی پیچیده‌ای شوند که احتمال بروز خطا در آنها وجود دارد. مدیریت و کنترل این زنجیره‌ها نیازمند دقت بالاست.

محدودیت در مقیاس‌پذیری:

در برخی موارد، مهندسی پرامپت ممکن است به خوبی در مقیاس بزرگ عمل نکند. به ویژه در وظایفی که نیاز به داده‌های زیادی دارند، ممکن است محدودیت‌هایی وجود داشته باشد.

جمع‌بندی

- مهندسی پرامپت با بهینه‌سازی پرامپت ها، قابلیت‌ها و کاربردهای مدل‌های زبانی و هوش مصنوعی را گسترش می‌دهد و به بهبود کیفیت و دقت نتایج کمک می‌کند.
- تکنیک‌های مهندسی پرامپت نشان داده‌اند که می‌توانند مدل‌ها را در حوزه‌های مختلف از جمله تولید زبان، پاسخ به سوالات، و استدلال منطقی بهبود بخشند. این تکنیک‌ها پتانسیل بالایی برای ارتقای عملکرد مدل‌ها دارند.

آینده‌پژوهی

- **تحقیقات بیشتر برای بهبود تکنیک‌ها:** نیاز به پژوهش‌های بیشتر برای بهبود و کارایی بالاتر تکنیک‌های موجود وجود دارد. پژوهش‌های آینده می‌توانند به کشف تکنیک‌های جدید و بهبود تکنیک‌های فعلی کمک کنند.
- **گسترش کاربردها:** انتظار می‌رود که مهندسی پرامپت در آینده در حوزه‌های جدیدی مانند علوم پزشکی، حقوق، و سایر زمینه‌های تخصصی مورد استفاده قرار گیرد. گسترش کاربردهای این تکنیک می‌تواند تأثیرات بزرگی در صنایع مختلف داشته باشد.
- **توسعه ابزارهای خودکار:** توسعه ابزارهای خودکار برای طراحی و بهینه‌سازی پرامپت ها می‌تواند فرآیند مهندسی پرامپت را ساده‌تر و کارآمدتر کند. این ابزارها می‌توانند به پژوهشگران و توسعه‌دهندگان کمک کنند تا به سرعت پرامپت ها مناسب را پیدا کرده و مدل‌ها را بهینه کنند.
- **ترکیب با تکنولوژی‌های جدید:** ترکیب مهندسی پرامپت با تکنولوژی‌های نوظهور مانند یادگیری عمیق و هوش مصنوعی توزیعی می‌تواند منجر به بهبود عملکرد و کارایی مدل‌ها شود. این ترکیب‌ها می‌توانند فرصت‌های جدیدی را برای بهبود و نوآوری در این حوزه فراهم کنند.

مرجع:

- A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024). Department of Computer Science And Engineering, Indian Institute of Technology Patna; Stanford University; Amazon AI.

✨ از نگاهتان سپاس گزارم ✨