한국어 문서 요약 효율화를 위한 Gemma2 모델 미세조정 및 프롬프트 튜닝

(Fine-Tuning and Prompt Tuning of the Gemma2 Model for Efficient Korean Document Summarization)

김승주*, 정세훈, 심춘보*****

(Seung Ju Kin, Se Hoon Jung, Chun Bo Shim)

요 약

본 논문에서는 Gemma2-9B-IT 모델의 한국어 문서 요약 성능을 향상하기 위한 미세조정 및 프롬프트 튜닝 방법을 제안한다. Gemma2 모델은 주로 영어 중심으로 사전학습돼 한국어와 같은 비영어권 언어에 대한 처리 능력이 제한적이다. 이를 개선하기 위해 AI Hub의 한국어 '문서 요약 텍스트' 데이터셋을 전처리 후 미세조정하고, 메모리 효율적인 학습을 위해 Unsloth 프레임워크와 LoRA 기반 파라미터 효율적 미세조정을 적용했다. 또한 "간결하고 핵심적인 내용만 포함해 자연스러운 요약문을 생성해"라는 최적화된 프롬프트를 적용해 요약 성능을 향상했다. 본 논문에서 제안한 모델의 성능을 평가한 결과 기본 모델 대비 BERTScore에서 1.68%, RDASS에서 34%의 성능 향상을 보였다. 제거 연구를 통해 미세조정이 20.18%, 프롬프트 튜닝이 9.39% 성능 향상에 기여도가 있음을 확인했으며, 두 기법 간 일부 중복되는 성능 향상 영역이 존재함을 발견했다. 특히 원문과 참조 요약문 모두에 대한 의미적 유사성 측면에서 큰 개선을 보였으나, 응답 시간은 56% 증가했다. 본 연구는 대형 언어 모델의 특정 언어 및 도메인 적응을 위한 효과적인 방법론을 제시하고, 미세조정과 프롬프트 튜닝의 개별 및 복합 효과를 체계적으로 분석해 향후 다양한 언어와 작업에 대한 최적화 전략 수립에 중요한 통찰을 제공한다.

■ 중심어 : 한국어 문서 요약 ; 대형 언어 모델 ; 파라미터 효율적 미세조정 ; 프롬프트 튜닝 ; 의미론적 평가 지표

Abstract

This paper proposes fine-tuning and prompt tuning methods to enhance the Korean document summarization performance of the Gemma2-9B-IT model. The Gemma2 model was primarily pre-trained on English data, resulting in limited performance on non-English languages such as Korean. To address this limitation, we preprocess the Korean 'Document Summarization Text' dataset provided by AI Hub and apply parameter-efficient fine-tuning using the Unsloth library and LoRA to enable memory-efficient training. Additionally, we design an optimized prompt: "Generate a concise and natural summary that includes only the essential information," to further improve summarization quality. For performance evaluation, we utilize BERTScore and RDASS, which rely on semantic embeddings. Experimental results show that the proposed approach outperforms the base model by 1.68% in BERTScore and 34% in RDASS. An ablation study reveals that fine-tuning contributes 20.18% and prompt tuning 9.39% to the overall improvement, with some overlapping effects observed between the two techniques. Notably, the model demonstrates substantial improvement in semantic similarity between the generated summary and the reference summary, although the response time increases by 56%. This study presents an effective methodology for adapting large language models to specific languages and domains, and provides systematic insights into the individual and combined effects of fine-tuning and prompt tuning, offering valuable guidance for optimization strategies in various languages and tasks.

■ keywords : Korean Document Summarization ; LLM ; LoRA ; Prompt Tuning ; Semantic Evaluation

Ⅰ. 서 론

인공지능을 활용한 문서 요약(Text Summariz

* 정회원, 국립순천대학교 스마트융합학부 멀티미디어공학전공
** 정회원, 국립순천대학교 컴퓨터공학과
*** 정회원, 국립순천대학교 인공지능공학부
This work was supported by a Research promotion program of SCNU
접수일자 : 2025년 05월 27일
게재확정일 : 2025년 07월 04일
교신저자 : 심춘보 e-mail : cbsim@scnu.ac.kr

ation) 방식에는 크게 추출적 요약(Extractive Summarization) 방식과 추상적 요약(Abstractive Summarization) 방식이 있다[1-2]. 추출적 요약 방식은 원문에서 중요한 문장이나 구절을 그대로 추출해 요약문을 생성하는 방식으로 간단하고 효율적이지만, 단어 간 연결이 부자연스러운 경우가 발생할 문제가 있다. 추상적 요약 방식은 원문의 의미를 이해하고 이를 바탕으로 새로운 문장을 생성해 요약문을 작성하는 방식으로 더 일관된 결과를 제공하지만, 계산 비용과 복잡도가 크다. 대형 언어 모델(Large Language Model, LLM)은 주로 추상적 요약 방식을 사용하는데, 대표적으로 GPT(Generative Pre-trained Transformer)는 입력 텍스트의 맥락을 이해하고 자연스러운 문장을 생성하는 추상적 요약을 주로 수행한다[3].

