Smart Media Journal / Vol.14, No.6 / ISSN:2287-1322
https://dx.doi.org/10.30693/SMJ.2025.14.6.50

소형 언어 모델의 특정 도메인에서의 파인튜닝과 RAG의 성능 비교 - 질의응답과 감정분석을 중심으로

(Comparison of Fine-tuning vs. RAG in sLLM - Focusing on Machine Reading Comprehension and Sentiment Analysis)

김가현*, 김도국**
(Ga Hyeon Kim, Doguk Kim)

요 약

거대 언어 모델에서 일부 특정 도메인에 대한 적응력 부족을 향상시킬 수 있는 기법인 파인튜닝(Fine-tuning)과 검색 증강 생성(RAG)의 장단점을 비교하는 연구가 활발히 진행되고 있다. 그러나 이러한 기존 연구는 질문과 정답의 출처 문서 쌍이 이미 입력된 환경에서도 여전히 RAG가 파인튜닝에 비해 특정 도메인의 성능을 손쉽게 향상시킬 수 있는 잠재력이 있는지에 대한 비교가 없었다. 이에 본 연구에서는 출처 문서가 이미 입력된 환경에서 소형 언어 모델(sLLM)을 활용한 한국어 기계독해(KorQuAD)와 감성 분석 작업(NSMC)에서 파인튜닝과 RAG의 성능을 비교하였다. 실험 결과, RAG는 기계독해에서 10.2%, 감성분석에서 32.3%의 성능 향상을 보였으며, 파인튜닝과 병행할 경우 각각 11.5%와 41.9%의 성능이 향상되었다. 이를 바탕으로, 컴퓨팅 자원이 부족하면 RAG를, 충분하다면 두 기법의 상호 보완 방식으로 성능을 향상시키는 방안을 제시하였다.

■ 중심어 : 검색 증강 생성 ; 파인튜닝 ; 거대 언어 모델 ; 생성형 AI

Abstract

Studies have actively compared fine-tuning and retrieval-augmented generation (RAG) to enhance the adaptability of large language models (LLMs) to specific domains. Although question and source document pairs are already available in certain environments, no comparative study has assessed whether RAG has the potential to improve domain-specific performance more easily than fine-tuning in such settings. In this study, we compare the performance of fine-tuning and RAG for Korean machine reading comprehension (KorQuAD) and sentiment analysis (NSMC) tasks using small language models (sLLM). Our results show that RAG improved performance by 10.2% on KorQuAD and 32.3% on NSMC. when RAG and fine-tuning were combined, performance improved by 11.5% and 41.9%, respectively. Our results indicate that RAG is advantageous when resources are limited, while a complementary use of both methods can outperform fine-tuning when sufficient resources are available.

■ keywords : RAG ; Fine-tuning ; LLM ; Generative Artificial Intelligence

Ⅰ. 서 론

최근의 GPT-4와 같은 강력한 거대 언어 모델(Large Language Models, LLM)의 발전은 NLP 작업의 성능을 크게 향상시켜 다양한 응용 분야에서 주목받고 있다[1]. 거대 언어 모델은 광범위한 도메인에서 다양한 작업을 수행할 수 있는 잠재력을 가지고 있지만, 특정 작업에 특화되도록 세밀하게 조정하지 않으면 적응력 부족으로 인해 성능이 저하될 수 있다. 따라서, 자원이 부족한 도메인에서 거대 언어 모델을 적용하려면 맞춤화가 필수적이다.

이를 극복하기 위해서는 파인튜닝(Fine-Tuning) 또는 검색 증강 생성(Retrieval-Augmented Generation, RAG)이라는 두 가지 주요 접근법이 필수적이다. 파인튜닝은 거대 언어 모델의 큰 규모로 인해 특정 작

* 준회원, 인하대학교 전기컴퓨터공학과
** 정회원, 인하대학교 전기컴퓨터공학과
본 연구는 인하대학교의 지원, 과학기술정보통신부 및 정보통신기획평가원의 생성AI선도인재양성사업(IITP-2025-RS-2024-00360227)의 지원, 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(NO.RS-2022-00155915, 인공지능융합혁신인재양성(인하대학교))을 받아 수행된 연구임.
접수일자 : 2025년 03월 19일
수정일자 : 2025년 04월 11일
게재확정일 : 2025년 05월 12일
교신저자 : 김도국 e-mail : dgkim@inha.ac.kr

업에 맞게 조정하는 데 많은 시간과 자원이 소모되는 단점이 있다. 그에 비해, RAG는 적은 컴퓨팅 자원을 사용한다. 언어 모델을 특정 작업에 맞게 맞춤화하려는 관심이 높아짐에 따라, 대부분의 실무 환경과 같이 제한된 컴퓨팅 자원을 가진 상황에서는 공정한 조건에서 두 기법을 비교해 주어진 상황과 도메인에 맞춰 둘 중 최적의 방안을 선택하는 것이 중요하다.

뿐만 아니라, RAG는 검색한 문서를 토대로 추론하여 답변을 생성하는 능력도 중요하다. 답변을 추론 가능한 질문의 출처 문서(골드 문서) 뿐 아니라, 질문과 관련된 RAG 문서 예시를 더 추가하여 입력하면 모델의 성능이 상당히 개선된다는 연구가 존재한다[2].

