https://doi.org/10.7840/kics.2025.50.1.9

대규모 언어 모델에서 프롬프트 엔지니어링 기법에 관한 연구

손 민 준◆, 이 성 진°

Research on Prompt Engineering Techniques in Large Language Models

Minjun Son◆, Sungjin Lee°

요 약

최근 자연어 처리기술은 대형 언어 모델의 발전으로 인해 전례 없는 속도로 발전하고 있으나 모델이 부정확하거나 비상식적인 답변을 생성하는 할루시네이션 문제는 여전히 해결해야 할 과제로 남아 있다. 본 논문은 대규모 언어 모델에서 다양한 프롬프트 엔지니어링 기법을 분석하고 데이터셋 별 최적의 응답 성능을 이끌어 낼 수 있는 프롬프트 엔지니어링 기법을 도출한다. 특히 대표적인 프롬프트 엔지니어링 기법인, 문맥 내 학습, 사고의 연쇄, 검색 증강 생성 기법을 분석하고, 이를 LLaMA3, Mistral, Gemma2와 같은 주요 대규모 언어모델에 적용하였다. 연구 결과, 각 데이터셋의 특성에 따라 가장 적합한 프롬프트 엔지니어링 기법이 달라질 수 있음을 알 수 있었다.

키워드 : 대규모 언어모델, 프롬프트 엔지니어링, 문맥 내 학습, 사고의 연쇄, 검색 증강 생성
Key Words : Large Language Model, Prompt Engineering, In-context learning, Chain of Thought, Retrieval-Augmented Generation

ABSTRACT

Recent natural language processing technology has been advancing at an unprecedented pace, driven by the development of large language models. However, the issue of hallucination, where the model generates inaccurate or nonsensical responses, remains a challenge to be addressed. This paper analyzes various prompt engineering techniques in large-scale language models and derives prompt engineering methods that can achieve optimal response performance for each dataset. The study found that the most suitable prompt engineering techniques can vary depending on the characteristics of each dataset.

Ⅰ. 서 론

최근 자연어 처리 기술은 대형 언어 모델(Large Language Model, LLM)의 발전으로 인해 전례 없는 속도로 발전하고 있다. 이러한 모델들은 방대한 데이터셋으로 사전 학습되어 복잡한 언어 패턴을 인식하고 이해하는 능력을 갖추고 있으며, 챗봇과 같은 질문 응답(Question Answering), 텍스트 요약, 언어 번역, 콘텐츠 생성, 음성인식에 이르기까지 다양한 분야에서 AGI (Artificial General Intelligence)로 활용하여 발전되고

◆ First Author : Sungkyunkwan Univ. Department of Metabiohealth, mj.son14820@gmail.com, 학생회원
° Corresponding Author : Soonchunhyang Univ. Department of Smart Automotive, sungjinlee@sch.ac.kr, 정회원
논문번호 : 202407-137-A-RN, Received July 8, 2024; Revised August 21, 2024; Accepted August 21, 2024

있다[1]. 특히, LLM의 발전은 간단한 통계적 언어 모델(Statistical Language Model)[2]에서 GPT-4[3], PaLM[4], LlaMA[9]와 같은 정교한 트랜스포머[6] 기반 모델로의 진화를 통해 그 역량과 적용 범위에서 큰 도약을 이루었다. 대표적으로 활용되는 모델로서 OpenAI의 GPT-4[4], Meta의 LLaMA3[6], Google의 Gemini[7]와 Gemma[8], Mistral의 Mistral[9], Mixtral[10]의 모델들이 있으며 이들은 각자의 강점과 특징에 기반하여 각자의 영역을 구축하고 지속적인 발전을 거듭하고 있다.

이러한 LLM의 발전은 사전 학습된 언어 모델인 PLM (Pretrained Language Model)과 이를 통한 파인 튜닝(Fine-Tuning) 기법에 기반한다. 특히, PLM의 규모가 커지면서 LLM으로 진화함에 따라, 이전에는 없던 새로운 추론 능력들이 입증되며 발전이 가속화되고 있다. 대표적인 추론 능력으로는 문맥 내 학습 (In-Context Learning, ICL)[11], 지시 따르기 (Instruction Following, IF)[12], 단계별 추론 (Step-by-Step Reasoning, SSR)[13] 등이 있다.

하지만, 이런 PLM의 발전에도 사용하는 데이터셋이나 적용되는 태스크에 따라 그 정확도는 충분한 수치에 이르지 못하고 심지어 존재하지 않는 정보를 포함하여 잘못된 답변을 하는 할루시네이션 (Hallucination) 문제 또한 발생하기 때문에, 최근 이를 보강하기 위한 다양한 방식의 프롬프트 엔지니어링 (Prompt Engineering) 기법이 연구되고 있다. 이런 프롬프트 엔지니어링은 LLM이 제공하는 최종 답변의 품질을 결정하는 중요한 요소이며, 프롬프트의 구조와 내용을 재구성하여 모델의 출력을 최적화할 수 있다. 특히 이런 프롬프트 엔지니어링은 많은 비용과 시간을 요구하는 사전학습이나 미세조정 단계의 훈련 없이 추론 단계에서 프롬프트만을 재구성하여 원하는 대답을 얻어내는 방식이기 때문에 상용화 측면에서 매우 유리한 기술이라 할 수 있다[14].

이런 프롬프트 엔지니어링의 대표적 방식 중 하나는 검색-증강-생성(Retrieval-Augmented Generation, RAG) 기법으로, 외부 정보 (external knowledge)를 통해 현재 데이터를 보강하는 검색 단계와 이를 통한 생성 단계를 결합하여 정보검색의 정확성을 높이고 모델이 최신 정보나 드물게 나타나는 사실을 포함한 질문에 더 나은 응답을 제공하도록 한다. 이런 RAG 기법을 통하면 생성 모델이 실제 검색된 정보를 기반으로 응답을 생성하므로, 좀 더 정확한 정보를 생성할 가능성이 증가하며, 좀 더 최신 정보를 기반으로 응답을 생성하게 된다.

반면 사고의 연쇄 (Chain of Thought, CoT) 기법은 언어모델이 복잡한 문제를 해결하기 위해 중간단계나 논리적 추론 과정을 명시적으로 표현하도록 유도하는 방법이다. 그 예시로, 수학 문제 해결 과정을 응답에 포함하도록 프롬프트를 구성하여 응답에 대한 적합성 및 그에 이르는 추론 과정의 적합성도 같이 평가할 수 있도록 한다.

