🟣 한 번에 요구 사항을 프롬프트에 넣어야 하는 까닭

20만 건의 대화가 증명한 것: AI 는 대화를 못 한다

LLM을 "대화형 도구"로 사용하지만, 정작 LLM은 대화에 최적화되어 있지 않습니다.

Microsoft Research와 Salesforce의 공동 연구는 "LLMs Get Lost In Multi-Turn Conversation" 에서 LLM의 치명적 약점을 정량적으로 증명했습니다.

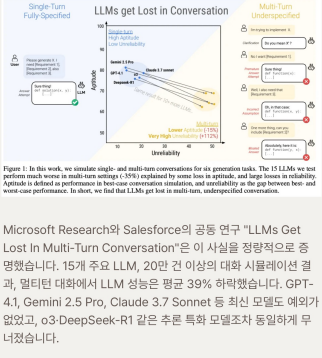

Figure 1: In this work, we simulate single- and multi-turn conversations for six generation tasks. The 15 LLMs we test perform much worse in multi-turn settings (-39%) explained by some loss in aptitude, and large losses in reliability. Aptitude is defined as performance in best-case conversation simulation, and unreliability as the difference between best- and worst-case performance. In short, we find that LLMs get lost in multi-turn, underspecified conversation.

Microsoft Research와 Salesforce의 공동 연구 "LLMs Get Lost In Multi-Turn Conversation"은 이 사실을 정량적으로 증명했습니다. 15개 주요 LLM, 20만 건 이상의 대화 시뮬레이션 결과, 멀티턴 대화에서 LLM 성능은 평균 39% 하락했습니다. GPT-4.1, Gemini 2.5 Pro, Claude 3.7 Sonnet 등 최신 모델도 예외가 없었고, o3·DeepSeek-R1 같은 추론 특화 모델조차 동일하게 무너졌습니다.

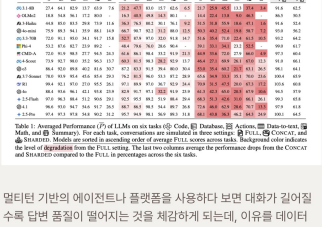

Table 1: Averaged Performance (P) of LLMs on six tasks (Code, Database, Actions, Data-to-text, Math, and Summary). For each task, conversations are simulated in three settings: FULL, CONCAT, and SHARDED. Models are sorted in ascending order of average FULL scores across tasks. Background color indicates the level of degradation from the FULL setting. The last two columns average the performance drops from the CONCAT and SHARDED compared to the FULL in percentages across the six tasks.

멀티턴 기반의 에이전트나 플랫폼을 사용하다 보면 대화가 길어질수록 답변 품질이 떨어지는 것을 체감하게 되는데, 이유를 데이터 기반으로 설명할 수 있습니다.

✅ 멀티턴 대화에서 품질이 떨어지는 4가지 원인

1. **정보가 부족한 초반에 성급하게 답변 시도**
2. **이전의 틀린 답변에 과도하게 의존**
3. **중간 턴의 정보를 잊어 버림**
4. **지나치게 장황한 응답으로 잘못된 가정 삽입**

✅ Lost in Conversation 현상

대화 초반, 정보가 부족한 상태에서 성급하게 답변을 시도하기 때문입니다. 초기에 세운 가정이 틀리면 이후 턴에서 그 틀린 답변에 오히려 더 의존하는 악순환이 생깁니다. Chain of Thought 같은 단일 추론 기법은 물론, Tree of Thought·Graph of Thought 같은 복합 추론 기법에서도 마찬가지입니다. 한 번 잘못 추론하면 이후 답변도 연쇄적으로 틀려버립니다.

☑️ 고품질 답변을 위한 프롬프트 엔지니어링 방법

1️⃣ 한 턴에 모든 요구사항을 전달하기

여러 턴에 걸쳐 조건을 추가할수록 답변 품질은 급격히 저하됩니다. 한 번에 많은 맥락을 프롬프트에 담아내려면 구조화는 필수입니다.

2️⃣ 대화가 꼬이면 이어가지 말고 새로 시작하기

"지금까지 말한 내용을 정리해줘"라고 요청한 뒤, 그 결과를 복사해 새 대화에 붙여넣기만 해도 품질이 눈에 띄게 회복됩니다.

3️⃣ 긴 대화 시 매 턴마다 요약 쌓기

매 턴마다 이전 정보를 반복 포함시키는 Snowball 방식은 답변 꼬임을 15~20% 회복시키는 효과를 보였습니다.

4️⃣ Temperature 를 낮춰도 근본적인 해결책은 아님

Temperature를 0으로 설정하면 일관성과 신뢰성은 높아지지만, 멀티턴 환경에서는 0으로 설정해도 여전히 높은 불안정성이 관찰됩니다.

LLM의 벤치마크의 높은 점수와 실제 사용 경험 사이의 간극, 이 논문이 그 간극을 숫자로 보여줍니다. 프롬프트 엔지니어링은 모델의 한계를 이해는 데서 시작합니다.

#LLM #AI #conversation#promptengineering #AIResearch #GenerativeAI #prompting

LLMs Get Lost In Multi-Turn Conversation

Large Language Models (LLMs) are conversational inter faces. As such, LLMs have the potential to assist their...

arxiv.org