입시 컨설턴트를 위한 진학상담 cQA

고동혁[1], 전승재[1], 양승원[2], 박진영[3], 김태훈[1], 박천음[1]*

국립한밭대학교[1], 바이브온[2], 성균관대학교[3]

{kohdh, sjjeon}@edu.hanbat.ac.kr, swyang@vibeon.ai, jy.bak@skku.edu, {thkim, parkce}@hanbat.ac.kr

Community QA of College Counseling for Admissions Consultant

Donghyeok Koh[1], Seungjae Jeon[1], Seungwon Yang[2], Jinyeong Bak[3], Taehoon Kim[1], Cheoneum Park[1]*

Hanbat National Univ[1], VIBEON[2], Sungkyunkwan Univ[3]

요 약

본 논문은 한국어 입시 상담 cQA에서 LLM을 통한 컨설턴트 스타일의 답변을 생성하기 위해, cQA로부터 모델 학습에 필요한 중간 추론 단계를 구성 및 활용하는 방법을 제안한다. 제안 방법은 cQA 데이터로부터 multi-hop 스타일 학습 데이터를 생성하여 모델별 LoRA 튜닝을 수행하며 질의 분해 모델, RAG 기반 하위 질의응답, 추론 결과 기반 최종 응답 생성으로 파이프라이닝된 프레임워크에 사용한다. 실험 결과, 제안 방법이 BLEU-1 0.351, BLEU-2 0.191를 달성하여 각 모델별 gpt-4o-mini를 활용한 결과 대비 높은 성능을 보이며 BERTScore 0.836를 달성하여 gpt-4o-mini에 근사한 의미적 유사도를 보인다.

I. 서론

거대 언어 모델(Large Language Model, LLM)의 발전으로 LLM과 Community QA (cQA)를 활용한 다양한 연구가 진행되고 있다 [1, 2, 3]. 실제 cQA 환경에서 사용자 질문은 단일 질의로 처리하여 답변을 검색하거나 생성한다. 이때 사용자 질문은 다수의 조건과 의도를 동시에 포함하며 명시적으로 일관된 구조 없이 서술되는 경우가 많아, 답변 생성에 필요한 정보를 누락하거나 질문의 의도를 정확히 파악하지 못하는 어려움이 있다.

복합적인 질문을 하위 의미 단위로 분해하여 태스크를 수행하는 방법 [4, 5]은 multi-hop QA나 복잡한 태스크와 같이 추론 기반 문제를 해결하는데 유의미한 결과를 보인다. 실제 cQA는 다양한 조건, 배경 정보 등 복합적인 정보를 포함하며 하나의 답변을 요구하는 질문으로 구성된다. 이는 multi-hop QA와 유사한 구조를 가지며, 질의 분해를 통한 추론 기반 응답 생성 방법이 적합하다. 그러나 cQA 데이터는 주로 질문과 답변만 제공되기 때문에 올바른 답변을 생성하기 위한 추론 과정을 해석하지 못하는 어려움이 있다.

이에 본 논문에서는 대한민국 입시 상담 cQA 데이터를 기반으로 중간 추론 과정으로 사용될 하위 질문-답변 쌍을 LLM으로 생성하고, 이를 학습에 활용하는 데이터 생성 방법을 제안한다. 또한 질문을 분해하고 하위 질문에 답변하는 추론 과정을 통해 올바른 최종 응답을 생성하는 cQA 질의응답 프레임워크를 제안한다. 실험 결과, 제안 방법이 BLEU-1 0.351, BLEU-2 0.191를 달성하여 정밀도 측면에서 각 모델별 gpt-4o-mini를 활용한 결과 대비 높은 성능을 보이며 BERTScore 0.836을 달성하여 의미적 유사도 측면에서 gpt-4o-mini에 근사한 성능을 보인다. 또한 ablation 실험을 통하여 모델 학습에 따른 성능을 분석하며 모든 모델에 LoRA tunning을 수행한 경우 성능 개선을 보여 제안 방법의 유효성을 입증한다.

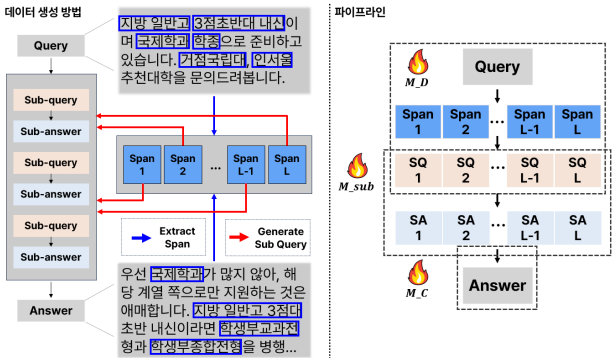

그림 1: cQA 기반 데이터 생성 방법 및 질의응답 파이프라인

II. 제안 방법

본 장에서는 한국어 입시 cQA에서 컨설턴트 스타일의 응답을 생성하기 위해 기존 cQA로부터 모델 학습 데이터를 생성하는 방법과 이를 활용한 학습 과정을 서술한다. 모델은 질문을 하위 질문로 분해하는 모델(M_D), 하위 질문에 답변을 생성하는 하위 답변 모델(M_{sub}), 최종 답변을 생성하는 모델(M_C)로 구성된다.

