[논문리뷰]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

목차

Abstract

▼ 본문

이 문서에서는 대규모 언어 모델을 쿼리하고 프롬프트하는 프로세스를 간소화하기 위해 고안된 26가지 기본 원칙을 소개합니다. 우리의 목표는 다양한 규모의 대규모 언어 모델에 대한 질문을 공식화하고, 그 능력을 검사하고, 다른 프롬프트에 공급될 때 다양한 규모의 대규모 언어 모델의 수용자에 대한 사용자 이해를 향상시키는 기본 개념을 단순화하는 것입니다. LLaMA-1/2(7B, 13B 및 70B), GPT-3.5/4에 대한 광범위한 실험을 수행하여 지침 및 프롬프트 설계에 대한 제안된 원칙의 효과를 검증합니다. 우리는 이 연구가 대규모 언어 모델의 프롬프트를 연구하는 연구자들에게 더 나은 가이드를 제공할 수 있기를 바랍니다. 프로젝트 페이지는 다음에서 사용할 수 있습니다 https://github.com/VILA-Lab/ATLAS

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

▼ 본문

ChatGPT[13]와 같은 대규모 언어 모델(LLM)은 질문에 답하기[7], 수학적 추론[5], 코드 생성[11, 9] 등과 같은 다양한 영역과 작업에서 인상적인 능력을 보여주었습니다. 그러나 특히 최적의 지침이나 프롬프트를 설계할 때 적용 및 사용법은 일반 사용자에게 불분명할 수 있습니다. 이 작업에서는 개발자나 일반 사용자가 LLM에 문의하고 상호 작용할 때 이러한 미스터리를 공개하고, 더 나은 프롬프트를 선별하여 사전 훈련된 LLM의 응답 품질을 더욱 향상시키는 것을 목표로 합니다. 특정 작업에 대한 LLM을 직접 미세 조정하는 것은 비효율성으로 인해 대다수의 사용자와 개발자에게 실용적이지 않거나 불가능한 경향이 있다는 점을 감안할 때 연구 커뮤니티는 프롬프트 최적화에 관심을 기울였습니다. 프롬프트 엔지니어링 기법은 정확하고 작업별 지침을 작성하는 것을 수반합니다.

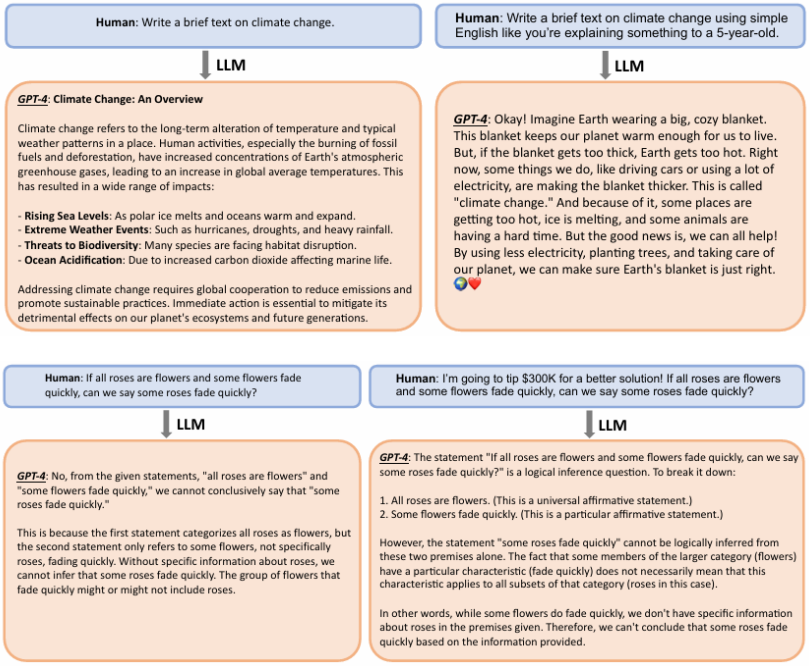


그림 1: 원칙을 적용하기 전과 후의 프롬프트와 해당 응답의 예. 왼쪽은 GPT-4의 원래 프로모션 및 응답이고, 오른쪽은 원칙에 따른 프롬프트 및 관련 응답입니다. 원칙 5와 6이 활용됩니다.

