생각의 사슬 (Chain of Thought)은 왜 동작하는가?

이동재

20240725

Abstract

생각의 사슬은 언어 모델이 복잡한 추론 문제를 해결할 때, 문제를 부분 문제들로 쪼개어 해결하는 프롬프트 엔지니어링 기법이다. 단순하면서도 효과적이어서 가장 널리 사용되는 방법 중 하나이다. 그러나, 여전히 생각의 사슬이 왜 동작하는지에 대한 명확한 이론적 근거는 부족하다. 본 글에서는 생각의 사슬이 동작하는 이유에 대해 두 가지 가설을 제시한다.

분할 정복 기법은 복잡한 문제를 작은 부분 문제들로 나누어 해결하는 방법이다. 문제의 종류에 관계없이 복잡하고 어려운 문제를 해결하는데 널리 사용되는 사고방식이다. 생각의 사슬은 이러한 분할 정복 기법을 언어 모델에 적용한 것이다.

생각의 사슬은 왜 언어 모델의 성능을 올려줄까? 나는 이 질문에 두 가지 관점에서 답하고자 한다. 첫 번째는 '생각의 용량'이다. 큰 수의 암산을 하다 보면 앞서 수행한 연산 결과를 까먹어 결국 암산에 실패하는 경험을 해보았을 것이다. 인간의 단기 기억은 빠르게 휘발되고 새로운 데이터로 덮어 쓰인다. 이 때문에 많은 데이터를 저장하고 유지할 수 없다.

언어 모델의 잠재 상태 (Hidden State) 역시 이러한 속성을 지닌다. 잠재 상태는 추론의 중간 과정을 저장하는 역할을 한다. 잠재 상태는 유한한 크기의 벡터이므로 저장할 수 있는 용량에 한계가 있다. 생각의 사슬은 추론의 중간 과정을 명시적으로 출력으로 내뱉도록 강제하며, 이는 잠재 상태의 용량을 초과하는 정보를 외부로 내보내도록 하는 효과를 가져온다.

두 번째는 지식의 의존 관계이다. 인간이 작성한 문서에는 일정한 형식이 있으며, 이는 곧 언어 모델의 학습한 지식 간에 의존 관계를 만든다. 예를 들어, 개인 정보가 담긴 서류를 학습한다고 가정해보자. 대부분은 개인의 이름이 다른 정보보다 선행할 것이다. 따라서 다른 정보와 이름 사이에는 의존 관계가 생긴다. 이러한 경우 어떤 사람의 정보를 물을 때 그 사람의 이름을 먼저 물어보고, 그 후 그 사람의 정보에 대해 물어보는 것이 더욱 효과적일 것이다.

특히 자가회귀모델은 'A는 B'를 학습했을 때 'B는 A'를 추론하는 것을 어려워한다. 따라서 생각의 사슬은 이러한 의존 관계를 명시적으로 표현하도록 강제함으로써 언어 모델의 성능을 향상하게 시킬 수 있다.