본 논문에서는 구글의 Gemma2 모델을 사용한다. Gemma2는 Text-to-Text 기반의 디코더(Decoder) 전용 대형 언어 모델로, 원문의 맥락과 흐름을 파악해 자연스럽고 부드러운 문장을 생성하며, 텍스트 생성, 요약, 번역, 질의응답 등 다양한 자연어처리(Natural Language Processing, NLP) 분야에서 응용할 수 있다[4]. 하지만, Gemma2는 주로 영어 웹 문서, 수학 및 코드 데이터로 구성된 토큰(Token)을 통해 훈련돼 한국어 처리 능력이 상대적으로 취약하다는 한계가 있다. 이에 Gemma2 모델에 한국어 문서 요약 데이터셋을 추가로 학습해 한국어 처리 능력과 요약 성능을 향상하도록 한다. 데이터셋은 AI Hub의 한국어 '문서 요약 텍스트' 데이터[5]를 전처리해 사용한다. 또한, 모델과 작업에 맞는 프롬프트(Prompt)를 활용해 한국어 문서 요약 성능을 향상하도록 한다. 모델 미세조정(Fine-Tuning) 시 메모리 효율적인 학습을 위해 Unsloth[6] 프레임워크(Framework)를 활용한다. 이를 통해 부족한 하드웨어 환경에서도 모델을 효율적으로 미세조정 하도록 한다.

본 논문의 구성은 다음과 같다. 2장은 메모리 효율적인 미세조정과 프롬프트 엔지니어링(Prompt Engineering) 관련 연구를 소개하고, 각 관련 연구가 본 연구에서 어떻게 활용되는지 기술한다. 3장에서는 제안하는 한국어 문서 요약 효율화를 위한 데이터 처리 방법, 미세조정 기법, 프롬프트 설계, 성능 평가 방법을 기술한다. 4장과 5장에서는 각각 실험 환경, 실험 결과 그리고 결론 및 기대 효과, 향후 연구에 관해 기술한다.

Ⅱ. 관련 연구

1. LoRA(Low-Rank Adaptaion)

LoRA[7]는 대형 언어 모델 및 기타 딥러닝 모델의 파라미터를 효율적으로 미세조정(Parameter-Efficient Fine-Tuning, PEFT)[8]하기 위해 제안된 방법론이다. 기존의 미세조정 방식은 전체 모델 파라미터(Parameter)를 업데이트해야 하므로, 계산 및 저장 비용이 많이 들고, 다양한 작업별로 각각의 전체 모델을 저장해야 하는 비효율성이 존재한다.

LoRA는 이러한 한계를 극복하기 위해, 사전학습(Pre-training)된 모델의 특정 선형 계층(Linear Layer)에 저랭크 행렬 분해(Low-Rank Matrix Decomposition)를 적용한다. 즉, 원래의 모델 파라미터는 고정(Frozen)하고, 추가로 매우 적은 수의 보정 파라미터만 학습한다. 따라서 실제로 업데이트되는 파라미터의 수가 기존 대비 대폭 감소하며, 이는 계산 효율성과 저장 효율성 모두를 크게 향상한다.

이 방식은 모델의 표현력을 유지하면서도, 미세조정 과정에서 추가되는 파라미터 수를 최소화할 수 있다는 이점을 갖는다. 또한 LoRA는 기존 사전학습 모델의 가중치를 변경하지 않았으므로, 여러 작업에 대해 독립적으로 미세조정된 추가 파라미터만을 저장 및 적용할 수 있다.

본 연구에서는 Unsloth 프레임워크와 LoRA를 활용해 미세조정 과정에서 연산 효율성과 자원 활용도를 극대화하고자 한다. Unsloth는 커널 최

적화, Flash Attention 등의 연산 기법을 통해 기존 미세조정 프레임워크 대비 훈련 속도와 메모리 효율성을 크게 향상한다.

따라서 LoRA의 파라미터 효율적 미세조정 기법을 Unsloth의 최적화된 연산 환경과 결합함으로써, 동일한 하드웨어 자원에서 더 빠르고 효율적인 모델 미세조정을 진행한다.

2. 프롬프트 엔지니어링

대형 언어 모델의 요약 성능을 높이기 위한 프롬프트 엔지니어링 연구의 대표적인 사례로 CoD(Chain-of-Density) 프롬프트[9]가 있다. 이는 요약문의 정보 밀도를 단계적으로 높이는 방식을 제안한다. CoD 프롬프트는 우선, 초기 요약에서 일부 핵심 개체(Entity)만 포함한 간결한 요약문을 생성한 뒤, 반복적으로 누락된 개체를 추가하면서 같은 길이를 유지하도록 유도한다. 이 과정을 5회 반복함으로써, 최종적으로 정보의 밀도가 높은 요약을 산출할 수 있다. [9]의 연구에서 CoD 프롬프트는 기존의 단순 요약 프롬프트에 비해 인간 평가 기준에서 25% 더 높은 선호도를 기록했으며, 키워드 누락률도 크게 감소하는 등 요약 품질 향상에 효과적임을 입증했다.

