



Home Work

Generative AI

Prompt engineering

1. فایل تمرین را در پنل خود آپلود کنید.
2. title فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد.
3. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.
4. فایل های پیوست را می توانید از [اینجا](#) دریافت کنید.

۱. بررسی اثر جزئیات پرامپت بر کیفیت محتوای تولید شده

در این تمرین، شما تأثیر میزان جزئیات ارائه شده در پرامپت را بر کیفیت خروجی مدل زبانی بررسی می کنید. این کار به شما کمک می کند درک بهتری از مهندسی پرامپت (Prompt Engineering) پیدا کنید و بتوانید در آینده، درخواست های بهینه ای برای تولید متن های علمی و دقیق از مدل های زبانی داشته باشید.

مرحله ۱: انتخاب یک موضوع علمی

یک موضوع علمی مرتبط با زمینه کاری یا تحصیلی خودتان انتخاب کنید. این موضوع می تواند مرتبط با رشته دانشگاهی، پروژه ای که روی آن کار می کنید یا حتی یک حوزه ای باشد که به آن علاقه دارید.

مرحله ۲: طراحی سه پرامپت مختلف

اکنون سه پرامپت برای مدل زبانی طراحی کنید که هر کدام میزان جزئیات بیشتری نسبت به قبلی داشته باشند. هدف این است که مشاهده کنید مدل زبانی چگونه با افزایش دقت پرامپت، محتوای بهتری تولید می‌کند.

پرامپت ۱ - کلی و بدون جزئیات

در این پرامپت، از مدل می‌خواهید که یک مقاله در مورد موضوع شما بنویسد، اما هیچ توضیحی درباره ساختار، محتوا یا نحوه ارائه اطلاعات نمی‌دهید.

مثال: یک مقاله درباره‌ی کاربرد یادگیری عمیق در پردازش زبان طبیعی بنویس.

پرامپت ۲ - مشخص کردن ساختار کلی مقاله

در این پرامپت، به مدل می‌گویید که مقاله باید چه ساختاری داشته باشد، اما هنوز توضیحی درباره محتوای دقیق هر بخش ارائه نمی‌دهید.

مثال: یک مقاله علمی درباره‌ی کاربرد یادگیری عمیق در پردازش زبان طبیعی بنویس. مقاله باید شامل مقدمه، توضیح روش‌های مختلف، مزایا و چالش‌ها، و نتیجه‌گیری باشد.

پرامپت ۳ - تعیین دقیق محتوا و جزئیات هر بخش

در این پرامپت، علاوه بر ساختار کلی، جزئیات مربوط به هر بخش را نیز مشخص می‌کنید تا مدل زبانی بداند که چه اطلاعاتی را در هر قسمت ارائه دهد.

مثال: یک مقاله علمی درباره‌ی کاربرد یادگیری عمیق در پردازش زبان طبیعی بنویس. مقاله باید شامل بخش‌های زیر باشد:

- **مقدمه:** معرفی پردازش زبان طبیعی و اهمیت آن در دنیای امروز.
- **روش‌ها:** توضیح مدل‌های معروف یادگیری عمیق مانند RNN و Transformer در پردازش زبان طبیعی.
- **مزایا و چالش‌ها:** بررسی نقاط قوت یادگیری عمیق در NLP و مشکلاتی مانند نیاز به داده‌های زیاد و هزینه محاسباتی بالا.

- نتیجه‌گیری: خلاصه‌ای از یافته‌های مقاله و پیش‌بینی آینده‌ی این حوزه.

مرحله ۳: مقایسه و تحلیل خروجی‌ها

خروجی تولید شده با سه پرامپت مختلف را با هم مقایسه کرده و اثر اضافه کردن جزییات به پرامپت را بررسی کنید.

۲. تاثیر اثر few-shot prompting در تقلید سبک نوشتاری

در این تمرین، شما بررسی خواهید کرد که چگونه ارائه‌ی چند نمونه‌ی راهنما می‌تواند باعث شود مدل زبانی سبک خاصی را بهتر تقلید کند.

بخش اول: انتخاب موضوع و سبک نوشتاری

ابتدا یک موضوع و یک سبک نوشتاری خاص انتخاب کنید. می‌توانید از پیشنهادهای زیر استفاده کنید یا سبک دلخواه خود را انتخاب کنید:

پیشنهادهای موضوع:

- خلاصه کردن یک مقاله‌ی علمی
- توضیح یک مفهوم تخصصی به زبان ساده
- نوشتن یک خبر مهم در حوزه‌ی فناوری، سیاست، یا پزشکی
- بازنویسی یک داستان کوتاه یا حکایت تاریخی

پیشنهادهای سبک نوشتاری:

- طنز آمیز و غیررسمی (مثلاً خلاصه کردن یک مقاله‌ی علمی به زبان توییت‌های طنز)

