Llama2 LLM과 prompting을 통한 Financial QA 풀이

이나경[1], 기경서[2], 권가진[3]
[1]서울대학교 지역시스템공학과 학부생
[2, 3]서울대학교 융합과학기술대학원 지능정보융합전공
lnktop@snu.ac.kr, kskee88@snu.ac.kr, ggweon@snu.ac.kr

Application of Llama2 LLM and prompting in Financial QA

NaKyung Lee[1], Kyung Seo Ki[2], Gahgene Gweon[3]
[1]Dept. of Rural System Engineering, Seoul National University
[2, 3]Dept. of Intelligence and Information, Seoul National University

요 약

본 논문에서는 RLHF 기반의 오픈소스 LLM인 llama-2-13b model을 FinQA task에 적용하여 그 성능을 확인해 보았다. 이때, CoT, few-shot과 같은 다양한 prompting 기법들을 적용해보며 어떤 방법이 가장 효과적인지 비교했다. 그 결과, 한 번(total)에 task를 수행한 경우 few-shot 예시를 2개 사용했을 때보다 3개 사용했을 때, subtask로 나누어 수행한 경우 prompt로 답(simple)만 제시했을 때보다 CoT 형식으로 주었을 때, 각각 24.85%의 정확도로 가장 높은 성능을 보였다.

1. 서론

최근 사람의 피드백을 강화학습의 reward 함수로 활용하는 RLHF (Reinforcement Learning with Human Feedback) 기법을 적용한 거대 언어 모델(Large Language Model)이 다양한 분야의 NLP task에서 신기록을 경신하고 있다[1]. 대표적으로 활용되고 있는 LLM으로 ChatGPT와 오픈소스 기반의 LLAMA-2[1]가 있다. 이에 따라, LLM이 어떤 task에 적절한지 다양한 시도들이 제시되고 있다.

LLM이 아직 시도되지 않은 유용한 task들 중 하나로 Financial QA(FinQA)가 있다. FinQA는 금융 분야의 전문적인 지식과 수학적 추론 능력을 요구는 task이다[2]. 하지만 지금까지의 FinQA 연구는 주어진 데이터로부터 수식을 세우는 데 필요한 정보를 파악하여 수식을 생성하는 retriever-generator 방식인 2-way 구조가 주로 사용되어왔다[2]. 또한, 지금까지 sota를 갱신한 다른 연구들 역시 2-way 방식으로 주로 모델을 개발해 왔다[3]. 그러나 LLM을 통해 해당 task를 해결하고자 하는 본격적인 시도는 아직 이루어지지 않은 상태다.

따라서, 본 논문에서는 RLHF 기반의 오픈소스 LLM인 Llama-2-chat model을 FinQA task에 적용하여 그 성능을 확인해 보고자 한다. 또한, LLM의 적용 과정에서 CoT(Chain of Thought), few-shot과 같은 다양한 prompting 기법들을 활용해 어떤 방법이 가장 효과적인지 비교해 볼 것이다.

2. 배경지식: Financial QA task와 Prompting 소개

이 논문의 task로 사용되는 Financial QA는 기업 공시정보 보고서로부터 적절한 숫자를 추출하여 주어진 질의를 풀이하기 위한 수식을 세우는 task이다[2]. 하지만 최근 들어 더 전통적이고 FinQA의 목적과 유사한 task인 MathQA는 LLM을 활용한 연구들이 많이 보고되고 있는 반면[1], FinQA는 LLM을 사용한 방법들이 보고된 바 없다. 따라서, MathQA의 분야를 확장하여 LLM을 응용해 풀이했던 다양한 방법들을 FinQA에도 적용해볼 수 있을 것이다.

본 논문에 적용된 LLM과 prompting에 대한 소개는 다음과 같다. 실험에 사용한 LLAMA-2 모델은 최근 LLM에서 제기되어왔던 unintended action이나 safety 관련 문제들을 개선할 수 있도록 학습되었다[1]. LLAMA-2는 chat 세팅의 별도로 튜닝된 모델을 제공하는데, 이 모델에는 각 user, assistant, system role마다 special token이 존재해 prompt engineering이 용이하다[1]. 따라서, system과 user를 통해 의도한 지시 사항을 prompt로 주입하면 원하는 방향으로 답변 생성이 가능하기 때문에 다양한 downstream task가 수행 가능할 것이다[4]. 이때, 최근 성능 개선에 기여하고 있는 prompting 기법들

인 CoT(Chain of Thought)[5], PoT(Program of Thought)[4] 등의 방법을 prompt로 사용함으로써 prompting의 성능 개선 또한 모색할 수 있다.

