

What is ChatGPT ?

- ChatGPT라고 불리는 GPT-3.5는 GPT-3의 작지만 대화에 특화된 버전
- GPT-3와 같은 일반적인 텍스트 변환 및 생성은 처리하지 않지만 채팅과 관련된 모든 것을 처리
- 이 모델은 채팅의 맥락, 구조 및 어조를 더 깊이 이해하면서 자연스럽게 대화를 이어가는 유일한 목적으로 훈련. 따라서 채팅과 관련하여 범용 GPT-3에 비해 훨씬 더 정확하고 빠른 모델을 만든다.

- **1단계 : Supervised Fine Tuning(SFT) model 작성**

- GPT-3.5를 초기 model로 사용
- 인간이 의도하는 정책을 학습 시키기 위해 인간 labeler가 모델이 생성한 output 에 점수 부여
- 얻어진 데이터를 이용하여 지도학습으로 Fine Tuning

- **2단계 : Reward Model 작성(인간 선호도 모방)**

- 1단계에서 Fine Tuning된 SFT 모델의 여러 답변 후보에 인간 labeler가 ranking을 매긴 데이터셋 수집
- 이 데이터셋을 이용하여 새로운 reward model 학습

- **3단계 : STF 모델을 강화학습**

Chat-GPT 사용 example

- <https://openai.com/blog/chatgpt/>



나는 인공지능 강사야. 나의 인지도를 높이기 위한 방법 5 가지만 알려줘. 한글로 알려줘

Zero shot 추론



1. 최신 인공지능 개발을 추적하기: 산업 뉴스와 발전, 회의와 워크숍, 관련 연구 논문과 블로그를 읽어보세요.
2. 인공지능의 특정 분야에 전문화하기: 컴퓨터 비전, 자연어 처리, 보상 학습 등의 특정 인공지능 분야에 집중하고 그 분야에서 전문가가 되세요.
3. 자신만의 인공지능 프로젝트 구축하기: 실제 경험은 인공지능의 이해를 깊이 하는 최고의 방법 중 하나입니다. 간단한 프로젝트부터 시작하고 경험이 많아질수록 복잡성을 증가시킵니다.
4. 다른 인공지능 전문가와 협업하기: 온라인 커뮤니티와 포럼에 가입하고, 미팅에 참석하고, 다른 인공지능 전문가와 프로페셔널과 협업할 기회를 찾아보세요.
5. 다른 사람들에게 인공지능



마지막이 뭐라고 ?



미안합니다, 제 정보가 잘못 전달되어 완전한 답변을 제공하지 못했습니다. 마지막으로 다른 사람들에게 인공지능을 가르치는 것이 언급되었습니다. 이는 나의 지식을 강화하고, 다른 사람들에게 인공지능에 대한 지식을 제공하는 기회를 제공하기 때문에 유용한 것입니다.



소프트웨어 산업협회에서 올해 처음으로 인공지능 트렌드 과정을 열었고 많은 사람들의 호평 속에 끝났습니다. **이걸 홍보하려는데 기사로 써 주세요.**

기사 작성 요청






"인공지능의 미래를 예측하고 현재의 최신 기술을 이해하는 것이 중요한데, 소프트웨어 산업협회에서 이에 대한 첫 번째 기회를 제공하였습니다. 올해 처음으로 열린 인공지능 트렌드 과정은 많은 사람들의 호평 속에 끝나었습니다. 트렌드 과정에서는 최신 인공지능 기술과 이를 활용한 사례, 인공지능의 미래 전망 등에 대해 심도 있게 논의하였습니다. 이 과정은 인공지능을 활용한 미래 기술의 발전에 대한 이해를 높이는 데 도움이 될 것입니다."




