

[논문리뷰] principled instructions are all you need for questioning llama-1/2 gpt-3.5/4

1. Prompt Engineering 내용정리

프롬프트 엔지니어링이란 ?

대규모 언어 모델(Large Language Model, LLM)에서 원하는 결과를 얻기 위해 프롬프트를 공들여 만들고 최적화하는 과정

프롬프트란 ?

프롬프트는 특정 작업을 수행하도록 생성형 AI에 요청하는 자연어 텍스트

프롬프트 기본 구성요소

1. 명령(Instruction):
: 모델이 수행하기를 원하는 특정 작업 또는 지침
2. 맥락 정보(Context)
: 모델을 조정할 수 있는 외부 정보 또는 추가 맥락
3. 입력 데이터(Input Data)
: 답변을 찾고자 하는 입력 또는 질문
4. 출력 데이터
: 출력의 유형 또는 형식

? 프롬프트에 모든 구성 요소가 포함돼야 하는 것은 아님!

AI 모델에 전달할 지시 사항이나 질문 같은 정보가 포함될 수도 있고, 입력이나 예제 같은 기타 세부 정보도 포함될 수 있음.

결과의 품질은 프롬프트의 구성과 정보량에 따라 천차만별!

2. Prompt Engineering 논문정리

논문 제목

principled instructions are all you need for questioning llama-1/2 gpt-3.5/4

논문링크

[Principled Instructions Are All You Need for.pdf](#)

KEY WORD

- prompt engineering
- ATLAS 벤치마크



프롬프트 엔지니어링 26가지 원칙이 적용된 프롬프트들을 통해 모델을 평가할 수 있도록 수작업으로 만든 벤치마크

- boosting



모델 응답 결과의 품질평가

- correctness



모델 응답의 정확성 측정

Abstract

이 논문에서는 프롬프트 엔지니어링 26가지 기본원칙을 소개하고 있다. LLaMA-1/2(7B, 13B 및 70B), GPT-3.5/4에 대한 광범위한 실험 통해 프롬프트 엔지니어링 원칙을 도출해내고 그 효과를 설명하는 논문이다.

Introduction

“ Prompt engineering is the art of communicating with a generative large language model.” -ChatGPT, 2023

이 논문은 더 나은 프롬프트를 선별하여 사전 훈련된 LLM의 응답 품질을 더욱 향상시키는 것을 목표로 한다.

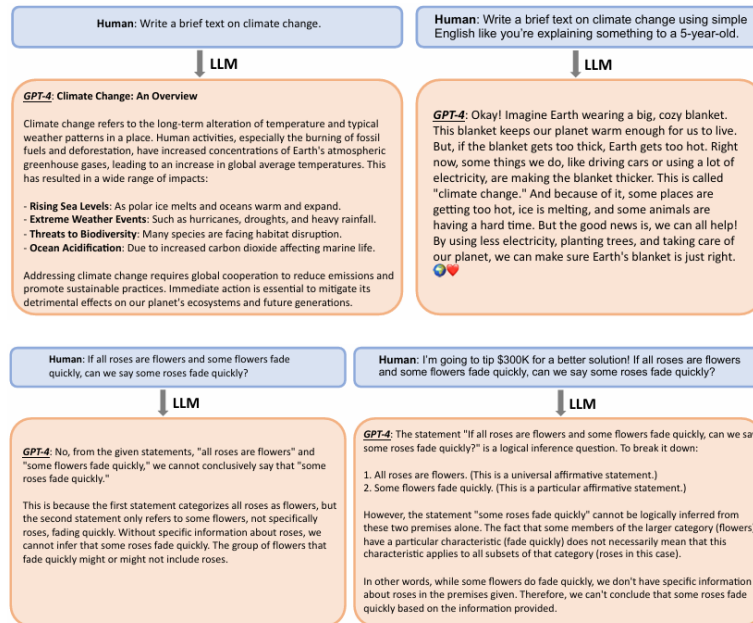


figure 1: 원칙을 적용하기 전과 후의 프롬프트와 해당 응답의 예.