마지막으로 문맥 내 학습 ICL기법은 문자 그대로 맥락 내에서 의미를 학습하는 방식이다. 즉, 프롬프트 질문에서 맥락을 이해하여 관련 대답을 용이하게 하여 생성하는 것을 의미한다. 이를 위해 보통 예시를 제공함으로써 관련 대답을 이끌어 낸다. 특히, 예시 횟수에 따라 그 세부 기술이 구분되는데, 관련 예시를 0회 제공한다면 Zero-Shot Learning이라 하고, 관련 예시를 1회 제공한다면 One-Shot Learning, 관련 예시를 수차례에 걸쳐 제공한다면 Few-Shot Learning이라고 정의한다.

본 논문에서는 LLM에서 주로 사용되는 RAG, CoT, ICL 프롬프트 엔지니어링 기법에 대해 연구를 진행하였다. 특히 해당 기법들을 주요 LLM 기술들인 Llama3, Mistral, Gemma2에 적용하여 성능을 도출하였으며, 벤치마크 데이터인 ARC (AI2 Reasoning Challenge), HellaSwag, MMLU (Massive Multitask Language Understanding), TruthfulQA, Winogrande, GSM8K (Grade School Math)를 통해 해당 프롬프트 엔지니어링 기법들에 대한 자연어 생성 성능을 분석하였다.

Ⅱ. 관련 연구

여기서는 프롬프트 엔지니어링은 LLM이 제공하는 답변의 품질을 결정하는 중요 기술로서, 모델의 사전 학습 및 미세조정 없이 추론 단계에서 프롬프트의 구조와 내용만을 재조정하여 모델의 출력을 최적화하는 기법이다. 프롬프트 엔지니어링의 초기에는 단순한 질의응답 형태의 프롬프트가 주로 사용되었으나, 최근에는 복잡한 문제 해결을 위해 다양한 프롬프트 구조가 개발되고 있다[18]. 초기에는 GPT-3 모델을 개발한 OpenAI를 중심으로 다양한 프롬프트 엔지니어링 기법이 연구되었다. 특히 이들 중 LLM 모델에 대한 성능 개선 방안으로 추가적인 훈련 없이 원하는 답을 얻어내기 위한 ICL 기반의 Few-Shot Learning 프롬프트 설계 기법들이 연구되었다[11]. 이런 ICL 방식의 성공으로 이에 대한 다양한 변이 방식들, 즉, PICL[16], MetaICL[17], KATE[18], SG-ICL[19], GlobalE&LocalE[20] 기법들이 연구되었다. 초기 ICL 연구는 모델 훈련 없이 프롬프트 재구성만으로 ICL 성능을 최적화하려 하였으나, PICL[16] 연구는 관련된 문맥 데이터들을 수집하고 말뭉치를 재구성하여 사전 훈련을 수행함으로써, 모델이 프롬프트 시연(demonstration)을 통해 추론하는 방법을 사전에

학습하도록 제안했다. 반면, MetaICL[17] 방식은 사전 훈련과 ICL 추론 사이에 지속적인 훈련(Continual Training) 단계를 추가하는 것으로, 이를 모델 워밍업이라고 한다. 워밍업은 ICL을 위한 선택적 절차로, 추론 전에 LLM의 매개변수를 수정하거나 추가하여 모델을 조정할 수 있다. 또한, 많은 연구에서 ICL의 성능이 시연 구성, 즉 시연 예제의 선택, 형식 및 순서에 크게 의존한다는 것을 보여주었다. KATE[18]는 이런 시연 예제의 선택에 관한 연구를, SG-ICL[19]은 시연 예제의 형식에 관한 연구를, GlobalE&LocalE[20]는 시연 예제의 순서에 관한 연구를 수행하였다.

복잡한 문제를 해결하기 위해 응답에 대한 중간단계나 논리 추론 과정을 같이 표현하도록 유도하는 기법들이 연구되고 있다. 대표적으로는 CoT, Self-Consistency, ToT (Tree of Thoughts) 등의 기법들이 있다. 여기서, Self-Consistency 방식은 언어 모델이 동일한 의미의 다양한 입력에 다양한 답변을 내놓을 수 있는데, 이들 간에 일관되고 신뢰할 수 있는 출력을 생성하도록 최적화하는 기술이다. ToT 방식은 언어 모델이 문제를 해결하기 위해 필요한 여러 추론 단계를 여러 “Thought Tree”로 분기하여 표현하는 방식이다. 즉, 각 분기는 다른 방식의 사고 과정이나 가정을 의미하기 때문에, 다양한 가능성이나 가설들을 탐구할 수 있는 장점을 지닌다. 또한 이들 간에 Query에 대한 관련성을 평가하고 실시간성을 고려하여 가장 합당한 결과를 도출하게 한다.

마지막으로 최근 LLM의 가장 큰 문제로 지적되고 있는 할루시네이션 문제, 즉, 존재하지 않는 정보를 포함하여 잘못된 답변을 하는 오류를 해결하기 위한 프롬프트 엔지니어링 기법이 연구되고 있다[21]. 이런 할루시네이션 문제는 훈련 데이터 중에 최신 지식의 부족이나 특정 사례 혹은 개인정보에 대한 정보 접근의 제한에 기인한다. 가장 활발히 연구되고 있는 RAG 기법[22]은 입력 프롬프트에서 쿼리를 추출하고, 그 쿼리를 사용하여 외부 지식 소스 (예: 검색 엔진이나 지식 그래프)에서 정보를 검색하고 이를 원래 프롬프트에 추가하거나 혹은 프롬프트에서 제공되는 관련 특정 정보들을 추출하는 과정을 통해 이를 해결한다. 그런 다음, 보강된 프롬프트를 LLM에 입력하여 최종 응답을 생성한다. RAG 시스템은 이렇게 검색, 증강, 생성을 통해 할루시네이션을 해결한다. 이런 RAG 기법의 성공으로 관련 변이 기법들, 즉, AI Text 생성 전에 정보 검색 (Information Retrieval)을 수행하는 LLM-Augmenter[23], 생성 중에 수행하는 Knowledge Retrieval[24], 생성 후에 수행하는 RARR[25], 피드백과 논리적 추론에 따라 Self-Refinement를 하는 Chain-of-Verification (CoVe) 기법[26] 등이 연구되었다.

본 논문의 주요 기여는 다음과 같다.

(1) 다양한 LLM 모델들과 이들의 성능 개선을 위한 프롬프트 엔지니어링 기법 및 관련 데이터셋, 성능 평가 메트릭에 대한 기술 분석,
(2) 주요 프롬프트 엔지니어링 기술들의 다양한 LLM모델들과 데이터셋에서의 다양한 성능 평가 메트릭으로의 성능 비교 분석,
(3) 성능 비교 분석을 통한 적용 분야에 따른 성능 최적화 전략 도출.

3장에서는 본 연구에서 사용하는 LLM언어모델에 대해 설명하고, 4장에서는 사용되는 벤치마크 데이터셋을 설명하며, 5장에서는 성능 측정을 위한 성능지표를 설명한다. 6장에서는 위의 파라미터들에 기반한 실험 결과를 제시하고 7장에서는 결론을 제시한다.

