The Transactions of the Korea Information Processing Society
TKIPS Vol.14, No.4, pp.239~247

eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2025.14.4.239

PEFT Methods for Domain Adaptation

Lee You Jin† · Yoon Kyung Koo†† · Chung Woo Dam†††

ABSTRACT

This study analyzed that the biggest obstacle in deploying Large Language Models (LLMs) in industrial settings is incorporating domain specificity into the models. To mitigate this issue, the study compared model performance when domain knowledge was additionally trained using MoRA, which enables learning more knowledge information, and LoRA, which is the most common among various PEFT methods. Along with this, training time was reduced through securing high-quality data and efficient data loading. The findings of this research will provide practical guidelines for developing efficient domain-specific language models with limited computing resources.

Keywords : Parameter-Efficient Fine-tuning, Domain Adaptation, Deep Learning, Large Language Model, LoRA, MoRA

도메인 데이터 추가 학습 PEFT 방법 비교

이 유 진† · 윤 경 구†† · 정 우 담†††

요 약

대규모 언어 모델(LLM)의 산업 현장 도입에 있어 가장 큰 장애물은 모델에 도메인 특수성을 반영하는 것이다. 해당 문제를 완화하기 위해 다양한 PEFT 방법 중 더 많은 지식 정보를 학습이 가능한 MoRA와 가장 보편적인 LoRA로 각각 학습하여 도메인 지식 추가 학습했을 때의 모델 성능을 비교하였다. 이와 함께 고품질 데이터 확보와 효율적인 데이터 로딩으로 학습 시간을 단축하였다. 이 연구 결과는 제한된 컴퓨팅 자원으로 효율적인 도메인 특화 언어 모델을 개발할 수 있는 실용적인 가이드라인을 제시한다.

키워드 : 매개변수 효율적 미세조정, 도메인 적응, 딥러닝, 대규모 언어 모델, LoRA, MoRA

1. 서 론

대규모 언어 모델(LLM, Large Language Models)의 초기 연구는 거대 데이터를 학습하여 모델 자체의 일반적인 언어 이해 능력을 향상시키는 사전학습 위주로 진행되었다. 특히, 스케일링 법칙[1, 2]과 창발적 능력[3]이 주목받으며 모델의 성능이 크게 향상되었고, ChatGPT[1]와 같은 거대 언어 모델 서비스들이 출시되었다. ChatGPT 서비스를 운영하고 있는 OpenAI는 2024년 8월 기준 주간 사용자가 작년 대비 2배 증가한 2억 명이라고 발표했다.

그러나, 실제 산업 현장에서는 위와 같은 일반적인 대규모 언어 모델을 그대로 적용하기 어렵다. 우선, 산업 현장의 전문적인 데이터를 학습하지 못한 언어 모델은 도메인 특수성을 반영하지 못하기 때문이다. 금융, 의료와 같이 일반 상식을 넘어 해당 산업에 특화된 전문 지식이 중요한 산업 현장에서 모델에게 요구되는 능력은 특정 도메인에 대한 전문성이므로 일반 대규모 언어 모델의 능력만으로는 충분치 않다. 또한, 기업 내 데이터는 기업 특화 기술과 경험이 축적된 중요 자산이기 때문에, 데이터가 조직 외부로 유출되지 않도록 주의를 기울여야 한다. 예를 들어 삼성전자 역시 내부 프로그램의 코드 유출을 방지하기 위하여 코딩 어시스턴트 서비스 코드아이(code.i)[2]를 자체 개발 및 도입하였다. 즉, 산업 도메인의 특수성, 용도에 맞게 조정된 폐쇄적인 도메인 특화 언어 모델로 구현될 필요가 있다.

따라서, 기업은 산업 현장에서 대규모 언어 모델을 도입하기 위해 Meta의 Llama[23], Google의 Gemma[26]와 같은 오픈소스 모델을 활용하여 데이터를 새로 학습하고 조정하는 과정을 거쳐야 한다. 그러나 이러한 학습이 아무리 사전학습에 비해 경제적이라 할지라도, 각 도메인에 특화된 모델로 거듭

※ 이 논문은 과학기술정보통신부 · 광주광역시가 공동 지원한 '인공지능 중심 산업융합 집적단지 조성사업'으로 지원을 받아 수행한 연구 결과입니다.
† 정 회 원 : 주식회사 파수 개발센터 전임 연구원
†† 정 회 원 : 주식회사 파수 개발센터 CTO
††† 비 회 원 : 주식회사 파수 개발센터 연구원
Manuscript Received : December 16, 2024
Accepted : January 28, 2025
* Corresponding Author : Yoon Kyung Koo(yoonforh@fasoo.com)

1) https://openai.com/chatgpt/overview/

2) https://techblog.samsung.com/blog/article/25

나기 위해서는 여러 번의 학습을 해야 한다는 어려움이 있다. 예를 들어, 80억 개의 매개변수를 갖는 모델 (8B 모델)에 평균 토큰 수가 3천 개인 데이터 샘플 100만 개를 NVIDIA H100 8대로 1 세대만큼 학습한다고 가정해 보자. Vast.ai에서 대여할 경우 시간당 약 3만원으로 학습 시간은 2일이 걸렸을 때 약 144만원으로 추정된다. 이처럼 학습을 통한 모델 개발이 단 한 번의 학습만으로는 어려우므로 도메인 특화 학습을 효율적으로 수행할 필요가 있다.

본 논문은 도메인 특화 언어 모델 학습 측면에서 다음과 같은 성과를 달성하였다.

1. PEFT(Parameter-Efficient Fine-Tuning)[5] 중 가장 보편적으로 사용되고 있는 LoRA와 계산 및 추론과 같은 작업에서 뛰어난 MoRA를 도메인 지식 학습 측면에서 학습 후 성능을 비교하였다.
2. 산업 현장에서 대규모 데이터 확보가 어려울 것을 감안하여 데이터의 규모에 따른 학습 실험을 수행하였다.
3. 학습 효율성을 높이기 위해 중복 제거, 저품질 데이터 필터링과 같은 전처리 작업을 수행하였고, 학습 과정에서 DDP(Distributed Data Parallel) 방식, Lazy 데이터 로딩을 적용하여 학습 시간을 절반 가량 줄였다.

따라서, 대표적인 PEFT 방법들의 성능 비교를 통해 제한된 컴퓨팅 자원 내에서 최적의 파인튜닝 전략을 수립하는 가이드라인의 역할을 할 것으로 기대한다.

