JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 3, pp. 43-54, March 2024
https://doi.org/10.9708/jksci.2024.29.03.043

# Enhancing LoRA Fine-tuning Performance Using Curriculum Learning

Daegeon Kim*, Namgyu Kim*

*Graduate Student, Graduate School of Business IT, Kookmin University, Seoul, Korea
*Professor, Graduate School of Business IT, Kookmin University, Seoul, Korea

## [Abstract]

Recently, there has been a lot of research on utilizing Language Models, and Large Language Models have achieved innovative results in various tasks. However, the practical application faces limitations due to the constrained resources and costs required to utilize Large Language Models. Consequently, there has been recent attention towards methods to effectively utilize models within given resources. Curriculum Learning, a methodology that categorizes training data according to difficulty and learns sequentially, has been attracting attention, but it has the limitation that the method of measuring difficulty is complex or not universal. Therefore, in this study, we propose a methodology based on data heterogeneity-based Curriculum Learning that measures the difficulty of data using reliable prior information and facilitates easy utilization across various tasks. To evaluate the performance of the proposed methodology, experiments were conducted using 5,000 specialized documents in the field of information communication technology and 4,917 documents in the field of healthcare. The results confirm that the proposed methodology outperforms traditional fine-tuning in terms of classification accuracy in both LoRA fine-tuning and full fine-tuning.

▸ **Key words**: Pre-training Language Models, Large Language Models, Curriculum Learning, LoRA, Data heterogeneity

## [요 약]

최근 언어모델을 활용하기 위한 연구가 활발히 이루어지며, 큰 규모의 언어모델이 다양한 과제에서 혁신적인 성과를 달성하고 있다. 하지만 실제 현장은 거대 언어모델 활용에 필요한 자원과 비용이 한정적이라는 한계를 접하면서, 최근에는 주어진 자원 내에서 모델을 효과적으로 활용할 수 있는 방법에 주목하고 있다. 대표적으로 학습 데이터를 난이도에 따라 구분한 뒤 순차적으로 학습하는 방법론인 커리큘럼 러닝이 주목받고 있지만, 난이도를 측정하는 방법이 복잡하거나 범용적이지 않다는 한계를 지닌다. 따라서, 본 연구에서는 신뢰할 수 있는 사전 정보를 통해 데이터의 학습 난이도를 측정하고, 이를 다양한 과제에 쉽게 활용할 수 있는 데이터 이질성 기반 커리큘럼 러닝 방법론을 제안한다. 제안 방법론의 성능 평가를 위해 국가 R&D 과제 전문 문서 중 정보통신 분야 전문 문서 5,000건, 보건의료 전문 문서 데이터 4,917건을 적용하여 실험을 수행한 결과, 제안 방법론이 LoRA 미세조정과 전체 미세조정 모두에서 전통적인 미세조정에 비해 분류 정확도 측면에서 우수한 성능을 나타냄을 확인했다.

▸ **주제어**: 사전학습 언어모델, 거대언어모델, 커리큘럼 러닝, LoRA, 데이터 이질성

• First Author: Daegeon Kim, Corresponding Author: Namgyu Kim
*Daegeon Kim (gun9809@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
*Namgyu Kim (ngkim@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
• Received: 2024. 02. 05, Revised: 2024. 03. 04, Accepted: 2024. 03. 04.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

## I. Introduction

컴퓨팅 자원의 발전으로 방대한 양의 데이터 저장 및 복잡한 연산 처리가 가능해지면서, 딥 러닝(Deep Learning) 기술을 활용한 연구가 활발히 이루어지고 있다. 딥 러닝 기술의 발전은 자연어 처리(Natural Language Processing) 분야에도 영향을 주며, 딥 러닝 기술을 활용한 언어모델(Language Model) 연구를 통해 혁신적인 성과를 달성하고 있다. 최근에는 언어모델을 활용하기 위한 연구가 활발히 이루어지면서, 언어모델을 분석 과제(Downstream Task)에 적용하여 문제를 해결할 때 모델의 규모가 커질수록 좋은 성능을 달성할 수 있다는 사례가 보고되고 있다[1]. 이에 따라 점차 큰 규모의 언어모델이 개발되고 있으며 특히, ChatGPT가 출시된 이후 제목이나 초록에 거대 언어모델(Large Language Model)이 포함된 arXiv 논문의 평균 발행 수가 하루에 0.40에서 8.58로 증가하는 등 거대 언어모델을 분석 과제에 활용하기 위한 관심이 급격히 증가하고 있다[2].