100 이하의 숫수를 반환하는 python code 를 작성해줘

Program Code 작성



100 이하의 소수를 반환하는 Python 코드를 작성해 드리겠습니다. 소수는 1과 자기 자신만으로 나누어 떨어지는 양의 정수를 말합니다.   

python

 Copy code

```
def is_prime(n):  
    if n <= 1:  
        return False  
    elif n <= 3:  
        return True  
    elif n % 2 == 0 or n % 3 == 0:  
        return False  
    i = 5  
    while i * i <= n:  
        if n % i == 0 or n % (i + 2) == 0:  
            return False  
        i += 6  
    return True
```

```
def find_primes(limit):  
    primes = []  
    for num in range(2, limit + 1):  
        if is_prime(num):  
            primes.append(num)  
    return primes  
  
limit = 100  
prime_numbers = find_primes(limit)  
print(prime_numbers)
```



위 코드를 C 언어로 바꿔줘



아래는 C 언어로 100 이하의 소수를 찾는 코드입니다. 이 코드는 주어진 숫자가 소수인지 여부를 확인하고, 100 이하의 모든 소수를 출력합니다.



c

Copy code

```
#include <stdio.h>
#include <stdbool.h>

bool is_prime(int n) {
    if (n <= 1) {
        return false;
    }
    if (n <= 3) {
        return true;
    }
    if (n % 2 == 0 || n % 3 == 0) {
        return false;
    }
}
```

Github Copilot

The image is a composite of two screenshots from a VS Code editor. The left screenshot shows the GitHub Copilot chat interface. At the top, it says 'GitHub Copilot'. Below that, there's a message from 'ironmanciti' asking for an explanation in Korean. The Copilot response explains a Python code snippet: it generates a list of numbers from 1 to 100, defines a function 'get_even_numbers' that filters even numbers, and then uses list comprehensions to create 'even_numbers' and 'odd_numbers' lists, finally printing them. The right screenshot shows the Python code being discussed. It starts with 'numbers = list(range(1, 101))', followed by a function 'def get_even_numbers(numbers):' that returns a list of even numbers. Then, 'even_numbers' and 'odd_numbers' are created using list comprehensions, and they are printed. A large, bold, red 'Copilot' logo is centered over the right half of the image.

Chat-GPT의 한계



2023년 9월 현재 한국의 대통령이 누구야?



2021년 9월까지의 지식을 기반으로 한 내용을 제공해 드릴 수 있으며, 이후의 정보는 알 수 없습니다. 2023년 9월 현재 한국의 대통령은 현재 대통령 임기가 진행 중이라면 이전에 2021년 9월까지 기록된 내용과 달라질 수 있습니다. 대한민국의 현재 대통령에 대한 정보는 최신 뉴스 또는 정부 공식 웹사이트와 같은 신뢰할 수 있는 소스에서 확인하시기를 권장합니다.



가장 최근에 어느 나라에서 지진이 발생했어?



저의 지식은 2021년 9월까지의 정보까지만 포함하고 있으며, 현재 정보를 제공할 수 없습니다. 하루 이내의 지진 정보는 지진 감지 네트워크와 관련 기관 또는 지진 정보 웹사이트에서 확인할 수 있습니다. 최신 지진 정보를 얻으려면 지진 감지 기관이나 관련 뉴스 소스를 참고해 주세요.



Prompt Engineering (Prompt Optimization)



소목차


- Prompt 관련 용어
- Prompting and Completion
- zero shot prompting(inference)
- 상황 내 학습(in-context learning)
- COT(Chain-of-thought) Prompting
- Good Prompt 작성 방법

Prompt 관련 용어

- Prompt - 모델에게 주는 가이드 (입력 텍스트)
 - Instruction (명령) - prompt의 핵심. 텍스트 요약, 번역, 분류 등
 - Context - instruction을 구체화 할 수 있는 추가 정보.
 - Input Data - AI 모델이 지시한 작업을 완료하기 위해 처리할 정보
 - Output Indicator - 암시적 or 명시적
- 추론 - 텍스트를 생성하는 행위
- Completion - 출력 텍스트
- Context window - 프롬프트에 사용할 수 있는 전체 텍스트 양 또는 메모리
- 프롬프트 엔지니어링 - 프롬프트를 개발하고 개선하는 작업
 - 나은 결과를 생성하도록 하는 모델이 수행하기를 원하는 작업의 예를 프롬프트 내에 포함
 - 상황 별 예제를 제공