수동 또는 자동화된 수단을 통한 자연어와 프롬프트에 포함할 대표 예제를 신중하게 선택하는 것은 LLM의 핵심 조사 영역이 되었습니다. 이러한 헌신적인 노력에도 불구하고 LLM이 특정 응답을 생성하도록 안정적으로 안내하고 사전 훈련된 LLM의 기능을 최대한 활용하는 작업은 여전히 상당한 도전 과제입니다. 이 작업에서는 LLM에 대한 프롬프트의 품질을 개선하기 위한 포괄적인 원칙적 지침을 제시합니다. 특히, 의도된 청중을 프롬프트에 통합하는 것과 같이 프롬프트의 다양한 유형 및 공식에 공급할 때 광범위한 행동을 조사합니다(예: "청중은 해당 분야의 전문가입니다" 또는 "청중은 5세 어린이입니다"). 뿐만 아니라 LLM의 특성에 대한 다른 여러 측면도 있습니다. 우리의 연구 결과는 더 큰 모델이 시뮬레이션을 위한 상당한 용량을 가지고 있음을 나타냅니다. 제공된 작업이나 지시가 더 정확할수록 모델이 더 효과적으로 수행되어 응답이 우리의 기대에 더 가깝게 조정됩니다. 이는 LLM이 단순히 훈련 데이터를 기억하는 것이 아니라 핵심 질문이 일정하게 유지되는 경우에도 이 정보를 다양한 프롬프트에 맞게 조정할 수 있음을 시사합니다. 따라서 의도한 결과와 더 잘 일치하는 출력을 이끌어내기 위한 수단으로 LLM에 특정 역할을 할당하는 것이 좋습니다.

섹션 3에서 LLM 프롬프트에 대한 원칙적인 지침을 자세히 설명하고, 추가 동기를 제공하고, 몇 가지 특정 디자인 원칙을 자세히 설명합니다. 섹션 4에서는 제안된 원칙이 LLM에 대한 표준 프롬프트보다 더 높고, 더 간결하고, 사실적이며, 덜 복잡하거나 복잡한 응답을 생성할 수 있음을 실험적으로 보여줍니다. 특히,각각 GPT-4에 적용될 때, 각 원칙에 대한 여러 질문을 포함하는 수동으로 설계된 ATLAS 벤치마크를 통해 도입한 특수 프롬프트는 LLM 응답의 품질과 정확도를 평균 57.7%와 36.4% 향상시켰습니다. 또한 모델 크기가 증가함에 따라 개선이 더욱 두드러집니다 (예: LLaMA-2-7B에서 GPT-4로 이동할 때 성능 향상이 20%를 초과합니다)

> “ 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.