그러나 파인튜닝과 RAG를 비교한 기존 연구들은 RAG가 관련 문서를 검색해 찾는 것이 중요하여, 찾는 것에 실패한다면 추론 정보가 부족해지도록 환경을 설정하였다[2-5]. 즉, 질문과 정답의 출처 문서 쌍이 이미 입력된 환경에서도 여전히 RAG가 파인튜닝에 비해 특정 도메인의 성능을 손쉽게 향상시킬 수 있는 잠재력이 있는지에 대한 비교가 없었다. 특히, 전통적인 기계독해와 같이 입력문에 정답이 포함된 문서가 항상 제공되는 작업에 대해서는 두 기술의 성능 비교 연구가 없었다. 또한, 주로 영어와 같은 주요 언어에 관련 연구가 집중되어 있기에 한국어에 초점을 맞춘 연구는 아직 부족한 실정이다.

이에 본 연구에서는 RAG 프롬프트 생성 시 골드 문서와 RAG 문서를 함께 사용하는 방식을 통해 파인튜닝과 RAG 두 기법의 성능을 비교하였다. 이를 통해 독해 및 추론이 중요한 작업에서 두 기법의 성능 차이를 더욱 명확하게 비교할 수 있다.

본 연구에서는 RAG와 파인튜닝 기법을 각각 적용하여 감성분석과 기계독해 작업에서의 성능을 비교 평가하고자 한다. 본 논문의 기여점은 다음과 같다.

1. 기계독해와 감성분석 작업에서 거대 언어 모델의 성능을 최적화하기 위한 3가지 접근 방식을 비교한다: (1) RAG, (2) 파인튜닝, (3) 파인튜닝된 모델에 RAG 단계를 추가하는 방법.

2. 기계독해와 감성분석 작업에서 RAG 기법이 추가적인 파인튜닝 없이도 성능을 개선할 수 있음을 입증하여, 컴퓨팅 자원이 제한된 상황에서 대안적 접근법으로서의 가능성을 제시하였다.

3. RAG를 활용한 퓨샷 프롬프트에서 각 작업의 최적 예시 개수를 도출하였다.

그 결과, RAG는 적은 컴퓨팅 자원으로도 성능을 개선할 수 있어 대안적 접근법으로서의 가능성을 확인하였다. 그리고 컴퓨팅 자원이 충분한 상황이라면 파인튜닝 기법이 RAG보다 일관적으로 더 높은 성능 향상을 보였다.

특히, 본 연구에서는 RAG와 파인튜닝 두 기법을 병행하여 실험하였다. 그 결과, 특정 도메인에 언어 모델을 적용해야 하는데 자원이 부족하다면 RAG를, 자원이 충분하다면 이 두 접근법을 병행하는 것이 상호보완적으로 최적의 전략이 될 수 있음을 확인하였다.

본 연구는 각 도메인의 특성에 따라 거대 언어 모델에서 RAG와 파인튜닝의 효과를 평가하고, NLP 연구와 실무에서 적용 가능한 선택 전략을 제안한다.

Ⅱ. 관련 연구

거대 언어 모델은 광범위한 일반 지식을 보유하고 있지만, 특정 작업에서 뛰어난 성능을 발휘하기 위해서는 파인튜닝 또는 RAG의 사용이 필수적이다. 이 두 개의 기법은 각기 다른 장단점이 있어서 비교하는 연구가 진행되고 있다.

1. 파인튜닝

전통적으로 사용되는 풀 파인튜닝(Full Fine-tuning) 방법은 모델 전체를 학습하는 데 많은 컴퓨팅 자원을 소모하기 때문에, 최근에는 더 효율적인 파라미터 효율적 파인튜닝(PEFT, Parameter-Efficient Fine-Tuning) 방법이 주로 사용된다.

LoRA(Low-Rank Adaptation)는 파라미터 공간

을 낮은 차원으로 변환하여 일부 매개변수만 추가로 학습함으로써 적은 비용으로 성능을 향상시키는 방법이다. QLoRA[6]는 이를 더욱 발전시켜 4비트 양자화와 페이징 기법을 통해 메모리 효율성을 극대화하였다. 이를 통해 개인용 컴퓨터에서도 거대 언어 모델을 파인튜닝하여 기존에 비해 더 적은 컴퓨팅 자원으로도 고성능을 구현할 수 있다.

2. RAG

거대 언어 모델은 강력한 능력을 가지고 있음에도 불구하고, 환각(hallucination) 문제, 불투명하고 추적할 수 없는 추론 과정 등의 한계를 가지고 있다. 이러한 문제를 해결하기 위해 RAG가 제안되었다.