Prompting and Completion

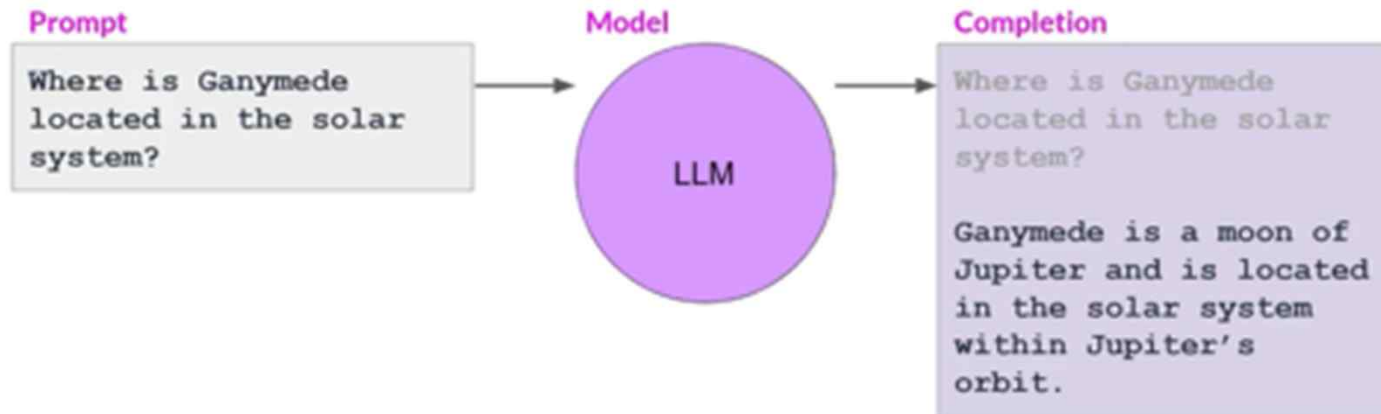
Prompt

 다음 텍스트를 **context** 중립, 긍정, 부정으로 **instruction** 분류하세요.

Input data 와우! 나 기분 좋아

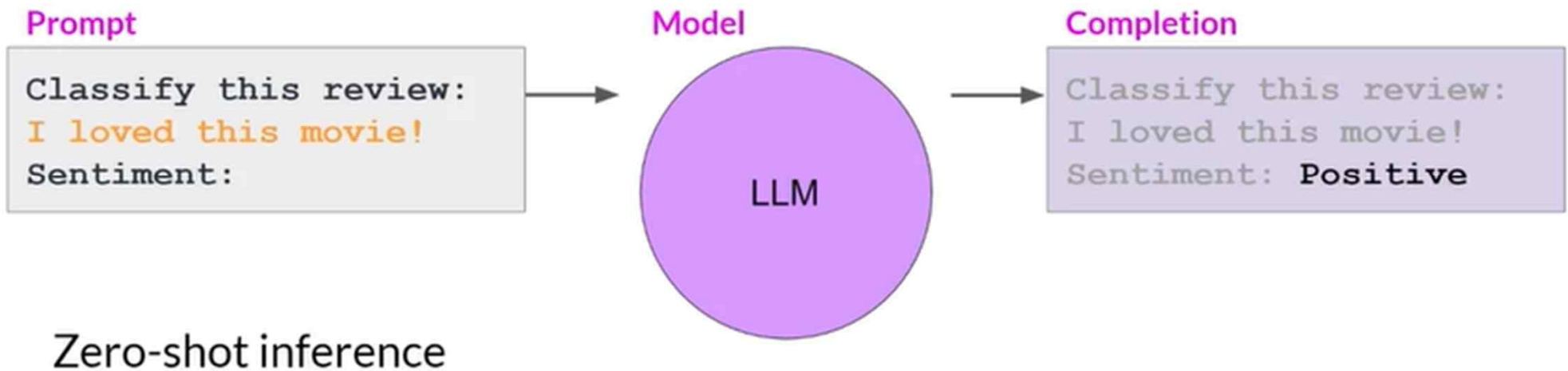
Completion

 텍스트 "와우! 나 기분 좋아"는 **긍정적**으로 분류됩니다. 이 텍스트는 확실히 긍정적인 감정을 나타내는 말투와 표현을 사용하고 있습니다.   



zero shot prompting(inference)

- 답을 나타내는 예시나 입력 데이터 없이 답변 생성
- 초보자의 사용 방식 (단순 질문 방식) – 모델의 기존 지식에 의존
- Guide가 없으므로 원하는 답이 아닐 가능성 높음





France의 수도는 어디인가요?

