

# 사실관계 확인을 위한 다중 대규모 언어모델의 입출력 제어 구조화 방법

정지수<sup>o</sup>, 이치현<sup>o</sup>, 강현석<sup>o</sup>, 정상근<sup>o\*</sup>, 노윤형<sup>o</sup>  
 충남대학교 컴퓨터융합학부<sup>o</sup>, 한국전자통신연구원<sup>o</sup>

{jisu.jung5, dlclgus1961, dnfldjaak11, hugmanskj}@gmail.com, yhroh@etri.re.kr

## Structuring Input and Output Control of Multiple Large Language Models for Fact-Checking

Jeesu Jung<sup>o</sup>, Chihyeon Lee<sup>o</sup>, Hyeonseok Kang<sup>o</sup>, Sangkeun Jung<sup>o†</sup>, Yoonhyung Roh<sup>o</sup>  
 Chungnam National University, Department of Computer Science  
 Engineering<sup>o</sup>, Electronics and Telecommunications Research Institute<sup>o</sup>

### 요약

최근 대규모 언어 모델의 활용 방법에 대한 많은 관심이 지속되고 있다. LLM이 생성한 정보의 가장 중요한 도전 과제는 출력 문장이 사실인지 판단하기 어렵다는 점이다. 본 논문은 하나 또는 복수의 LLM를 구조화하여, 생산되는 다양한 결과를 활용하는 방안을 탐구한다. LLM이 생성한 출력 정보를 고려하여 사실 관계 확인 과제를 수행함에 있어서, 성능 향상 가능성을 평가하기 위한 실험을 진행했다. 대규모 언어모델의 구조화를 통해 입력 정보의 제어를 할 경우, 기존 최고 성능보다 4.75의 정확도 향상을 관찰할 수 있었다.

주제어: 대규모 언어모델, Prompt Engineering, 대규모 언어모델 구조화

### 1. 서론

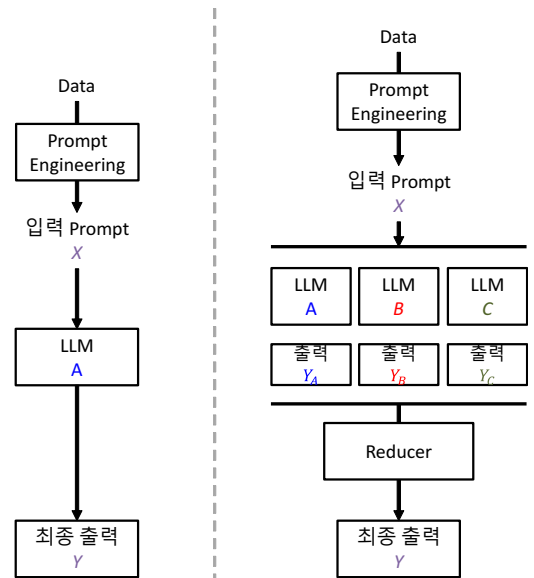
최근 ChatGPT를 포함한 다양한 대규모 언어 모델의 개발 이후, 인공 지능 및 자연어 처리 분야에서 정보 생성에 관련된 다양한 고민과 연구가 진행되고 있다 [1]. 특히, 언어 모델로 생성된 정보의 가장 큰 문제점 중 하나는 생성된 문장의 진실성을 판단하는 문제이다. 이에 대한 신뢰성 있는 판별 기준과 방법론을 개발하는 것은 핵심적인 과제 중 하나이다.

우리 연구는 이러한 사실 확인 문제에 주목하였다. 대규모 언어 모델(Large Language Model, LLM)를 구조화하여 다수의 결과를 종합적으로 활용하여 정보 생성 및 이에 대한 사실 확인의 품질을 향상시키는 방안을 모색한다. 이러한 방법론은 다양한 언어 모델이 생성한 결과물을 유용하게 활용하는 새로운 접근법을 제시한다.

입력과 출력 정보의 제어에 대한 연구 역시 우리 연구의 주요 관심사 중 하나이다. 대규모 언어모델의 다수의 출력 정보를 고려하여 사실 관계 확인을 진행할 때, 얼마나 성능이 향상되는지에 대한 실험을 진행한다. 이를 통해 언어 모델의 활용 가능성과 한계를 더 깊이 이해하고자 한다.