2. PEFT의 연구 동향

대규모 언어 모델은 여러 분야에서 높은 활용가능성을 보여주었다. 그러나 이러한 모델은 수십억 개의 매개변수를 지니고 있기에 모델 학습뿐만 아니라 모델을 사용하여 추론할 때에도 상당한 컴퓨팅 자원을 필요로 했다. 따라서, 비용적인 문제를 완화하고 대규모 언어 모델을 실제로 도입하기 위해 효율적인 모델 학습에 대한 관심이 높아졌다. 이와 관련된 연구인 PEFT(Parameter-Efficient Fine-Tuning)에서는 사전 학습된 모델의 매개변수를 특정 작업 혹은 도메인에 맞게 조정하는 과정에서 추가되는 매개변수의 수와 연산 자원을 최소화하는 학습 방식을 제안하였다. 〈Table 1〉에 보이는 것처럼 학습, 추론에 사용되는 매개변수를 제한하는 방식에 따라 네 가지 방법으로 분류할 수 있다.

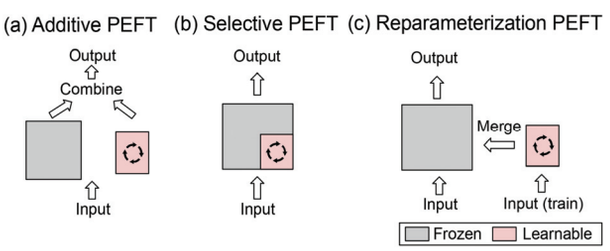

Fig. 1. Different types of PEFT algorithms. [4]

Table 1. Parameter-Efficient Fine-Tuning Methods

Methodology	Learning principles	Related works
Additive Fine-tuning	adapter	[5], [6], [7], [8]
	prompting	[9]
Selective Fine-tuning	parameter masking	[11], [12]
Reparameterized Fine-tuning	low-rank decomposition	[13], [14], [15], [16], [17]
Hybrid Fine-tuning	combine advantages	[18], [19]

부가적 기법(additive)은 원래 모델에 새로운 데이터 정보를 주입하는 방법에 따라 어댑터, 소프트 프롬프트 방식으로 나뉜다. 어댑터 계열 연구는 새로 학습 가능한 모듈이나 매개변수를 원본 모델 아키텍처에 주입한다. 다시 말해, 트랜스포머 블록 내에 작은 어댑터 계층을 삽입하는 방식으로, 대표적으로 직렬(serial) 어댑터[5, 6], 병렬(parallel) 어댑터[7, 8]가 있다. 한편, prefix-tuning[9]과 같은 소프트 프롬프트 계열 연구는 프롬프트 내에서 추가적인 정보를 주입하는 방법이다. 이는 문맥 내 학습을 통한 이산적인 토큰 표현을 최적화하는 것보다 프롬프트 내 연속적인 임베딩 공간이 더 많은 정보를 포함할 수 있음을 전제로 한다. 부가적 기법은 모델의 전체 매개변수가 아닌 어댑터 모듈만 학습하므로 효율적이며, 어댑터나 소프트 프롬프트를 필요에 따라 추가 및 제거할 수 있다는 점에서 모델을 유연하게 확장할 수 있다. 그러나, 성능 향상의 폭이 제한적일 수 있고, 어댑터 구조 설계가 어렵다. 또한, 소프트 프롬프트는 모델별, 작업별로 잘 작동하는 프롬프트 최적화가 어렵다. 해당 방법은 각 작업에 대해 복잡한 제어가 필요하거나 디버깅과 해석이 중요한 시스템에 적용하는 것이 유용하다.

선택적 기법(selective)은 일부 매개변수에 대해서만 학습 가능하도록 선택한다. Diff pruning[10]은 모델 매개변수의 학습 여부를 이진 마스크를 적용하여 마스킹 방식으로 선택하며, PaFi[11], FishMask[12]가 이에 속한다. 선택적 기법은 중요한 매개변수만 선별적으로 학습하므로 계산 효율성을 향상시키며 어떤 매개변수가 특정 태스크에 중요한지 파악할 수 있어 모델에 대한 해석가능성을 높인다. 하지만, 최적 마스크를 선정하기 어렵고, 구현이 복잡하다.

재매개변수화 기법(reparameterized)은 원래 모델 매개변수를 저차원으로 재매개변수화하여 학습한 후, 추론할 때 이를 다시 동등한 차원으로 변환하는 방식이다. 따라서, 추론 속도가 변하지 않는다는 장점이 있다. Intrinsic SAID[13]와 같은 초기 연구들은 사전 학습된 모델의 매개변수가 실제로는 저차원의 공간에 분포되어 있다는 것을 밝혀냈다. 이러한 통찰을 바탕으로 LoRA[14]가 개발되었다. LoRA는 모델 적응 과정에서의 파라미터 변화도 낮은 본질 차원(intrinsic rank)을 가질 것이라 보고, 저차원에서 학습하는 방식을 제안하였다.

이에 대한 파생 연구로 DyLoRA[15], DoRA[16], MoRA[17] 등이 있다. 재매개변수화 기법은 저차원 공간에서 학습이 이루어지므로 메모리와 계산 효율적이며, 기본 모델 구조를 크게 변경하지 않고도 적용 가능하여 구현이 쉽다. 그러나, 차원 축소 과정에서 정보를 손실할 수도 있고, 복잡한 작업에서 저차원 표현만으로 충분하지 않을 수 있다. 해당 방법으로 학습된 대표적인 사례로는 Alpaca[24], Vicuna[27] 등이 있다.

하이브리드 기법(hybrid)은 서로 다른 PEFT 방법들의 장점들을 결합해 더 효과적인 모델을 구축하는 방식이다. 각 PEFT 방법별 특화된 작업이 있을 것이라 가정하고 PEFT 방법들의 유사성을 분석하여 다수의 방법들을 복합적으로 사용하였다. 관련 연구로 UniPEFT[18], S4[19]이 있다. 하이브리드 기법은 각 PEFT 방법의 장점을 결합하여 더 높은 성능을 달성할 수 있고, 작업의 특성에 따라 다른 방법을 조합해볼 수 있다는 높은 적용성을 띄지만, 구현 복잡도가 그에 따라 증가하고 이로 인한 최적화가 어렵다는 단점이 있다.

이처럼 효율적인 언어 모델 학습을 위해 다양한 PEFT 방법 연구들이 진행되었다. 대규모 언어 모델 활용의 중요성이 대두됨에 따라, 기업의 효율적인 AI 도입을 위해서는 도메인 지식 학습 연구가 필수적이다. 특히 데이터 보안 문제 그리고 산업별 특화된 전문 지식의 필요성이 증가하면서 도메인 특화 학습의 중요성이 더욱 부각되고 있다. 따라서, 각 기업의 특수한 도메인에 모델을 최적화하는 과정에서 발생하는 비용과 시간을 절감하기 위해, 체계적인 도메인 지식 학습 방법론과 가이드라인 개발은 중요한 이슈이다. 이는 단순한 기술적 최적화를 넘어 기업의 경쟁력과 직결되기 때문이다.