일반적으로 언어모델은 방대한 양의 데이터로 사전학습(Pre-training) 과정을 거치고, 사전학습이 완료된 모델은 분석 과제에 적용하기 위해 미세조정(Fine-tuning)을 수행하는 방법으로 활용한다[3]. 미세조정은 사전학습 모델을 분석 과제에 활용하기 위해, 특정 과제에 최적화하는 방향으로 모델의 가중치를 갱신시키는 방법이다. 하지만, 실제 현장에서는 언어모델의 미세조정에 필요한 비용과 자원이 한정적이라는 한계가 있다. 특히, 거대 언어모델의 경우, 큰 규모의 가중치를 갱신하기 위해 필요한 시간과 비용의 부담이 더욱 클 수밖에 없다. 이에 따라, 미세조정의 효율 및 효과를 극대화하기 위해 다양한 연구가 이루어지고 있는데, 구체적으로 모델의 사전학습 가중치를 고정하고 특정 가중치를 삽입한 뒤 학습 가능한 가중치의 규모를 줄여 학습 효율을 높이는 방법[4], 한 개의 모델로 여러 개의 과제를 학습할 때의 성능을 유지 및 개선하는 방법[5-7], 그리고 데이터의 학습 순서를 조정하여 미세조정의 성능을 개선하는 방법[8] 등의 연구를 들 수 있다.

딥 러닝 모델의 크기가 증가함에 따라 갱신해야 하는 가중치의 수가 기하급수적으로 증가하고, 이로 인해 충분한 학습이 이루어지기 어렵다는 한계[9]가 지적되어 왔다. 이를 극복하기 위한 효과적인 학습 전략의 필요성이 주목받는 가운데, 최근에는 학습 데이터의 순서를 조정하여 성능을 개선하기 위한 연구가 이루어지고 있다. 이 중 가장 괄목할 만한 접근으로, 학습 난이도에 따라 순서를 조정하여 점차적으로 학습하는 방법론인 커리큘럼 러닝(Curriculum Learning)을 들 수 있다. 커리큘럼 러닝의 핵심은 인간이 기초적인 개념을 습득한 뒤 이를 바탕으로 어려운 문제를 해결하는 것처럼, 딥 러닝 모델도 쉬운 난이도의 학습을 먼저 수행하고 어려운 난이도의 학습을 나중에 수행함으로써 학습 성능을 향상시킬 수 있다는 것이다. 최근 커리큘럼 러닝을 적용하기 위한 다양한 관점의 연구가 이루어지고 있으며, 특히 학습 난이도를 정의하고 측정하기 위한 시도가 다수 제안되고 있다.

초기의 커리큘럼 러닝 방법론은 이미지 데이터의 모양이라는 사전 지식을 통해 학습 난이도를 구분하였다. 구체적으로 원, 정사각형 등 기본 이미지는 쉬운 난이도의 이미지로 가정하고, 타원, 직사각형 등 변형된 모습의 이미지는 상대적으로 어려운 난이도의 이미지로 가정한다. 하지만 이와 같은 가정 및 사전 지식으로 학습 데이터의 난이도를 구분할 수 있는 경우는 매우 제한적이며, 실제 학습 환경에서는 데이터의 형태나 해결할 과제 등 다양한 요인을 함께 고려하여 난이도가 측정되어야 한다. 또한 인간이 생각하는 난이도와 모델이 이해하는 난이도 간 괴리가 발생하는 경우도 존재할 수 있다.