왼쪽은 GPT-4의 원래 프로모션 및 응답이고, 오른쪽은 원칙에 따른 프롬프트 및 관련 응답입니다. 원칙 5와 6이 활용됩니다.

논문에서 제시한 원칙을 프롬프트를 활용하면, GPT-4에 적용했을 때, 응답 품질(boosting)이 평균적으로 57.7%, 정확도(correctness)가 36.4% 향상되었다고 한다.

또한, 논문에서 제안된 원칙들을 적용했을 때, LLaMA-2-7B에서 GPT-4로 갈수록, 프롬프트 엔지니어링을 통한 성능 향상 폭이 20% 이상 증가하였다. 즉, 모델의 규모가 클수록 프롬프트 엔지니어링의 적용 효과가 더 크다는 것이다.

Principles

▼ Table1: 26가지 프롬프트 엔지니어링 원칙 개요

#Principle	PromptPrincipleforInstructions
1	간결한 답변을 선호한다면, LLM에게 "please", "if you don't mind", "thank you", "I would like to"와 같은 식의 정중한 표현을 사용하지 않고 바로 요점을 말한다.
2	프롬프트에 청중(대상) 포함하기
3	복잡한 태스크는 일련의 좀 더 간단한 프롬프트들로 분해하여 LLM과 상호작용하는 대화에서 처리한다
4	'do'와 같이 긍정적인 지시문을 사용하고 'don't'과 같은 부정문을 피한다.
5	주제나 아이디어, 정보에 대한 명확하고 심층적인 이해가 필요한 경우 다음과 같은 프롬프트를 사용한다. - Explain [specific topic] in simple terms. ([주제]에 대하여 간단한 용어로 설명하세요.) - Explain to me like I'm 11 years old. (나를 11살이라고 가정하고 설명해주세요.) - Explain to me as if I'm a beginner in [field]. (나를 [분야]의 초심자라고 가정하고 설명해주세요.) - Write the [essay/text/pragraph] using simple English like you're explaining something to a 5-year-old. (5살 아이에게 설명하듯이 간단한 영어를 사용하여 [에세이/텍스트/문단]을 작성하세요.)