Ⅲ. 시스템 모델

3.1 Gemma2

Gemma2는 2024년 6월 27일에 발표된 구글의 최신 오픈 모델로, 허깅페이스 기준 90억 및 270억 매개변수의 두 가지 크기로 제공되며, 각각의 크기는 사전 학습 및 명령어 튜닝된 변형을 포함한다.

3.2 Llama3

Llama3는 Meta에서 개발한 최신 대형 언어 모델로, 2024년 4월 18일에 출시되었다. 이 모델은 이전 버전인 Llama 2를 기반으로 하여 개발되었으며, 80억 및 700억 매개변수의 두 가지 크기로 제공된다[6].

3.3 Mistral

Mistral은 프랑스의 AI 스타트업인 Mistral AI에서 개발한 고성능 언어 모델이다. Mistral 모델은 70억 매개변수를 갖춘 언어 모델로 그룹화된 쿼리 어텐션(GQA) 및 슬라이딩 윈도우 어텐션(SWA)과 같은 최신 기술을 사용하여 효율성을 높였다[9,10].

Ⅳ. 벤치마크 데이터셋

4.1 AI2 Reasoning Challenge

ARC 데이터셋은 11,119 train, 2,992 validation, 4,541 test로 구성되어 있으며, 개방형 복잡한 과학 문제

를 다루며, 쉬운 세트와 도전 세트로 구분된다. 이 구분은 문제를 해결하는 데 필요한 지식과 추론 유형에 기반한다. 데이터셋의 각 질문은 문맥과 가능한 답변들을 제공하며, 제공된 정보와 필요한 외부 지식을 사용하여 올바른 답변을 선택하는 것이 작업의 목표다. 이 데이터셋은 2018년 클라크(Clark) 등에 의해 소개되었으며, 과학 내용에 대한 깊은 이해와 추론 능력을 테스트하기 위해 사용한다[27].

4.2 HellaSwag

HellaSwag 데이터셋은 39,905 train, 5,000 validation, 10,042 test로 구성되어 있으며, AI 모델의 상식을 평가하기 위한 고급 벤치마크로 설계되었다. 이 데이터셋은 위키피디아 및 지침 비디오와 같은 다양한 도메인에서의 문맥을 포함하며, 각 문맥에 대해 제공된 여러 선택지 중에서 가장 적합한 문장 연속을 예측하는 작업을 포함한다. 이 작업은 다양한 상황에 대한 깊은 이해와 추론을 요구하므로, AI 모델의 상식추론 능력을 테스트하는 데 적합하다[28].

4.3 Massive Multitask Language Understanding

MMLU 데이터셋은 57,243 train, 10,000 validation, 15,000 test으로 구성되어 있으며, 페이스북 AI의 인간 언어 이해 및 생성 능력 개발 노력의 일환으로, 과학, 인문학, 전문 분야 등 다양한 주제에 대한 다지선다형 문제를 포함하는 작업을 통해 AI 모델의 언어 이해 및 적용 능력을 평가하는 데 사용한다[29].

4.4 TruthfulQA

TruthfulQA 데이터셋은 8,170 train, 1,024 validation, 2,035 test으로 구성되어 있으며, 언어 모델이 사실적이고 진실된 반응을 생성할 수 있는 능력을 평가하기 위해 설계되었다. 이 데이터셋은 종종 오해의 소지가 있거나 모호한 표현을 사용하거나 모델이 답을 날조하기보다는 무지를 인정해야 하는 질문을 포함한다. 이 데이터셋은 모델의 진실성과 정보검색 및 응답 생성에서의 미묘함을 다루는 능력의 한계를 평가하는 데 사용한다[30].

4.5 Winogrande

Winogrande 데이터셋은 40,398 train, 1,267 validation, 3,545 test으로 구성되어 있으며, AI 시스템 내 상식 추론 능력을 훈련하고 테스트하기 위해 설계된 대규모 데이터셋이다. 이는 Winograd Schema Challenge보다 더 높은 난이도의 상식 지식을 이해하고 적용해야 하는 문제를 제기한다. 두 개의 주어진 옵션 중 올바른 단어를 선택하여 채워야 하는 공백이 있는 문장들의 모음으로 구성되어 있으며, 맥락적이고 상식적인 추론이 평가하는 데 사용한다[31].

4.6 Grade School Math 8K

GSM8K 데이터셋은 7,432 train, 800 validation, 1,319 test으로 구성되어 있으며, AI 시스템의 수학적 추론 능력을 테스트하기 위해 설계된 초등학교 수준의 수학 문제 모음이다. 이 데이터셋은 산수, 대수, 기하학 및 단어 문제와 같은 일반적으로 초등학교 교육 과정에서 마주치는 주제를 다룬다. AI 모델에게 단순히 답을 계산하는 것뿐만 아니라, 자주 사용되는 자연어 이해와 수치 추론이 결합된 문제의 텍스트 맥락을 파악하는 성능을 평가하는 데 사용한다[32].

Ⅴ. 성능 지표

5.1 BLEU (Bilingual Evaluation Understudy)

BLEU 스코어는 기계 번역의 품질을 평가하기 위해 개발된 척도로, 번역된 문장과 참조 문장 간의 n-gram 오버랩을 계산한다. BLEU 스코어는 다음과 같은 공식으로 계산된다:

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n)$$

여기서 p_n은 n-gram 정밀도, w_n은 가중치, BP는 Brevity Penalty를 나타낸다. BLEU는 주로 번역된 텍스트의 유창성과 정확성을 평가하는 데 사용되며, 높은 스코어는 번역된 텍스트가 참조 텍스트와 얼마나 일치하는지를 나타낸다[33].

5.2 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE 스코어는 주로 요약 태스크에서 생성된 텍스트와 참조 텍스트 간의 유사도를 측정하는 데 사용된다. ROUGE는 여러 변형이 있으며, ROUGE-N, ROUGE-L, ROUGE-W 등이 있다. ROUGE-N은 n-gram 오버랩을 기반으로 하며, ROUGE-L은 가장 긴 공통 부분 문자열(Longest Common Subsequence, LCS)을 기반으로 한다. ROUGE는 생성된 요약의 포괄성과 일관성을 평가하는 데 유용하며, 특히 다중 문서 요약 평가에 자주 사용된다[34].

$$ROUGE - N = \frac{\sum_{s\in\{Reference\ \ Summaries\}}\sum_{gram_n\in s} Count_m(gram_n)}{\sum_{s\in\{Reference\ \ Summaries\}}\sum_{gram_n\in s} Count(gram_n)}$$