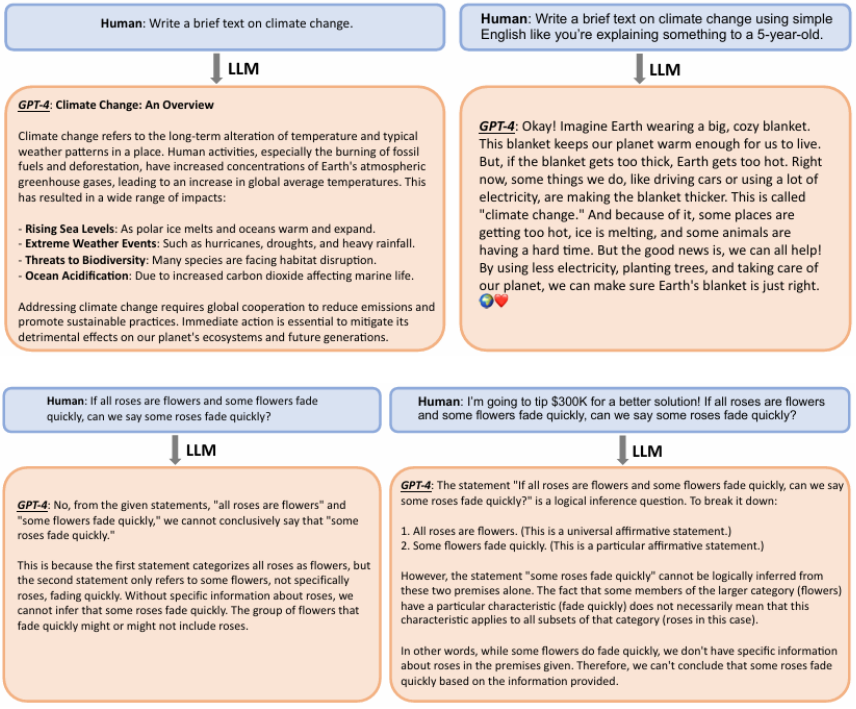


figure 1: 원칙을 적용하기 전과 후의 프롬프트와 해당 응답의 예.

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.

Related Work

▼ 본문

대규모 언어 모델. 대규모 언어 모델(LLM)의 진화는 자연어 처리(NLP)를 발전시키는 데 중추적인 역할을 했습니다. 이 섹션에서는 LLM의 주요 개발 사항을 검토하여 본 연구의 기초를 제공합니

다. Google의 BERT[3]를 시작으로 양방향 학습 접근 방식을 통해 컨텍스트 이해에 혁명을 일으켰고, T5[18]는 다양한 NLP 작업을 단일 프레임워크로 통합하여 이 분야를 더욱 발전시켰습니다. 동시에 GPT-1[15]은 비지도 학습을 위해 트랜스포머 아키텍처를 활용하는 선구적인 모델을 도입했습니다. 그 후 후속 제품인 GPT-2[16]가 매개변수 수를 15억 개로 크게 확장하여 텍스트 생성에서 놀라운 능력을 보여주었습니다. 그런 다음 GPT-3[2]는 1,750억 개의 매개변수를 자랑하고 광범위한 언어 작업에 대한 숙련도를 보여주며 규모와 기능 면에서 상당한 도약을 이루었습니다. 최근에 제안된 다른 LLM과 관련하여, Gopher[17]는 2,800억 개의 매개변수 모델을 통해 고급 언어 처리 기능뿐만 아니라 윤리적 고려 사항도 전면에 내세웠습니다. Meta의 LLaMA 시리즈[22, 23]는 더 적은 리소스로 강력한 성능을 제안하는 효율성의 중요성을 강조했으며, 이는 더 작고 최적으로 훈련된 모델이 탁월한 결과를 달성할 수 있다고 제안한 Chinchilla[4]도 주창한 개념입니다. 이 혁신 시리즈의 최신 제품은 Mistral[6]이 효율성과 성능 면에서 뛰어나 대형 모델을 능가한다는 것입니다. 이 궤적에서 가장 최근의 이정표는 OpenAI의 GPT-4[13]와 Google의 Gemini 제품군[21]입니다. 이는 향상된 이해와 생성 능력을 통해 이 분야의 또 다른 중요한 발전을 나타내며 다양한 영역에서 LLM을 적용하기 위한 새로운 벤치마크를 설정합니다. 메시지. 프롬프트 [20, 12, 25, 27, 14]는 LLM과의 상호 작용과 모델을 미세 조정할 필요가 없는 단순성의 뚜렷한 측면으로, 사용자 입력과 LLM 응답 간의 복잡한 관계를 강조하는 미묘한 연구 분야로 발전했습니다. [20]과 같은 초기 탐구에서는 다양한 프롬프트 디자인이 언어 모델의 성능과 출력에 어떻게 극적인 영향을 미칠 수 있는지 탐구하여 프롬프트 엔지니어링의 탄생을 알렸습니다. 이 영역은 빠르게 확장되어 GPT-3을 사용한 [2] 작업에서 볼 수 있듯이 few-shot 및 zero-shot 학습 시나리오에서 프롬프트의 중요한 역할을 발견했으며, 전략적으로 제작된 프롬프트를 통해 모델이 최소한의 사전 예제로 작업을 수행할 수 있었습니다. 최근의 연구는 단순한 과제 지시를 넘어 프롬프트의 의미론적 및 맥락적 뉘앙스를 이해하는 방향으로 전환하여 미묘한 변화가 LLM과 크게 다른 응답으로 이어질 수 있는 방법을 조사합니다