II.I. cQA 기반 데이터 생성 방법

[그림 1]의 데이터 생성 방법은 실제 cQA 데이터를 사용하며 M_D, M_{sub}, M_C 각 모델 학습에 필요한 데이터를 gpt-4o-mini를 통하여 생성한다. 우선 질문 Q와 답변 A에서 $span$을 추출하고 각 $span$을 기반으로 하위 질문(Q_{sub})를 생성한다. 이때 Q 기반 $span_Q$은 사용자가 묻는 정보들을 답변 생성에 필요한 하위 질문로 분해하는 과정을 학습하기 위하여 사용하며, A기반 $span_A$은

*Corresponding author

표 1: 생성 답변 성능 비교

Method	R-1	R-L	B-1	B-2	BS	Token
w/ gpt	0.206	0.184	0.301	0.162	0.837	398
Ours	0.158	0.145	0.351	0.191	0.836	309

표 2: Ablation 실험 결과

M_D & M_C	R-1	R-L	B-1	B-2	BS
Base & Base	0.092	0.085	0.297	0.144	0.829
LoRA & Base	0.119	0.111	0.278	0.135	0.827
Base & LoRA	0.114	0.107	0.347	0.191	0.838
LoRA & LoRA	0.158	0.145	0.351	0.191	0.836

컨설턴트가 답변을 생성할 때 참고한 주요 내용을 학습하기 위하여 사용한다. $span_Q$, $span_A$으로부터 생성한 Q_{sub}를 병합하여 최종 Q_{sub} 집합을 구성한다. 이후 각 Q_{sub}에 대해 하위 답변(A_{sub})을 생성해 중간 추론 단계를 구성한다. A_{sub}를 생성할 때, RAG기반 방법을 사용하며 검색에는 BM25와 SentenceTransformer를 사용한다. 하나의 (Q, A) 쌍으로부터 생성되는 학습 데이터는 모델별 목적에 따라 다음과 같이 생성된다.

- 1) M_D 학습 데이터: $Q, span, Q_{sub}$
- 2) M_{sub} 학습 데이터: Q_{sub}, 근거 문서, A_{sub}
- 3) M_C 학습 데이터: Q, Q_{sub}, A_{sub}, A

II.II. 모델 학습

모든 모델은 Llama-3.1-Korean-8B-Instruct을 백본으로 사용하여 LoRA tuning을 수행한다. M_D는 Q를 입력으로 받아 적절한 $span$을 추출하고 각 $span$ 대한 Q_{sub}을 생성하도록 학습하며, M_{sub}는 각 Q_{sub}와 검색된 문서를 입력으로 받아 A_{sub}를 생성하도록 학습한다. 마지막으로 M_C는 Q와 Q_{sub}, A_{sub}을 입력으로 받아, 컨설턴트 스타일의 최종 답변 A를 생성하도록 학습한다.

III. 실험 및 결과

본 장에서는 제안 방법(Ours)의 유효성을 검증하기 위해 대한민국 입시 상담 도메인의 cQA 데이터를 기반으로 학습 데이터를 구축하고 모델을 학습하여 성능을 평가한다. 실험은 제안 방법과 각 모델별 gpt-4o-mini를 사용한 비교 실험과 질의 분해 모델과 답변 생성 모델에 대한 ablation 실험을 진행한다.

III.I. 데이터셋 및 실험 설정

실험에는 수험생 질문에 입시 컨설턴트가 답변을 작성한 VIBEON cQA 데이터를 사용한다. 데이터는 총 831개로 구성되어 학습 데이터 구축에 700개의 데이터를 사용하고 성능 측정에 131개의 데이터를 사용한다. 데이터 생성 방법을 통하여 700개의 cQA 데이터를 기반으로 M_D와 M_C 학습 데이터 각각 1400개, M_{sub} 학습 데이터 12,297개를 생성하여 모델 학습에 사용한다.