- رسمی و آکادمیک (مثلاً تبدیل یک متن عامیانه به زبان مقالات دانشگاهی)
- روایی و داستانی (مثلاً بازنویسی یک رویداد تاریخی به صورت داستان کوتاه)
- هیجان‌انگیز و تبلیغاتی (مثلاً تبدیل یک خبر علمی به تبلیغی جذاب)
- زبان نوجوانان و عامیانه (مثلاً توضیح یک مفهوم اقتصادی به زبان مکالمات روزمره)

بخش دوم: اجرای پرامپت بدون Few-shot prompting

اکنون پرامپت خود را بنویسید و مدل را وادار کنید که بدون نمونه‌های آموزشی، متن مورد نظر را تولید کند.

مثال: اگر موضوع شما توضیح نظریه‌ی نسبیت به سبک زبان عامیانه‌ی نوجوانان باشد، پرامپت اولیه می‌تواند به این صورت باشد:

Explain Einstein's theory of relativity in a way that a teenager would understand.

انتظار داریم که مدل خروجی نسبتاً عمومی ارائه دهد که سبک موردنظر شما را به‌خوبی تقلید نمی‌کند.

بخش سوم: اضافه کردن چند نمونه‌ی راهنما

اکنون ۳ تا ۵ نمونه‌ی راهنما در سبک انتخابی خود به مدل اضافه کنید تا ببینید چگونه پاسخ بهبود پیدا می‌کند.

مثال: در صورت انتخاب سبک زبان عامیانه‌ی نوجوانان

Explain scientific concepts in a way that a teenager would understand:

- Quantum mechanics? It's basically like the universe is playing a giant lottery, and particles just teleport around randomly.
- Black holes? Think of them as cosmic vacuum cleaners that eat everything, even light. Super messy eaters!
- The Big Bang? Imagine pressing 'start' on a universe-making video game. Boom! Everything appears out of nowhere.

Now, explain Einstein's theory of relativity in a similar way:

انتظار داریم که مدل این بار پاسخی بسیار بهتر و متناسب‌تر با سبک موردنظر ارائه دهد.

بخش چهارم: مقایسه و تحلیل نتایج

- خروجی‌های دو مرحله Zero-shot و Few-shot را مقایسه کنید.
- بررسی کنید که آیا مدل بعد از اضافه کردن نمونه‌ها، سبک انتخابی را بهتر رعایت کرده است؟
- آیا هنوز هم به بهبود نیاز دارد؟ اگر بله، چه تعداد نمونه‌ی دیگر می‌توان اضافه کرد؟

۳. تاثیر few-shot prompting در تولید خروجی ساختار یافته

در این تمرین، شما یاد می‌گیرید که چگونه با استفاده از پرامپت‌نویسی، مدل زبانی را وادار کنید تا یک فایل رزومه را پردازش کند و اطلاعات موردنیاز را در قالب یک ساختار JSON استاندارد استخراج کند.

بخش اول: تعیین اطلاعات موردنیاز

ابتدا تصمیم بگیرید که چه اطلاعاتی را می‌خواهید از رزومه استخراج کنید. به عنوان مثال، می‌توانید اطلاعات زیر را در نظر بگیرید:

- نام و اطلاعات تماس (نام، ایمیل، شماره تلفن)
- تحصیلات (مقطع، رشته، دانشگاه، سال فارغ‌التحصیلی)
- تجربیات شغلی (عنوان شغلی، شرکت، مدت زمان اشتغال، وظایف اصلی)
- مهارت‌ها (فنی و نرم)
- گواهینامه‌ها و دوره‌های آموزشی

بخش دوم: نوشتن پرامپت برای استخراج اطلاعات (Zero-shot)

ابتدا یک پرامپت ساده بنویسید که از مدل بخواهد اطلاعات را از رزومه‌ی ورودی استخراج کند.

انتظار داریم که مدل اطلاعات را به‌صورت JSON تولید کند، اما ممکن است فرمت دقیق و ساختار استاندارد نداشته باشد.

بخش سوم: مشخص کردن دقیق ساختار JSON (Few-shot Learning)

اکنون مدل را وادار کنید که اطلاعات را در یک ساختار JSON دقیق و استاندارد ارائه دهد.

انتظار داریم که این‌بار خروجی مدل دقیقاً مطابق با ساختار داده‌ی مشخص شده باشد.

نحوه‌ی تحویل تمرین:

دانشجوها باید برای هر تمرین یک فایل پایتون `py` یا `ipynb` تحویل دهند که در آن با استفاده از کتابخانه‌ی `LangChain` یک `instance` از `LLM` ایجاد کرده و پرامپت‌هایی که طراحی کرده‌اند را اجرا کنند. برنامه باید خروجی تولید شده توسط مدل را نمایش داده و در صورت نیاز ذخیره کند.