이러한 어려움을 극복하고자 학습 과정에서의 손실 값을 난이도로 가정하고, 모델이 능동적으로 데이터의 난이도를 측정하는 방법론[10]이 등장하였다. 구체적으로 본 방법론은 특정 임계값을 넘지 않는 손실 값의 데이터만 초기 학습에 반영하고, 이후 높은 손실 값을 가진 데이터도 학습에 반영하는 방식으로 난이도를 학습에 반영한다. 또한 사전 정보 없이 데이터 난이도를 측정하는 방법과 사전 정보를 이용하여 난이도를 측정하는 방법을 결합하여 모델의 학습 효과를 향상시킨 방법론[11] 등 데이터의 학습 난이도를 측정하여 학습 성능을 향상시키기 위한 다양한 시도가 이루어지고 있다.

한편, 실제 현장에서는 모델의 효과성뿐 아니라 효율성[12]도 함께 고려되어야 하므로, 모델의 경량화를 통해 학습의 시간과 비용을 절약하는 방법이 널리 사용되고 있다. 이처럼 경량화를 위해 모델의 특징이나 형태가 변경된 경우, 커리큘럼 러닝을 통한 성능 향상이 일반 모델과 경량 모델에서 상이한 양상으로 나타날 가능성이 있다. 이는 일반 모델에 커리큘럼 러닝을 적용한 연구의 결과를 경량 모델 등 최신 모델의 활용에 그대로 대입할 수 없으며, 최신 경량 모델에 대해 커리큘럼 러닝을 효과적으로 적용하는 방식에 대한 연구가 해당 환경에서 별도로 수행될 필요가 있음을 의미한다.

이에 본 연구는 다음의 두 가지 측면에서 기존 연구의 한계를 극복하고 현장 수요에 부응하고자 한다. 우선 텍스트

데이터에 대한 사전학습 언어모델의 미세조정 과제에서 각 데이터의 학습 난이도를 측정하는 방안을 제시하고, 제시한 기준에 따라 커리큘럼 러닝을 적용했을 때의 모델 성능 개선 여부를 평가한다. 또한 제안 방법론을 모델의 가중치를 효율적으로 갱신하여 훈련하는 최신 방법론인 LoRA의 미세조정에 적용함으로써, 커리큘럼 러닝을 통한 일반 모델과 경량 모델의 성능 개선 양상을 비교하고자 한다.

본 논문의 이후 구성은 다음과 같다. 2장에서는 사전학습 언어모델과 거대 언어모델 및 커리큘럼 러닝의 기존 연구를 소개하고, 모델을 효율적으로 훈련시키는 LoRA 방법론을 소개한다. 본론인 3장에서는 본 연구에서 제안하는 언어모델의 데이터 이질성 기반 커리큘럼 러닝 방법론을 소개한다. 4장에서는 제안 방법론의 성능 평가 결과를 제시하고, 5장에서는 본 연구의 기여와 한계를 요약한다.

## II. Preliminaries

### 1. Pre-trained Language Model and Large Language Model

사전학습 언어모델이란 언어의 범용적인 지식을 모델이 이해하도록 학습시킨 언어모델이다. 사전학습이 완료된 모델은 분류, 요약, 개체명 인식, 기계 번역, 그리고 생성 등 구체적인 과제에 대한 미세조정을 통해 문제 해결에 적용한다. 이러한 언어모델은 사전에 대용량 텍스트 데이터를 학습했기 때문에 언어의 범용적인 지식을 이해하고 있으며, 추후 상대적으로 작은 규모의 데이터를 사용한 미세조정을 통해서도 만족스러운 성과를 나타낼 수 있다. 이러한 사전학습 언어모델은 트랜스포머 모델[13]의 인코더 구조에서 파생된 모델과 디코더 구조에서 파생된 모델의 두 가지 유형으로 구분된다.