3. 실험 설계

다양한 prompting 방법에 따라 모델의 task 수행 성능이 얼마나 달라지는지 비교하기 위한 실험을 수행하였다. 실험 데이터로 FinQA 논문에서 발표한 1147개의 test data를 사용했다[2]. 이때, prompt로 자연어 질의인 question, 풀이에 필요한 금융 정보인 gold_inds, 연산 과정을 나타낸 program, 그리고 계산 결과 answer 카테고리를[2] 추출하여 사용했다.

실험에 사용한 모델은 llama-2-13b-chat으로 총 5번의 실험을 수행했다. 모든 실험에서 system에는 task에 대한 instruction과 일정한 answer format을, user와 assistant에는 질문, 금융 정보-모범답안 pair의 few-shot 예시를 prompt로 제시했다. <표 1>에 표시된 task 풀이 구조(total/subtask) 및 prompt 형식(simple/CoT)은 5개의 실험을 구분하는 기준이다. Total 풀이 구조는 한 번에 FinQA 문제를 풀이하는 세팅이고, subtask 구조는 FinQA task를 다시 1. 필요한 수 반환하기, 2. 수식 세우기, 3. 계산하기의 세부 task로 나누어 순차적으로 문제를 풀이하는 방법이다. Simple prompting에서는 단순히 수식 과정, program만을 assistant에게 모범답안으로 주었고, CoT는 program이 도출된 logic을 제시함으로써 reasoning step까지 제공했다. 추가로, Sympy는 문자열로 제시된 수식을 실제로 계산해주는 python library로 계산 과정을 code로 대체하여 정확도를 높이기 위해 실험 4, 5에 적용했다. 이때, Total은 1-3번 과정을 한 번에 진행하기 때문에 수식 세우기에서 CoT와 계산단계에서 Sympy를 적용하지 않았다.

4. 결과 및 토의

<표 1>은 각 실험 별로 FinQA의 accuracy를 비교한 결과이다. Total 구조의 실험2와 subtask 구조에 CoT를 적용한 실험5에서 24.85%로 가장 높은 정답률을 보였다. 또한, prompt의 token 개수를 간소화하기 위해 subtask 구조를 취하면 정답률이 높아질 것이라고 예상한 것과 달리 실험3, 4에서 정답률이 가장 낮았다. 이는 앞 task에서 반환된 결과를 다음 task에 사용하기 때문에 이전 task의 오차가 점차 누적된 결과로 추정된다. 또한, 실험3의 수식 계산에서 오류가 많았는데, 원인을 찾던 중 llama tokenizer가 수의 자릿수를 인식하지 못하고 각 숫자를 개별 token으로 인식함을 알 수 있었다. 모델이 token을 처리하는 방식 자체의 한계로 인해 여러 방식의 정제된 연산 규칙을 prompt로 지시하더라도 성능이 유의미하게 증가하지 못한 것으로 추정된다.

<표 1> Prompting 방법에 따른 performance.

실험	Task 풀이구조	Prompt 형식	Few-shot	Sympy	Accuracy
1	total	simple	2	X	22.58 %
2	total	simple	3	X	**24.85** %
3	subtask	simple	3	X	17.26 %
4	subtask	simple	3	O	19.97 %
5	subtask	CoT	3	O	**24.85** %

5. 결론

Prompting을 적용한 LLM으로 FinQA task를 수행한 결과, 성능은 LLAMA-2에서 제시한 MathQA 성능 수준과 유사하게[1] 나타났다. 기존 FinQA 풀이 방법인 2-way[2] 모델에 비해 reasoning에 있어서 LLM은 아직 한계를 지닌 것으로 보인다. 따라서, subtask에 맞도록 fine-tuning을 하거나 더 적합한 prompt method를 찾는 후속 연구가 필요하다.

6. Acknowledgement

이 성과는 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2020R1C1C1010162).

참고문헌

[1] Hugo Touvron, et.al., “Llama 2: Open Foundation and Fine-Tuned Chat Models”, Meta, arXiv:2307.09288v2, 2023

[2] Zhiyu Chen, et.al., "FINQA: A Dataset of Numerical Reasoning over Financial Data", EMNLP, Dominican Republic, 2021, p3697 - 3711

[3] Jiaxin Zhang, et.al., “ELASTIC: Numerical Reasoning with Adaptive Symbolic Compiler”, NeurIPS, New Orleans, 2022

[4] Pengfei Liu, et.al., “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”, ACM Computing Surveys, V55, N9, p195-230, 2023

[5] Jason Wei, et.al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, NeurIPS, New Orleans, 2022