PEFT는 도메인 특화 모델 개발에 있어 제한된 컴퓨팅 자원으로도 효율적인 모델 학습의 패러다임을 열어주었다. 서론에 언급했듯이 대규모 언어 모델 학습에는 상당한 컴퓨팅 비용이 소요되는데, PEFT 방법론을 적용하면 메모리와 연산량 측면에서 학습을 효율적으로 진행할 수 있게 된다. 예를 들어, 8B 모델을 FFT(Full Fine-tuning)할 때는 모델 매개변수만으로 32GB의 메모리가 필요하며, 옵티마이저 상태, 그래디언트, 활성화 값까지 고려하면 A100 80GB 2대는 확보되어야 학습이 가능해진다. 반면, PEFT 방법 중 하나인 LoRA를 순위(rank) 8로 적용하면 학습 파라미터 수가 전체의 0.1%인 8M로 줄어들어 필요 메모리가 100MB 수준으로 감소하므로, RTX 3090 24GB 2대에서도 학습이 가능해진다. 비록 모델의 성능은 FFT 대비 다소 낮을 수 있으나, 학습 효율성 측면의 이점이 성능 차이를 상쇄할 만큼 크기 때문에 제한된 자원 환경에서는 PEFT 방법을 적용한 학습이 필수적이다.

본 논문은 금융 도메인 데이터를 LoRA와 MoRA 방법으로 각각 학습시킨 후 PPL(perplexity) 측정을 통해 금융 도메인 지식의 학습 정도를 비교하였다. 또한, 실제 산업 환경에서 고품질의 대규모 데이터 확보가 어려운 현실을 감안하여, 데이터 규모에 따른 학습 효율성을 비교하였다. 이는 효율적인 도메인 추가 학습을 진행함에 있어 유용한 가이드라인이 될 것이다.

3. 도메인 지식 학습 방법

3.1절은 도메인 지식을 학습하는 두 가지 주요 전략인 추가 학습과 지시 학습에 대해 설명하고, 본 연구에서 추가 학습 방식을 선택한 배경을 제시한다. 3.2절에서는 재매개변수화 기법의 선정 배경과 이론적 근거를 살펴본다. 특히 저차원 접근법인 LoRA와 고차원 접근법인 MoRA의 수학적 특성을 비교 분석하여, 도메인 학습에서 효과적인 학습 전략을 제안한다.

3.1 도메인 학습 전략

1) 추가 학습 (further training)

추가 학습은 도메인 텍스트 말뭉치 등의 데이터로 사전 학습한 모델을 학습하여 모델 자체의 기본 지식을 확장한다. 대표적으로 법률, 의료, 금융 등 전문 분야의 지식 습득, 학술 연구나 전문 서적에 대한 이해를 목적으로 수행된다. 도메인 특화 전문 용어나 지식이 내재된 텍스트 자체를 학습하기 때문에, 도메인 특유의 언어 패턴, 표현, 문맥 등을 안정적으로 학습할 수 있다는 장점이 있다. 그러나 많은 양의 학습 데이터가 필요하며, 실제 현장 적용을 위해서는 특정 작업별 추가 조정이 요구된다.

2) 지시 학습 (instruction tuning)

지시 학습은 입력-출력 쌍 형태의 데이터를 통해 지시사항과 그에 적절한 응답을 학습하는 방식이다. 예를 들어 고객 서비스 챗봇, 특정 양식 보고서 생성 서비스 등 특정한 작업을 수행하기 위해 지시 학습을 진행한다. 추가 학습과 달리 작업 범위 내에서 지식을 적재적소에 활용하는 방법과 출력 제약을 동시에 학습하며, 더 적은 양의 데이터로도 학습이 가능하다. 다만 고품질 데이터의 구축이 어렵고, 학습된 도메인 지식이 특정 작업에만 한정된다는 한계가 있다.

위 두 방법은 학습 목적이 다르고 이에 따라 사용하는 학습 데이터와 직면하는 문제점이 상이하다. 추가 학습의 목표는 사전 학습한 언어 모델의 일반적 지식에 도메인 지식을 보강하는 것이고, 지시 학습은 명확한 지시를 잘 수행하는 능력을 향상시키는 것이다. 따라서, 추가 학습은 학습 데이터로 도메인 관련 전문 서적, 논문 등과 같은 대량의 비구조적 텍스트 데이터를, 지시 학습은 전문가에 의해 고품질의 데이터로 검증된 입력(지시)-출력 쌍의 구조화된 텍스트 데이터를 필요로 한다. 결과적으로 추가학습은 대규모 데이터를 학습하기에 더 많은 컴퓨팅 리소스를 확보해야하는 어려움을 가지며, 지시학습은 데이터 수집 과정에서 전문가의 검증과 같은 과정이 필요하므로 데이터 확보의 난이도에 따른 데이터 구축 비용이 높다.

실제 현장에서 도메인 특화 언어 모델을 개발하기 위해 추가 학습과 지시 학습을 순차적으로 활용하는 것이 일반적이나, 본 연구는 추가 학습에만 초점을 맞추었다. 이는 지시 학습에서는 LoRA의 우수성이 이미 입증 되었지만, 도메인 지식

습득 과정인 추가 학습에서의 효과적인 방법론에 대한 연구는 진행되지 않았기 때문이다. 따라서 본 실험에서는 금융 도메인 데이터를 활용하여 인과적 언어 모델링(CLM, Causal Language Modeling) 학습을 진행하였다.

3.2 재매개변수화 방법

도메인 지식 추가 학습을 위한 PEFT 방법으로 재매개변수화 기법을 선택하였다. 이는 추론 시 지연이 없어 실용적이며, [13]에서 밝힌 것처럼 도메인 지식 학습에서도 매개변수의 본질 차원이 낮을 것이라는 이론적 근거가 있기 때문이다. 이러한 장점은 LoRA와 같은 재매개변수화 기법의 우수한 성능을 통해 나타난다.

재매개변수화 방법의 학습 원리는 Equation (1)과 같다:

$$W' = W + \Delta W \tag{1}$$

여기서 W는 전체 매개변수 행렬이며, ΔW는 저차원 매개변수로 생성되는 업데이트 행렬이다. ΔW 생성에 사용되는 중간 행렬들은 W보다 작은 차원을 가지며, 이러한 중간 행렬들의 재매개변수화 방식이 핵심이다.