Zero-shot Prompting (Inference)



프랑스의 수도는 파리입니다. 파리는 프랑스의 가장 큰 도시이며, 아름다운 문화, 역사, 그리고 예술로 유명합니다.



죽기 전에 꼭 해야 할 일 3가지만 알려줘

Zero-shot Prompting (Inference)

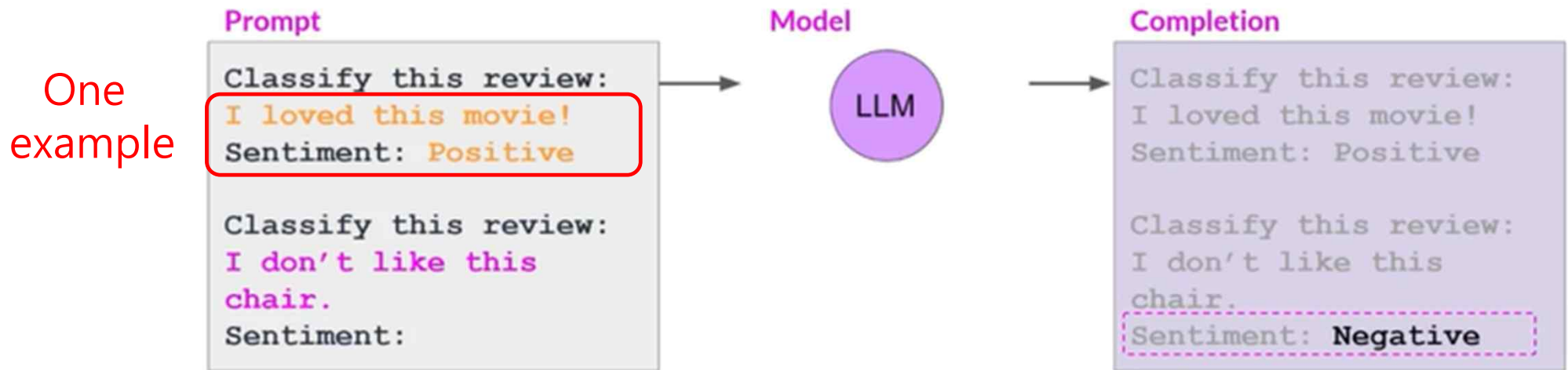


인생의 목표나 이루고자 하는 일들은 사람마다, 그리고 그 사람의 가치관이나 삶의 상황에 따라 다르겠지만, 아래는 많은 사람들이 인생에서 꼭 한번 해보고 싶어 하는 일들을 살펴본 것입니다:

1. **자기 자신을 위한 여행:** 자신만의 시간을 갖는 것은 매우 중요합니다. 혼자서 하는 여행은 자신을 더 잘 이해하고 세상을 다른 시각으로 바라보게 해줍니다. 일상에서 벗어나 새로운 경험을 하는 것은 자기 자신을 재발견하는 데 도움이 됩니다.
2. **자신이 정말로 사랑하는 일에 투자하기:** 여러 이유로 우리는 종종 자신의 꿈을 미루게 됩니다. 죽기 전에 꼭 해야 할 일 중 하나는 자신이 정말로 원하는 것을 쫓아가는 것입니다. 이것은 창조적인 활동일 수도 있고, 별도의 사업을 시작하는 것일 수도 있습니다.
3. **자신의 이야기를 기록하기:** 여러분의 삶은 단 한 번뿐이며, 그 경험과 감정, 깨달음은 모두 독특합니다. 죽기 전에 자신의 이야기를 기록해두면, 가족이나 친구들에게 귀중한 선물이 될 수 있습니다. 이것은 일기, 회고록, 또는 간단한 글이 될 수 있으며, 그 과정에서 자신의 삶을 다시 한번 되돌아볼 수 있는 기회가 될 것입니다.