트랜스포머 모델의 인코더 구조에서 파생된 모델은 대표적으로 BERT[1]를 들 수 있다. BERT는 다양한 자연어 처리 과제에서 SOTA(State-of-the-art)를 달성하며 우수한 성능을 입증하고 있으며, 특히 자연어 이해 과제에서 좋은 수행 능력을 보이며 활발히 사용되고 있다. BERT의 핵심 아이디어는 후속 연구에서도 꾸준히 다루어지고 있으며, 구체적으로 RoBERTa[14], ALBERT[15], DistilBERT[16], 그리고 DeBERTa[17] 등 BERT의 성능을 다양한 방법으로 개선하는 연구가 보고되고 있다.

한편 트랜스포머 모델의 디코더 구조에서 파생된 모델의 대표적 예로 GPT를 들 수 있다. GPT는 디코더 구조의 특징에 따라 이전 토큰 정보를 통해 다음 토큰을 예측하는 방식으로 학습한다. GPT도 역시 사전학습 언어모델이기 때문에, 대용량 텍스트 데이터에 대해 사전학습을 완료한 후 미세조정을 통해 특정 과제에 최적화하는 형태로 사용된다. GPT-1[18] 모델이 등장한 이후, 모델 구조를 부분적으로 수정하고 모델의 가중치 및 학습 데이터 규모를 증가시킨 GPT-2[19] 모델이 발표되었다. 또한 기존의 미세조정 방식을 벗어난 제로 샷 러닝(Zero-Shot Learning)[원 샷 러닝(One-Shot Learning), 그리고 퓨 샷 러닝(Few-Shot Learning)[20-22] 등 새로운 학습 방식을 적용한 GPT-3[23] 모델은 매우 우수한 성능을 보이며 학계와 업계에서 널리 사용되고 있다.

GPT의 발전에 따라 모델의 규모도 동시에 커지면서 GPT와 유사한 크기의 다양한 언어모델이 발표되고 있으며, 언어모델은 규모가 커질수록 더 우수한 성능을 달성할 수 있다는 사례가 보고되면서 최근에는 거대 언어모델에 대한 관심이 급증하고 있다. Meta AI에서 만든 모델인 LLaMA[24] 역시 대표적인 거대 언어모델 중 하나이다. LLaMA는 GPT-3와 같은 트랜스포머의 디코더 구조를 따르고 있으나, 부분적인 구조 변경과 모델 규모에 따른 학습 데이터 규모 조정을 통해 성능을 향상시켰다. LLaMA의 등장과 더불어 Alpaca[25], Vicuna[26], 그리고 LLaMA를 발전시킨 LLaMA2[27] 등 거대한 규모의 언어모델이 속속 발표되고 있으며, 국내에서도 거대 언어모델에 대한 연구가 활발히 이루어지고 있다. 국내 거대 언어모델 연구의 대표적인 예로 개인 연구자의 KoLLaMA[28]와 KoAlpaca[29], NAVER의 HyperCLOVA[30], KAKAO의 KoGPT[31], 그리고 KT의 믿음[32] 등을 들 수 있으며 이러한 연구를 필두로 거대 언어모델의 개발 및 활용에 관심이 꾸준히 높아지고 있다.

### 2. Curriculum Learning

커리큘럼 러닝[8]이란 데이터의 난이도에 따라 순서를 조정한 뒤, 이를 순차적으로 학습하여 딥 러닝 모델의 성능을 개선하는 방법론이다. 커리큘럼 러닝의 원리는 인간의 학습 과정과 매우 유사하다. 예를 들어, 인간은 초기에 기초적인 개념과 쉬운 문제 해결을 학습하며 문제 해결 능력을 향상시키고, 이후 점차 어려운 문제의 해결에 대한 학습을 수행한다. 하지만 일반적인 딥 러닝 모델은 학습 순서에 대한 별도의 고려 없이, 즉 데이터의 순서를 고려하지 않고 랜덤하게 데이터를 섞어 학습을 수행한다. 커리큘럼 러닝은 일반적인 모델 학습 방법과 달리 쉬운 난이도부터 점차 어려운 난이도의 데이터를 통해 학습을 수행하고자 하는 시도이며, 난이도에 따른 학습 전략과 관련된

방법론[33]이 다수 등장하고 있다.