1) LoRA(Low-Rank Adaptation)

Microsoft가 제안한 LoRA[14]는 매개변수가 저차원 공간에 분포한다는 이전 연구[13]에 착안하여, 도메인 적응 학습도 저차원 공간에서 이루어질 수 있음을 밝혔다. 이에 따라 원본 모델의 매개변수는 고정한 채, 매개변수를 저차원의 학습 가능한 행렬로 분해하는 재매개변수화를 수행하였다. 이를 수식으로 나타내면 다음과 같다:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{2}$$

위 식의 B, A가 이에 해당되는데, 이 행렬들은 원본 매개변수 행렬의 업데이트를 효과적으로 표현할 수 있으며, 훨씬 적은 매개변수만으로 유사한 성능을 달성하였다. 예를 들어, $d \times k$ 크기의 원본 매개변수 행렬을 가진 모델에서 LoRA는 이를 $d \times r$과 $r \times k$ 크기를 가진 두 행렬의 곱으로 표현한다. 여기서 순위(rank)를 나타내는 r은 일반적으로 d와 k보다 훨씬 작은 값을 사용하며, 이를 통해 매개변수 수와 메모리 사용량을 크게 줄일 수 있다.

2) MoRA (High-Rank Adaptation)

MoRA[17]는 LoRA와 달리 B, A를 정방행렬 M으로 표현한다. 수식화하면 다음과 같다:

$$h = W_0 x + f_{decomp}(M f_{comp}(x)) \tag{3}$$

Equation (2) 내 학습 가능한 행렬 A, B는 MoRA에서 Equation (3)의 $f_{decomp}(M f_{comp}(x))$로 표현된다. LoRA의 매개변수가 $(d+k)r$개의 매개변수를 가진다면, MoRA는 $\hat{r}$인 정방행렬 M을 사용하여 $\hat{r} \times \hat{r}$개의 매개변수를 가진다. f_{comp}와 f_{decomp}는 각각 입력 차원을 줄이고, 출력 차원을 늘리는 함수이다. 해당 함수는 모델 내에서 학습하는 것이 아니라 실무자가 직접 새로 함수를 생성하여 적용할 수 있다. (Fig. 2)에서 LoRA와 MoRA를 비교하여 설명하였다.

LoRA와 MoRA를 각각 사용하여 순위(rank)를 8로 지정하고 모델 학습을 진행한다고 하자. 예를 들어 hidden size가 4096이라면 LoRA는 크기가 각각 4096×8인 행렬 A와 8×4096인 행렬 B로 순위(rank, ΔW)는 8로 제한된다. 반면, MoRA는 256×256 크기의 정사각 행렬 M이므로 순위(rank, ΔW)는 256가 된다. MoRA의 학습 가능한 매개변수는 $256 \times 256 = 65536$이고, LoRA의 학습가능한 매개변수는 $4096 \times 8 \times 2 = 65536$로 동일한 값을 가진다, 따라서, 순위(rank)를 8로 줬을 경우, MoRA는 32배 더 높은 랭크를 가지므로 복잡한 패턴과 관계를 표현할 수 있게 된다. LoRA의 순위가 8일 때 MoRA는 256을, 32일 때 512가 될 것이므로, 순위(rank)가 커지면 커질수록 정보 표현력 차이가 커질 것이다. 그러므로 많은 양의 도메인 지식을 학습해야할 경우 MoRA가 더 효과적일 것으로 추정할 수 있다.

4. 실 험

MoRA에 따르면 LoRA는 대규모 언어 모델의 기존 지식을 활용하는 데는 탁월하지만, 모델의 능력을 향상시키는 작업에서는 한계가 있다. 이는 LoRA가 사용하는 저차원 행렬의 rank로 인해 독립적인 정보의 양이 제한되어 복잡한 데이터 표현이 어렵다고 판단했다. MoRA는 이러한 한계를 극복하기 위해 사용된 파라미터의 개수에 비해 상대적으로 고차원의 행

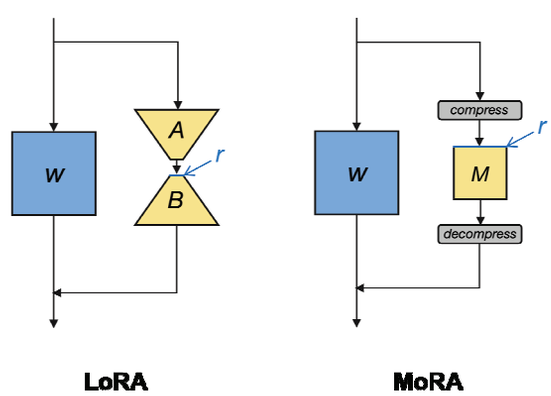

Fig. 2. An Overview of LoRA and MoRA. W is the frozen weight from model. B and A are trainable low-rank matrices in LoRA, while M is trainable matrix in MoRA. *decompress* and *compress* are non-parameter operators for increasing output dimension and reducing the input dimension. [17]

렬을 도입하여, 복잡한 추론이나 수학적 작업, 암기 등에서도 더 나은 성능을 보여준다.

도메인 지식 추가 학습에서 모델은 도메인에 특화된 전문 용어의 암기뿐만 아니라 복잡한 문맥 및 다양한 표현까지 많은 양의 정보를 학습해야 한다. 이전 섹션에서 LoRA와 MoRA의 차이를 비교하며, 모델 학습 시 MoRA 학습에서 더 큰 순위(rank)를 가지므로 정보 표현력이 뛰어날 것을 도출했다. 본 연구는 MoRA의 특성을 바탕으로 도메인 지식 추가 학습에서도 MoRA가 더 적합할 것이라는 가설을 세웠다. 이 가설의 가능성을 검토하기 위해 평균 토큰 수가 4천 개인 데이터 샘플 1만 개로 LoRA, MoRA 방식에 대하여 학습하여 PPL을 측정하였다. 실험을 통해 도출된 PPL 수치는 MoRA에서 8.025, LoRA에서 9.546으로 MoRA의 성능이 더 좋았고, 이를 토대로 MoRA의 도메인 지식 학습에 대한 잠재성을 확인하여 실험을 확장하였다.

실험은 두 가지로 분류하여 진행되었다. 첫 번째 실험은 금융 도메인 데이터로 LoRA와 MoRA를 각각 학습시킨 후 PPL 측정을 통해 금융 도메인 지식 학습 정도를 비교하였다. 두 번째 실험은 실제 산업 현장에서 고품질의 데이터 확보가 어려운 것을 반영하여, 데이터 규모에 따른 LoRA와 MoRA의 학습 유효성을 측정하였다. 첫 번째 실험과 마찬가지로, 비교를 위한 척도로 PPL을 사용하였다. 또한, 원본 데이터의 품질을 높이기 위해 전처리 과정을 수행했고, DDP(Distributed Data Parallel), Lazy 데이터 로딩을 적용하여 학습의 효율성을 높였다.