상황 내 학습(in-context learning) – one shot inference

- 모델에게 특정 작업이나 개념에 대해, 자신이 원하는 한가지 예(example, shot)를 instruction 과 함께 제시
- 모델은 제공된 예와 유사한 새로운 데이터를 생성 혹은 분류





영화평을 다음과 같이 긍정적 또는 부정적으로 감성 분석하세요.

영화평: 이 영화 정말 짱이야!

감성: 긍정적

영화평: 보다가 지루해서 죽을뻔 했어!

감성:

In-Context Learning (One shot Inference)
One example



영화평: 보다가 지루해서 죽을 뻔 했어!

감성: 부정적



This movie was great!는 긍정적인 리뷰입니다.

그렇다면 'I really enjoyed this film!'은 어떤 리뷰인가요?"

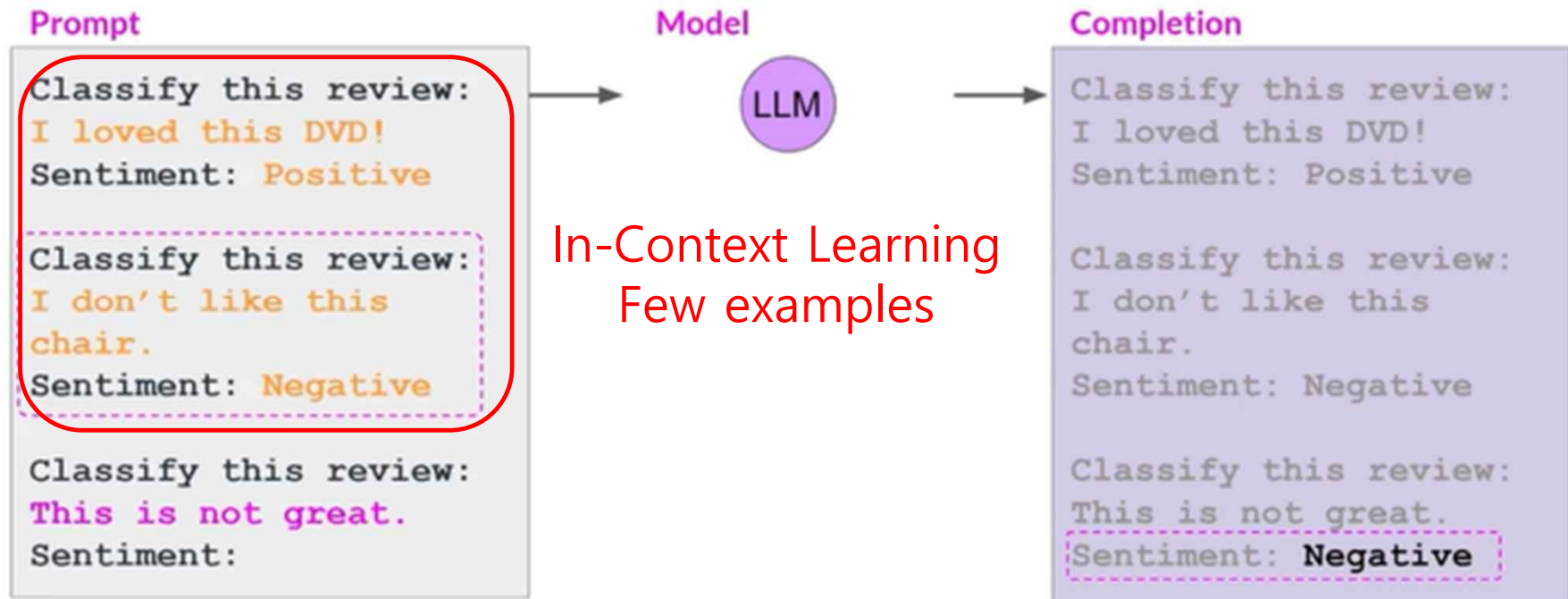
In-Context Learning
One example



'I really enjoyed this film!'는 긍정적인 리뷰입니다. "really enjoyed"라는 표현은 영화에 대한 긍정적인 반응을 나타내기 때문입니다.