난이도 기반 커리큘럼 러닝을 적용하기 위해서는, 당연히 데이터의 난이도 측정 방안이 마련되어야 한다. 데이터의 난이도는 사전 지식을 미리 제공하여 학습에 활용하는 방법, 또는 학습 과정에서 딥 러닝 모델이 직접 난이도를 측정하면서 학습하는 방법으로 측정된다. 커리큘럼 러닝의 전통적인 방법론은 난이도에 대한 사전 정보를 통해 데이터를 나누어 학습한다. 구체적으로 난이도에 따라 데이터를 미리 구분하고, 학습 시 난이도에 따라 데이터의 순서를 조정하여 쉬운 난이도에서 점차 어려운 난이도의 학습을 수행한다. 예를 들어 <Fig. 1>은 쉬운 난이도의 이미지를 먼저 학습하고 점차 어려운 이미지를 학습하는 커리큘럼 러닝을 통해, 난이도의 구분 없이 혼합된 전체 이미지를 한꺼번에 학습할 때 비해 성능 향상을 가져올 수 있음을 보인다.

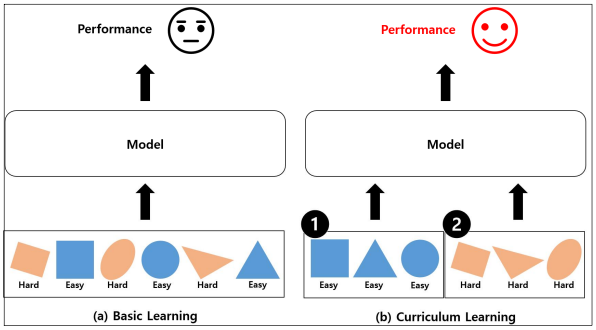


Fig. 1. An Example of Curriculum Learning

한편, 모델에 데이터의 난이도 정보를 사전에 제공하지 않고, 모델이 직접 데이터 난이도를 측정하며 학습하는 방법[10]도 제안된 바 있다. 일반적으로 모델은 순전파를 통해 예측값을 출력하고, 예측값과 실제 정답 간 손실을 계산하여 해당 손실값을 최소화하는 방향으로 학습을 수행한다. 구체적으로 해당 방법론은 입력된 데이터의 손실값이 임계값보다 낮은 경우, 해당 데이터를 난이도가 낮은 데이터로 가정하여 학습에 우선 반영한다. 학습이 진행됨에 따라 임계값을 점차 증가시키며, 이에 따라 학습 후반부에는 손실값이 다소 큰 데이터도 학습에 포함된다. 해당 방법론은 모델이 데이터 난이도를 직접 측정할 수 있는 방법을 제안했다는 점에서 기존의 방법론과 차이를 보인다. 이후 사전 정보를 제공하면서 동시에 난이도를 직접 측정할 수 있는 방법론[11]이 등장하는 등, 학습 난이도를 고려한 학습 최적화 전략 연구가 활발히 이루어지고 있다.

## 3. LoRA: Low-Rank Adaptation of Large Language Models

언어모델은 방대한 양의 텍스트 데이터를 통해 사전학습되어, 전이학습을 통해 다양한 과제에 활용되고 있다. 하지만 언어모델의 규모가 점차 커짐에 따라, 전이학습 수행에 막대한 비용과 시간이 발생한다는 치명적인 단점이 더욱 부각되고 있다. 이로 인해 거대한 규모의 모델을 효율적으로 학습할 수 있는 연구가 주목받고 있으며, 최근에는 모델 전체가 아닌 모델의 일부만 학습하는 다양한 방안이 제안되고 있다.

이러한 연구 중 대표적인 것으로, 언어모델의 사전학습 가중치를 고정하고 학습 가능한 더 낮은 랭크(Rank)의 가중치를 삽입하여 학습하는 방법론인 LoRA(Low-Rank Adaptation)[3]를 들 수 있다. LoRA는 모델의 가중치 규모에 비해 실제 가중치 행렬의 내재적인 차원은 더 낮다는 연구[34,35]를 배경으로 제안되었으며, 학습 과정에서 모델의 가중치 변화량도 내재적으로 더 낮은 랭크를 가진다는 특징에 기반을 두고 동작한다.