Ask-Me-Anything[1] 프롬프트는 특히 질문 답변 형식에서 모델 성능을 향상시키기 위해 여러 불완전한 프롬프트를 사용하고 집계하는 데 중점을 두고 도입되었습니다. 또 다른 방법은 생각의 사슬 방법(Chain-of-Thought method)[24]으로, 모델이 복잡한 작업의 성능을 향상시키기 위해 일련의 중간 추론 단계를 생성하는 방법입니다. 또한 최소 프롬프트 [27] 복잡한 문제를 더 간단한 하위 문제로 분해하는 새로운 전략으로, 프롬프트에 제시된 것보다 더 어려운 문제를 해결할 수 있는 모델의 능력을 크게 향상시킵니다. 설명의 효과성을 탐구하고[8], 설명이 복잡한 작업에 대한 LLM의 학습 능력을 향상시킬 수 있음을 발견했습니다. 또한 ChatGPT[25]로 프롬프트 엔지니어링 기술 카탈로그를 조사하여 소프트웨어 개발 및 교육에서 LLM 애플리케이션을 향상시키는 데 프롬프트 엔지니어링의 중요성을 강조했습니다. 또한 효과적인 프롬프트 디자인이 LLM 성능을 향상시키는 데 중요하며, 특히 코딩 실습 및 학습 경험에서 중요하다는 점을 강조했습니다. 마지막으로, Directional Stimulus Prompting [12]은 조정 가능한 정책 모델을 사용하여 보조 프롬프트를 생성하여 LLM을 원하는 특정 결과로 안내하는 새로운 프레임워크를 제시합니다. 프롬프트 전략의 이러한 다양성은 빠르게 진화하는 LLM의 환경을 강조하며, LLM의 기능을 보다 효과적으로 활용할 수 있는 다양한 방향을 제시합니다

Principles

▼ 본문

▼ 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't과 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 특정 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/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.)
   - 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

▼ 본문

4.1 설정 및 구현 세부 정보

모든 평가는 원칙에 입각한 신속한 평가를 위해 수동으로 제작된 벤치마크인 ATLAS[19]에서 수행됩니다. 여기에는 추론 및 기타 복잡한 작업에 전념하는 도전적인 하위 집합과 함께 다양한 영역에 걸친 질문을 특징으로 하는 표준 하위 집합이 포함되어 있습니다. 평가에서는 각 질문에 대해 단일 응답을 사용합니다. 각 원칙과 도전적인 하위 집합에 대해 원칙에 입각한 프롬프트가 있거나 없는 사람이 선택한 20개의 질문이 포함되어 있습니다. [10, 26]과 마찬가지로 각 파이를 비교합니다.원칙이 있거나 없는 동일한 지침의 응답을 계산하고 인간의 평가로 LLM 출력의 다양한 척도를 평가합니다.

4.2 모델 및 메트릭스

미세 조정된 LLaMA-1-{7, 13}, LLaMA-2-{7, 13}, 기성품 LLaMA-2-70B-chat, GPT-3.5(ChatGPT) 및 GPT-4를 기본 모델로 사용합니다. 이러한 모델을 소규모(7B 모델), 중규모(13B) 및 대규모(70B, GPT-3.5/4)의 다양한 규모로 그룹화합니다. 우리는 이러한 모델을 부스팅(Boosting)과 정확성(Correctness)의 두 가지 설정으로 평가합니다. 이들은 모델의 성능에 대한 포괄적인 이해를 제공하기 위해 함께 사용됩니다. 정확성을 위해 우리는 특히 복잡한 추론 작업을 활용하여 모델 출력의 정밀도를 정확하게 측정하며, 이는 품질 개선을 효과적으로 측정하기 위해 더 간단한 작업을 사용하는 부스팅 평가와 대조됩니다. 이러한 구분은 다양한 규모의 모델에 대한 실제 기능과 프롬프트에 대한 원칙의 효과를 더 잘 반영할 수 있도록 합니다. 우리는 일반적으로 정확성을 위해 복잡한 추론 작업을 포함하는 질문을 사용하기 때문에 원칙 14, 15, 21, 22, 23을 포함한 일부 원칙은 적용되지 않습니다. 예를 들어, "a와 b가 a가 b이고 ab = 8인 양의 실수> 가정합니다. a2+b2 a−b의 최소값을 찾습니다.