실험에 사용된 모든 모델은 Llama-3.1-Korean-8B-Instruct을 백본으로 사용하며 LoRA tuning을 수행한다. Q_{sub}에 대한 답변을 생성하기 위한 검색에는 BM25와 SentenceTransformer를 통한 re-ranking 방법을 사용하며 re-ranking 모델은 jhgan/ko-sroberta-multitask를 사용한다. 최종 생성 결과에 대한 평가 지표로는 ROUGE (R), BLEU (B), BERTScore (BS)를 사용한다.

III.II. 실험 결과

[표 1]는 파이프라인의 모든 모델을 gpt-4o-mini를 사용한 결과(w/ gpt)와 제안 방법(Ours)의 성능 결과이다. 제안 방법은 ROUGE에서 R-1 0.158, R-L 0.145를 기록하여 w/ gpt 대비 낮은 성능을 보이지만 BLEU Score에서 B-1 0.351, B-2 0.191을 달성하여 w/ gpt 대비 높은 성능을 보인다. 생성된 답변의 평균 길이(Token)에서 w/ gpt는 398 토큰을 생성하여 제안 방법보다 긴 답변을 생성하는 것을 보인다. 따라서 참조 답변의 표현을 얼마나 포함하는지에 대한 재현율(Recall) 측면의 평가 지표인 ROUGE에서 긴 답변을 생성한 w/ gpt가 높은 성능을 기록한 것으로 보인다. 반면 BLEU는 정밀도(Precision) 측면의 평가 지표로 제안 방법이 짧은 길이에도 정답의 핵심 표현을 포함하여 높은 성능을 보인다. 이는 제안 방법의 답변 길이가 짧지만 원본 답변의 핵심 정보를 포함하는 답변을 생성하는 것을 시사한다. 의미적 유사도를 측정하는 BERTScore에서는 두 방법 모두 유사한 성능을 보이며, 제안 방법이 비교적 짧은 답변에도 의미 수준의 정합성을 유지함을 보인다.

[표 2]는 제안 방법의 질의 분해 모델 M_D와 답변 생성 모델 M_C의 학습 효과를 분석하기 위한 ablation 실험 결과이다. 먼저 학습을 진행하지 않은 모델을 사용한 경우(Base & Base)에서 가장 낮은 성능을 보이며, 전반적으로 학습이 안된 상태에서는 end-to-end 결과 생성이 어렵다는 것을 확인할 수 있다. M_D만 학습한 경우(LoRA & Base)는 Base & Base 대비 R-1, R-L이 개선되지만 B-1, B-2는 오히려 감소한다. 이는 질의를 적절히 분해해도 최종 답변을 생성하는 M_C가 학습되지 않으면 출력 표현의 정밀도를 확보하기 어렵다는 것을 의미한다. 반대로 M_C만 학습한 경우(Base & LoRA) 모든 평가 지표에서 Base & Base 대비 향상된 성능을 보인다. 이는 M_C의 학습이 문장 생성 능력 및 표현 정합성에 기여한 것으로 사료된다. 마지막으로 M_D, M_C 모두 학습한 경우(LoRA & LoRA)가 모든 지표에서 가장 높은 성능을 보이며 이는 M_D의 질의 분해와 M_C가 결합될 때, 의미적 유사도를 유지하며 답변의 핵심 정보를 더 포함하는 방향으로 성능이 개선됨을 의미한다.

IV. 결론

본 논문에서는 대한민국 입시 상담 cQA 데이터를 기반으로, 중간 추론 과정에 활용될 하위 질의-정답 쌍을 LLM으로 생성하는 학습 데이터 구축 방법과 cQA 답변 생성 프레임워크를 제안한다. 실험 결과, 제안 방법이 BLEU-1 0.351, BLEU-2 0.191을 기록하여 정밀도 측면에서 gpt-4o-mini 대비 높은 성능을 보이며 BERTScore 0.836을 기록하여 의미적 유사도 측면에서 gpt-4o-mini에 근사한 성능을 기록하였다. 추가로 ablation 실험을 통해 학습 유무에 따른 성능 변화를 분석하였으며, 모델별 LoRA tunning을 수행한 경우 가장 높은 성능을 달성하여 제안 방법에 유효성을 입증하였다.

참 고 문 헌

[1] Z. Li, J. Zhang, C. Yin, Y. Ouyang, and W. Rong, "Procqa: A large-scale community-based programming question answering dataset for code search," 2024.

[2] L. Zhong and Z. Wang, "Can chatgpt replace stackoverflow? a study on robustness and reliability of large language model code generation," 2024.

[3] Q. Chen, W. Tao, Z. Zhu, M. Xi, L. Guo, Y. Wang, W. Wang, and Y. Lan, "ComRAG: Retrieval-augmented generation with dynamic vector stores for real-time community question answering in industry," 2025.

[4] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," 2023.

[5] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. Chi, "Least-to-most prompting enables complex reasoning in large language models," 2023.