RAG의 주요 과정은 두 가지 단계로 구성된다[5]:

1. Retrieval(문서 검색): 사용자가 입력한 쿼리를 벡터화하여 문서와의 유사도를 계산한 후, 유사한 상위 K개의 문서를 검색하는 단계이다.

2. Generation(응답 생성): 검색된 문서를 입력 프롬프트에 추가해 언어 모델이 최종 응답을 생성하는 단계이다.

특히, 질의응답(QA) 작업에서 RAG의 성능을 최적화하기 위해 DPR(Dense Passage Retrieval)과 BM25 알고리즘을 결합하여 사용하는 접근이 주목받아 왔다. [7-9]. DPR은 문서 내의 의미적 유사성을 기반으로 관련 문서를 검색하는 반면, BM25는 텍스트 기반의 전통적인 역문서 빈도(TF-IDF) 방법을 사용한다. 두 기법을 결합한 하이브리드 검색을 사용함으로써, 의미적 일치와 텍스트 기반 검색의 강점을 모두 활용하여 보다 정확하고 포괄적인 문서 검색이 가능해진다.

3. RAG와 파인튜닝의 비교

RAG와 파인튜닝 모두 언어 모델의 성능을 향상시키는 강력한 기술이지만, 근본적으로 다른 기술이며 최적화 과정에서 서로 다른 요구사항을 충족한다. 따라서, 제한된 컴퓨팅 자원을 가진 상황에서 성능을 위해 둘 중 가장 적합한 기법을 선택할 때는 둘의 장단점을 고려해야 한다.

사전 학습된 언어 모델을 특정 작업에 맞게 정제하려는 관심이 높아짐에 따라, 공정한 조건에서 파인튜닝과 RAG 전략을 비교하는 것이 점점 더 중요해지고 있다. 최근 연구들은 다양한 분야에서 파인튜닝과 RAG의 성능을 비교해 왔다.

예를 들어, 농업 및 지리 도메인에서 긴 질문에 대한 답변에서 파인튜닝과 RAG를 비교한 연구[4]에서는 RAG는 새로운 지식의 경우 농장 데이터 해석과 같이 데이터가 맥락적으로 관련성이 있는 경우에 매우 효과적임을 보였다. 반면, 파인튜닝은 작물 수확량 예측 개선이나 기상 패턴에 따른 일정 최적화와 같이 RAG로는 성능이 보완되지 않는 도메인에서도 학습 기반으로 모델의 성능을 향상시킬 수 있었다. 결론적으로 RAG와 파인튜닝은 모두 효과적인 기술이지만 특정 작업, 데이터 세트의 특성 및 크기, 모델 개발에 사용할 수 있는 자원에 따라 달라짐을 보였다.

해부학, 천문학, 대학 생물학, 선사 시대와 같은 특수 영역에서의 다중 선택 질문에 대해 비지도 학습 파인튜닝과 RAG의 성능을 평가한 연구[5]에서는 RAG가 훈련 중에 접한 기존 지식과 완전히 새로운 지식 모두에서 비지도 파인튜닝보다 일관되게 우수한 성과를 보였다.

오픈 도메인 질의응답 작업인 POPQA 데이터셋의 덜 알려진 지식에 대해 파인튜닝과 RAG를 비교한 연구[3]에서는, 그 결과 파인튜닝은 모든 엔터티에 대해 일관된 성능 개선을 보여주며, 가장 인기 있는 엔터티와 가장 인기 없는 엔터티 모두의 성능을 크게 개선하며, RAG가 그 외의 부분에서 우세하다고 보여줬다. 또한, RAG가 파인튜닝과 결합될 때 더욱 효과적임을 보여주었다.

RAG는 주어진 문서를 토대로 추론하여 독해, 분석하여 생성하는 성능 또한 중요하다. 또한 RAG에서는 골드 문서를 더욱 보강하여 해당 성능의 향상이 가능하므로, 파인튜닝과의 비교가 필요하다. 그

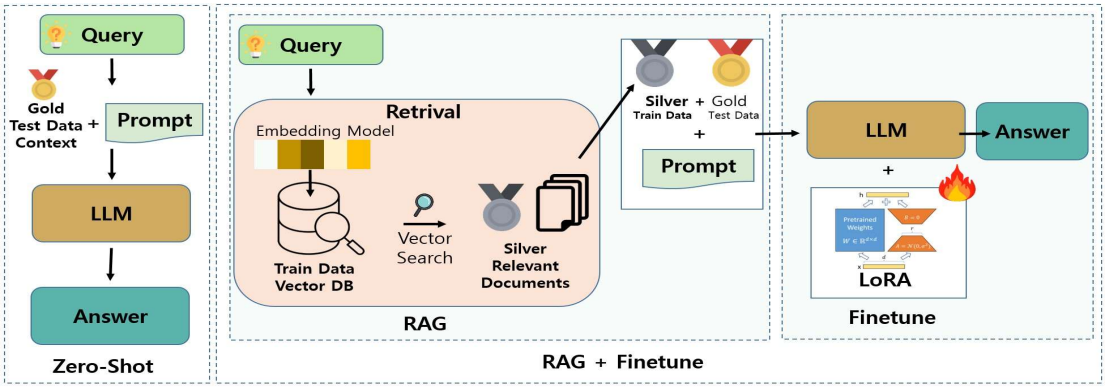

그림 1 실험에 사용한 파인튜닝과 RAG 과정
Figure 1. Fine-tuning and RAG processes used in the experiments

러나, 기존 연구들은 RAG의 외부 지식 검색에만 초점을 맞추어, 질문과 정답의 출처 문서 쌍이 이미 입력된 환경에서도 여전히 RAG가 파인튜닝에 비해 특정 도메인의 성능을 손쉽게 향상시킬 수 있는 잠재력이 있는지 비교하지 않았다.