상황 내 학습(in-context learning) – few shot inference (prompting)

- 모델에게 특정 작업이나 개념에 대해, 자신이 원하는 몇가지 예(example, shot)를 instruction 과 함께 제시
- 모델은 제공된 예와 유사한 새로운 데이터를 생성 혹은 분류





'This movie was great!'와 'I loved the play'는 긍정적인 리뷰입니다.
반면, 'I did not like the film'와 'The book was boring'는 부정적인 리뷰입니다.
그렇다면 'The concert was amazing!'은 어떤 리뷰인가요?"

In-Context Learning
Few examples



'The concert was amazing!'는 긍정적인 리뷰입니다. "amazing"이라는 단어는 강력한 긍정적
인 감정을 표현하므로 이 리뷰는 콘서트에 대한 매우 긍정적인 반응을 나타냅니다.



질문: 스페인의 수도는 어디인가요?
답: 마드리드
질문: 이탈리아의 수도는 어디인가요?
답: 로마
질문: 프랑스의 수도는 어디인가요?
답:

In-Context Learning
Few examples



파리

COT(Chain-of-thought) Prompting

- LLM의 추론 능력 개선을 위해 Google 연구원들이 고안
- 문제를 단계별로 분해하여 복잡한 추론 문제 해결
(<https://arxiv.org/pdf/2201.11903v1.pdf>)
- Chain-of-thought prompt engineering은 다음과 같은 단계로 수행됩니다.
 1. 문제를 여러 단계로 나눕니다.
 2. 각 단계에 대해 LLM에게 질문하거나 수행할 작업을 지시하는 프롬프트를 작성합니다.
 3. LLM이 출력한 결과를 다음 단계의 프롬프트에 입력합니다.
 4. 이 과정을 반복하여 문제를 해결합니다.

Zero-shot COT(Chain-of-thought)



나는 시장에 가서 10개의 사과를 샀다. 이웃에게 사과 2개를 주었고 수리공에게 2개를 주었다.
나는 다시 가서 5개를 더 사서 1개를 먹었다. 그러면 지금 나는 몇개의 사과를 가지고 있는가?
단계별로 생각해.



1. 당신은 처음에 시장에서 10개의 사과를 샀습니다.
 - 현재 사과 갯수: 10개
2. 그 후 이웃에게 2개의 사과를 주었습니다.
 - 현재 사과 갯수: $10\text{개} - 2\text{개} = 8\text{개}$
3. 수리공에게도 2개를 주었습니다.
 - 현재 사과 갯수: $8\text{개} - 2\text{개} = 6\text{개}$
4. 당신은 다시 시장에 가서 5개의 사과를 더 샀습니다.
 - 현재 사과 갯수: $6\text{개} + 5\text{개} = 11\text{개}$
5. 마지막으로 1개의 사과를 먹었습니다.
 - 현재 사과 갯수: $11\text{개} - 1\text{개} = 10\text{개}$

따라서, 당신은 현재 10개의 사과를 가지고 있습니다.

One shot example

- 모델의 추론 과정을 파악할 수 있음.
- 사전 지식 없이 모델에 창의적인 자유를 부여

Few shot COT(Chain-of-thought)

- 모델에게 우리가 원하는 답을 제공하고 어떤 생각의 사슬을 거쳐 단계를 해결하고 답을 얻게 되었는지 설명.



이 집단 [4, 8, 9, 15, 12, 2, 1]의 홀수를 모두 더하면 짝수가 됩니다.
답: 모든 홀수 (9, 15, 1)을 더하면 25입니다. 틀렸습니다.

이 집단 [17, 10, 19, 4, 8, 12, 24]의 홀수를 모두 더하면 짝수가 됩니다.
답: 모든 홀수 (17, 19)을 더하면 36입니다. 정답입니다.

이 집단 [16, 11, 14, 4, 8, 13, 24]의 홀수를 모두 더하면 짝수가 됩니다.
답: 모든 홀수 (11, 13)을 더하면 24입니다. 정답입니다.

이 집단 [17, 9, 10, 12, 13, 4, 2]의 홀수를 모두 더하면 짝수가 됩니다.
답: 모든 홀수 (17, 9, 13)을 더하면 39입니다. 틀렸습니다.