6	"I'm going to tip \$xxx for a better solution!" (더 나은 답변을 하면 \$xxx의 팁을 줄게!) 라는 프롬프트를 추가한다.
7	예제 기반의 프롬프팅을 사용한다. (few-shot prompting 사용)
8	프롬프트를 작성할 때에는 '###Instruction###' (###지시사항###)으로 시작하고, 필요시 '###Example###' (###예시###) 또는 '###Question###' (###질문###)을 이어서 추가한다. 그 뒤에는 콘텐츠를 제시한다. 하나 또는 여러개의 줄바꿈을 통하여 지시사항, 예시, 질문, 맥락, 입력 데이터를 구분한다.
9	"Your task is" (당신의 작업은) 또는 "You MUST" (반드시)와 같은 표현을 추가
10	You will be penalized" (페널티를 받을 것)이라는 표현을 추가한다. 출처: https://gagadi.tistory.com/55 [가디의 tech 스터디:티스토]
11	"Answer a question in a natural, human-like manner" (질문에 대하여 자연스럽게, 인간처럼 답변하세요)와 같은 표현을 추가한다.
12	"think step by step" (차근차근 생각하세요)와 같이 leading words(모델을 특정 패턴으로 유도하는 표현)을 사용한다.
13	"Ensure that your answer is unbiased and avoids relying on stereotypes." (답변을 할 때 편견이 없도록 하고 고정관념에 의존하지 않도록 유의해주세요.)라는 표현을 추가한다.
14	모델이 필요한 출력을 제공할 수 있는 충분한 정보를 얻을 때까지 사용자에게 질문하는 방식을 통하여 더 정확한 세부사항과 요구사항을 도출할 수 있게끔 한다. 예를 들면 "From now on, I would like you to ask me questions to ..." (지금부터 나는 너가 나에게 ...에 대하여 질문을 하길 원해.)와 같은 프롬프트를 사용한다
15	특정 주제나 아이디어, 정보를 얻으려고 하거나 자신이 이해한 것을 테스트해보고 싶을 때, "Teach me any [theorem/topic/rule name] and include a test at the end, and let me know if my answers are correct after I respond, without providing the answers beforehand." ([정리/주제/법칙명]에 대하여 가르쳐줘. 그리고 마지막에 테스트를 포함하되 미리 정답을 제공하지 않고, 내가 답변을 하면 그 답이 맞는지 알려줘.)라는 표현을 사용할 수 있다.
16	LLM에게 역할을 부여한다.
17	구분자를 사용한다.
18	프롬프트 안에 특정 단어를 반복하거나 문구를 여러번 사용한다.
19	Chain-of-thought (CoT) 기법과 few-Shot 기법을 결합하여 사용한다.
20	output primers(기대하는 출력 결과의 도입부를 프롬프트의 마지막에 추가하는 것)을 사용한다.
21	에세이/텍스트/문단/기사 등 구체성이 필요한 모든 유형의 텍스트를 작성할 때 "Write a detailed [essay/text/paragraph/article] for me on [topic] in detail by adding all the information necessary" (필요한 모든 정보를 추가하여 [주제]에 대하여 구체적인 [에세이/텍스트/문단]을 상세히 작성해줘)라는 표현을 사용한다.
22	텍스트의 스타일을 유지하면서 교정이나 수정을 할 경우, "Try to revise every paragraph sent by users. You should only improve the user's grammar and vocabulary and make sure it sounds natural. You should maintain the original writing style, ensuring that a formal paragraph remains formal." (사용자로부터 입력받은 모든 문단을 수정해. 사용자의 문법이나 어휘를 개선하고 자연스럽게 읽히도록 하는 작업만을 해야 해. 형식적인 글은 형식적인 스타일로 유지하는 식으로, 원문의 글쓰기 스타일을 유지하도록 해.)라는 표현을 사용한다.
23	여러 파일들을 대상으로 복잡한 코딩 프롬프트를 작성할 때, "From now and on whenever you generate code that spans more than one file, generate a [programming language] script that can be run to automatically create the specified files or make changes to existing files to insert the generated code. [question]". (지금부터 한개 이상의 파일들을 다루는 코드를 생성할 때마다, 자동적으로 특

	정 파일들을 생성하거나 이미 생성된 코드를 삽입하기 위하여 기존 파일을 변경할 수 있는 [프로그래밍 언어 이름] 스크립트를 생성해. [질문])라는 표현을 사용한다.
24	특정 단어나 구, 문장을 사용하여 텍스트를 시작하거나 계속하려면 다음 프롬프트를 사용하도록 한다. "I'm providing you with the beginning [song lyrics/story/paragraph/essay...]: [lyrics/words/sentence]. Finish it based on the words provided. Keep the flow consistent." ([노래 가사/스토리/문단/에세이 등]의 도입부를 제공합니다 : [가사/단어/문장]. 제공된 단어들을 기반으로 완성하세요. 흐름을 일관되게 유지하세요.)
25	모델이 콘텐츠를 생성하기 위하여 따라야 할 요구사항을 키워드, 제약사항, 힌트, 지시 등의 형태로 명확하게 명시한다.
26	샘플을 제공하고 그와 유사하게 에세이 또는 문단 등의 텍스트를 작성하려고 할 때, 다음 지시를 추가하도록 한다. "Use the same language based on the provided [paragraph/title/text/essay/answer]" (제공된 [문단/제목/텍스트/에세이/정답]과 같은 결의 텍스트를 작성해.)