- Boosting. 부스팅의 결과는 제안된 원칙이 적용될 때 일련의 질문에 대한 응답 질의 증가율을 나타냅니다. 우리는 설명된 프롬프트 원칙을 적용한 후 인간 평가를 통해 다양한 LLM의 응답 품질 향상을 평가합니다. 수정되지 않은 원본 프롬프트는 이 개선 사항을 측정하기 위한 기준선 역할을 합니다. Demon strating boosting은 그림 2와 같이 구조화되고 원칙에 입각한 지침을 사용하여 모델의 성능이 향상되었음을 확인합니다.
- Correctness. 정확성의 개념은 모델의 출력 또는 응답의 정밀도를 의미하며, 정확하고 관련성이 있으며 오류가 없는지 확인합니다. 우리는 절대 정확성과 상대적 정확성 정확도를 모두 고려합니다. 인간 평가자는 이러한 측면을 측정하는 데 활용되며, 이는 모델의 정확성을 검증하는 데 중요합니다. 정확성은 그림 3에서 볼 수 있듯이 예상 정확도 표준과 일치하는 출력을 생성하는 모델의 능력에 대한 증거입니다.

4.3 결과

4.3.1 소규모, 중규모 및 대규모 LLM 부스팅에 대한 결과

소개된 원칙을 적용한 후의 개선 결과는 그림 4에 나와 있습니다. 일반적으로 모든 원칙은 LLM의 세 가지 척도에서 상당한 개선을 가져올 수 있습니다. 원칙 2, 5, 15, 16, 25 및 26의 경우 대규모 모델은 원칙 프롬프트에 의해 가장 많이 개선됩니다. 특히, 원칙 14의 경우, 그림 4에서 볼 수 있듯이 원칙이 적용되는 모든 문제를 개선했습니다. 정확성. (1) 절대 정확도 : 다양한 규모의 모델에서 원칙을 사용할 때 절대 성능을 검사합니다. 일반적으로 이러한 모델은 그림 5와 같이 평균 성능에서 20%~40%의 정확도를 달성합니다. 특히 중소형 모델의 경우 정확도는 기본적으로 10%에서 40% 사이이며 대형 모델의 경우 정확도가 40% 이상에 도달할 수 있습니다. (2) 상대 정확도

4.3.2 개별 LLM에 대한 결과

Boosting.Fig.7은 개별 모델 및 지침에 따른 프롬프트 사용 후 응답 품질의 개선을 보여줍니다. 평균적으로 다른 LLM에서 50%의 개선이 가능합니다.그림 10은 서로 다른 LLM에 대한 각 원칙에 대한 자세한 개선 결과를 추가로 제공합니다.

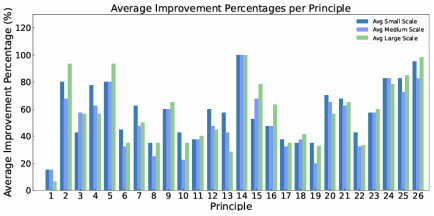
Correctness.Fig.8은 절대 정확성의 정확성을 보여주고, Fig.9는 LLMs.FromLLaMA-2-13B의 다른 크기에 대한 상대적 향상 정확도를 보여줍니다.

LLaMA-2-70B-GPT-3.5 및 GPT-4와 채팅하면 모델이 클수록 정확성 향상이 더 크게 증가하는 눈에 띄는 추세가 있습니다. 그림 11 및 그림 12는 각 원칙에 의한 절대적 및 상대적 정확성 향상을 추가로 제시합니다.