그러나 고도화된 프롬프트 설계는 몇 가지 한계를 동반한다. CoD 프롬프트와 같은 요약 기법은 반복적인 단계로 인해 토큰 소모량과 계산 비용이 많이 증가한다. 또한, 프롬프트가 지나치게 길거나 복잡할 경우, 대형 언어 모델의 어텐션 메커니즘(Attention Mechanism)이 분산돼 오히려 핵심 정보 전달력이 떨어질 수 있으며, 프롬프트 과적합(Prompt Overfitting) 현상이 발생할 수 있다[10-11].

본 연구에서는 이러한 한계를 극복하기 위해 프롬프트의 간결성과 성능 간의 균형을 맞춘 간략한 요약 요청 프롬프트를 설계하고 모델에 적용한다.

Ⅲ. 제안 방법

본 연구에서는 Google의 Gemma2-9B(Billion)-IT(Instruct) 모델에 프롬프트 엔지니어링과 한국어 문서 요약 데이터를 추가로 학습시키고, 메모리 효율적으로 미세조정해 모델의 한국어 문서 요약 성능을 향상하고자 한다. 모델 선정 이유는 다음과 같다.

Gemma2-9B-IT 모델은 Llama3-70B, GPT-3.5-175B와 같은 대형 언어 모델들보다 현저히 적은 매개변수를 가지고 있어 상대적으로 제한된 컴퓨팅 자원으로도 실험할 수 있다. 또한, Gemma2는 공식 문서[4]에 따르면 영어 중심의 데이터셋으로 사전학습 됐으며, 한국어와 같은 비영어권 언어에 대한 처리 능력이 상대적으로 제한적이다. Gemma2는 연구 및 상업적인 용도로 모델 가중치가 공개된 오픈 모델[4]로 다양한 프레임워크에서 미세조정하기 용이하다. 이러한 요소들을 종합적으로 고려할 때, Gemma2-9B-IT 모델은 제한된 컴퓨팅 자원 환경에서 특정 작업에 관한 미세조정 연구에 적합하다.

모델의 미세조정을 위한 데이터셋은 AI Hub의 한국어 '문서 요약 텍스트' 데이터[5]를 사용하며, 전처리를 진행한 후 사용한다. 메모리 효율적 미세조정을 위해 Unsloth 프레임워크와 LoRA를 사용한다. 미세조정 시 사용되는 프롬프트는 간결성과 성능 간의 균형을 맞춘 프롬프트를 사용한다. 모델의 성능 평가를 위한 지표로는 BERT Score(Bidirectional Encoder Representations from Transformers Score)[12], RDASS(Reference and Document Aware Semantic Score)[13]를 사용한다.

1. 데이터 전처리

Gemma2-9B-IT 모델의 미세조정을 위한 한국어 문서 요약 데이터셋을 구축하고자, AI Hub에서 제공하는 한국어 '문서 요약 텍스트' 데이터[5]를 기반으로 체계적인 전처리 및 품질 분석을 수행했다. 전처리를 위해 원본 JSON(Java Script Object Notation) 데이터를 통합하고, 모델 학

습에 적합한 Annotation Format인 'text'-'target' 쌍을 구성했다. 각 문서의 본문은 문단 내 문장 단위를 공백 기준으로 연결해 하나의 연속적인 입력 텍스트로 구성했다. 요약은 생성 요약문으로 선택했다. 불필요하거나 잡음이 될 수 있는 특수 문자, 공백과 같은 요소들을 제거하고, 'text'-'target' 쌍을 이루지 못하는 비정상 데이터를 삭제해 데이터를 정제했다. 이후 학습 효율성과 품질 향상을 위해 입력 텍스트는 토큰 수 기준으로 300 이상 1,024 이하, 요약문은 10 이상 256 이하의 조건으로 필터링(Filtering)했다. 전처리 후 데이터 분포는 그림 1과 같다.

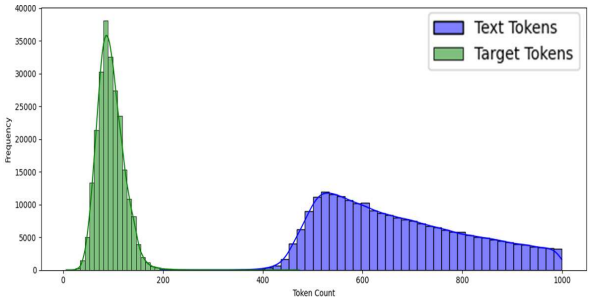

그림 1. 전처리 후 데이터 분포

토큰 수 계산에는 추후 학습 대상 모델과의 호환성을 고려해 사용하는 모델인 Gemma2-9B-IT의 토크나이저(Tokenizer)를 사용했다.

