

AI와 제대로 대화하는 법



“ AI와 대화할 때 원하는 답변을 얻지 못해 엉뚱한 답변으로 답답했던 경험이 발생합니다. "곰에 대해 알려달라"고 했는데 시카고 베어스 야구팀 이야기가 나왔다거나, 간단하게 질문했지만 횡설수설 하기도 합니다..

이런 문제의 해답이 바로 프롬프트 명령 기술이기도 합니다. 이는 복잡한 기술이 아닙니다. 단지 AI와 명확하게 소통하는 방법을 아는 것입니다.

이 내용은 'Google의 69페이지 백서'를 바탕으로 한 종합 가이드를 참고하여 작성 했습니다. -이기상-

1. 직접 질문하기 (Zero-Shot)

가장 간단한 방법입니다. 추가 설명 없이 바로 원하는 것을 요청하세요.

예시:

ai 에게 영화를 보고 쓴 다음 리뷰를 '좋았다' , '보통이다, '별로였다'로 평가해 줘라.

리뷰: "정말 재미있게 봤습니다!"

답변: 좋았다

답변처럼 ai는 빠르고 효율적으로 답변하지만, "멋지지만 지루했다"같은 복잡한 표현에서는 평가가 부정확할 수 있다는 것을 알고 질문해야 한다.

2. 예시로 가르치기 (Few-Shot)

몇 가지 예시를 보여준 다음 같은 방식으로 처리하도록 요청하는 방법입니다. 데이터를 일정한 형태로 가공할 때 특히 유용합니다.

다음 형식을 따라 문장을 만들어라.

예시:

입력: 사과

출력: 사과는 달콤하고 비타민이 풍부한 과일입니다.

입력: 바나나

출력: 바나나는 부드러운 식감과 단맛이 특징이고 칼륨 함량이 높은 과일입니다.

→ 모델은 바나나에 대한 문장을 예시 형식대로 작성하게 됩니다.

3. 역할 부여하기 (Role Prompt)

AI에게 특정한 역할을 맡겨 답변의 톤과 관점을 조정하는 방법입니다.

예시:

당신은 여행 가이드입니다. 로마의 필수 관광지 3곳을 친근하게 추천해주세요.

답변:

- 콜로세움: 거대한 고대 경기장, 꼭 봐야 할 웅장함
- 루브르 박물관: 모나리자 미소를 직접 확인하세요
- 몽마르트: 맛있는 크레페와 함께 산책하기 좋은 골목들

코치, 전문가, 친구 등 상황에 맞는 다양한 역할 설정이 가능합니다.

4. 단계별 사고 유도하기 (Chain of Thought)

시가 사고 과정을 단계별로 보여주도록 하는 방법입니다. 수학 문제나 복잡한 추론에서 실수를 줄일 수 있습니다.

예시:

단계별로 풀어보세요: 내가 5살일 때 누나는 나보다 2배 나이가 많았습니다.

내가 지금 20살이면 누나는 몇 살인가요?

답변:

1단계: 내가 5살일 때 누나 나이 = $5 \times 2 = 10$ 살

2단계: 누나와 나의 나이 차이 = $10 - 5 = 5$ 살

3단계: 내가 20살이면 누나 = $20 + 5 = 25$ 살

정답: 누나는 25살입니다.

답변이 다소 길어지지만, 복잡한 문제에서 정확도가 크게 향상됩니다.

흔한 문제들과 해결 방법

모호한 질문의 함정

문제: "별에 대해 말해달라" → 할리우드 스타 정보가 나옴 해결: "우주의 별에 대한 천문학적 정보를 알려달라"

잘못된 정보 생성

문제: "고양이가 날 수 있다"는 식의 거짓 정보 해결: "사실에 근거한 정보만 제공해달라" 또는 "정확한 리스트로 정리해달라"

반복적인 답변

문제: "네 네 네"처럼 같은 말 반복 해결: 단어 수 제한을 설정하거나 더 구체적인 지시 추가

효과적인 프롬프트 작성 원칙

명확성이 핵심입니다. "파리의 카페 3곳"이 "프랑스의 재미있는 장소들"보다 훨씬 좋은 결과를 만들어냅니다.

예시를 적극 활용하세요. AI에게 지도 역할을 해주는 가장 효과적인 방법입니다.

구체적으로 지시하세요. "이런 형식으로 답해달라"고 명시하면 원하는 결과를 얻기 쉽습니다.

긍정적으로 표현하세요. "소설을 쓰지 마라" 대신 "짧은 이야기를 써달라"가 더 효과적입니다.

성공 패턴을 기록하세요. 잘 작동한 프롬프트는 따로 정리해두면 나중에 유용합니다.

프롬프트 명령 기술의 한계

이런 기법이나 기술은 종류에 따라 시기에 따라 달라질 수 있어 완벽한 기술은 아닙니다. AI는 여전히 실수할 수 있고, 특히 최신 정보나 매우 전문적인 분야에서는 제한이 있습니다. 중요한 결정에 사용할 정보는 반드시 추가 검증이 필요합니다.

또한 프롬프트 엔지니어링은 AI 모델의 성능 한계를 극복하는 마법이 아닙니다. 모델 자체의 능력을 최대한 끌어내는 도구일 뿐입니다.

🔄Revision #3

★Created 23 August 2025 07:29:29 by pajuwiki

✎Updated 23 August 2025 07:48:23 by pajuwiki