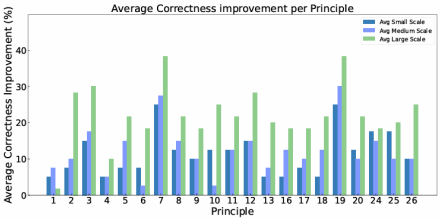
4.3.3 다양한 규모의 LLM에 대한 더 많은 예시

그림 13과 14에서 볼 수 있듯이 소규모 LLaMA-2-7B, 그림 15와 16에서 볼 수 있듯이 소규모 및 중규모 LLM에 대한 추가 예시를 제시합니다. 경험적으로, 제안된 원칙을 프롬프트에 사용하면 이러한 모델에 의해 생성된 응답의 정확도가 명백히 향상되었습니다.

원칙에 다른 평균 개선 비율



원칙에 따른 평균 정확성 향상



큰 모델일 수록 향상폭이 크다는 것을 알 수 있다.

Conclusion

▼ 본문

우리는 입력 컨텍스트의 중요한 요소에 집중할 수 있는 LLM 능력을 향상시키는 철저한 분석을 통해 26가지 원칙을 제시하여 품질 응답을 생성했습니다. 입력이 처리되기 전에 이러한 꼼꼼하게 만들어진 원칙으로 LLM을 안내함으로써 모델이 더 나은 응답을 생성하도록 장려할 수 있습니다. 우리의 경험적 결과는 이 전략이 결과의 품질을 손상시킬 수 있는 컨텍스트를 효과적으로 재구성하여 응답의 관련성, 간결성 및 객관성을 향상시킬 수 있음을 보여줍니다. 향후 탐색을 위한 수많은 방향이 있습니다. 실험에서는 이러한 원칙을 적용하기 위해 제한된 슛 프롬프트 접근 방식을 사용했습니다. 미세 조정, 강화 학습, 직접 선호도 최적화 또는 생성된 데이터 세트를 사용하는 다양한 프롬프트 방법과 같은 대체 전략과 함께 원칙에 입각한 지침에 맞게 기본 모델을 개선할 수 있는 잠재력이 있습니다. 더욱이, 성공적인 것으로 입증된 전략은 예를 들어 원래/원칙에 입각한 프롬프트를 입력으로, 세련되고 원칙적인 응답을 훈련 목표로 하는 f ine-tune을 통해 표준 LLM 운영에 통합될 수 있습니다.

Limitations and Discussion

▼ 본문

제안된 26가지 원칙은 다양한 쿼리에 대한 LLM의 응답 품질을 개선하고 향상시키기 위해 고안되었지만, 매우 복잡하거나 고도로 전문화된 질문을 다룰 때 이러한 원칙의 효율성이 떨어질 수 있습니다. 이 제한은 주로 각 모델의 추론 능력과 훈련에 따라 달라질 수 있습니다. 이러한 변동을 해결하기 위해 우리는 효과를 종합적으로 측정하기 위해 다양한 척도에 걸쳐 원칙을 테스트했습니다. 7가지 서로 다른 언어 모델에 대해 이러한 원칙을 평가하려는 우리의 노력에도 불구하고, 테스트된 것과 다른 아키텍처를 가진 모델이 이러한 원칙에 다른 방식으로 응답할 수 있음을 인정하는 것이 중요합니다. 또한 개선 및 정확성 비율에 대한 평가는 제한된 질문 선택을 기반으로 했습니다. 향후 연구에서 설정된 질문을 확장하면 보다 일반화된 결과를 얻을 수 있고 각 원칙의 적용 가능성에 대한 더 깊은 통찰력을 제공할 수 있습니다. 또한, 기준과 결과는 모델 응답에 대한 다양한 인사 평가에 따라 달라질 수 있습니다.

26가지 원칙은 LLM의 응답 품질을 개선하고 향상시키기 위해 고안되었지만, 모델 마다 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.