4.1 실험 환경

이 연구에서는 Llama 3.1 8B 모델을 사용했으며, 학습에는 5e-5의 학습률과 cosine-decay 스케줄러를 적용하였다. 데이터 품질에 따라 배치사이즈를 차등 적용했는데, 이는 전처리 이후 데이터의 평균 길이가 줄었기 때문이다. 일반 데이터는 64, 고품질 데이터는 128의 배치사이즈를 사용했다.

PEFT 방법 중 LoRA와 MoRA를 채택했으며, 두 방법 모두 순위(rank)는 32, alpha는 64로 동일한 값을 주었다. 학습 데이터는 AI-hub의 금융 분야 다국어 병렬 말뭉치와 허깅페이스 Bloomberg Financial News, Financial-news-articles를 활용하여 총 93만 개의 샘플(약 2.3GB)을 확보했다. 실험은 NVIDIA H100 80GB HBM3 GPU 8개를 사용하였고, CUDA 12.4, PyTorch 2.4, Python 3.10 환경에서 구현되었다.

4.2 실험 방법

1) 고품질 학습 데이터 구축

고품질 학습 데이터 구축은 초거대 언어 모델의 성능에 큰 영향을 미친다. OpenAI의 'Dalle-3[3]'과 'Dalle-2'의 성능 차이는 학습 데이터 주석 품질에 의한 것이었고, AllenAI 연구소 모델 'OLMo[4]'와 Meta의 'Llama 3'은 구조적으로 매우 유사하지만 학습 데이터 차이에 의해 성능 차이가 발생했다고 분석되었다. 따라서, 고품질 학습 데이터 구축을 위해 중복 제거와 모델 기반 데이터 필터링 작업[23]을 수행했다.

a) 중복 제거

웹 크롤링 및 다양한 공개 데이터를 통합하여 학습 데이터로 사용할 경우, 중복된 데이터 문자열이 포함될 수 있다. 학습 데이터 내 중복 제거는 암기 현상을 줄이고 데이터의 다양성이 증가시키므로 모델 성능을 향상시킬 수 있다. 또한, 중복 제거의 수준은 문서 단위, 문장 단위로 수행될 수 있다.

본 논문에서 활용한 중복 제거 방법은 AllenAI 연구소에서 공개한 bff[5] 도구이다. 먼저, 텍스트 내 문단과 문서를 n-gram 단위로 분절하여 조회하면서 해당 n-gram의 블룸 필터에 존재 여부를 확인한다. 다음으로, 블룸 필터 내에 존재하는 n-gram의 비율이 필터링 임계값을 넘는 경우, 해당하는 문단 혹은 문서를 제거한다. 블룸 필터의 초기화에 필요한 n-gram 개수는 중복 제거를 실행할 텍스트를 공백 기반으로 분절했을 때의 단어 수로 할당하였다. 또한, 거짓 양성 오류 비율은 [22]에서 사용한 기본 값인 0.01로, n-gram 크기와 필터링 임계값은 [22]에서 사용한 13과 0.8로 설정하였다.

b) 모델 기반 데이터 필터링

학습 가능한 모델을 품질 필터로 사용했을 때, 실제 특정 작업에 대한 모델 성능이 향상되는 것[20, 21]을 발견했다. [22]는 이를 토대로 다양한 데이터 품질 필터링 방법을 비교하였고, fastText[25] 모델이 가장 뛰어난 성능을 보여주었다. fastText는 word2vec과 유사하나, 단어를 n-gram으로 분절하고, 하위 단어를 고려하여 유사성을 학습하는 모델이다. 이에 근거하여 본 논문은 fastText 모델을 적용한 모델 기반 데이터 필터링을 수행하였다. 이를 위해 KoCommercial, Eli5에서 수집한 텍스트 데이터를 긍정으로, c4에서 수집한 텍스트 데이터를 부정으로 분류하도록 학습하였다. 해당 모델을 기반으로 추론 시, 부정으로 분류 된 데이터 중 하위 20%만이 제거되도록 하였다.

word2vec와 달리 fastText 기반 필터링은 텍스트의 맥락적 특성을 고려하면서 동시에 과도한 데이터 손실을 방지한다. fastText는 하위 단어 정보를 고려한 연속 skip-gram의 확장이기에 형태학적으로 풍부한 언어 벡터 표현이 가능하기 때문이다. 이 특징으로 인해 fastText는 제한된 데이터로 학습을 통해서도 단어 집합에 존재하지 않는 데이터(Out Of Vocabulary, OOV)를 처리할 수 있으며, 단어 순서와 지역 문맥을 잘 포착한 분류를 할 수 있다. 본 실험에서는 [22]가 제시한 상위 10%만 남기고 필터링하는 것이 아닌 하위 20%에 해당하는 데이터에 대해 필터링하는 완화된 기준을 채택하였

3) https://openai.com/index/dall-e-3/

4) https://allenai.org/olmo

5) https://github.com/allenai/bff

다. 이는 사용한 데이터가 단순 웹크롤링이 아니며 [22]와 도메인 데이터를 상위 10%만 남기고 제거했을 때 소량의 데이터만 남는 최악의 상황을 고려하여 필터링의 기준을 저품질 데이터 필터링에 더 초점을 뒀기 때문이다. 따라서, 부정으로 분류된 데이터 중 하위 20%만 제거하는 기준을 적용하여 모델의 판단이 불확실한 경계 영역의 데이터는 보존하며 품질이 현저히 낮은 데이터만 선별적으로 제거하였다.

우리는 Rust로 작성된 bff에 fasttext classification을 적용하였다. 12th Gen Intel(R) i7-12700 12코어 20스레드 CPU, 32GB RAM 환경에서 실행한 결과, 기존 2.26GB에서 15.4%가 제거되어 1.91GB의 데이터를 남기는 데 총 29초가 소요되었다.

2) 학습 데이터 샘플 로드 모듈

대규모 텍스트 데이터를 효율적으로 처리하고 학습에 적합한 형태로 변환하기 위해 데이터 처리 방식을 변경하였다. 기존 방식은 한 번에 전체 데이터를 로드한 다음, 그 데이터를 토큰화(tokenizing)하는 과정에서 불필요한 자원을 많이 소모했다. 이를 개선하기 위해 데이터가 필요할 때마다 로드하는 방식으로 변경했다.