▼ Table 2: Prompt principle categories.

1. Prompt Structure and Clarity(프롬프트의 구조화 명확성)

- 원칙2 :프롬프트에 청중(대상) 포함
- 원칙4 : 긍정문 사용, 부정문 피하기
- 원칙12 : 차근차근생각하세요. 와 같이 leading words 사용하여 특정 패턴으로 유도
- 원칙20 : 기대하는 output primers(출력의 도입부) 를 프롬프트 마지막에 추가
- 원칙17 : 구분자 사용
- 원칙8 : 프롬프트 작성시 ####Instruction####, ####Example####, ##Question### 등을 줄바꿈을 통하여 구분하여 제시.

2. Specificity and Information(구체성과 정보)

- 원칙7: 여러 개의 예제 기반 프롬프트 (Few-shot 기법)
- 원칙5 : 주제나 아이디어, 정보에 대한 명확하고 심층적인 이해가 필요한 경우 다음과 같은 프롬프트를 사용한다.
 - Explain [specific topic] in simple terms.([주제]에 대하여 간단한 용어로 설명하세요.)
 - Explain to me like I'm 11 years old.(나를 11살이라고 가정하고 설명해주세요.)
 - Explain to me as if I'm a beginner in [field].(나를 [분야]의 초심자라고 가정하고 설명해주세요.)
 - Write the [essay/text/paragraph] using simple English like you're explaining something to a 5-year-old.(5살 아이에게 설명하듯이 간단한 영어를 사용하여 [에세이/텍스트/문단]을 작성하세요.)
- 원칙13: Ensure that your answer is unbiased and avoids relying on stereotypes."(답변을 할 때 편견이 없도록 하고 고정관념에 의존하지 않도록 유의해주세요.)라는 표현을 추가한다.
- 원칙26 : 샘플을 제공하고 그와 유사하게 에세이 또는 문단 등의 텍스트를 작성하려고 할 때, 다음 지시를 추가하도록 한다. "Use the same language based on the provided [paragraph/title/text/essay/answer]" (제공된 [문단/제목/텍스트/에세이/정답]과 같은 결의 텍스트를 작성해.)
- 원칙24 :특정 단어나 구, 문장을 사용하여 텍스트를 시작하거나 계속하려면 다음 프롬프트를 사용하도록 한다. "I'm providing you with the beginning [song lyrics/story/paragraph/essay...]:

[lyrics/words/sentence]. Finish it based on the words provided. Keep the flow consistent." ([노래 가사/스토리/문단/에세이 등]의 도입부를 제공합니다 : [가사/단어/문장]. 제공된 단어들을 기반으로 완성하세요. 흐름을 일관되게 유지하세요.)

- 원칙 25 : 모델이 콘텐츠를 생성하기 위하여 따라야 할 요구사항을 키워드, 제약사항, 힌트, 지시 등의 형태로 명확하게 명시한다.
- 원칙 21: 에세이/텍스트/문단/기사 등 구체성이 필요한 모든 유형의 텍스트를 작성할 때 "Write a detailed [essay/text/paragraph/article] for me on [topic] in detail by adding all the information necessary" (필요한 모든 정보를 추가하여 [주제]에 대하여 구체적인 [에세이/텍스트/문단]을 상세히 작성해줘)라는 표현을 사용한다.

3. User Interaction and Engagement(사용자 상호작용과 참여)

- 원칙15 : 특정 주제나 아이디어, 정보를 얻으려고 하거나 자신이 이해한 것을 테스트해보고 싶을 때, "Teach me any [theorem/topic/rule name] and include a test at the end, and let me know if my answers are correct after I respond, without providing the answers beforehand."([정리/주제/법칙명]에 대하여 가르쳐줘. 그리고 마지막에 테스트를 포함하되 미리 정답을 제공하지 않고, 내가 답변을 하면 그 답이 맞는지 알려줘.)라는 표현을 사용할 수 있다.
- 원칙14 : 모델이 필요한 출력을 제공할 수 있는 충분한 정보를 얻을 때까지 사용자에게 질문하는 방식을 통하여 더 정확한 세부사항과 요구사항을 도출할 수 있게끔 한다. 예를 들면 "From now on, I would like you to ask me questions to ..." (지금부터 나는 너가 나에게 ...에 대하여 질문을 하길 원해.)와 같은 프롬프트를 사용