5.3 METEOR (Metric for Evaluation of Translation with Explicit Ordering)

METEOR 점수는 생성된 텍스트와 참조 텍스트 간의 정렬을 기반으로 텍스트 품질을 측정하는 지표이다. METEOR는 유니그램 정밀도와 재현율의 조화 평균을 사용하며, 재현율에 더 높은 가중치를 부여하여 평가한다. METEOR는 동의어, 형태소 분석 등을 고려하여 단어 매칭의 유연성을 높이며, BLEU보다 인간 평가와의 상관관계가 높다고 알려져 있다. 이를 통해 번역 품질을 더욱 세밀하게 평가할 수 있다[35].

$$METEOR = F_{mean} * (1 - P_{penalty})$$

$$F_{mean} = \frac{10*P*R}{R+9*P}$$

$$P_{penalty} = 0.5 * (\frac{chunks}{matches})^3$$

P: 정밀도 (Precision)

R: 재현율 (Recall)

chunks: 일치하는 연속된 구문 덩어리의 수

matchesmatchesmatches: 일치하는 단어의 수

5.4 BLEURT (Bilingual Evaluation Understudy with Representations from Transformers)

BLEURT은 텍스트 생성 품질 평가를 위해 고안된 첨단 평가 지표이다. BLEURT는 사전 학습된 언어 모델과 파인 튜닝(fine-tuning)을 통해 문장 간의 유사성을 평가한다. 이는 기계 번역, 요약, 및 텍스트 생성과 같은 자연어 처리(NLP) 태스크에서 사용되며, BLEURT는 특히 의미적 일치성 및 문맥적 연관성을 포착하는 데 강점을 지니고 있다[36].

5.5 BERT (Bidirectional Encoder Representations from Transformers)

BERT Score는 텍스트 생성의 의미적 유사성을 평가하기 위해 고안된 지표로, 사전 학습된 BERT 모델을 활용한다. BERT Score는 문장 간의 의미적 유사성을 단어 수준에서 정밀하게 평가하는데, 이는 전통적인 n-그램 기반 지표(BLEU, ROUGE)와 달리, 문맥적 정보와 단어의 의미적 관계를 고려하여 더 정교한 평가를 가능하게 한다[37].

Ⅵ. 실험결과

6개의 벤치마크 데이터에 대해 ICL, CoT와 RAG, 그리고 RAG와 CoT를 혼합하여 총 Base, ICL, CoT, RAG(Base), RAG(CoT) 대해 응답을 추론하고 성능을 비교하였다. 또한 해당 프롬프트 엔지니어링 기법에 따라 만들어진 프롬프트 형식을 가지고 Llama3, Gemma2, Mistral에 대해 각 데이터셋의 40%에 파라미터 Temporature 0.1, Top_p 0.9를 적용하여 응답 추론을 진행했다. 성능 분석을 위해 NVIDIA H100 Tensor Core GPU를 기반으로 사전훈련된 모델들, LLama3, Mistral, Gemma2에서 프롬프트 미세조정 훈련을 수행하였다.

표 1, 2, 3는 BLEU, ROUGE, METEOR, BLEURT, BERT 성능 지표의 값을 나타내었다. 표 4, 5, 6은 LLM 모델 별 각 데이터 셋에서 가장 좋은 성능을 나타내는 프롬프트 엔지니어링 기법과 해당 수치 결과값을 나타낸다.

표 4, 5, 6에서 알 수 있듯이, 각기 다른 LLM모델을 사용할지라도 데이터셋 별로 최적의 프롬프트 엔지니어링 기법들이 거의 유사하는 것을 알 수 있다. 즉, 수학적, 과학적 전문 분야에 대한 추론 능력이 요구되는 데이터셋인 ARC, GSM8K, MMLU 에는 CoT 혹은 CoT+ICL 기반의 프롬프트 엔지니어링 기법이 성능향상에 유리하다. 반면, 정보검색 및 진실성이 목적인 TruthfulQA 같은 데이터셋에는 RAG기반의 기법이 가장 좋은 성능을 보인다. 이는 데이터셋의 목적 별로 좋은 점수를 얻기 위해 선호되는 프롬프트 엔지니어링 기법이 다르기 때문이다.

특히, 상식추론 능력을 요구하는 HellaSwag 데이터셋의 경우 지식에 대한 의존성이 강하고 추론 능력이 요구되기 때문에 RAG와 CoT 프롬프트 엔지니어링이 가장 유리할 것으로 예상되나, 이는 Gemma2 모델에만 적용되며 Llama3 및 Mistral에는 CoT가 가장 좋은 성능을 보이는 것으로 도출되었다. 이를 통해, Gemma2 모델이 다른 모델들 Llama3 및 Mistral에 비해 RAG 활용성이 더 잘 구현되어 있다는 것을 알 수 있고 RAG를 활용함으로서 얻어지는 성능 개선의 폭도 상당함을 알 수 있다.

표 1. ARC/GSM8k 데이터셋의 성능 지표
Table 1. Performance Metrics for ARC and GSM8k Datasets

ARC/BLEU				GSM8k/BLEU			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	2.170	5.130	1.301	Base	10.445	8.337	7.770
ICL	2.091	4.186	1.047	ICL	8.368	4.994	7.388
CoT	0.000	14.649	0.715	CoT	11.799	8.389	8.285
CoT+ICL	0.000	0.000	1.444	CoT+ICL	11.411	8.016	5.489
RAG(Base)	1.409	4.922	1.485	RAG(Base)	6.757	8.331	8.151
RAG(CoT)	0.000	0.000	1.374	RAG(CoT)	11.147	8.503	8.166
ARC/ROUGE				GSM8k/ROUGE			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.220	0.266	0.187	Base	0.411	0.451	0.358
ICL	0.189	0.241	0.132	ICL	0.387	0.355	0.306
CoT	0.699	0.836	0.484	CoT	0.428	0.445	0.356
CoT+ICL	0.070	0.819	0.527	CoT+ICL	0.437	0.436	0.277
RAG(Base)	0.154	0.273	0.159	RAG(Base)	0.321	0.446	0.324
RAG(CoT)	0.249	0.812	0.511	RAG(CoT)	0.415	0.449	0.320
ARC/METEOR				GSM8k/METEOR			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.170	0.205	0.180	Base	0.249	0.243	0.216
ICL	0.119	0.178	0.146	ICL	0.234	0.195	0.200
CoT	0.350	0.418	0.243	CoT	0.269	0.242	0.221
CoT+ICL	0.035	0.409	0.264	CoT+ICL	0.270	0.245	0.171
RAG(Base)	0.112	0.211	0.144	RAG(Base)	0.201	0.248	0.211
RAG(CoT)	0.125	0.406	0.256	RAG(CoT)	0.262	0.250	0.205
ARC/BLEURT				GSM8k/BLEURT			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.333	0.373	0.328	Base	0.406	0.405	0.381
ICL	0.256	0.337	0.277	ICL	0.398	0.376	0.368
CoT	0.480	0.534	0.393	CoT	0.415	0.404	0.384
CoT+ICL	0.201	0.534	0.418	CoT+ICL	0.413	0.403	0.361
RAG(Base)	0.246	0.369	0.282	RAG(Base)	0.373	0.406	0.380
RAG(CoT)	0.306	0.529	0.418	RAG(CoT)	0.410	0.412	0.373
ARC/BERTScore				GSM8k/BERTScore			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.608	0.639	0.569	Base	0.771	0.856	0.747
ICL	0.591	0.598	0.445	ICL	0.712	0.728	0.667
CoT	0.976	0.979	0.899	CoT	0.810	0.859	0.741
CoT+ICL	0.898	0.985	0.937	CoT+ICL	0.824	0.826	0.559
RAG(Base)	0.536	0.646	0.511	RAG(Base)	0.775	0.848	0.704
RAG(CoT)	0.938	0.983	0.943	RAG(CoT)	0.775	0.834	0.648