2. 미세조정

본 연구에서는 Gemma2-9B-IT 모델을 기반으로 한국어 문서 요약 작업에 특화된 미세조정을 수행했다. 모델은 다양한 작업에 대한 범용성을 갖추고 있지만 한국어 문서 요약과 같은 특정 언어 및 도메인(Domain)에 대해 최적의 성능을 발휘하기 위해서는 추가적인 미세조정이 필요하다. 모델 미세조정 과정에서 가장 핵심적으로 적용된 기법은 LoRA 기반의 파라미터 효율적 미세조정이다. LoRA는 대형 언어 모델의 전체 파라미터 모두를 학습하지 않고, 전체 파라미터 대비 극히 일부만을 업데이트하게 되며, GPU(Graphic Processing Unit) 메모리 사용량과 연산량을 크게 줄이면서도 모델의 성능 저하 없이 효율적인 미세조정이 가능하다.

모델 파라미터는 4비트(Bit) 양자화(Quantization)로 로드했다. 4비트 양자화는 모델 파라미터를 4비트 정밀도로 변환해 저장 및 연산함으로써, 동일한 하드웨어 환경에서 더 큰 모델을 학습하거나 더 빠른 연산을 가능하게 한다. QLoRA(Quantized LoRA)[14]는 4비트 양자화와 LoRA를 결합해, 대형 언어 모델의 성능 저하 없이 메모리 사용량을 크게 줄일 수 있음을 실험적으로 입증했다. 특히 NF4(Normal Float 4-bit) 데이터 타입은 기존 FP4(4-bit Float Point)보다 정밀도가 높아, 16비트 미세조정과 동등한 성능을 보인다.

3. 프롬프트 튜닝

모델이 입력을 요약 작업으로 인식하도록 유도하며, 프롬프트의 명확성과 일관성을 높이기 위해 최적의 프롬프트를 찾는 것은 중요하다. 최적의 프롬프트를 찾기 위해 우선, Google의 AI Studio Playground[15]에서 Gemma2 9B 모델에 다양한 프롬프트를 직접 입력해보고, 그에 대한 응답을 확인한다. 이를 통해 얻은 프롬프트 후보군에 대해 미세조정 시 각 입력 텍스트 앞에 프롬프트를 적용해 가며 최적의 프롬프트를 탐색했다. 실험을 통해 탐색한 최적의 프롬프트는 "간결하고 핵심적인 내용만 포함해 자연스러운 요약문을 생성해"이고, 이를 각 입력 텍스트 앞에 위치시켜 모델이 좋은 요약문을 만들도록 유도했다.

4. 성능 지표

본 연구에서는 한국어 문서 요약 모델의 성능 평가 지표로 BERTScore[12]와 RDASS[13]를 사용했다. 텍스트 요약 작업에서 대표적인 성능 지표로 사용되는 ROUGE(Recall-Oriented Understudy for Gisting Evaluation)[16]를 사용하지 않은 이유는 다음과 같다. ROUGE는 n-gram 기반의 자동 평가 지표로 생성 요약문과 참조 요약

문 간의 단어 혹은 어절의 중첩 빈도를 측정한다. ROUGE-1(Unigram), ROUGE-2(Bigram), ROUGE-L(Longest Common Sequence) 등이 널리 쓰이며, 영어권에서 추출적 요약이나 단순한 정보 요약 평가에는 여전히 표준적인 지표로 활용된다. 그러나 한국어와 같은 교착어에서는 단어의 형태가 다양하게 변형될 수 있고, 같은 의미를 가진 표현도 다양한 어휘와 조합으로 나타난다. 예를 들어 “책을 읽었다”와 “독서를 했다”라는 표현은 의미상으로 동일하지만, 단어 중첩이 없어 ROUGE 점수는 낮게 나온다. 이러한 현상은 추상적 요약에서 더욱 두드러진다. 추상적 요약은 원문의 단어를 그대로 복사하지 않고, 의미를 재구성하거나 유의어 및 동의어를 활용하기 때문에 n-gram 기반의 ROUGE로는 의미적 일치도를 제대로 평가할 수 없다는 한계가 있다[17].

이러한 한계점을 극복하기 위해 본 연구에서는 의미론적 임베딩(Embedding)을 활용하는 BERTScore와 RDASS를 사용했다. BERTScore는 사전학습된 BERT 계열 모델의 문장 임베딩을 활용해, 생성 요약문과 참조 요약문 간의 의미적 유사도를 측정한다. 단어가 다르더라도 문맥상 의미가 유사하면 높은 점수를 반환하므로, 추상적 요약의 평가에 매우 적합하다. 특히 본 연구에서는 XLM(Cross-lingual Language Model)-RoBERTa(Robustly Optimized BERT Approach)-Large와 같은 다국어 모델을 활용해, 한국어의 다양한 표현을 효과적으로 포착할 수 있도록 했다.