우리 연구는 정보 생성과 판별 분야에서 대규모 언어모델의 출력물을 반영하여 사실 확인 문제에 적용하는 방법론을 연구한다. 본 논문의 방법론에는 1) 입력 정보를 적절한 입력 패턴(prompt)를 통해 제어(prompt engineering [2])하고, 2) 이러한 prompt를 통해 대규모 언어모델로부터 다수의 결과를 얻어낸다. 3) 대규모 언어모델의 다수의 출력물을 하나로 종합하기 위해 서로의 출력을 반영하거나, 다수결을 통해 최종 출력을 결정한다. 이러한 기본 구성은 그림 1를 통해 확인할 수 있다.

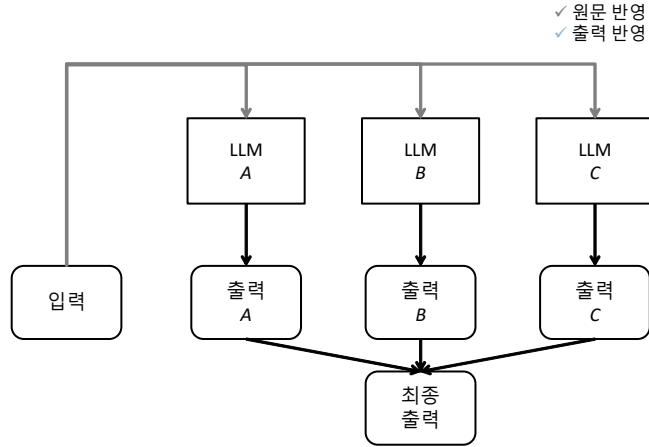
\*, †: 교신저자(Corresponding Author)



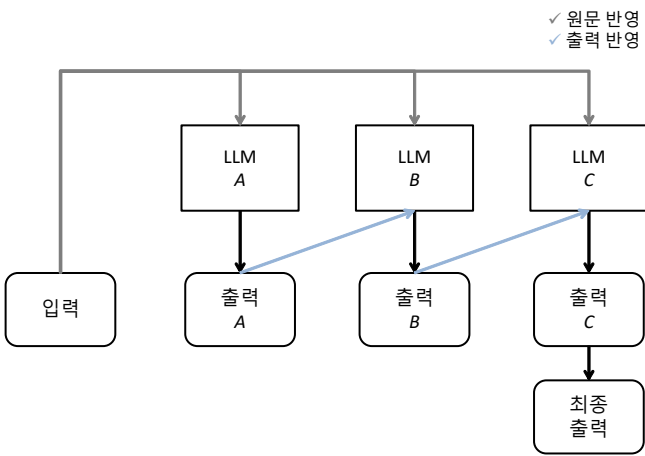
(a) 기존 대규모 언어모델 활용 방식  
 (b) 구조화를 통한 다중 대규모 언어모델 활용 방식  
 그림 1. 다중 대규모 언어모델의 입출력 제어 구조화

대규모 언어 모델의 출력 형태를 사실 확인 및 정보 검색의 형태로써 활용 가능성을 확장하는데 기여할 것으로 기대한다. 실제로, 기존 최고 성능 모델[3]에 비해, 대규모 언어모델을 구조화하여 활용할 때, 기존보다 4.75의 정확도 향상을 관찰할 수 있었다

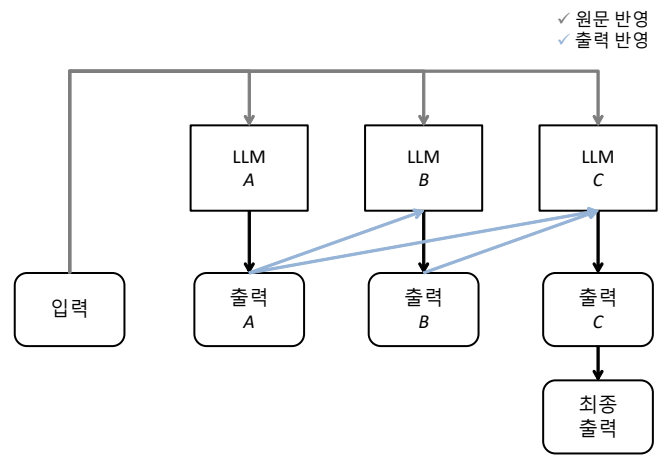
본 논문은 2. 장에서 관련 연구를 소개하고, 3. 장에서 연구의 방법론을 설명한다. 4. 장에서는 실험의 구체적 방법을 설명하며, 5.에서 이에 대한 결과와 분석을 보여준다. 6. 장에서는