본 연구에서는 전통적인 기계독해와 같이 골드 문서가 주어진 상황에서 RAG를 이용해 추론 능력을 향상시킬 수 있는 방안을 평가하고, 이를 파인튜닝과 비교하고자 한다.

Ⅲ. 제안 방법

RAG와 파인튜닝은 각각 고유한 장단점이 있다. 이러한 차이 때문에 두 방법을 적절한 상황에 활용하는 것이 중요하다. 또한, 두 방법을 결합하면 각 접근법의 장점을 극대화할 수 있음을 제안한다.

본 연구는 두 접근법의 성능을 비교하기 위해 (그림 1)과 같이 세 가지 접근 방식을 비교한다: (1) RAG, (2) 파인튜닝, (3) 파인튜닝된 모델에 RAG 단계를 추가하는 방법. 공정하게 평가하기 위해, 프롬프트에 제공된 문단은 세 가지로 변형되었다: (a) 골드 문단, (b) 골드 문단에 무작위 문단을 추가(퓨샷 프롬프트), (c) 골드 문단에 실버 문단을 추가.

이러한 접근 방식과 문맥 설정의 조합으로 총 6개의 고유한 구성(표 3)이 만들어졌다. 각 구성에 대한 자세한 설명은 아래에 설명한다.

1. RAG 접근법

골드 문단(Gold Passage)은 질문과 짝을 이루는 핵심 문단으로써, 해당 문단만을 이용해서도 질문에 대한 정답 구간을 추론할 수 있다[9]. 기계독해 작업의 경우, 원래 주어지는 질문의 출처 문단이 골드 문단으로 간주된다. 반면, 실버 문단(Silver Passage)은 RAG 검색을 통해 얻은 관련성이 높은 상위 문단으로, 골드 문단에 비해 직접적으로 답변을 포함하지 않을 수 있다[10].

본 실험에서는 학습용 데이터셋 중에서 질문과 관련된 문단을 검색한 후, 입력 프롬프트에 퓨샷 프롬프트로 k개의 검색된 관련 실버 문단 예시를 추가 제공한다.

RAG는 질문에 골드 문단이 포함된 작업에도, 추가로 검색한 해당 질문과 관련된 질문, 정답 쌍의 실버 문단 예시를 추가하면 모델의 성능을 향상시켰다는 연구가 존재한다[2]. 그러나 이전 연구에서는 파인튜닝과 비교한 RAG는 골드 문단이 주어지지 않은 상태에서 관련된 실버 문단을 검색하여 찾는 것이 중요했다. 반면 추론이 중요한 작업들의 단일 골드 문단이 이미 주어지는 경우, RAG와 파인튜닝 성능 비교 연구가 없었다.

질의응답(QA) 작업은 RAG 기법 중 BM25와 DPR을 결합한 하이브리드 검색기가 주목받아왔으며[7-9], 파인튜닝과 RAG를 비교하는 연구[8]에서도 해당 검색기를 사용하여, 본 연구에 동일한 방법을 사용했다.

아래에 BM25와 DPR의 하이브리드 계산 식을 나타내었다.

$$Hybrid(q,d) = \alpha \cdot BM25(q,d) + (1-\alpha) \cdot DPR(q,d) \quad (1)$$

실험 시 BM25 상위 10개, DPR 상위 10개 점수를 위 수식을 적용해 α=0.5의 비율로 결합했다. 여기서 BM25 점수는 쿼리와 문서의 단어 빈도 사이의 내적으로 나타낼 수 있다. 아래에 BM25 계산 식을 나타내었다.

$$BM25(q,d) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{(f(q_i,d) \cdot (k_1+1))}{f(q_i,d) + k_1 \cdot (1-b+b \cdot \frac{|d|}{avgdl})} \quad (2)$$

여기서 f(q_i, d)는 문서 d에서 쿼리의 i번째 토큰 q_i의 빈도, IDF(q_i)는 q_i의 역문서 빈도다. 실험 시 변수 k_1과 b는 k_1=1.5, b=0.75로 진행했다.

또한 DPR 점수는 이중 인코더(dual-encoder) 구조가 널리 사용되며, 임베딩 모델을 사용하여 질의 q와 패시지 p를 각각 벡터로 변환한 후, 내적을 사용하여 유사도를 측정한다. 임베딩 모델은 HuggingFace Models 라이브러리의 thingsu/koDPR_question모델과 thingsu/koDPR_context모델을 사용했다. 아래에 DPR 계산식을 나타내었다.

$$DPR(q,p) = E_Q(q)^T E_P(p) \quad (3)$$

답변 생성 시에는 하이브리드 검색기를 사용해 학습용 데이터셋 중에서 질문에 대해 가장 관련성 높은 문단을 검색한 후, 검색된 실버 문단을 입력 프롬프트에 k개의 추가 예시로 제공하는 퓨샷 프롬프트를 적용하였다.

2. 파인튜닝 접근법

풀 파인튜닝 방법은 많은 컴퓨팅 자원을 소모하기 때문에, 효율적인 학습을 위해 본 연구에서는 Hugging Face PEFT 라이브러리의 QLoRA 방식을 사용해 파인튜닝했다.