RDASS는 생성 요약문이 참조 요약문뿐만 아니라 원문 문서와도 얼마나 의미적으로 일치하는지를 동시에 평가한다. 이는 SBERT(Sentence-BERT) 기반의 임베딩을 활용해, 생성 요약문과 참조 요약문, 그리고 생성 요약문과 원문 문서 각각의 코사인 유사도(Cosine Similarity)를 산출하고, 그 평균을 최종 점수로 사용한다. 이 방식은 생성 요약문이 참조 요약문과 의미적으로 유사할 뿐 아니라, 원문 정보의 핵심을 정확히 반영하고 있는지까지 점검할 수 있다는 점에서, 단순 참조 일치에만 의존하는 기존 평가 지표의 한계를 극복한다.

이와 같이, 본 연구에서 ROUGE 대신 BERTScore와 RDASS를 요약 성능 평가 지표로 선택한 것은, 한국어와 같은 교착어에서 ROUGE의 평가 신뢰도가 낮고, 추상적 요약의 의미적 품질을 제대로 반영하지 못하는 한계에 기반한다. BERTScore와 RDASS는 의미론적 임베딩을 활용해, 생성 요약문의 실제 품질과 정보의 포함 여부를 더 정확히 평가할 수 있다.

Ⅳ. 실험 및 결과

본 연구에서 실험환경의 세부 설정은 표 1과 같다.

표 1. 실험환경 세부 설정

Component	Details
CPU	Intel Core i9-12900K
GPU	NVIDIA GeForce RTX 3090 Ti, 24GB
RAM	DDR4 128GB
Operating System	Ubuntu 20.04.6 LTS
CUDA Version	11.8.0
cuDNN Version	8.6.0
Programming Language	Python 3.11.8
Deep Learning Library	Pytorch 2.2.0

실험은 Google의 Gemma2-9B-IT 모델을 기반으로 했으며, Unsloth 프레임워크를 활용해 효율적인 미세조정을 진행했다. 데이터셋은 3장 제안 방법의 데이터 전처리 방법을 거쳐 구축한 데이터셋을 사용한다. 전체 데이터셋은 236,362개의 샘플(Sample)을 갖는다. 이를 Train 191,511

샘플, Validation 21,211 샘플, Test 23,640 샘플로 나누어 사용한다. 각각의 비율은 약 81:9:10이다. 데이터셋 파일은 JSON 형식이고, 각 샘플은 'text'-'target' 쌍으로 원문과 참조 생성 요약문으로 구성돼 있다.

모델의 한국어 문서 요약 작업을 위한 미세조정 과정에서 설정된 하이퍼파라미터(Hyperparameter)들은 모델의 성능과 효율성을 최적화하기 위해 선택됐다. 이러한 설정은 제안하는 모델의 메모리 효율성, 학습 안정성, 과적합 방지, 그리고 한국어 문서 요약이라는 특정 작업에 최적화된 성능을 달성하기 위한 전략을 반영한다.

모델을 효율적으로 학습시키기 위해 LoRA 기법을 적용한다. LoRA 하이퍼파라미터 설정값은 표 2와 같다.

표 2. LoRA 하이퍼파라미터 설정

Hyperparameter	Value
r	32
lora_alpha	64
target_modules	["q_proj", "k_proj", "v_proj", "o_proj"]
lora_dropout	0.1
use_rslora	True

랭크 파라미터 r을 32로 설정해 모델의 표현력을 확보하고, 이와 균형을 맞추기 위해 스케일링 파라미터 lora_alpha를 64, 즉 r 값의 2배로 설정해 안정적인 학습을 유도했다[7]. 특히 어텐션 메커니즘의 핵심 프로젝션 레이어(Projection Layer)들만을 타겟팅해 미세조정의 효율성을 극대화했으며, 랭크 안정화 LoRA인 RSLoRA(Rank-Stabilized LoRA)[18]를 통해 학습 안정성을 더욱 강화했다.

학습 과정에서의 효율성과 안정성을 위해 배치 크기와 그래디언트(Gradient) 누적 스텝을 적절히 조정했고 하이퍼파라미터와 설정값은 표 3과 같다.

표 3. 배치 및 학습 단계 하이퍼파라미터 설정

Hyperparameter	Value
per_device_train_batch_size	6
per_device_eval_batch_size	4
gradient_accumulation_steps	8
max_steps	12,000
eval_steps / save_steps	1,200

디바이스당 6개의 훈련 샘플과 8단계의 그래디언트 누적을 통해 실질적인 배치 크기는 48이돼, 메모리 사용량을 관리하면서도 충분한 학습 안정성을 확보했다. 또한 12,000 스텝이라는 충분한 학습 기간을 설정하고, 전체 학습 과정의 10% 간격인 1,200 스텝마다 모델 평가 및 저장을 진행했다.

과적합 방지 및 모델 성능 최적화를 위한 옵티마이저(Optimizer) 및 학습률(Learning Rate)은 표 4와 같다.