(a) 입력 정보만 제공(Majority Voting)



(b) 순차적 출력 반영(Recursive Output Offer)



(c) 순차적 다중 출력 반영(Recursive Multiple Output Offer)

그림 2. 대규모 언어모델의 정보 제어를 위한 구조. 그림 2a은 동일한 입력만을 고려하는 구조(Majority Voting, MV), 그림 2b은 바로 앞 순서의 모델의 판단 근거를 반영하는 구조(Recursive Output Offer, ROO), 그림 2c은 모든 모델의 판단 근거를 순차적으로 반영하는 구조(Recursive Multiple Output Offer, RMOO)이다.

방법론에 대한 실험으로 입증된 결론을 정리한다.

## 2. 관련 연구

### 2.1 사실 관계 확인

사실 관계 확인 문제의 경우, 하나의 주장과 이에 관련된 사실을 가지고 지지(Support), 반박(Refute) 두 가지 라벨로 분류하는 문제이다. 이러한 문제를 위한 데이터로는 일반적인 주제에 대해 문장과 근거 사실이 결합되어 있는 FEVER[4]와 과학적 주장과 근거 사실이 결합되어 있는 SciFact[5] 데이터가 대표적으로 활용된다.

이러한 데이터를 활용하여, 미세조정(Fine-tuning)을 통한 자연어 처리 딥러닝 모델을 사용하는 경우가 대다수이다. 이러한 Fine tuning 방법에 대해서는, 해당 문제에 대해 다양한 접근법을 사용하고 있다. 그 중 높은 성능을 기록하고 있는 제시된 주장을 쪼개어 근거가 되는 사실을 대조하는 형식으로

학습을 진행하거나[3], 근거 문장에 대해 그래프 형태로 구성하여, 주장이 모델에 입력될 경우, 근거 문장과와의 거리를 판단하여 지지, 혹은 반박으로 라벨을 분류한다[6].

### 2.2 대규모 언어 모델 정보 제어

대규모 언어 모델의 출력을 원하는 형태로 얻기 위해, 입력 정보 제어(prompt engineering [2])를 진행한다. 원하는 글의 형태, 문체, 문맥 등을 제어하기 위해 이러한 prompt engineering을 다방면에서 활용하고 있다.

유사한 문맥을 가진 정보를 먼저 제공하여 원하는 출력을 얻는 문맥 학습(in-context learning [7])의 입력 형태를 볼 수 있다. 원하는 출력 형태를 제어하기 위한 명령문의 형태를 활용하기도 한다.

Prompt 범주	Prompt 패턴	
언어	영문	국문
번역	-	한국어로 번역해줘.: {입력}
주장+근거	Is this Evidence support the claim? Answer 'supports' or 'refutes'. And, explain the rationale for the answer. Claim: {주장} Evidence: {근거}	Evidence와 비교했을 때, claim이 사실입니까? 'supports' 혹은 'refutes'로 답하고, 그 이유를 대시오. Claim: {주장} Evidence: {근거}
주장+근거+LLM 판단 근거	Is this Evidence support the claim? Answer 'supports' or 'refutes'. And, explain the rationale for the answer. There is others opinion on this. Claim: {주장} Evidence: {근거} Other opinions: {LLM 판단 근거}	Evidence와 비교했을 때, claim이 사실입니까? 'supports' 혹은 'refutes'로 답하고, 그 이유를 대시오. 이에 대한 Opinion을 참고하세요. Claim: {주장} Evidence: {근거} Other opinions: {LLM 판단 근거}
주장+근거+LLM 판단 근거 다수	Is this Evidence support the claim? Answer 'supports' or 'refutes'. And, explain the rationale for the answer. There is others opinion on this. Claim: {주장} Evidence: {근거} Opinion 1: {LLM 판단 근거1} Opinion 2: {LLM 판단 근거2}	Evidence와 비교했을 때, claim이 사실입니까? 'supports' 혹은 'refutes'로 답하고, 그 이유를 대시오. 이에 대한 Opinion을 참고하세요. Claim: {주장} Evidence: {근거} Opinion 1: {LLM 판단 근거1} Opinion 2: {LLM 판단 근거2}

표 1. 사실 관계 확인 문제를 위한 영문 및 국문 prompt 패턴

### 3. 다중 대규모 언어모델 활용 사실 확인

본 논문은 3개의 서로 다른 GPT-3.5[8] 기반의 대규모 언어 모델들을 세 가지의 구조로 구성하였다. 이에 알맞게 prompt 패턴을 만들어 다중 대규모 언어모델의 입력 형태를 제어하였다.