3. 파인튜닝 + RAG 접근법

사전에 파인튜닝된 모델 프롬프트에 RAG를 추가해, k개의 관련성 높은 문단을 검색한 후, 입력 프롬프트에 검색된 관련 예시 k개를 제공했다.

Ⅳ. 실험

1. 데이터 셋 및 실험 환경

본 연구에서 제안한 방법을 실험하기 위하여, 골드 문서가 항상 제공되는 대표적 한국어 기계독해 및 감성분석 벤치마크인 KorQuAD와 NSMC 데이터셋을 사용하여 실험을 설계하였다. 다음으로, 본 실험에 사용된 데이터셋, 평가 지표, 그리고 실험 결과의 세부 사항을 아래에 설명하였다.

가. 데이터셋

KorQuAD 1.0[11]: 주어진 문단에서 정답을 추론하는 기계독해 작업을 위해 만들어진 데이터셋이다. Wikipedia article에 대해 학습용 데이터셋 60,407개, 평가용 데이터셋 5,774개의 질의응답 쌍으로 이루어져 있다. KorQuAD 2.0[12]에는 표나 그래프 등 비정형 정보가 있어, 본 연구에서는 순수 텍스트 기반의 성능을 분석하기 위해 KorQuAD 1.0을 사용하였다.

NSMC[13]: 영화 리뷰 감성분석 데이터셋이다. 총 20만 개로 해당 리뷰가 긍정인 경우 1, 부정인 경우 0을 표시한 레이블로 구성되어져 있다. NSMC 데이터셋은 랜덤 샘플링을 통해 파인튜닝과 검색을 위한 학습용 72,000개, 평가용 4,000개를 구성했다. 또한, 긍정 및 부정의 비율이 1:1이 되도록 데이터셋을 구성했다.

나. 평가 지표

NSMC는 데이터가 균등하게 구성했으므로 정확도(Accuracy)라는 분류 성능평가 지표를 통해 비교한다.

KorQuAD 1.0은 예측한 답변을 주어진 정답 답변과 비교하여 성능 평가 점수인 EM(Exact Match)와 F1을 구한다. EM은 모델이 정답을 변화 없이 정확히 동일하게 맞추면 1, 그렇지 않으면 0으로 측정한다. F1 점수는 모델이 낸 답안과 정답을 음절 단위로 비교해서 정답과 겹치는 부분을 고려한다. 생성형 언어모델은 특성상 답변의 길이 차이 때문에 기계독해에서 EM이 F1 점수에 비해 낮게 나올 수 있다.

다. 모델 학습

NSMC 데이터셋은 기존 연구[14]에서 10epoch 학습한 사례를 바탕으로, 본 연구에서는 과적합 방지를 고려하여 2 epoch로 설정하였다. KorQuAD 데이터셋은 이전 연구[15]에서 1 epoch만으로도 우수한 성능을 보였던 점을 고려하여 동일하게 1 epoch로 진행했다. 하이퍼파라미터는 KorQuAD의 이전 연구를 기반으로 배치 크기 8, Rank 32, LoRa-Alpha 16로 했으며 일부 부분은 학습률은 5e-4, LoRa-Dropout 0.2로 일부 조정했다. 학습 시 표 2에 있는 프롬프트에서 예시(Example_k)를 제외한 제로샷 프롬프트를 사용하여 가의 학습용 데이터셋으로 학습을 수행하였다. 모든 실험은 NVIDIA Geforce RTX 4090 GPU에서 진행되었다. 이를 위해 사용되는 템플릿은 표 2와 같다.

라. 비교 모델

실험에 사용한 소형 언어 모델(Small Language Models, sLLM)은 Meta의 Llama-3-8B 모델과 beomi의 Llama-3-Open-Ko-8B 모델을 사용했다. Llama-3-Open-Ko는 Llama-3-8B를 기반으로 한국어 사전학습되어 한국어 문장 생성에 우수한 성능을 보인다.

마. 프롬프트 구성

프롬프트는 본 논문의 RAG에서 유일하게 변화하는 요소다. RAG는 검색된 k개의 관련 예시를, 파인튜닝은 랜덤한 k개의 학습 데이터를 퓨샷 프롬프트에 추가해 비교했다. 이를 위해, 본 실험에서는 다음 세 가지 RAG, 파인튜닝, 그리고 두 기법을 결합한 방법에 퓨샷과 제로샷 프롬프트를 각각 적용하여 비교했다.

이를 위해 사용되는 템플릿은 표 2와 같다.