표 4. 옵티마이저 및 학습률 하이퍼파라미터 설정

Hyperparameter	Value
learning_rate	4e-6
warmup_ratio	0.1
optim	adamw_torch
weight_decay	0.05
lr_scheduler_type	cosine

모델의 안정적인 미세조정을 위한 4e-6이라는 학습률과 초기 10% 동안 학습률을 점진적으로 증가시켜 안정성을 확보했다. 또한, 학습 후반부의 학습률을 점진적으로 감소하도록 하는 코사인 스케줄링(Scheduling)을 통해 안정적인 학습 진행을 유도했다. 또한, AdamW(Adaptive Moment Estimation with Weight Decay)[19] 옵티마

이저와 0.05 가중치 감쇠를 통해 모델의 정규화를 강화했다. 3회의 연속된 평가에서 성능 향상이 없을 때 학습을 조기에 종료하는 전략을 통해 과적합을 방지하고 최적의 모델을 선택했다.

하드웨어 자원의 효율적인 활용을 위해 메모리 최적화 기법을 적용했다. BFloat16(Brain Floating Point 16) 데이터 타입과 4비트 양자화를 통해 9B 파라미터 모델의 메모리 사용량을 크게 줄이면서도 학습 정확도를 유지했다. 또한, 2,048 토큰이라는 충분한 시퀀스 길이를 설정해 요약 작업에 필요한 컨텍스트(Context) 길이를 확보했다.

본 연구에서는 모델의 미세조정 및 프롬프트 튜닝을 적용해 한국어 문서 요약 효율을 향상했다. 표 5는 제안하는 방법을 적용한 모델과 기본 모델의 성능을 BERTScore, RDASS, 평균 응답 시간을 통해 비교한다.

표 5. 제안 방법 적용 모델과 기본 모델 간 성능 비교

Model	BERTScore	RDASS	Average Response Time(s)
Base Model	0.8690	0.5617	4.1738
Our Model	0.8836	0.7527	6.5117

본 연구에서 제안한 모델은 BERTScore에서 기본 모델 대비 약 1.68% 향상된 0.8836을 기록했다. 특히 주목할 만한 개선은 RDASS에서 나타났는데, 기본 모델의 0.5617에서 0.7527로 약 34% 증가했다. 이는 제안하는 모델이 원본 문서와 참조 요약문 모두에 대해 의미적으로 더 높은 유사성을 가진 요약문을 생성할 수 있음을 시사한다. 그러나 이러한 성능 향상은 계산 비용 측면에서 상충관계를 보인다. 제안하는 모델의 평균 응답 시간이 기본 모델 대비 약 56% 증가했다. 이는 미세조정 및 프롬프트 튜닝 과정에서 추가된 파라미터와 계산 복잡성 증가에 기인한 것으로 보인다[20]. 요약이라는 특정 작업에 특화된 표현 학습으로 어텐션 연산의 복잡도가 높아졌을 가능성이 있으며, 기본 모델은 추론 최적화가 적용됐지만, 미세조정된 모델은 최적화 없이 원시 상태로 실행돼 시간이 더 길어졌을 수 있다. 이러한 결과는 모델의 정확도뿐만 아니라 추론 효율성까지 함께 고려해야 함을 시사한다.

표 6은 제안하는 모델의 단순한 성능 비교를 넘어, 미세조정과 프롬프트 튜닝의 타당성과 개별적인 기여도 및 복합 효과를 분석하고 결과를 해석하기 위한 제거 연구(Ablation Study) 결과를 제시한다.

표 6. 제거 연구 결과

Model	BERTScore	RDASS
Full Model	0.8836	0.7527
W/O Fine-Tuning	0.8713	0.6008
W/O Prompt-Tuning	0.8789	0.6820
W/O Both	0.8693	0.5621

미세조정만 제거한 경우, BERTScore는 0.8836에서 0.8713으로 약 1.39% 감소했으며, RDASS는 0.7527에서 0.6008로 약 20.18% 감소했다. 이는 미세조정이 특히 RDASS에서 모델 성능에 기여했음을 보여준다.

프롬프트 튜닝만 제거한 경우, BERTScore는 0.8836에서 0.8789로 약 0.53% 감소했으며, RDASS는 0.7527에서 0.6820으로 약 9.39% 감소했다. 이는 프롬프트 튜닝이 BERTScore보다 RDASS에서 더 큰 영향을 미쳤음을 시사한다.

미세조정과 프롬프트 튜닝의 개별 기여도 합이 BERTScore 0.0170, RDASS 0.2226으로 실제 두 기법을 모두 적용했을 때의 성능 향상인 BERTScore 0.0143, RDASS 0.1906과 유사하나 일치하지 않는데, 이는 두 기법 사이에 상호작용 효과가 존재함을 시사한다. 특히 RDASS에서 개별 기여도의 합이 0.2226으로 실제 복합 효과 0.1906

보다 약 16.8% 크게 나타난 것은 두 기법이 일부 중복되는 성능 향상 영역을 가지고 있을 가능성을 보여준다.