#### 3.1 복수 대규모 언어모델의 입출력 구조화

본 연구에서 다수의 대규모 언어모델을 제어하기 위해, 세 가지의 구조를 사용하였다. 첫째로, 서로 의견을 주고받지 않고, 각 모델별 독립적 결과를 추합하여 다수결로 최종 결론을 출력하는 구조(Majority Voting, MV)와 둘째로 바로 앞의 LLM의 근거를 참고하여 사실관계를 판단하는 구조(Recursive Output Offer, ROO), 마지막으로 모든 LLM의 근거를 참고하여 사실 관계를 판단하는 구조(Recursive Multiple Output Offer, RMOO)이다. 그림 2에서 각 구조별 입출력 정보의 흐름을 볼 수 있다.

사실 관계 확인 문제에 대해 추가적인 훈련 없이, 본 논문은 prompt engineering만을 통해 해당 입력을 제어하여 다중 LLM을 구조화하였다. 이를 위해, 사실 관계 확인 문제를 위한 세 가지 형태의 입력 prompt 형태를 구성하였다.

#### 3.2 복수 대규모 언어모델의 순차 정보 제어를 위한 Prompt Engineering

사실 관계 확인은 정확한 판단을 내리는 것이 중요한 문제이다. 이를 위해, 대규모 언어모델의 상호작용을 통해 판단을 내릴 때, 정확도가 높아질 수 있다 [9]. 본 논문에서 정보 제어를 위한 입력 형태로써, 다음과 같은 prompt를 활용하여 실험을 진행하였다. Prompt의 경우, 입력 정보 제어를 위해 세 가지 구조에 맞추어 세 가지의 패턴을 사용하였다. 첫째로, 주장과 근거만 받는 경우(주장+근거), 둘째로, 주장과 근거, LLM 하나의 판단 근거를 받는 경우(주장+근거+LLM 판단 근거), 셋째로, 주장과 근거, 다수의 LLM의 판단 근거를 받는 경우(주장+근거+LLM 판단 근거 다수) 세 가지에 대해 prompt를 구성하였다. 이러한 범주에 대해, prompt 패턴의 구성은 표 1에서 확인할 수 있다.

또한, 데이터를 국문으로 번역하기 위해, 대규모 언어 모델을 활용하였다. 이에 대해 사용된 번역 prompt 패턴 역시, 표 1에서 확인할 수 있다.

### 4. 실험

#### 4.1 데이터

영문 문장과 이에 대한 뒷받침 사실 근거 데이터에 대해 지지(Support), 반박(Refute) 두개의 라벨로 구성되어 있는 FEVER[4] 데이터셋을 사용하였다. 전체 185,445개의 데이터 중 1000개를 샘플링하여 진행하였다. 이에 대한 표본 오차는

	FEVER	KorFEVER
gpt-3.5-turbo-16k-0613	81.60	67.97
gpt-3.5-turbo	81.90	68.49
gpt-3.5-turbo-0613	84.15	68.23
MV	82.78	72.06
ROO	<b>84.22</b>	<b>74.73</b>
RMOO	83.40	72.95

표 2. 사실 확인 관계 문제에 대한 정확도. **굵은 글씨**는 각 데이터별 가장 높은 성능을 뜻한다. MV(Majority Voting)는 다수결을 통해 최종 결론을 출력하는 구조, ROO(Recursive Output Offer)는 바로 앞의 LLM의 근거를 참고하여 사실관계를 판단하는 구조, RMOO(Recursive Multiple Output Offer)는 모든 LLM의 근거를 참고하여 사실 관계를 판단하는 구조이다.

신뢰수준 99%에 대해 4.07%이다.

현재 사실 관계 확인의 국문 실험을 위해, 이번 연구에서는 GPT-3.5-turbo를 통해 번역한 데이터를 활용하였다. 해당 데이터는 영문 데이터의 명칭을 본따, KorFEVER(Korean FEVER) 데이터셋으로 명명하였다.