LoRA 방법론의 개요는 <Fig. 2>와 같다. 먼저 사전학습 가중치로부터 더 낮은 랭크를 갖는 가중치 A와 B를 생성한다. 입력 데이터는 사전학습 가중치와 가중치 A와 B에 전달되고, 두 가중치의 출력을 합하는 방법으로 순전파가 이루어진다. 이후 학습 과정에서 모델은 사전학습 가중치를 고정하고, 더 낮은 랭크의 가중치 A와 B만 갱신하며 학습을 진행한다.

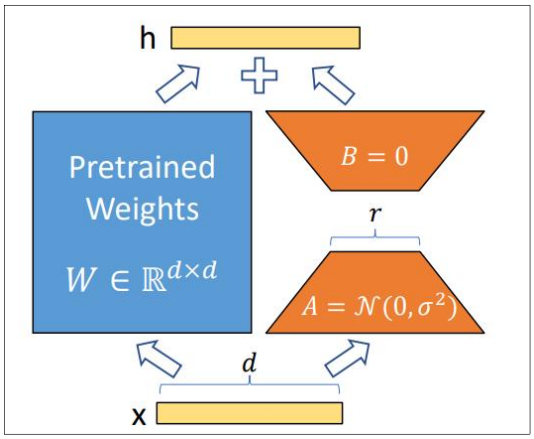


Fig. 2. Architecture of LoRA Method

학습 가능한 가중치를 삽입하는 기존 연구[36]는 모델의 특정 층을 추가함에 따라 연산량이 증가하여 추론 시 지연이 발생하는 단점이 있다. 하지만 LoRA는 추론 시 고정된 사전학습 가중치와 학습 가능한 가중치를 병합하여 사용

하기 때문에 추론 지연을 방지할 수 있다. 또한, 기존 가중치보다 더 작은 규모로 대체된 작은 가중치만을 학습하기 때문에 학습 시간 및 비용이 단축된다. 이러한 특징으로 인해 LoRA 방법론은 일반적인 학습 방법에 비해 더욱 효율적으로 좋은 성능을 달성할 수 있음이 알려져 있으며, 구체적으로 GLUE 벤치마크에서 제공하는 과제에 대한 성능 실험 결과 LoRA 방법론이 정확도 지표 측면에서 우수함을 보였다.

## III. Proposed Method

### 1. Research Process

본 장에서는 서로 다른 도메인의 데이터 간 유사도를 통해 난이도를 정의하고, 이를 반영하여 LoRA 미세조정을 수행하는 커리큘럼 러닝 방법론을 소개한다. 제안 방법론의 전체적인 과정은 <Fig. 3>과 같다.

제안 방법론은 전통적인 방식에 따라 사전학습을 수행하는 Phase 1, 사전학습용 데이터의 중심 벡터를 계산하고 이 중심 벡터와 각 미세조정용 데이터와의 코사인 유사도(Cosine Similarity)를 비교하는 Phase 2, 그리고 산출한 유사도에 따라 커리큘럼 러닝 기반 LoRA 미세조정을 수행하는 Phase 3의 세 단계로 구성된다. 구체적으로 Phase 1은 사전학습용 데이터를 통해 일반적인 방식에 따라 사전학습을 수행한다(1). 이후 Phase 2에서는 (2) 사전학습용 데이터와 미세조정용 데이터 각각에 대해 전처리 및 임베딩을 진행하고, (3) 사전학습용 데이터의 벡터 중심을 계산하여 기준점(Reference Point)을 설정한다. 다음으로 (4) 미세조정용 데이터의 각 벡터값과 이 기준점의 코사인 유사도를 기준으로 각 데이터의 난이도를 계산한다. 이후 Phase 3에서는 (5) 각 데이터의 난이도에 따라 미세조정용 데이터의 순서를 재정렬하고, (6) 마지막으로 이 순서에 따라 미세조정을 수행하는 커리큘럼 학습 기반 LoRA 미세조정을 수행한다.