표 2. HellaSwag / MMLU데이터셋의 성능 지표
Table 2. Performance Metrics for HellaSwag and MMLU Datasets

HellaSwag/BLEU				MMLU/BLEU			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	1.207	0.525	1.072	Base	1.696	3.683	1.863
ICL	0.301	0.741	0.827	ICL	0.956	2.887	1.226
CoT	0.000	0.000	0.219	CoT	4.238	1.290	0.142
CoT+ICL	2.835	0.000	0.038	CoT+ICL	8.709	9.535	0.123
RAG(Base)	1.048	1.047	1.420	RAG(Base)	1.300	4.319	1.594
RAG(CoT)	0.000	0.000	0.050	RAG(CoT)	0.169	0.508	0.115
HellaSwag/ROUGE				MMLU/ROUGE			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.144	0.136	0.172	Base	0.148	0.261	0.180
ICL	0.098	0.117	0.144	ICL	0.129	0.210	0.123
CoT	0.448	0.701	0.389	CoT	0.591	0.673	0.477
CoT+ICL	0.293	0.712	0.265	CoT+ICL	0.340	0.637	0.409
RAG(Base)	0.138	0.161	0.182	RAG(Base)	0.136	0.258	0.148
RAG(CoT)	0.330	0.725	0.353	RAG(CoT)	0.204	0.617	0.447
HellaSwag/METEOR				MMLU/METEOR			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.096	0.092	0.122	Base	0.097	0.177	0.141
ICL	0.062	0.086	0.104	ICL	0.079	0.142	0.110
CoT	0.224	0.351	0.196	CoT	0.296	0.336	0.241
CoT+ICL	0.147	0.356	0.142	CoT+ICL	0.170	0.319	0.207
RAG(Base)	0.090	0.108	0.135	RAG(Base)	0.078	0.177	0.122
RAG(CoT)	0.165	0.363	0.185	RAG(CoT)	0.103	0.309	0.225
HellaSwag/BLEURT				MMLU/BLEURT			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.229	0.251	0.283	Base	0.255	0.348	0.298
ICL	0.185	0.235	0.258	ICL	0.202	0.302	0.248
CoT	0.362	0.442	0.338	CoT	0.445	0.474	0.395
CoT+ICL	0.357	0.444	0.257	CoT+ICL	0.329	0.467	0.356
RAG(Base)	0.227	0.266	0.285	RAG(Base)	0.220	0.346	0.271
RAG(CoT)	0.342	0.449	0.289	RAG(CoT)	0.271	0.458	0.382
HellaSwag/BERTScore				MMLU/BERTScore			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.518	0.492	0.555	Base	0.546	0.624	0.575
ICL	0.336	0.440	0.395	ICL	0.522	0.558	0.478
CoT	0.871	0.935	0.861	CoT	0.968	0.973	0.935
CoT+ICL	0.838	0.938	0.778	CoT+ICL	0.918	0.972	0.924
RAG(Base)	0.482	0.537	0.541	RAG(Base)	0.507	0.610	0.519
RAG(CoT)	0.847	0.940	0.825	RAG(CoT)	0.888	0.964	0.924

표 3. TruthfulQA/Winogrande 데이터셋의 성능 지표
Table 3. Performance Metrics for TruthfulQA and Winogrande Datasets

TruthfulQA/BLEU				Winogrande/BLEU			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	4.819	6.304	6.667	Base	3.967	2.490	0.541
ICL	2.901	2.088	9.657	ICL	6.242	15.209	0.276
CoT	8.911	9.828	13.184	CoT	0.000	0.000	2.924
CoT+ICL	1.467	12.530	15.071	CoT+ICL	0.000	0.000	10.175
RAG(Base)	8.363	9.123	11.759	RAG(Base)	2.648	13.179	0.280
RAG(CoT)	8.835	21.232	17.870	RAG(CoT)	0.000	0.000	7.085
TruthfulQA/ROUGE				Winogrande/ROUGE			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.149	0.199	0.243	Base	0.388	0.548	0.391
ICL	0.113	0.096	0.245	ICL	0.401	0.544	0.239
CoT	0.192	0.234	0.290	CoT	0.225	0.685	0.548
CoT+ICL	0.067	0.257	0.349	CoT+ICL	0.551	0.661	0.574
RAG(Base)	0.206	0.217	0.288	RAG(Base)	0.367	0.506	0.233
RAG(CoT)	0.201	0.341	0.398	RAG(CoT)	0.469	0.621	0.497
TruthfulQA/METEOR				Winogrande/METEOR			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.118	0.148	0.249	Base	0.204	0.291	0.224
ICL	0.069	0.066	0.254	ICL	0.209	0.282	0.157
CoT	0.161	0.189	0.273	CoT	0.114	0.350	0.287
CoT+ICL	0.049	0.217	0.352	CoT+ICL	0.283	0.339	0.297
RAG(Base)	0.155	0.177	0.293	RAG(Base)	0.188	0.269	0.149
RAG(CoT)	0.163	0.314	0.399	RAG(CoT)	0.242	0.319	0.259
TruthfulQA/BLEURT				Winogrande/BLEURT			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.233	0.278	0.344	Base	0.386	0.511	0.384
ICL	0.187	0.208	0.347	ICL	0.401	0.510	0.255
CoT	0.257	0.321	0.380	CoT	0.247	0.611	0.515
CoT+ICL	0.152	0.344	0.439	CoT+ICL	0.521	0.597	0.541
RAG(Base)	0.258	0.320	0.379	RAG(Base)	0.366	0.482	0.267
RAG(CoT)	0.279	0.421	0.467	RAG(CoT)	0.459	0.573	0.487
TruthfulQA/BERTScore				Winogrande/BERTScore			
Method	Llama	Gemma	Mistral	Method	Llama	Gemma	Mistral
Base	0.427	0.489	0.558	Base	0.779	0.835	0.723
ICL	0.410	0.444	0.546	ICL	0.783	0.837	0.608
CoT	0.473	0.527	0.594	CoT	0.700	0.897	0.843
CoT+ICL	0.322	0.559	0.624	CoT+ICL	0.844	0.887	0.859
RAG(Base)	0.470	0.521	0.588	RAG(Base)	0.749	0.824	0.613
RAG(CoT)	0.479	0.625	0.672	RAG(CoT)	0.815	0.876	0.825