## 4.2 모델

활용한 대규모 언어 모델로는 현재 오픈된 API를 가진 GPT-3.5-turbo, GPT-3.5-turbo-16k-0613, GPT-3.5-turbo-0613를 사용하였다. ROO, RMOO 구조를 활용한 경우 순차적으로 출력된 근거 정보를 다시 입력으로 받아들이므로, 이에 대한 순서는 개별 모델의 성능에 따라, 정확도가 낮은 순에서 높은 순으로 고정하였다. 모델을 정확도순으로 사용한 이유는, 정확도가 높은 모델이 더 많은 정보에 대해 제어 능력이 있을 것으로 판단했기 때문이다.

## 5. 결과

본 논문에서는 사실 관계 확인 문제에 대해 영문 FEVER 데이터셋과 국문으로 번역한 KorFEVER 데이터셋에 대하여 실제 라벨과 LLM이 판단한 라벨을 비교하였다. 이에 대한 평가 지표로서, 정확도와 분류결과표(confusion matrix)를 관찰하였다.

### 5.1 정확도

FEVER 데이터셋에 실험에 대한 정확도는 표 2에서 확인할 수 있다. 기존 최고 성능 모델[3]의 79.47에 비해, 대규모 언어모델을 ROO 구조의 형태로 활용할 때, 84.22의 성능을 기록하며 기존보다 4.75의 성능 향상을 관찰할 수 있었다. 이는 단독모델의 최고성능인 84.15에 비해 0.7의 성능 향상으로,

단순 다수결을 차용한 MV 구조에 비해 1.44 높은 성능이다.

KorFEVER 데이터셋에 대한 실험에 대해서도 단독모델을 사용했을 때보다, 구조화를 활용했을 때 성능이 더욱 높아진다. ROO 구조를 사용할 때 단독모델의 최고 성능보다 6.24 가량 높은 성능을 보인다.

### 5.2 분류결과표

FEVER의 결과는 그림 3, KorFEVER의 결과는 그림 4에서 확인할 수 있다. 데이터는 Supports 라벨이 전체의 약 77.08% 가량으로 치우쳐져 있다. 영문의 경우 실험에 활용된 모든 결과에 대해 Refutes 라벨을 Supports로 표기되는 오류가, Supports 라벨을 Refutes로 표기하는 경우보다 많다. KorFEVER에 대한 실험의 경우, 구조2a을 활용할 경우를 제외하면 유사한 경향성을 보인다.

### 5.3 분석

표 3는 각 구조별 정답으로 예측된 데이터에 평균적인 문장의 평균 길이 및 어절의 평균 개수를 보여준다. 특히, 대규모 언어모델의 출력 결과를 반영한 경우에는 단순 다수결에서 해결하지 못한 데이터에 한하여 평균을 추산하였다. 대규모 언어모델의 판단 근거에 노출될 경우, 단순 다수결을 활용할 때보다 짧은 길이의 데이터에 대해 대응 가능성을 유추할 수 있다.

구조화를 활용했을 때의 예측 결과가 달라짐을 표 4에서 확인할 수 있다.

## 6. 결론

본 연구는 다중 대규모 언어 모델의 활용을 통해 사실 확인 문제를 개선하기 위한 방법론을 탐구하였다. 우리는 대규모 언어 모델을 적절한 입력 형태로 제어하고, 이를 통해 다양한 결과를 얻어냄으로써 정보 생성 및 사실 확인의 품질을 향상시킬 수 있는 새로운 접근법을 제안했다.

실험 결과, 우리의 방법론이 기존의 최고 성능 모델에 비해 4.75의 정확도 향상을 이끌어냄을 확인하며, 대규모 언어모델의 구조화가 출력 결과 성능을 향상시킴을 입증하였다.

또한, 본 연구는 대규모 언어 모델의 출력 형태를 정보 검색 및 사실 확인의 용도로 활용할 수 있는 새로운 방법을 제시하여, 언어 모델의 활용 가능성을 확장하고자 한다.

이러한 연구 결과를 통해, 다중 대규모 언어 모델을 활용하여 정보 생성과 사실 확인 분야에서 더 나은 성과를 달성할 수 있는 더 나은 구조와, 출력 결과의 활용 방안에 대해 연구하고자 한다. 앞으로의 연구에서는 이러한 방법론을 더 발전시켜, 실제 사람이 활용할 수 있는 사용자 인터페이스, 혹은 결과 제어 방법론에 대해 확장하여, 활용성을 높이고자 한다.