표 1. 모델 설정 및 프롬프트 구성
Table 1. model settings and prompt usage

	파인튜닝	RAG
적용 기법	QLoRA	BM25 + DPR
데이터셋	학습 데이터 KorQuAD: 60,407개 NSMC: 랜덤 72,000개	검색 데이터베이스 KorQuAD: 60,407개 NSMC: 랜덤 72,000개
방법	학습 하이퍼파라미터 NSMC: 2 epoch, Korquad: 1 epoch, 배치 크기: 8, 학습률: 5e-4, Rank: 32, LoRa-Alpha: 16, LoRa-Dropout: 0.2	BM25 상위 10개 + DPR 상위 10개 점수 결합 후 NSMC: 상위 3개 예시, KorQuAD: 상위 8개 예시를 프롬프트에 적용
프롬프트 유형	제로샷 프롬프트, 랜덤 문서 예시 퓨샷 프롬프트	관련된 문서 예시 퓨샷 프롬프트

표 2. 퓨샷 프롬프트를 이용한 추론 예제
Table 2. Inference examples using few-shot prompts

KorQuAD	
프롬프트	너는 주어진 CONTEXT에서 Question에 가장 잘 답하는 짧은 단어를 추출 후 그대로 단어를 ANSWER하는 챗봇이야. 주의할 점은 다음과 같아. 찾은 0ANSWER할 때 문장으로 ANSWER하지 말고 1단어로 ANSWER해, 질문 내용을 쓰지 마, '은', '는', '~다.', '~이다.', '~것', '?'으로 문장 끝내지 마. RAG_Example_1: RAG_Context_1: {... 정부는 재원확보를 위해 723년(요로 7년)에는 삼세일신법을 시행하여 개간을 장려하였다.} RAG_Question: {나라시대에 정부가 삼세일신법을 시행한 때는 언제인가?} RAG_Answer_1: {723년} 다음 CONTEXT에서 Question에 가장 잘 답하는 짧은 단어를 추출해서 한 단어로 최대한 짧고 간결하게 답하도록 해. Context:{복을 하고 난 직후에 내시가 왕이 입고 있던 옷을 재빨리 지붕 아래로 ... 그 옷을 덮고 5일간 살아나기를 기다렸다.} Question:{복의식 직후 왕의 옷을 아래에 있는 내시에게 던지면 곧장 죽은 왕의 몸 위에 덮고 며칠간을 기다렸는가?} Answer:
답변	5일간, 국무장관 등 추출한 답변
NSMC	
프롬프트	너는 주어진 CONTEXT를 분석하여 긍정적인 감정을 표현하는지, 부정적인 감정을 표현하는지 분류하는 챗봇이야. RAG_Example_1: CONTEXT_1: [{내용하며배우들연기력하며두루두루매력넘침}] = **분석한 감정**: [{긍정적}] 이제 다음 CONTEXT 대괄호 안에 있는 영화 리뷰의 감정을 분석하고, 그것이 긍정적인지, 부정적인지를 판단한 후, "긍정적", "부정적" 둘 중에서 하나로 감정 레이블을 분류하세요. 요구사항: "긍정적", "부정적" 레이블을 분류하십시오. "긍정적" 또는 "부정적"레이블만으로 분석한 감정을 반환하고 다른 텍스트는 반환하지 마세요. ##다음 분석해서 답할 CONTEXT: [{액션이 없는데도 재미 있는 몇안되는 영화}] = **분석한 감정:**
답변	긍정적 / 부정적

표 3 파인튜닝과 RAG 성능 비교

Table 3 Performance comparison of Fine-tuning and RAG

Model	Llama3			Llama3-Ko		
Dataset	KorQuAD		NSMC	KorQuAD		NSMC
	EM	F1	Acc	EM	F1	Acc
0-shot	0.575	0.771	0.530	0.416	0.620	0.360
few-shot prompt	0.621	0.859	0.826	0.483	0.816	0.855
RAG (few-shot prompt)	0.695	0.873	0.853	0.573	0.845	0.844
finetune (0-shot prompt)	0.684	0.883	0.558	0.709	0.886	0.920
finetune (few-shot prompt)	0.695	0.882	0.907	0.719	0.892	**0.927**
finetune + RAG	**0.718**	**0.886**	**0.922**	**0.742**	**0.893**	0.916

2. 파인튜닝과 RAG 비교 실험 결과

본 연구에서는 기계독해와 감성분석 작업에 대해 RAG와 파인튜닝의 영향을 네 가지 구성으로 평가하였다: (1) 파인튜닝과 RAG 모두 사용하지 않음, (2) RAG, (3) 파인튜닝, (4) 파인튜닝된 소형 언어 모델에 RAG 단계를 추가하는 방법. 실험 결과는 표 3에서 확인할 수 있다.

Llama-3의 경우 RAG는 KorQuAD에서 10.2%, NSMC에서 32.3%의 성능 향상을 보였으며, 파인튜닝과 RAG를 병행할 경우 각각 11.5%와 41.9%의 성능 향상이 나타났다.

파인튜닝이 RAG에 비해 일관되게 더 높은 성능 향상을 보였으며, 각 모델별로 KorQuAD는 88.2%, 90.7%, NSMC는 89.2%, 92.7%의 성능을 보였다. KorQuAD 작업의 경우 파인튜닝이 제로샷 프롬프트만으로도 RAG보다 성능이 높다. 이는 파인튜닝이 주어진 문서의 세부 정보를 정확하게 찾아내고, 출력 형식을 맞춤화하는 데 효과적이기 때문으로 해석된다.