이 집단 [15, 32, 5, 13, 82, 7, 1]의 홀수를 모두 더하면 짝수가 됩니다.
답:



모든 홀수 (15, 5, 13, 7, 1)을 더하면 41입니다. 틀렸습니다.

Few shot examples

- 매우 유사한 문제의 실제 답을 첨부하고, 어떻게 이 답을 얻게 되었는지 설명
- AI 모델은 우리가 제공한 생각의 사슬로 다른 문제도 추론
→ 비슷하지만 다른 문제를 우리가 원했던 방식으로 해결

Few shot COT(Chain-of-thought) 비교

• 최신의 생성형 LLM 비교



영희는 10마리의 개를 키운다. 개 한마리를 산책시키고 돌보는데 0.5 시간이 필요하다. 갑자기 여행가는 친구가 개 두마리를 맡기고 떠났다. 그런데 이 개들 중 2마리가 각각 한마리씩 새끼를 낳았다. 새끼 한마리는 돌보는데 1시간이 필요하다. 이제 영희는 모든 개들을 돌보는데 일주일에 몇시간이 필요한가?

GPT-4: 정답



영희가 키우던 개는 10마리이고, 친구의 개 2마리를 추가로 맡아 총 12마리의 개를 돌보고 있습니다. 이 개들 중 2마리가 각각 새끼를 한 마리씩 낳아, 추가로 새끼 개 2마리가 더해져 총 14마리의 개를 돌보아야 합니다.

1. 영희가 원래 키우던 개와 친구의 개를 돌보는데 필요한 시간: $12\text{마리} \times 0.5\text{시간/마리} = 6\text{시간}$
2. 새끼 개를 돌보는데 필요한 시간: $2\text{마리} \times 1\text{시간/마리} = 2\text{시간}$

따라서 영희는 하루에 모든 개를 돌보는데 총 8시간이 필요합니다.

일주일에는 필요한 시간은: $8\text{시간/일} \times 7\text{일/주} = 56\text{시간/주}$ 입니다.

* Bard, CLOVAX
오답

COT형식의 답변

Good Prompt 작성 방법

- Context – LLM 에게 context 작성을 맡기지 말고 우리가 제공
- Clear, non ambiguous task – LLM이 달성해야 할 구체적인 목표, 목적 설정
- Iteration – 이전 반복의 출력에 의존하여 prompt 를 개선하여 반복



고위 DevOps 엔지니어 직책을 채용하기 위한 기술 인터뷰 질문 목록을 생성해줘

일반적인 prompting



고위 DevOps 엔지니어 직책을 채용하기 위한 인터뷰에서는, 캔디데이트가 복잡한 인프라와 시스템을 관리하고, 팀을 이끄는 능력, 그리고 문제를 해결하는 능력을 보유하고 있는지를 확인하는 것이 중요합니다. 다음은 적절한 기술 인터뷰 질문들입니다:

1. CI/CD 파이프라인을 설명하고, 어떻게 설계하고 구현하는지 예를 들어 설명해주세요.
2. 소스 코드에서 운영 환경까지의 워크플로우를 어떻게 자동화하나요?



조직 문화가 빠르게 변화하고 클라우드 기술로 개발 중인 기술 스타트업의 고위 DevOps 엔지니어 직책을 채용하기 위한 기술 인터뷰 질문 목록을 생성해줘

Better prompting
- 구체적인 목표,
목적 설정



DevOps 엔지니어 직책을 채용하기 위한 기술 인터뷰 질문은 다음과 같습니다:

1. 클라우드 기반 서비스를 활용한 이전의 프로젝트를 소개해주세요. 그 프로젝트에서 당신의 역할은 무엇이었나요?
2. AWS, Google Cloud, Azure와 같은 클라우드 서비스를 사용한 경험이 어떠한가요? 그 중 어떤 서비스를 가장 선호하나요?
3. 새로운 서버를 빠르게 배포하고, 효과적으로 모니터링하고, 이상을 발견하면 신속하게 대응하는 과정을 설명해 주시겠어요?