실험 결과는 한국어 문서 요약 작업에서 Gemma2-9B-IT 모델의 성능 향상을 위한 미세조정과 프롬프트 튜닝의 효과를 명확히 보여준다. 두 기법 모두 모델 성능 향상에 기여했으나, 그 중에서도 미세조정이 더 큰 영향을 미쳤음을 확인할 수 있다. 특히 RDASS의 현저한 개선은 본 연구에서 제안한 접근법이 원본 문서와 참조 요약 모두에 대한 의미적 유사성을 효과적으로 향상했음을 시사한다.

또한 BERTScore보다 RDASS에서 더 큰 개선이 관찰된 것은 제안하는 모델이 표면적인 텍스트 유사성보다 심층적인 의미 이해와 표현에서 더 뛰어난 성능을 보이고 있음을 시사한다. 그러나 성능 향상은 응답 시간 증가라는 비용을 수반한다는 점을 고려할 때, 실제 응용환경에서는 성능과 효율성 사이의 적절한 균형점을 찾는 것이 중요할 것이다.

이러한 결과는 대형 언어 모델의 미세조정과 프롬프트 엔지니어링의 특정 도메인 및 언어에 대한 모델 성능을 효과적으로 향상할 수 있음을 입증하며, 향후 연구에서는 계산 효율성을 유지하면서도 이러한 성능 향상을 달성할 방법에 대한 탐구가 필요함을 시사한다.

Ⅴ. 결 론

본 논문에서는 Gemma2-9B-IT 모델을 한국어 문서 요약 작업에 최적화하기 위한 미세조정과 프롬프트 튜닝 접근법을 제안하고, 그 효과를 실증적으로 검증했다. 실험결과를 종합적으로 고려할 때, 다음과 같은 결론을 도출할 수 있다.

첫째, 제안한 미세조정과 프롬프트 튜닝을 결합한 방법론은 Gemma2-9B-IT 모델의 한국어 문서 요약 성능을 유의미하게 향상했다. 특히 BERTScore는 약 1.68%, RDASS는 약 34%라는 상당한 개선이 이루어졌다. 이는 대형 언어 모델의 특정 언어 및 도메인 적응에 있어 적절한 최적화 전략의 중요성을 강조한다.

둘째, 미세조정과 프롬프트 튜닝의 개별적인 기여도를 분석한 결과, 두 기법 모두 성능 향상에 긍정적인 영향을 미쳤으나, 그중에서도 미세조정이 더 큰 기여도가 있음을 확인했다. 구체적으로, 미세조정은 RDASS 기준 약 20.18%의 성능 향상에 기여한 반면, 프롬프트 튜닝은 약 9.39%의 성능 향상을 가져왔다. 이는 한국어와 같은 특정 언어에 대한 요약 작업에서 모델 파라미터의 적응이 입력 프롬프트의 최적화보다 더 핵심적인 역할을 할 수 있음을 시사한다.

셋째, 두 기법의 상호작용 분석을 통해 미세조정과 프롬프트 튜닝 사이에는 일정 부분 중복되는 성능 향상 영역이 존재함을 확인했다. RDASS의 개별 기여도의 단순합은 0.2226으로 실제 두 기법을 모두 적용했을 때의 성능 향상 0.1906보다 크게 나타난 것은 두 접근법이 모델의 일부 동일한 측면을 개선하는 데 중첩된 효과를 가질 수 있음을 보여준다. 이러한 발견은 최적의 성능과 효율성 간의 균형을 위한 미세조정 전략 설계에 중요한 시사점을 제공한다.

넷째, 본 논문에서 제안한 모델이 BERTScore보다 RDASS에서 더 큰 개선을 보인 점은 주목할 만하다. 이는 본 접근법이 표면적인 텍스트 유사성보다 원본 문서와 참조 요약문에 대한 의미적인 유사성을 더 효과적으로 향상했음을 의미한다. 이러한 결과는 제안하는 모델이 한국어 텍스트의 심층적인 의미 이해와 표현 능력을 크게 개선했음을 시사한다.

다섯째, 성능 향상은 계산 비용의 증가를 수반했다. 응답 시간이 56% 증가한 점은 실제 응용환경에서 배포 시 고려해야 할 중요한 요소이다. 특히 실시간 서비스나 제한된 컴퓨팅 자원 환경에서는 성능과 효율성 사이의 적절한 균형점을 찾는 것이 필수적이다.

본 논문에서는 대형 언어 모델의 한국어 문서 요약 능력 향상을 위한 효과적인 방법론을 제시

하고, 그 효과를 실증적으로 검증했다는 점에서 의의가 있다. 특히 미세조정과 프롬프트 튜닝의 개별 및 복합 효과에 대한 체계적인 분석은 향후 다양한 언어와 작업에 대한 대형 언어 모델 최적화 전략 수립에 중요한 통찰을 제공할 것으로 기대한다.