RAG는 랜덤 퓨샷 프롬프트보다 일관되게 베이스 모델의 성능을 개선하지만, 파인튜닝 모델의 랜덤 퓨샷 프롬프트만큼 효과적이지는 못했다. 하지만, RAG 성능은 Llama-3의 NSMC작업의 경우 파인튜닝에 제로샷 프롬프트를 적용한 것보다 29% 더 높았고, KorQuAD 작업의 경우에도 0.1%밖에 성능이 크게 뒤처지지 않았다. 특히, NSMC작업의 경우 정확도가 제로샷 프롬프트에 비해 Llama-3의 경우 32%, Llama-3-Open-Ko의 경우 48% 증가하는 것을 확인할 수 있었다. 제로샷 프롬프트의 파인튜닝 모델은 긍정과 부정을 포함하지 않은 채 감정 단어만 응답하는 경우가 많았으며, 이는 형식 불일치로 인해 오답 처리되는 원인이 되었다. 반면 RAG는 이러한 형식 오류가 적고, 응답의 일관성이 높았다. 영화 리뷰 분류라는 구체적인 도메인에 한정된 데이터를 다루고 있기 때문에, 검색된 외부 예시가 맥락적으로 관련성이 있어 모델의 판단을 더욱 직접적으로 도와주었음을 확인할 수 있었다.

이처럼 RAG가 적은 컴퓨팅 자원으로도 성능을 개선했으므로 적은 컴퓨팅 자원을 사용하고자 하면 대안적 접근법으로 고려될 수 있음을 확인하였다.

파인튜닝과 RAG를 통합하면 Llama-3-Open-Ko의 NSMC 작업을 제외한 모든 작업에서 일관되게 더 높은 성능을 발휘해 도메인별 작업에서 이 두 가지 접근 방식의 상호 보완적임을 보였다. 이러한 결과는 RAG가 전반적인 정확도를 향상시키는 동시에 이를 파인튜닝과 결합하면 도메인별 최적의 성능을 달성하기 위해 함께 사용할 때 가장 좋은 결과를 낼 수 있음을 시사한다. 또한, 파인튜닝과 RAG 방식이 상호 보완적이며 이 둘을 병행하는 것이 최적의 전략이 될 수 있음을 제안한다.

3. 예시 수(k)에 걸친 성능 비교 실험 결과

표 4는 Llama-3의 KorQuAD, NSMC 데이터의 RAG 적용 시 예시 수 k에 걸친 성능 영향을 보여준다. 최적의 예시 수는 하위 작업마다 다르다.

프롬프트에 한 개의 예시만 추가해도 모델의 성능이 상당히 향상되었다. 예시 수가 늘어날수록 성능

표 4. 퓨샷 프롬프트의 k샷에 따른 성능 비교
Table 4. Selection of the number of shots in the prompt

	KorQuAD		NSMC
k	EM	F1	Acc
0	0.575	0.771	0.530
1	0.639	0.823	0.819
2	0.703	0.862	0.826
3	**0.695**	**0.873**	0.835
4	0.629	0.849	0.842
6	0.619	0.858	0.852
8	0.611	0.850	**0.853**
12	0.622	0.851	0.852

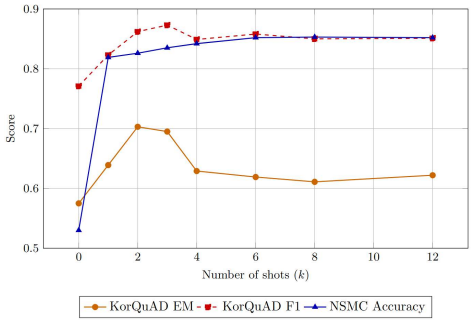

그림 2. 퓨샷 프롬프트의 k샷에 따른 성능 비교
Figure 2. Selection of the number of shots in the prompt

이 증가하다가, KorQuAD 데이터는 k=3, NSMC 데이터는 k=8 이후에는 오히려 성능이 하락했다. 이는 프롬프트 예제의 수가 너무 많을 시 모델이 혼란을 야기하거나 관련 없는 정보에 초점을 맞추게 되어, 쿼리의 관련 세부 사항에 제대로 집중할 수 없기 때문이다[16].

결과에서 KorQuAD 데이터의 경우 k=3, NSMC 데이터의 경우 k=8에 가장 높음을 보여주기 때문에, 제안된 퓨샷 프롬프트 및 RAG 실험에서는 데이터별로 각각 예시 수를 적용했다.

V. 결론

골드 문서를 보강하여 RAG 예시 문서를 추가하면 독해, 분석 작업의 성능을 향상시킬 수 있음에도 불구하고, 기존 연구들은 RAG의 외부 지식 검색에만 초점을 맞추어 연구하였다. 이에 따라, 골드 문서가 포함된 작업에 대한 RAG 기반의 보강과 파인튜닝의 성능 비교 연구가 필요했음에도 아직 관련된 연구는 전무한 실정이다. 이에 본 연구에서는 감성분석과 기계독해 작업에 대해 파인튜닝과 RAG를 비교 평가함으로써, 향후 연구와 실무에 중요한 방향을 제시한다.

실험 결과, 파인튜닝이 RAG보다 일관된 더 좋은 성능 향상을 가져왔다. 반면, RAG는 파인튜닝 없이도 적은 컴퓨팅 자원으로도 성능을 개선할 수 있어, 적은 컴퓨팅 자원을 사용하고자 하면 효과적으로 사용될 수 있다. 이를 통해 RAG 방식이 대안적 접근법으로서의 가능성을 확인하였다. 특히, 감성분석과 같이 주제가 한정적인 태스크나, 한국어 사전학습이 충분히 되지 않은 모델일 때 RAG는 효과적인 해결책임을 확인했다.