표 4. Llama3 모델 기반 데이터셋 별 성능 지표 별 최고 성능의 프롬프트 엔지니어링
Table 4. Optimal Prompt Engineering for Achieving Peak Performance Metrics Across Datasets Based on the Llama3 Model

Llama3	ARC	GSM8k	HellaSwag	MMLU	TruthfulQA	Winogrande
데이터셋 목적	복잡한 과학문제	수학적 추론능력	문장 연속성, 상식추론 능력	과학, 인문, 전문분야 다지선다형	진실성, 정보검색, 응답 생성	맥락적, 상식적 추론 능력
BLEU	Base{2.170}	CoT{11.799}	CoT+ICL{2.835}	CoT+ICL{8.709}	CoT{8.911}	ICL{6.242}
ROUGE	CoT{0.699}	CoT+ICL{0.437}	CoT{0.448}	CoT{0.591}	RAG(Base){0.206}	CoT+ICL{0.551}
METEOR	CoT{0.350}	CoT+ICL{0.270}	CoT{0.224}	CoT{0.296}	RAG(CoT){0.163}	CoT+ICL{0.283}
BLEURT	CoT{0.480}	CoT{0.415}	CoT{0.362}	CoT{0.445}	RAG(CoT){0.279}	CoT+ICL{0.521}
BERTScore	CoT{0.976}	CoT{0.824}	CoT{0.871}	CoT{0.968}	RAG(CoT){0.479}	CoT+ICL{0.844}
Best	CoT	CoT	CoT	CoT	RAG(CoT)	CoT+ICL

표 5. Gemma2 모델 기반 데이터셋 별 성능 지표 별 최고 성능의 프롬프트 엔지니어링
Table 5. Prompt Engineering for Optimal Performance Metrics Across Datasets Using the Gemma2 Model

Gemma2	ARC	GSM8k	HellaSwag	MMLU	TruthfulQA	Winogrande
데이터셋 목적	복잡한 과학문제	수학적 추론능력	문장 연속성, 상식추론 능력	과학, 인문, 전문분야 다지선다형	진실성, 정보검색, 응답 생성	맥락적, 상식적 추론 능력
BLEU	CoT{14.649}	RAG(CoT){8.503}	RAG(Base){1.047}	CoT+ICL{9.535}	RAG(CoT){21.232}	ICL{15.209}
ROUGE	CoT{0.836}	Base{0.451}	RAG(CoT){0.725}	CoT{0.673}	RAG(CoT){0.341}	CoT{0.685}
METEOR	CoT{0.418}	RAG(CoT){0.250}	RAG(CoT){0.363}	CoT{0.336}	RAG(CoT){0.314}	CoT{0.350}
BLEURT	CoT{0.534}, CoT+ICL{0.534}	RAG(CoT){0.412}	RAG(CoT){0.449}	CoT{0.474}	RAG(CoT){0.421}	CoT{0.611}
BERTScore	CoT+ICL{0.985}	CoT+ICL{0.859}	RAG(CoT){0.940}	CoT{0.973}	RAG(CoT){0.625}	CoT{0.897}
Best	CoT	RAG(CoT)	RAG(CoT)	CoT	RAG(CoT)	CoT

표 6. Mistral 모델 기반 데이터셋 별 성능 지표 별 최고 성능의 프롬프트 엔지니어링
Table 6. Prompt Engineering for Optimal Performance Metrics Across Datasets Using the Mistral Model

Mistral	ARC	GSM8k	HellaSwag	MMLU	TruthfulQA	Winogrande
데이터셋 목적	복잡한 과학문제	수학적 추론능력	문장 연속성, 상식추론 능력	과학, 인문, 전문분야 다지선다형	진실성, 정보검색, 응답 생성	맥락적, 상식적 추론 능력
BLEU	RAG(Base){1.485}	CoT{8.285}	RAG(Base){1.420}	Base{1.863}	RAG(CoT){17.870}	CoT+ICL{10.175}
ROUGE	CoT+ICL{0.527}	Base{0.358}	CoT{0.389}	CoT{0.477}	RAG(CoT){0.398}	CoT+ICL{0.574}
METEOR	CoT+ICL{0.264}	CoT{0.221}	CoT{0.196}	CoT{0.241}	RAG(CoT){0.399}	CoT+ICL{0.297}
BLEURT	CoT+ICL{0.418}, RAG(CoT){0.418}	CoT{0.384}	CoT{0.338}	CoT{0.395}	RAG(CoT){0.467}	CoT+ICL{0.541}
BERTScore	RAG(CoT){0.943}	Base{0.747}	CoT{0.861}	CoT{0.935}	RAG(CoT){0.672}	CoT+ICL{0.859}
Best	CoT+ICL	CoT	CoT	CoT	RAG(CoT)	CoT+ICL

표 7. ARC Dataset(Gemma2)
Table 7. ARC Dataset(Gemma2)

ARC Dataset(Gemma2)	
Label	covered by water
BASE	underwater
ICL	underwater
CoT	covered by water
CoT+ICL	covered by water
RAG(BASE)	underwater
RAG(CoT)	covered by water

또한, LLM 모델 중에는 거의 모든 성능 metric에서 Gemma2가 가장 좋은 결과를 갖는다는 것을 알 수 있다. 또한, 주목할만한 사실은, Mistral은 7B 모델임에도 불구하고 Llama3 9B보다 더 좋은 성능을 보이며, Gemma2 9B와도 필적할 만한 성능을 보인다는 점이다. 이를 통해 Gemma2는 높은 성능을 필요로 하는 어플리케이션에 더 적합하며, Mistral은 경량화 성능을 필요로 하는 On-Device AI에 더 적합하다고 할 수 있다.

LLM 데이터셋에 대한 성격을 잘 모를 경우, 즉, 일반적인 경우를 생각할 때, 표 4, 5, 6의 최고 프롬프트 엔지니어링 기법 출현 수를 합산한다면 일반적인 데이터셋에서 가장 효과적으로 쓰일 수 있는 프롬프트 엔지니어링 기법을 알 수 있을 것이다. 실험 결과, CoT (Chain of Thought) 기법이 총 14회로 가장 우수한 성능을 보인다. CoT+ICL 기법과 RAG(CoT) 기법은 각각 4회로 특정 데이터셋에서 좋은 성능을 보였으나, CoT 기법만큼의 일관된 성능을 보이지는 않았다. 이와 같은 결과는 CoT 기법이 다양한 데이터셋에서 복잡한 문제 해결 및 추론 능력을 향상시키는 데 효과적임을 시사한다. 반면, RAG(CoT) 기법은 사실 기반 응답 생성에서 우수한 성능을 보여, 특정한 응답 생성 환경에 효과적일 수 있음을 나타낸다.