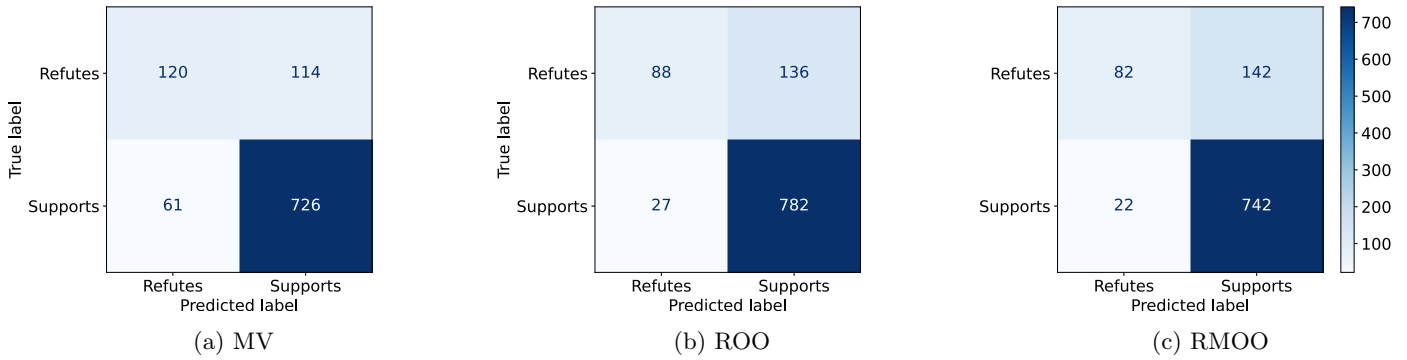


그림 3. FEVER[4] 데이터셋에 대한 정답 라벨과 예측 라벨로 구성된 분류결과표(confusion matrix). MV(Majority Voting)는 다수결을 통해 최종 결론을 출력하는 구조,ROO(Recursive Output Offer)는 바로 앞의 LLM의 근거를 참고하여 사실관계를 판단하는 구조, RMOO(Recursive Multiple Output Offer)는 모든 LLM의 근거를 참고하여 사실 관계를 판단하는 구조이다.

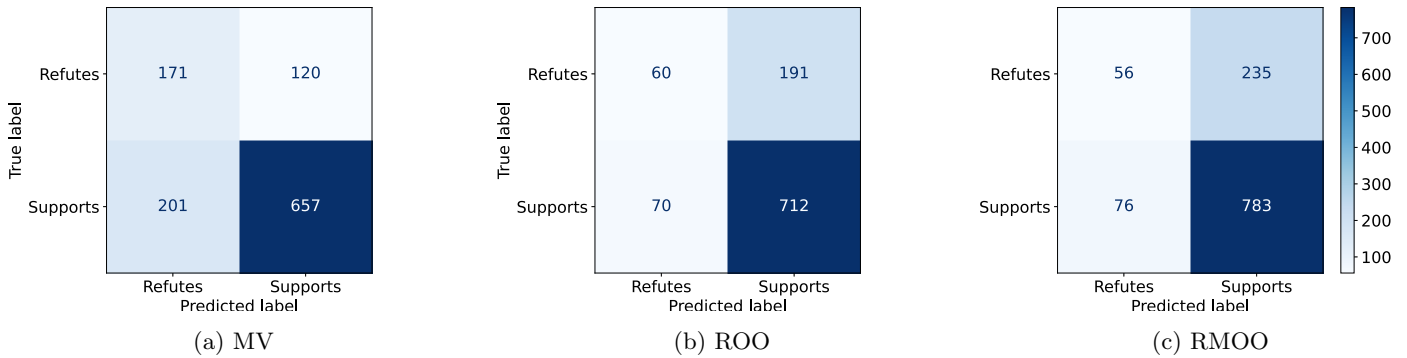


그림 4. KorFEVER 데이터셋에 대한 정답 라벨과 예측 라벨로 구성된 분류결과표(confusion matrix). MV(Majority Voting)는 다수결을 통해 최종 결론을 출력하는 구조,ROO(Recursive Output Offer)는 바로 앞의 LLM의 근거를 참고하여 사실관계를 판단하는 구조, RMOO(Recursive Multiple Output Offer)는 모든 LLM의 근거를 참고하여 사실 관계를 판단하는 구조이다.