먼저 추가적인 Dataset Class를 생성하고, json 데이터를 jsonl과 jsonl.idx 파일로 변환하였다. jsonl 형식은 각 행이 독립적인 json 객체이므로 전체 파일을 메모리에 로드할 필요 없이 필요한 값만 읽을 수 있다. 여기서, idx 파일은 각 행의 시작 위치를 지정하는 역할을 하여, 전체 파일을 순차적으로 읽지 않고도 원하는 데이터에 빠르게 접근할 수 있게 된다. 따라서, 데이터 샘플 로딩 방식을 필요한 데이터만 메모리에 로드하는 Lazy Loading을 구현할 수 있다.

이는 특히 대규모 학습 과정에서 발생하는 데이터 로딩 병목 현상을 효과적으로 해소하여 학습 시간을 획기적으로 단축시켰다. 허깅페이스 Accelerate 라이브러리[6]를 사용한 DDP(Distributed Data Parallel) 학습에서 Lazy Loading 도입 전후를 비교한 결과는 〈Table 3〉에 있다. MoRA의 1 세대 학습 시간이 7일, LoRA의 1 세대 학습 시간이 4일이었던 것이 각각 6.25일, 3.4일로 평균 10%의 시간을 단축하였다.

Table 2. Data Processing Impact on Training Time

Model (Llama 3.1 8B)	Data preprocessing	Data sample loading methods	training time per epoch (days)
MoRA	O	lazy	2.7
MoRA	X	lazy	6.25
MoRA	X	normal	7
LoRA	O	lazy	1.5
LoRA	X	lazy	3.4
LoRA	X	normal	4

6) https://huggingface.co/docs/accelerate/index

Table 3. Perplexity of LoRA and MoRA

model	r	# of samples	hold-out ppl	trained ppl
Llama 3.1 8B (base)	-	-	12.7490	6.0055
Llama 3.1 8B MoRA	32	917,846	5.6132	4.7544
Llama 3.1 8B MoRA	32	831,387	5.3602	4.3907
Llama 3.1 8B LoRA	32	917,846	6.0924	4.9484
Llama 3.1 8B LoRA	32	831,387	5.9001	4.6263

Lazy Loading 방식의 효율성을 검증하기 위해 전처리된 86만개의 데이터를 각각 전체를 한 번에 로드하는 Pre Loading과 인덱스 파일 기반으로 로드하는 Lazy Loading 방식으로 각각 로드한 다음 메모리 증가량과 파일 입출력시간을 측정하였다. 그 결과, Pre Loading의 메모리 사용 증가량은 4575.79MB, File I/O 시간은 21.51초인 반면, Lazy Loading은 배치 당 메모리 사용 증가량 0.04MB, File I/O 시간은 81.93초로 측정되었다.

Pre Loading은 File I/O 측면에서 한 번만 데이터를 로드하기에 시간적으로 이득이지만, 항상 고정된 메모리를 차지하기에 물리적 메모리가 넘치는 경우 스왑 메모리에 대한 오버헤드가 발생할 수 있다. 또한, 메모리를 모델과 결합하여 사용하기 때문에 모델 학습에도 부정적인 영향을 미칠 수 있다. 반면, Lazy loading은 File I/O가 여러 번 발생하여 더 느리지만 인덱스 파일 기반으로 접근하며 불필요한 I/O를 최소화하므로 오버헤드가 작다. 또한 Lazy Loading을 할 경우 데이터로 인해 차지하는 물리적 메모리가 적기에 스왑 메모리가 발생하지 않으며, 모델이 학습 시 거의 다 쓸 수 있다는 장점이 있다.

3) 모델 학습

본 실험은 스탠포드 Alpaca[24], Vicuna[27]의 학습 방식을 참고하여 진행하였다. Alpaca는 2023년 3월 스탠포드에서 LLaMA 7B을 self-instruct 방식을 통해 생성한 고품질 데이터로 LoRA 방식을 도입해 지시학습을 수행한 모델로, 대규모 컴퓨팅 리소스가 필요한 언어모델을 $600 미만의 비용으로 학습시켜 주목받았다. Vicuna는 Alpaca에 영감을 받은 UC Berkerley, UCSD, CMU, MBZUAI가 공동으로 개발한 대화형 언어 모델이다. LLaMA 7B, 13B 모델을 고품질 대화 데이터로 Alpaca와 동일하게 LoRA로 지시학습을 수행하였다. 실험은 Vicuna 팀에서 공개한 Fastchat 프레임워크를 토대로 추가학습을 위한 데이터 전처리 코드와 MoRA 학습에 필요한 코드를 각각 추가하였다. 모델은 데이터의 품질에 따라 배치사이즈가 각각 64, 128인 것과 모델 최대 길이가 16k에서 8k로 변경된 것을 제외하고 모든 학습에서 동일한 설정 값을 갖는다. 5e-5의 학습률, cosine-decay 학습 스케쥴러 방식을 채택하였으며, LoRA, MoRA 모두 각각 rank는 32, alpha 값은 64이다. 또한, 두 방법 모두 어텐션과 피드포워드 네트워크에 대하여 학습하였다.

본 논문에서는 데이터 품질 개선과 로딩 방식의 최적화를 통해 학습 효율성을 높였다. 〈Table 2〉은 고품질 데이터를 확보하고 데이터 샘플 로딩 방식을 개선한 결과이다. CPU와 GPU 메모리를 효율적으로 활용하면서 MoRA와 LoRA의 1 세대 학습 시간이 각각 7일에서 2.7일, 4일에서 1.5일로 대폭 감소하였다. 또한, 전처리를 통해 평균 데이터 샘플의 길이가 1만 6천개에서 8천개로 줄어들면서 디바이스 당 배치사이즈를 1에서 2로, 최대 토큰 수를 1만 6천에서 8천으로 조정할 수 있었다. 이러한 과정을 통해 전체적인 학습 과정의 효율성이 약 62% 정도 향상되었다.

4.3 실험 결과

첫 번째 실험은 도메인 지식 추가 학습에서 LoRA, MoRA의 학습 효율성을 확인하기 위해 금융 도메인 데이터로 추가 학습을 진행하였다. 과학기술정보통신부 및 NIPA에서 운영하는 국내 AI 통합 플랫폼 AIhub와 허깅페이스 FinBERT[28], BloombergGPT[29]와 같은 모델 학습에 활용된 Bloomberg Financial News, Financial-news-articles에 의해 확보되었다. 해당 데이터는 공시정보, 뉴스기사, 규제정보, 보고서, 학술논문과 같은 금융 문서에서 발췌한 텍스트 데이터로 전체 93만개의 샘플 중 대략 50만개는 한국어, 나머지 43만개는 영어로 구성되어 있다. 해당 데이터는 한국어와 영어 데이터를 함께 학습시킨 이유로는 영어 데이터 확보 용이성과 사전 학습된 모델 LLaMA가 지닌 강력한 영어 기반 지식과 언어 이해 능력의 망각(Catastrophic Forgetting) 현상을 방지하기 위함이다. 학습 효율성을 평가하기 위해 PPL을 측정했으며, 학습 데이터에서 랜덤 샘플링으로 추출된 10만개의 샘플과 학습에 사용되지 않은 hold-out 샘플 2만 개에 대한 PPL을 비교하였다.