각 단계에 대한 구체적인 내용은 다음 절에서 예시와 함께 설명하며, 실제 데이터에 대해 제안 방법론을 적용한 성능 평가 결과는 4장에서 제시한다.

### 2. Pre-training

본 절에서는 언어모델의 일반적인 사전학습 수행 과정(단계 1)을 소개한다. 본 방법론에서는 사전학습 언어모델인 RoBERTa 구조의 모델을 사용하였으며, 특히 한국어로

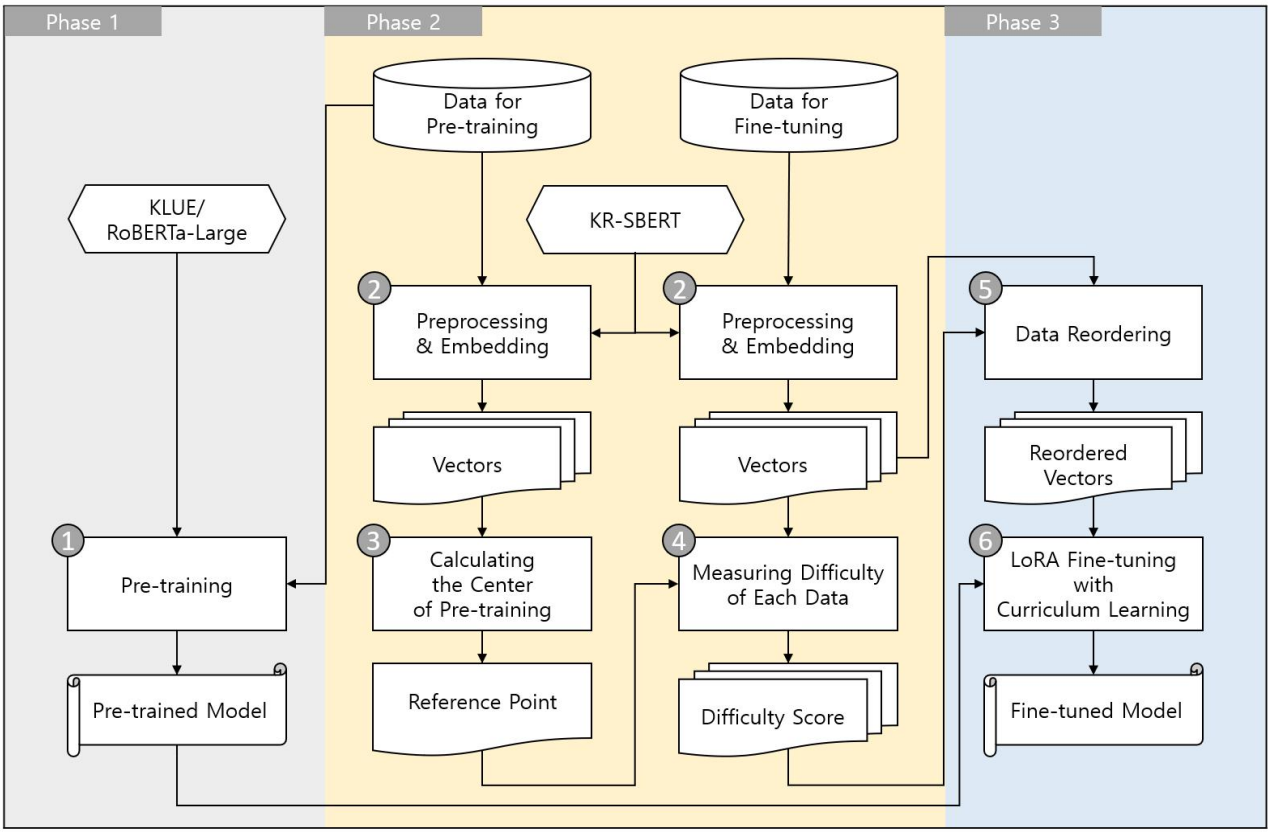


Fig. 3. Overall Research Process

이루어진 데이터에서 좋은 분석 성능을 내고자 한국어 특화 언어 모델인 KLUE/RoBERTa[37] 모델을 사용하였다.