이제 가장 좋은 성능을 보이는 Gemma2에 대한 프롬프트 엔지니어링 적용 예시를 각 데이터셋에 대해 표 7-12에 걸쳐 보여주었다.

표 7는 CoT 기법을 포함하는 CoT, CoT+ICL, RAG(CoT)는 의미적으로도 유사하며, 응답 추론 형식도 유사하게 답변을 하는 것을 보여준다. BASE, ICL, RAG(BASE) 경우 의미적으로는 유사하나 응답 형식

표 8. HellaSwag Dataset(Gemma2)
Table 8. HellaSwag Dataset(Gemma2)

HellaSwag Dataset(Gemma2)	
Label	lifts it up to his chest and pauses.
BASE	is likely about to lift weights.
ICL	This document doesn't say.
CoT	lifts it up to his chest and pauses.
CoT+ICL	lifts it up to his chest and pauses.
RAG(BASE)	lifts it
RAG(CoT)	lifts it up to his chest and pauses.

표 9. MMLU Dataset(Gemma2)
Table 9. MMLU Dataset(Gemma2)

MMLU Dataset(Gemma2)	
Label	the total welfare is maximized.
BASE	This statement is generally considered true.
ICL	It allocates resources efficiently
CoT	the total welfare is maximized.
CoT+ICL	the total welfare is maximized.
RAG(BASE)	It allocates resources efficiently and promotes innovation.
RAG(CoT)	the total welfare is maximized.

표 10. TruthfulQA Dataset(Gemma2)
Table 10. TruthfulQA Dataset(Gemma2)

TruthfulQA Dataset(Gemma2)	
Label	Light behaves as both a particle and a wave
BASE	Both
ICL	The Bible doesn't specify
CoT	Light exhibits properties of both particles and waves.
CoT+ICL	Light is both a particle and a wave.
RAG(BASE)	This document does not contain information about the nature of light.
RAG(CoT)	Light can be both a particle and a wave.

표 11. WinograndeDataset(Gemma2)
Table 11. WinograndeDataset(Gemma2)

WinograndeDataset(Gemma2)	
Label	house
BASE	trainer
ICL	hammer
CoT	training
CoT+ICL	training
RAG(BASE)	trainer
RAG(CoT)	house

표 12. GSM8k Dataset(Gemma2)
Table 12. GSM8k Dataset(Gemma2)

GSM8k Dataset(Gemma2)	
Label	Carol and Jennifer have already collected 20 + 44 signatures = <<20+44=64>>64 signatures.\nSince their goal is 100, they need to collect 100 - 64 signatures. 100 - 64 = <<100-64=36>>36 signatures\n#### 36
BASE	They need 100 total signatures and currently have 20 + 44 = 64 signatures.
ICL	They need 100 signatures total and Carol has 20 and Jennifer has 44 for a total of 64. 100-64=36.
CoT	They need 100 total signatures and currently have 20 + 44 = 64 signatures. Therefore, they need 100 - 64 = 36 more signatures.
CoT+ICL	They need 100 signatures total and Carol has 20 and Jennifer has 44 for a total of 64. 100 - 64 = 36.
RAG(BASE)	'They need 100 signatures total and currently have 64 (20 + 44). Therefore, they need 36 more signatures (100 - 64 = 36).': 'Carol and Jennifer have already collected 20 + 44 signatures = <<20+44=64>>64 signatures.\nSince their goal is 100, they need to collect 100 - 64 signatures. 100 - 64 = <<100-64=36>>36 signatures\n#### 36
RAG(CoT)	They need 100 total signatures and currently have 20 + 44 = 64 signatures. Therefore, they need 100 - 64 = 36 more signatures.': 'Carol and Jennifer have already collected 20 + 44 signatures = <<20+44=64>>64 signatures.\nSince their goal is 100, they need to collect 100 - 64 signatures. 100 - 64 = <<100-64=36>>36 signatures\n#### 36

을 맞추지 못한 것을 알 수 있다.

표 8을 통해 CoT 기법을 포함하는 CoT, CoT+ICL, RAG(CoT)는 정확한 응답 추론을 하는 것을 확인할 수 있다. BASE와 RAG(BASE) 경우 유사하게 답변을 했으나 의미적으로 비교했을 때는 적합하지 않은 것을 확인할 수 있다. ICL은 문에 대해 주어진 예제를 이해하지 못해 응답을 추론하지 못하는 결과를 보였다.

표 9를 통해 CoT 기법을 포함하는 CoT, CoT+ICL, RAG(CoT)가 정확한 응답을 추론하는 것을 확인할 수 있다. BASE 경우 잘못된 응답을 추론했다. ICL과 RAG(BASE) 같은 경우 정답과 비슷한 응답을 추론하는 것을 확인할 수 있다.

표 10을 통해 CoT 기법을 포함하는 CoT, CoT+ICL, RAG(CoT)는 의미적으로도 유사하며, 응답 추론 형식도 유사하게 답변을 하는 것을 확인할 수 있다. BASE 경우 의미적으로는 유사하나 응답 추론 형식을 맞추지 못하였다. ICL과 RAG(BASE) 같은 경우 질문에 대해 주어진 예제를 이해하지 못해 단어가 명시되지 않았다, 예제에 관련 내용이 포함되지 않았다 같이 응답을 추론하지 못하는 결과를 보였다.

표 11을 통해 RAG(CoT)만 정확한 응답을 추론하는 것을 확인할 수 있다. 그 외의 프롬프트 엔지니어링 기법들은 틀린 응답을 추론하는 것을 확인할 수 있다

표 12를 통해 RAG 기반 기법인 RAG(BASE), RAG(CoT)들이 정확한 답을 추론했을 뿐만 아니라 응답 형식도 맞춰서 추론하는 것을 확인할 수 있다. CoT, CoT+ICL, ICL은 정확한 답을 추론했으나 응답 형식은 다른 것을 확인했다. BASE 경우 틀린 답을 추론하는 것을 확인할 수 있다.

Ⅶ. 결 론

본 논문은 6가지 벤치마크 데이터셋을 통해 5가지 프롬프트 엔지니어링의 성능을 비교하였다. 실험을 통해 각 데이터셋의 성격, 목적, 특성에 따라 가장 적합한 프롬프트 엔지니어링 기법이 달라질 수 있음을 알 수 있었다. 또한, Gemma2가 일반적으로 성능이 가장 우수함을 알 수 있었고, 제약된 자원 환경하에서는 Mistral이 좋은 대안이 될 수 있음을 알 수 있었다.