향후 연구에서는 계산 효율성을 유지하면서도 유사한 성능 향상을 달성할 수 있는 최적화 기법의 탐구가 필요하다. 또한 다양한 도메인과 작업에 걸친 미세조정과 프롬프트 튜닝의 일반화 가능성, 그리고 모델 크기에 따른 두 기법의 효과 차이에 대한 추가적인 조사도 의미 있는 연구 방향이 될 것으로 생각한다.

REFERENCES

[1] Y. Liu, "Fine-tune BERT for Extractive Summarization," arXiv preprint arXiv:1903.10318, 2019.

[2] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 11328-11339, Jul. 2020.

[3] L. Basyal, and M. Sanghvi, "Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models," arXiv preprint arXiv:2310.10449, 2023.

[4] Team Gemma, et al. "Gemma 2: Improving Open Language Models at a Practical Size," arXiv preprint arXiv:2408.00118, 2024.

[5] AI-Hub(n.d.). https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=data&dataSetSn=97 (accessed May. 16, 2025).

[6] Unsloth(n.d.). https://docs.unsloth.ai/ (accessed May. 16, 2025).

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "Lora: Low-Rank Adaptation of Large Language Models," *International Conference on Learning Representations*, Apr. 2022.

[8] Z. Hu, L. Wang, Y. Lan, W. Xu, E. Lim, L. Bing, X. Xu, S. Poria, and R. K. Lee, "LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models," arXiv preprint arXiv:2304.01933, 2023.

[9] G. Adams, A. R. Fabbri, F. Ladhak, E. Lehman, and N. Elhadad, "From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting," *In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, vol. 2023, no. 4, pp. 68-74, Dec. 2023.

[10] M. S. Aissi, C. Romac, T. Carta, S. Lamprier, P. Oudeyer, O. Sigaud, L. Soulier, and N. Thome, "Reinforcement Learning for Aligning Large Language Models Agents with Interactive Environments: Quantifying and Mitigating Prompt Overfitting," arXiv preprint arXiv:2410.19920, 2024.

[11] S. U. Park, and J. Y. Kang, "Analysis of Prompt Engineering Methodologies and Research Status to Improve Inference Capability of ChatGPT and Other Large Language Models," *Journal of Intelligence and Information Systems*, vol. 29, no. 4, pp. 287-308, Dec. 2023.

[12] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," arXiv preprint arXiv:1904.09675, 2019.

[13] D. Y. Lee, M. C. Shin, T. S. Whang, S. W. Cho, B. G. Ko, D. Lee, E. G. Kim, and J. C. Jo, "Reference and Document Aware Semantic Evaluation Methods for Korean Language Summarization," arXiv preprint arXiv:2005.03510, 2020.

[14] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," *Advances in Neural Information Processing Systems*, Vol. 36, pp. 10088-10115, 2023.

[15] Google AI Studio(n.d.). https://aistudio.google.com/app/prompts/new_chat?model=gemma-2-9b-it (accessed May. 16, 2025).

[16] C. Y. Lin, "ROUGE: A package for Automatic Evaluation of Summaries," *Text Summarization Branches Out*, pp. 74-81, Jul. 2004.

[17] S. S. Lee, and S. W. Kang, "Empirical Study for Automatic Evaluation of Abstractive Summarization by Error-Types," *Korean Journal of Cognitive Science*, vol. 34, no. 3, pp. 197-226, Sep. 2023.

[18] D. Kalajdzievski, "A Rank Stabilization Scaling Factor for Fine-Tuning with LoRA," arXiv preprint arXiv:2312.03732, 2023.

[19] I. Loshchilov, and F. Hutter, "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101, 2017.

[20] H. Zhou, X. Lu, W. Xu, C. Zhu, T. Zhao, and M. Yang, "LoRA-drop: Efficient LoRA Parameter Pruning based on Output Evaluation," *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5530-5543, Jan. 2025.

저 자 소 개

김승주(정회원)

2025년 국립순천대학교 인공지능공학부 졸업(공학사).
2025년~현재 국립순천대학교 스마트융합학부 멀티미디어공학전공 석사과정.

<주관심분야 : 대형언어모델, 자연어처리, 딥러닝>

정세훈(정회원)

2012년 국립순천대학교 멀티미디어공학과 졸업(공학석사).
2017년 국립순천대학교 멀티미디어공학과 졸업(공학박사).
2018년 영산대학교 빅데이터융합전공 조교수.
2020년 안동대학교 창의융합학부 조교수.
2022년~현재 국립순천대학교 컴퓨터공학과 조교수.

<주관심분야 : 소프트웨어공학, 강화학습, 블록체인, 딥러닝, 데이터 마이닝, 빅데이터 분석 및 예측>

심춘보(정회원)

1996년 전북대학교 컴퓨터공학과 졸업(공학사).
1998년 전북대학교 컴퓨터공학과 졸업(공학석사).
2003년 전북대학교 컴퓨터공학과 졸업(공학박사).
2005년~현재 국립순천대학교 인공지능공학부 교수.

<주관심분야 : 빅데이터, 블록체인, 딥러닝, 생성모델, 자연어처리, 강화학습>