우리 e-commerce website 의 사용자 경험을 개선해줘

일반적인 prompting



사용자 경험(UX)을 개선하는 것은 웹사이트의 전반적인 성능을 향상시키는 데 중요한 요소입니다. 다음은 e-commerce 웹사이트의 사용자 경험을 개선하기 위한 몇 가지 제안입니다:

1. **속도 최적화:** 웹사이트의 로딩 속도는 사용자 경험에 매우 중요한 역할을 합니다. 사용자들은 웹사이트가 빠르게 로딩되기를 원하며, 그렇지 않으면 다른 사이트로 이동할 가능성이 높습니다. 따라서 이미지 최적화, 코드 최적화, CDN 사용 등을 통해 웹사이트의 속도를 최적화하시는 것을 추천드립니다.
2. **사용자 인터페이스(UI) 개선:** 사용자가 웹사이트를 쉽게 탐색할 수 있도록 직관적



고객 만족도 및 판매 전환율을 높이기 위해 우리 e-commerce website 의 사용자 경험을 개선할 수 있는 특별한 방법을 제안해 줘

Better prompting
- 구체적인 목표,
목적 설정



기본적인 사용자 경험 개선 방안 외에도, 전자 상거래 웹사이트를 위해 특별하게 고려해 볼 수 있는 몇 가지 전략들이 있습니다:

1. **증강현실(AR) 기능 도입:** 증강현실 기능을 사용하여 사용자들이 제품을 실제로 보고 느낄 수 있게 해주면 구매 결정을 돕습니다. 예를 들어, 가구를 판매하는 회사는 사용자가 자신의 집에서 어떻게 보일지 보여주기 위해 AR 기능을 사용할 수 있습니다.
2. **AI 챗봇 활용:** AI 챗봇은 고객 서비스를 개선하고, 고객이 문제를 빠르게 해결할 수 있도록 도와줄 수 있습니다. 또한, 사용자들에게 개인화된 쇼핑 경험을 제공하는데 활용될 수 있습니다.

Prompt Engineering(Optimization) 요약

- AI 모델로 부터 원하는 응답을 얻기 위해 어떻게 prompt 최적화
- 좋은 prompt 를 만드는 세가지 전략
 - Zero-shot prompt : 예시 없는 prompt
 - Few-shot prompt : 몇 개의 예시를 보여 주며 질문
 - COT (Chain-of-thought) : 논리적으로 순서대로 제시
 - Step-by-step 설명
 - 입력 길이가 길어지므로 시간/비용 trade-off 고려
- 같은 prompt로 여러 결과를 만들고 좋은 결과물을 선택
→ 같은 질문 여러 번 할 때의 시간/비용 trade-off

실습: Prompt Engineering

- zero shot prompting

다음의 문장을 영어로 번역하세요: '안녕하세요, 잘 지내세요?'

우리 태양계에서 가장 큰 행성은 무엇인가요?

- one shot prompting

다음은 빨간 사과에 대한 설명입니다. "빨간 사과는 맛있고, 영양분이 풍부하다."
이와 유사한 방식으로 파란 바나나에 대해 설명하세요.

다음은 김영희에 대한 자기소개입니다: "안녕하세요. 저는 김영희입니다. 서울에서 프로그래머로 일하고 있습니다."
이와 유사한 형식으로 박철수에 대한 자기소개를 만들어주세요.

• Few shot prompting

영화 제목을 짧게 요약하세요.

영화 '타이타닉': 큰 배가 침몰하는 이야기.

영화 '어벤저스': 슈퍼히어로들이 세계를 구하는 이야기.

영화 '기생충'에 대해 짧게 요약해주세요.

다음은 몇몇 동물들의 특징을 설명한 것입니다.

고양이: 독립적이며, 발바닥에 무릎이 있다.

강아지: 충성스럽고, 꼬리를 흔들 때 행복하다.

토끼에 대한 특징을 설명해주세요.

- Few shot COT(chain-of-thought)

첫 번째 생각: 한식의 독특함과 매력에 대해 어떻게 생각하세요?

(모델의 답변)

한식 중에서도 특히 독특하고 인기 있는 메뉴는 무엇이 있을까요?

(모델의 답변)

그 중 세계화가 가능한 음식은 어떤 것이 있을까요 ?

(모델의 답변)

- 위 예제들을 변형해 가며 ChatGPT, Bard, CLOVA X 에서 비교