〈Table 3〉를 보면 학습된 데이터의 PPL(5열)에서 LoRA, MoRA로 학습된 모델들의 PPL이 모두 원본 모델 PPL보다 낮으면서, 동시에 hold-out 샘플에 대한 PPL(4열)이 5열과 비교하여 차이가 크지 않다. 따라서, LoRA, MoRA의 학습은 유효함을 시사한다. 또한, hold-out 샘플에 대한 MoRA의 PPL은 5.6132, LoRA의 PPL은 6.0924임을 알 수 있다. 마찬가지로, 고품질 데이터로 학습한 모델을 기준으로 MoRA는 5.3602, LoRA는 5.9001을 가진다. 두 유형의 데이터에 대한 평균 PPL 개선율을 비교했을 때, MoRA는 57.96%, LoRA는 53.72%임을 알 수 있다. 따라서, MoRA가 LoRA보다 약 4.24%p 더 높은 개선율을 달성하였으므로 MoRA를 적용한 학습이 도메인 지식 추가 학습에 더 효과적이다.

두 번째 실험은 실제 산업 환경에서 고품질 데이터 샘플의 확보가 어렵다는 점을 고려하여, 고품질 데이터 샘플의 수를 점차 줄여가면서 MoRA 학습을 수행하였다. 데이터 샘플 수를 고품질 데이터 샘플 수 83만 개부터 40만 개, 20만 개로 점차 줄여 진행하였다. 데이터는 랜덤 샘플링으로 추출되었다. 첫 번째 실험과 모두 동일한 환경에서 진행되었으며, 평가 방법도 동일하다. 다만, PPL 측정은 첫 번째 실험에서의 hold-out 샘플에 대한 PPL만 비교하여 학습 유효성을 평가하였다.

그 결과, 데이터 샘플을 1/2씩 줄여 학습한 모델의 PPL은 (Fig. 3)에 보이듯이 점차 증가하는 경향을 보인다. PPL은 40만 개일 때 약 4%, 20만 개일 때 약 27% 정도 증가했다. 그러나 같은 상황에서도 MoRA로 학습된 모델의 PPL은 데이터가 40만 개일 때 5.9102, 데이터가 20만 개일 때 6.7950로 확인되었고, LoRA로 학습된 모델의 PPL은 데이터가 40만 개일 때 6.2849, 데이터가 20만 개일 때 7.6322로 측정되었다. 따라서, 데이터 샘플을 충분히 확보하기 어려운 상황에서는 LoRA보다 MoRA로 학습하는 것이 더 낫다고 볼 수 있다.

결과적으로 학습 데이터 샘플의 수와 관계없이 MoRA와 LoRA의 학습 가능한 매개변수 수는 동일할 때, MoRA가 더 효과적으로 정보를 포착함을 확인할 수 있다. 예를 들어 학습 과정에서 지정해준 순위(rank)가 8이라고 가정해보자. 순위가 8로 설정되었을 때, LoRA는 순위가 8인 저차원 행렬로 정보가 압축되는 반면, MoRA는 순위가 256인 행렬로 정보가 표현되어 더 풍부한 특징을 포착할 수 있게 된다. 따라서, 제한된 학습 데이터 환경에서도 MoRA는 LoRA와 동일한 매개변수 효율성을 유지하면서 더 높은 표현력을 달성할 수 있음을 입증한다. 이는 MoRA가 데이터의 미세한 특징과 복잡한 패턴을 더 정교하게 학습할 수 있는 구조적 이점을 가지고 있음을 시사한다.

5. 결 론

이 논문은 대규모 언어 모델(LLM)을 산업 현장에 효과적으로 도입하기 위한 도메인 특화 학습 전략에 대한 연구를 수행하였다. 일반적인 대규모 언어 모델들이 가진 한계로 인해 기업은 오픈소스 모델을 자체 데이터로 학습시켜야 하는데, 이

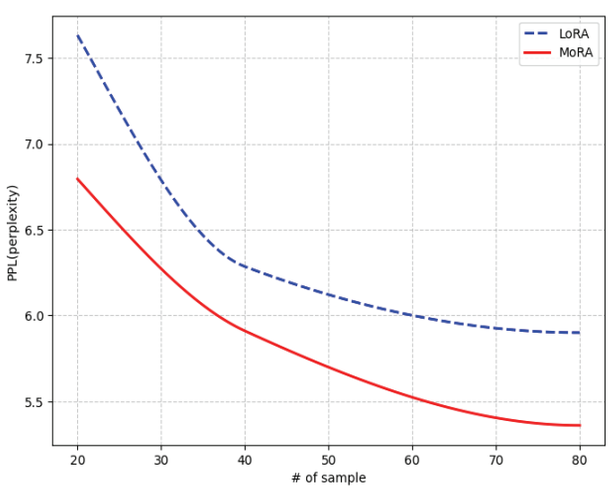

Fig. 3. PPL according to Data Sample Size

과정에서 상당한 비용과 시간이 소요된다. 이러한 문제를 해결하기 위해 본 논문은 데이터 전처리를 통한 고품질 데이터 확보와 LoRA, MoRA 같은 효율적인 파라미터 튜닝 방법(PEFT)을 비교 분석하여, 제한된 자원 내에서 최적의 도메인 특화 학습 전략을 제시하고자 하였다.

금융 도메인 데이터에 대한 LoRA, MoRA의 학습 능력을 비교하기 위해 본 실험은 고품질 데이터, 원본 데이터로 데이터 품질에 따라 분류하여 각각 학습 후 PPL 성능을 비교하였다. 또한, LoRA와 MoRA의 rank와 alpha를 둘 다 32, 64로 맞춰주어 비슷한 양의 파라미터를 추가 학습하도록 설정했다. 이에 따라, LoRA의 학습 파라미터 수는 83,886,080개, MoRA의 학습 파라미터 수는 83,846,144개로 8B 모델 기준 1%에 달하는 파라미터를 학습했다.

그 결과, 고품질 데이터로 학습한 MoRA의 PPL은 hold-out 샘플에 대하여 5.3602, LoRA의 PPL은 5.9001로 MoRA가 9.15% 더 낮게 측정되었다. 이를 통해 MoRA가 더 많은 도메인 지식을 암기했음을 확인할 수 있었다. 비록 원본 데이터에 대한 MoRA의 학습시간은 7일로 상당히 긴 시간을 소요했으나, 데이터 전처리와 DDP, Lazy Loading 등을 통해 2.7일까지 단축하여 활용 가능성을 높였다.