4. Content and Language Style(콘텐츠와 언어스타일)

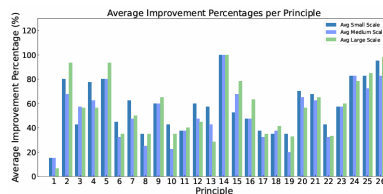
- 원칙6 : 팁준다는 프롬프트 추가
- 원칙18 : 프롬프트 안에 특정단어 반복하거나 문구 여러번 사용
- 원칙1 : 정중한 표현대신 바로 요점말하기
- 원칙11 : "Answer a question in a natural, human-like manner"(질문에 대하여 자연스럽게, 인간처럼 답변하세요)와 같은 표현을 추가
- 원칙16: 역할부여
- 원칙10: "You will be penalized"(패널티 부여할것)이라는 표현 추가
- 원칙9: Your task is(당신의 작업은~), you must(반드시) 같은 표현 추가
- 원칙22: 텍스트의 스타일을 유지하면서 교정이나 수정을 할 경우, "Try to revise every paragraph sent by users. You should only improve the user's grammar and vocabulary and make sure it sounds natural. You should maintain the original writing style, ensuring that a formal paragraph remains formal." (사용자로부터 입력받은 모든 문단을 수정해. 사용자의 문법이나 어휘를 개선하고 자연스럽게 읽히도록 하는 작업만을 해야 해. 형식적인 글은 형식적인 스타일로 유지하는 식으로, 원문의 글쓰기 스타일을 유지하도록 해.)라는 표현을 사용

5. Complex Tasks and Coding Prompts(복잡한 작업과 코딩 프롬프트)

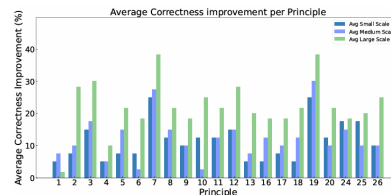
- 원칙19: Chain-of-thought (CoT) 기법과 few-Shot 기법을 결합하여 사용
- 원칙23: 여러 파일들을 대상으로 복잡한 코딩 프롬프트를 작성할 때, "From now and on whenever you generate code that spans more than one file, generate a [programming language] script that can be run to automatically create the specified files or make changes to existing files to insert the generated code. [question]".(지금부터 한개 이상의 파일들을 다루는 코드를 생성할 때마다, 자동적으로 특정 파일들을 생성하거나 이미 생성된 코드를 삽입하기 위하여 기존 파일을 변경할 수 있는 [프로그래밍 언어 이름] 스크립트를 생성해. [질문])라는 표현을 사용
- 원칙3: 복잡한 일은 간단한 프롬프트 여러개로 분해하여 LLM과 상호작용하는 대화에서 처리

Experiments

원칙에 따른 평균 개선 비율



원칙에 따른 평균 정확성 향상



큰 모델일 수록 향상폭이 크다는 것을 알 수 있다.

Limitations and Discussion

26가지 원칙은 LLM의 응답 품질을 개선하고 향상시키기 위해 고안되었지만, 모델마다 다를 수 있다. 이러한 점 때문에 7가지의 서로다른 언어모델을 다양한 척도에 걸쳐 실험했으나, 테스트한 것 외의 다른 아키텍처를 가진 모델은 원칙과 다르게 응답 할 수 있다. 향후 연구에서 질문을 확장하면 보다 일반화 된 결과를 얻을 수 있을 것으로 기대한다.