적용 예시를 분석하였을 때 Base, ICL, RAG(BASE) 비해 CoT, CoT+ICL, RAG(CoT)가 응답 형식과 가장 유사한 응답을 추론하는 것을 확인할 수 있었고 RAG나 ICL 같은 경우 예제를 이해하지 못해 응답하지 못하거나 질문의 범위를 벗어나는 답변을 진행하는 경우를 발견할 수 있었다.

References

[1] J. Shin, "Development and commercialization challenges of large language models," *J. KICS*, vol. 40, no. 6, pp. 3-11, 2015.

[2] Google DeepMind, "Gemini: A family of highly capable multimodal models," *arXiv. https://arxiv.org/abs/2312.11805*, 2023.

[3] OpenAI, et al., "GPT-4 technical report," *arXiv preprint arXiv.2303.08774*, 2023. (https://doi.org/10.48550/arXiv.2303.08774).

[4] A. Chowdhery, et al., "PaLM: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS 2017*, 2017. (https://doi.org/10.48550/arXiv.1706.03762)

[6] Meta, "*Llama 3. Meta AI Blog*," https://ai.meta.com/blog/meta-llama-3, 2024.

[7] Gemini Team Google, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023. (https://doi.org/10.48550/arXiv.2312.11805)

[8] AliTech, "Pali Gemma and Gemma 2: Google breakthrough in vision-language models," *AliTech*, 2024

[9] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, and G. Lample, "Mistral 7B," *arXiv preprint arXiv.2310.06825*, 2023. (https://doi.org/10.48550/arXiv.2310.06825)

[10] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de Las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. Le Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mixtral of experts," *arXiv preprint arXiv.2401.04088*, 2024. (https://doi.org/10.48550/arXiv.2401.04088)

[11] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z Sui, "A survey on in-context learning," *arXiv preprint arXiv.2301.00234*, 2023. (https://doi.org/10.48550/arXiv.2301.00234)

[12] L. Ouyang, et al., "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022. (https://doi.org/10.48550/arXiv.2203.02155)

[13] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's verify step by step," *arXiv preprint arXiv:2305.20050,* 2023. (https://doi.org/10.48550/arXiv.2305.20050)

[14] F. Liu and M. Lapata, "Text summarization with pretrained encoders," *arXiv preprint arXiv:1908.08345,* 2019.

[15] L. Reynolds and K. McDonell, "Prompt programming for large language models: Beyond the few-shot paradigm," *arXiv preprint arXiv:2102.07350*, 2021.

[16] Y. Gu, L. Dong, F. Wei, and M. Huang, "Pre-training to learn in context," in *Proc. 61st Annu. Meeting of the Assoc. for Comput. Linguistics* (vol. 1: Long Papers), pp. 4849-4870, Toronto, Canada, 2023.

[17] S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi, "MetaICL: Learning to learn in context," in *Proc. 2022 Conf. North Am. Chapter of the Assoc. Comput. Linguistics: Human Language Technol.*, pp. 2791-2809, Seattle, USA, 2022.

[18] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" in *Proc. Deep Learn. Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learn. Architectures (DeeLIO@ACL 2022)*, pp. 100-114, Dublin, Ireland and Online, May 2022.

[19] H. J. Kim, H. Cho, J. Kim, T. Kim, K. M. Yoo, and S. Lee, "Self-generated in-context learning: Leveraging autoregressive language models as a demonstration generator," *arXiv preprint arXiv:2206.08082,* 2022.

[20] J. Wei, et al., "Chain of thought prompting elicits reasoning in large language models," *arXiv preprint arXiv:2201.11903*, 2022.

[21] S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das, "A comprehensive survey of hallucination mitigation techniques in large language models," *arXiv preprint arXiv:*

2401.01313, Jan. 2024. (https://arxiv.org/pdf/2401.01313)

[22] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *arXiv preprint arXiv:2005.11401v4*, 2021.

[23] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, and J. Gao, "Check your facts and try again: Improving large language models with external knowledge and automated feedback," *arXiv preprint arXiv:2302.12813*, 2023.

[24] N. Varshney, W. Yao, H. Zhang, J. Chen, and D. Yu, "A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation," *arXiv: 2307.03987,* 2023.

[25] L. Gao, Z. Dai, P. Pasupat, A. Chen, A. T. Chaganty, Y. Fan, V. Zhao, N. Lao, H. Lee, D.-C. Juan, et al., "Rarr: Researching and revising what language models say, using language models," in *Proc. 61st Annu. Meeting of the Assoc. for Computat. Linguistics* (vol. 1: Long Papers)," pp. 16477-16508, 2023.

[26] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-verification reduces hallucination in large language models," *arXiv preprint arXiv:2309.11495*, 2023.

[27] M. Boratko, et al., "A systematic classification of knowledge, reasoning, and context within the ARC dataset," *arXiv preprint arXiv: 1806.00358*, 2018.

[28] R. Zellers, et al., "HellaSwag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[29] D. Hendrycks, et al., "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.0330*, 2021.

[30] S. Lin, et al., "TruthfulQA: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

[31] K. Sakaguchi, et al., "Winogrande: An adversarial winograd schema challenge at scale," *arXiv preprint arXiv:1907.10641*, 2020.

[32] K. Cobbe, et al., "Training verifiers to solve math word problems," *arXiv preprint arXiv: 2105.04187*, 2021.

[33] D. Steele and L. Specia, "Vis-eval metric viewer: A visualisation tool for inspecting and evaluating metric scores of machine translation output," *North Am. Chapter of the Assoc. for Comput. Linguistics*, 2018.

[34] S. Sreelekha and P. Bhattacharyya, "Indowordnet's help in Indian language machine translation," *AI & Soc.*, 2017.

[35] M. Popovic and H. Ney, "Syntax-oriented evaluation measures for machine translation output," WMT@EACL, 2009.

[36] T. Sellam, D. Das, and A. P. Parikh, "BLEURT: Learning robust metrics for text generation," *Annu. Meeting of the Assoc. Comput. Linguistics*, 2020.

[37] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," *Int. Conf. Learn. Representations*, 2020.

손 민 준 (Minjun Son)

2024년 2월 : 동서울대학교 전자 공학과 졸업

2024년 3월~현재 : 성균관대학교 메타바이오헬스학과 석사과정

<관심분야> 인공지능, LLM

이 성 진 (Sungjin Lee)

2011년 8월 : 연세대학교 전자전기공학과 박사

2012년 9월~2016년 7월 : 삼성전자 DMC연구소 책임연구원

2016년 7월~2024.12월 : 동서울대학교 전자공학과 교수

2025년 1월~현재: 순천향대학교 스마트자동차학과 교수

<관심분야> 자율주행, 멀티모달러닝, 이동통신