게다가, 학습 데이터 규모에 따른 실험 결과를 보면 MoRA와 LoRA 모두 80만 개에서 20만 개까지 점차 데이터의 규모가 작아짐에 따라 PPL의 성능이 크게 감소하는 추세를 보였으나, 항상 MoRA의 성능이 LoRA의 성능보다 좋았다. 40만 개에서 MoRA가 5.9102, LoRA가 6.2849로 6% 더 뛰어났으며, 20만 개에서는 6.7950, 7.6322로 10%로 확인되었다. 따라서, 데이터가 충분하지 않은 경우에도 MoRA로 학습하는 것이 더 적절할 것이다.

본 논문의 실험은 추가 학습에 국한되어 도메인 특화 작업에 대한 모델의 성능을 평가하지 못한 아쉬움을 남긴다. 후속 연구가 수행된다면 해당 모델에 대해 지시학습을 수행하고 각 작업별 모델 성능을 비교하여, 효율적인 도메인 학습 전략의 범위를 넓힐 것이다.

References

[1] J. Kaplan et al., “Scaling Laws for Neural Language Models,” *arXiv preprint arXiv:2001.08361*, 2020.

[2] J. Hoffmann et al., “Training Compute-Optimal Large Language Models,” *arXiv preprint arXiv:2203.15556*, 2022.

[3] J. Wei et al., “Emergent Abilites of Large Language Models,” *arXiv preprint arXiv:2206.07682*, 2022.

[4] Z. Han, C. Gao, J. Liu, J. Zhang and S. Q. Zhang, “Parameter-Efficient Fine-Tuning for Large Models,” *arXiv preprint arXiv:2403.14608*, 2024.

[5] N. Houlsby et al., “Parameter-efficient transfer learning for nlp,” *International Conference on Machine Learning, PMLR*, pp.2790-2799, 2019.

[6] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho and I. Gurevych, “AdapterFusion: Non-Destructive Task Composition for Transfer Learning,” *arXiv preprint arXiv:2005.00247*, 2021.

[7] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, “Towards a unified view of parameter-efficient transfer learning,” *in The Tenth International Conference on Learning Representations, ICLR*, 2022.

[8] T. Lei et al., “Conditional adapters: Parameter-efficient transfer learning with fast inference,” *arXiv preprint arXiv:2304.04947*, 2023.

[9] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” *arXiv preprint arXiv:2101.00190*, 2021.

[10] D. Guo, A. M. Rush and Y. Kim, “Parameter-efficient transfer learning with diff pruning,” *arXiv preprint arXiv:2012.07463*, 2020.

[11] B. Liao, Y. Meng, and C. Monz, “Parameter-efficient fine-tuning without introducing new latency,” *arXiv preprint arXiv:2305.16742*, 2023.

[12] Y. Sung, V. Nair, and C. Raffel, “Training neural networks with fixed sparse masks,” *Advances in Neural Information Processing Systems*, Vol.34, pp.24193-24205, 2021.

[13] A. Aghajanyan, L. Zettlemoyer, and S. Gupta, “Intrinsic dimensionality explains the effectiveness of language model fine-tuning,” *arXiv preprint arXiv:2012.13255*, 2020.

[14] E. J. Hu et al., “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.

[15] M. Valipour, M. Rezagholizadeh, I. Kobyzev, and A. Ghodsi, “Dylora: Parameter efficient tuning of pre-trained models using dynamic search- free low-rank adaptation,” *arXiv preprint arXiv:2210.07558*, 2022.

[16] S. Liu et al., “Dora: Weight-decomposed low-rank adaptation,” *arXiv preprint arXiv:2402.09353*, 2024.

[17] T. Jiang et al., “MoRA: High-Rank Updating for Parameter-Efficient Fine-Tuning,” *arXiv preprint arXiv:2405.12130*, 2024.

[18] Y. Mao et al., “Unipelt: A unified framework for parameter-efficient language model tuning,” *arXiv preprint arXiv:2110.07577*, 2021.

[19] J. Chen, A. Zhang, X. Shi, M. Li, A. Smola, and D. Yang, “Parameter-efficient fine-tuning design spaces,” *arXiv preprint arXiv:2301.01821*, 2023.

[20] D. Brandfonbrener, H. Zhang, A. Kirsch, J. R. Schwarz, and S. Kakade. “Color-filter: Conditional loss reduction filtering for targeted language model pre-training,” *arXiv preprint arXiv:2406.10670*, 2024.

[21] A. Fang, A. M. Jose, A. Jain, L. Schmidt, A. Toshev, and V. Shankar, "Data filtering networks," *arXiv preprint arXiv:2309.17425*, 2023.
[22] J. Li et al., "DataComp-LM: In search of the next generation of training sets for language models," *arXiv preprint arXiv:2406.11794*, 2024.
[23] A. Dubey et al., "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.
[24] R. Taori et al., "Alpaca: A Strong, Replicable Instruction-Following Model," Stanford Center for Research on Foundation Models. https://crfm.stanford.edu/2023/03/13/alpaca. html, 2023.
[25] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, "Advances in Pre-Training Distributed Word Representations," *arXiv preprint arXiv:1712.09405*, 2017.
[26] M. Riviere et al., "Gemma 2: Improving Open Language Models at a Practical Size," *arXiv preprint arXiv: 2408.00118*, 2024.
[27] W. Chiang et al., "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality," https://lmsys.org/blog/2023-03-30-vicuna/, 2023.
[28] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," *CoRR, vol.abs/1908.10063*, 2019.
[29] S. Wu et al., "BloombergGPT: A Large Language Model for Finance," *arXiv preprint arXiv:2303.17654*, 2023.

이 유 진
https://orcid.org/0009-0003-2673-6243
e-mail : jinleey0@fasoo.com
2018년 부경대학교 시스템경영공학부 (학사)
2023년 한양대학교 컴퓨터소프트웨어학과 (석사)
2024년~현 재 주식회사 파수 개발센터 전임연구원
관심분야 : Generative AI & XAI

윤 경 구
https://orcid.org/0009-0003-0048-3938
e-mail : yoonforh@fasoo.com
1991년 서울대학교 원자핵공학과(학사)
2019년~현 재 주식회사 파수 개발센터 CTO
관심분야 : Generative AI, Middleware

정 우 담
https://orcid.org/0009-0007-6785-6179
e-mail : elasticdavid@fasoo.com
2021년 연세대학교 심리학과(학사)
2023년~현 재 주식회사 파수 개발센터 연구원
관심분야 : Big Data System