## 감사의 글

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구이자 (2019-0-00004, 준지도학습형 언어지능 원천기술 및 이에 기반한 외국 인 지원용 한국어 튜터링 서비스 개발), 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2022R1F1A1071047)

## 참고문헌

- [1] S. Mohamadi, G. Mujtaba, N. Le, G. Doretto, and D. A. Adjeroh, "Chatgpt in the age of generative ai and large language models: A concise survey," *arXiv preprint arXiv:2307.04251*, 2023.
- [2] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance

prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

- [3] A. Krishna, S. Riedel, and A. Vlachos, "Proofver: Natural logic theorem proving for fact verification," *Transactions of the Association for Computational Linguistics*, Vol. 10, pp. 1013–1030, 2022.
- [4] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira, "No parameter left behind: How distillation and model size affect zero-shot retrieval," *arXiv preprint arXiv:2206.02873*, 2022.
- [5] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," *arXiv preprint arXiv:2004.14974*, 2020.
- [6] W. Zhong, J. Xu, D. Tang, Z. Xu, N. Duan, M. Zhou, J. Wang, and J. Yin, "Reasoning over semantic-level graph for fact checking," *arXiv preprint*

	FEVER				KorFEVER			
	문장 길이		어절 개수		문장 길이		어절 개수	
	주장	근거	주장	근거	주장	근거	주장	근거
MV	45.99523	293.1265	7.856802	49.80549	46.26439	43.78496	9.451234	9.379553
ROO	48.67361	188.5903	8.451389	31.88194	47.91005	43.66138	10.04233	9.428571
RMOO	48.5188	185.4211	8.383459	31.51128	47.48276	41.05747	9.810345	8.856322

표 3. 정답으로 판단된 데이터의 문장과 어절의 평균 길이이다. 출력 결과를 반영한 경우(ROO, RMOO)의 경우, 단순 다수결(MV)에서 예측하지 못한 경우에 대해 계산하였다. MV(Majority Voting)는 다수결을 통해 최종 결론을 출력하는 구조, ROO(Recursive Output Offer)는 바로 앞의 LLM의 근거를 참고하여 사실관계를 판단하는 구조, RMOO(Recursive Multiple Output Offer)는 모든 LLM의 근거를 참고하여 사실 관계를 판단하는 구조이다.

*arXiv:1909.03745*, 2019.

- [7] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.
- [8] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen *et al.*, "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," *arXiv preprint arXiv:2303.10420*, 2023.
- [9] C. Zhang, L. Liu, J. Wang, C. Wang, X. Sun, H. Wang, and M. Cai, "Prefer: Prompt ensemble learning via feedback-reflect-refine," *arXiv preprint arXiv:2308.12033*, 2023.

		label
claim	Alex Rodriguez never hit a home run.	
evidence	During his 22-year career, Rodriguez has amassed a .297 batting average, 696 home runs, over 2,000 runs batted in "RBI", over 2,000 runs scored, and over 3,000 hits.	REFUTES
GPT-3.5-turbo	The evidence supports the claim. Rationale: The evidence states that Alex Rodriguez has accumulated 696 home runs in his 22-year career, which contradicts the claim that he never hit a home run.	SUPPORTS
GPT-3.5-turbo-16k-0613	The evidence supports the claim. The evidence states that Alex Rodriguez has amassed 696 home runs, which contradicts the claim that he never hit a home run.	SUPPORTS
GPT-3.5-turbo-0613	The evidence supports the claim that Alex Rodriguez did hit a home run. The evidence states that he has amassed 696 home runs, which indicates that he has indeed hit home runs throughout his career.	SUPPORTS
ROO	This evidence refutes the claim that Alex Rodriguez never hit a home run. The evidence clearly states that Rodriguez amassed 696 home runs during his career. The other opinions that state the evidence supports the claim are incorrect, as they are misinterpreting the evidence.	REFUTES
RMOO	Supports. The evidence provided clearly states that Alex Rodriguez has amassed 696 home runs over his 22-year career. This directly contradicts the claim that he never hit a home run, therefore supporting the claim that he did indeed hit a home run.	SUPPORTS

표 4. FEVER 데이터셋의 단독 모델의 예측 결과와, 구조화를 활용했을 경우의 예측 결과 예시이다. ROO(Recursive Output Offer)는 바로 앞의 LLM의 근거를 참고하여 사실관계를 판단하는 구조, RMOO(Recursive Multiple Output Offer)는 모든 LLM의 근거를 참고하여 사실 관계를 판단하는 구조이다.