결과적으로, 파인튜닝과 RAG 방식이 상호 보완적이며, 이 둘을 병행하는 것이 최적의 전략으로 확인되었다. 자원이 부족하다면 RAG를, 충분하다면 이 두 접근법을 병행하는 것이 바람직한 선택이 될 수 있다.

본 연구에서 프롬프트 형식을 고정해 사용하였지만, 다른 프롬프트 형식이 결과에 어떤 영향을 미칠 수 있는지에 대한 추가 분석도 향후 과제로 남아 있다.

후속 연구로는 소프트 프롬프트(Soft Prompt) 방식을 도입하여 더 효율적이고 효과적인 프롬프트 자동화 기법을 활용하거나, 최신 RAG 기법인 Advanced RAG를 사용하여 비교할 필요가 있다. 또한, 다른 모델을 사용한 실험을 통해 더 다양한 접근 방식을 탐색할 필요가 있다.

REFERENCES

[1] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann et al., "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research,* Vol. 24, No. 240, pp. 1-113, 2023.

[2] M. Suppa, D. Skala, D. Jass, S. Sucik, A. Svec, and P. Hraska, "Bryndza at ClimateActivism 2024: Stance, Target and Hate Event Detection

via Retrieval-Augmented GPT-4 and LLaMA," in Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-Political Events from Text (CASE 2024), St. Julians, Malta, Association for Computational Linguistics, 2024, pp. 166-177.

[3] H. Soudani, E. Kanoulas, and F. Hasibi, "Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge," arXiv preprint, arXiv:2403.01432, 2024.

[4] M. A. de L. Balaguer, V. Benara, R. L. de F. Cunha, R. de M. Estevão Filho, T. Hendry, D. Holstein, J. Marsman, N. Mecklenburg, S. Malvar, L. O. Nunes, R. Padilha, M. Sharp, B. Silva, S. Sharma, V. Aski, and R. Chandra, "RAG vs Fine-Tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture," arXiv preprint, arXiv:2401.08406, 2024.

[5] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-tuning or Retrieval? Comparing Knowledge Injection in LLMs," arXiv preprint, arXiv:2312.05934, 2023.

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," arXiv preprint, arXiv:2305.14314, 2023.

[7] A. Asai, S. Min, Z. Zhong, and D. Chen, "Retrieval-based language models and applications," in Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Toronto, Canada, Association for Computational Linguistics, pp. 41-46, 2023.

[8] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval,* Vol. 3, No. 4, pp. 333-389, 2009.

[9] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), *Association for Computational Linguistics,* pp. 6769-6781, 2020.

[10] H. Phan, A. Acharya, S. Chaturvedi, S. Sharma, M. Parker, D. Nally, A. Jannesari, K. Pazdernik, M. Halappanavar, S. Munikoti, and S. Horawalavithana, "RAG vs. Long Context: Examining Frontier Large Language Models for Environmental Review Document Comprehension," arXiv preprint, arXiv:2407.07321, 2024.

[11] S. Lim, M. Kim, and J. Lee, "KorQuAD1.0: Korean QA Dataset for Machine Reading Comprehension," arXiv preprint, arXiv:1909.07005, 2019.

[12] 김영민, 임승영, 이현정, 박소윤, and 김명지, "KorQuAD 2.0: 웹문서 기계독해를 위한 한국어 질의응답 데이터셋," *정보과학회논문지,* 제47권, 제6호, 577-586쪽, 2020년

[13] *Naver Sentiment Movie Corpus*, https://github.com/e9t/nsmc, (accessed Apr., 25, 2025).

[14] J. M. Kim, Y.-J. Lee, H.-J. Choi, and S. Jung, "LANGALIGN: Enhancing Non-English Language Models via Cross-Lingual Embedding Alignment," arXiv preprint, arXiv:2503.18603, 2025.

[15] P. Devine, "ALoFTRAG: Automatic Local Fine Tuning for Retrieval Augmented Generation," arXiv preprint, arXiv:2501.11929, 2025.

[16] H. Bahak, F. Taheri, Z. Zojaji, and A. Kazemi, "Evaluating ChatGPT as a Question Answering System: A Comprehensive Analysis and Comparison with Existing Models," arXiv preprint, arXiv:2312.07592, 2023.

저 자 소 개

김가현(준회원)

2023년 인하대학교 컴퓨터공학과 학사 졸업.
2023년 ~ 현재 인하대학교 인공지능학과 석사 과정

<주관심분야 : 자연어 처리, 질문 응답 , 텍스트 데이터 증강>

김도국(정회원)

2012년 한국과학기술원 컴퓨터공학과 졸업 (학사)
2014년 한국과학기술원 정보보호대학원 졸업 (석사)
2018년 한국과학기술원 전산학부 졸업 (박사)
2018년~2020년 하나금융융합기술원 책임
2020년~2021년 카카오엔터프라이즈 책임
2021년~현재 인하대학교 인공지능공학과 조교수
<주관심분야 : 시계열예측, 데이터증강, 딥러닝자동화>