

prompt templates

2023년 10월 4일 수요일 오전 2:47

prompt engineering : optimizing the communication between AI and human

clarity : prompt should be clear and easily understood

relevance : prompts should be directly related to the information sought(seek과거분사)

context : prompts should fit within the broader scope of the conversation or topic at hand

iterative refinement : user feedback을 받아서 refined prompt생성

prompting AI 에 대한 발전 가능성과 앞으로 적용하기 위해 봐야 할것들

Emerging Trends in Prompting and AI

Conversational AI and NLP Improvements: Advancements in natural language processing (NLP) are making conversational AI more nuanced and sophisticated, allowing for more natural and human-like interactions.

Context-Aware Prompting: AI systems are becoming better at understanding context, which allows for more effective prompting. This means AI can maintain the thread of a conversation and understand the nuances of different scenarios or user needs.

Personalized Prompts: Personalization in prompting is a growing trend, where AI systems tailor prompts based on a user's past interactions, preferences, and behaviors.

Ethical Prompting: As AI becomes more integrated into daily life, there's an increased focus on ethical prompting. This involves crafting prompts that avoid reinforcing biases and respect user privacy and consent.

Multimodal Interactions: Prompts are no longer limited to text; emerging trends include the integration of multimodal data (e.g., voice, visual cues) into prompting, enabling AI to respond more effectively to a wider range of human inputs.

Autonomous Prompt Generation: AI systems are beginning to autonomously generate their own prompts to gather additional information or clarify ambiguous user input, leading to more efficient problem-solving.

Collaborative AI: The future of prompting includes collaborative AI, where AI systems help users refine their prompts in real-time, essentially teaching users how to interact with AI more effectively.

Prompting for Creativity: There's a growing use of AI for creative tasks, where prompting is used to generate new ideas, content, or solutions in fields like writing, design, and entertainment.

Interactive Learning and Prompting: AI systems are being designed to learn from user prompts and subsequent interactions, allowing them to improve their performance over time without requiring explicit retraining.

Cross-Domain Prompting: AI systems are getting better at handling prompts that span multiple domains or areas of knowledge, allowing for more complex and comprehensive interactions.

화면 캡처: 2023-12-15 오전 10:04

특히 오른쪽 부분은 robotics ,강화 학습 부분에서 생각해보고 적용할 방법을 찾아서 적용해보면 재미있을것 같다

prompt 작성 방법

Six strategies for Prompt Engineering

▪ **Prompt engineering**: To improve the performance of large language models.

▪ **Six strategies**

1. Write clear instructions
2. Provide reference text
3. Split complex tasks into simpler subtasks
4. Give the model time to think
5. Use external tools
6. Test changes systematically

화면 캡처: 2023-12-15 오전 10:02

[Prompt templates](#) | [Langchain](#)

PromptTemplate를 통해 prompt적용 한다

input_variables 파라미터에 prompt에서 {}한 변수 적용 -> PromptTemplate or ChatPromptTemplate사용한다
return StringPromptValue

ChatPromptTemplate

: prompt to chat models -> sys, human, ai 등 role에 따른 prompt적용 가능하다

LCEL(langchain expression language) 최근에 추가된 것

support that calls : invoke, ainvoke, stream, astream, batch , abatch, astream_log

return ChatPromptValue

custom prompt template

prompt수정해서 적용하는거가 custom prompt 아닌가? 공식 문서만 보았을때는 뭔자인지 잘 모르겠다

Few shot prompt template

몇가지 example을 주면 그 형식을 배우는것이다

FewShotPromptTemplate를 이용해서 만든다 -> examples와 example_prompt를 넣어주면 됨

zero shot은 아무 예제 없지만 input을 어떻게 다르게 주면 더 학습한게 없어도 좋은 답안을 준다

ExampleSelector

주어진 example 기반으로 SemanticSimilarityExampleSelector는 비슷한 example을 생성해준다 -> 유사도를 봐야 하기 때문에 임베딩 모델이 필요하다

```
example_selector = SemanticSimilarityExampleSelector.from_examples(  
    # This is the list of examples available to select from.  
    examples,  
    # This is the embedding class used to produce embeddings which are used to measure semantic similarity.  
    OpenAIEmbeddings(),  
    # This is the VectorStore class that is used to store the embeddings and do a similarity search over.  
    Chroma,  
    # This is the number of examples to produce.  
    k=1  
)
```

출처: <https://python.langchain.com/docs/modules/model_io/prompts/prompt_templates/few_shot_examples>

5 type

```
Langchain provides 5 types of example samplers:  
1. Select by Length : LengthBasedExampleSelector  
2. Select by Maximal Margin Relevance : MaxMarginalRelevanceExampleSelect  
3. Select by n-gram similarity : NGramOverlapExampleSelector :  
4. Select by Similarity : SemanticSimilarityExampleSelector  
5. Custom Selector : CustomExampleSelector
```

화면 캡처: 2023-12-15 오전 6:58

Few shot for chat model

The basic components of the template are:

- `examples`: A list of dictionary examples to include in the final prompt.
- `example_prompt`: converts each example into 1 or more messages through its `format_messages` method. A common example would be to convert each example into one human message and one AI message response, or a human message followed by a function call message.

화면 캡처: 2023-10-23 오후 4:28

`format_messages`는 인간의 질문과 ai의 결과를 묶어서 표현한 예제들이다

Composition

prompt를 여러 개 결합 할 수가 있다

[Composition](#) | [Langchain](#)

outputparser

pydantic output parser : attribute의 type을 정한다 : pydantic 사용해서 output parser 실행하는게 좋다 -openai model만 적용됨

chain of thought prompt (COT)

prompt과정을 논리적으로 작성하는것이다

(a) Few-shot	(b) Few-shot-CoT
<p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?</p> <p>A:</p> <p>(Output) The answer is 8. ✗</p>	<p>Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?</p> <p>A:</p> <p>(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓</p>
(c) Zero-shot	(d) Zero-shot-CoT (Ours)
<p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?</p> <p>A: The answer (arabic numerals) is</p> <p>(Output) 8 ✗</p>	<p>Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?</p> <p>A: Let's think step by step.</p> <p>(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓</p>

화면 캡처: 2023-12-15 오전 10:20

automatic chain of thought (auto cot)

1. question clustering : 주어진 데이터에서 질문을 분할해서 cluster만들
2. demonstration sampling : 만든 cluster중에서 (많은 sampling중에서 대표를 선택)후 zero shot cot적용

self consistency

multimodal cot prompting

React

reason and acting

[ReAct: Synergizing Reasoning and Acting in Language Models – Google Research Blog](#)

Agent에서 사용시 일반적으로(모든 경우는 아니다) 어떤 tool을 선택할지 React과정에서 결정할수가 있다