RoBERTa의 사전학습 방법은 BERT의 사전학습 방법 동일하지만 몇 가지 수정 사항이 있다. BERT의 사전학습은 MLM(Masked Language Mode)과 NSP(Next Sentence Prediction)라는 두 가지 방법을 통해 이루어진다. 먼저, MLM은 학습할 모든 입력 토큰 중에서 일부의 토큰을 무작위로 마스킹 처리한 뒤, 주변 문맥을 보고 마스킹 처리된 토큰을 예측하는 방식으로 학습을 수행한다. 하지만 BERT는 학습이 다시 진행될 때 똑같은 마스킹 토큰을 사용한다는 한계를 갖는다. 이와 달리 RoBERTa는 학습할 때마다 동일한 데이터에 대해 마스킹을 다르게 적용하는 동적 마스킹을 활용하여 사전학습을 수행한다.

BERT의 두 번째 사전학습 방법인 NSP는 문장 두 개로 이루어진 데이터를 입력받아 문장의 관계성을 예측하는 학습을 수행한다. 하지만 RoBERTa는 NSP를 수행하지 않는 것이 성능 개선에 도움을 준다는 연구 결과에 따라, NSP를 제외한 방식으로 사전학습을 수행한다.

## 3. Preprocessing and Embedding

본 절에서는 입력할 데이터의 전처리 및 임베딩 진행 과정(단계 2)을 설명한다. 텍스트 데이터의 경우 언어의 다양성, 비표준적 언어 표현, 그리고 비구조화된 형식으로 인해 분석 데이터의 품질이 보장되지 않는다. 따라서, 분석에 적합한 형태로 데이터를 가공하기 위해, 불필요한 정보를 제거하는 등의 전처리 과정이 필수적이다. 본 연구에서는 한국어의 자음과 모음만 존재하는 데이터 제거, 문자나 공백이 아닌 특수 문자 제거, HTML 태그 제거, 연속된 공백을 단일 공백으로 대체, 그리고 텍스트의 맨 앞 혹은 맨 뒤의 공백을 제거하는 전처리를 수행한다.

전처리를 통해 가공이 완료된 데이터는 모델 입력 및 코사인 유사도 계산이 가능한 형태로 변환되어야 하며, 이를 위해 텍스트 데이터를 벡터 형태로 변환하는 임베딩을 수행한다. 구체적으로 텍스트 데이터를 토큰 단위로 구분한 후, 해당 토큰 단위의 데이터를 모델에 입력하여 벡터로 변환한다. 본 연구에서는 문장 임베딩의 성능이 우수하다고 알려진 SBERT[38]를 통해 텍스트 데이터 임베딩을 수행했으며, 구체적으로 한국어 데이터에 특화된 KR-SBERT[39]를 사용하였다. <Fig. 4>는 텍스트 데이터의 전처리 및 임베딩 결과를 나타내는 예시이다.

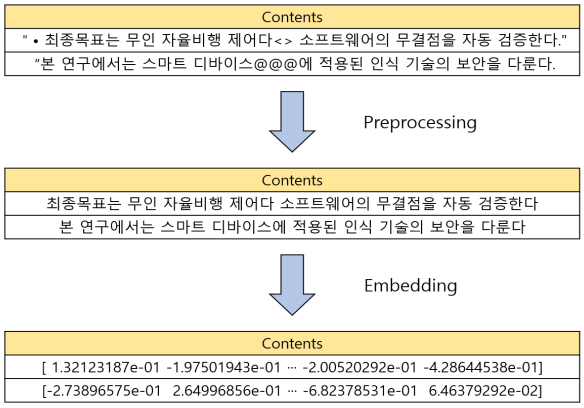


Fig. 4. Example of Data Preprocessing and Embedding

## 4. Measuring Difficulty of Each Data

본 절에서는 사전학습용 데이터의 벡터 중심을 계산(단계 3)하고, 미세조정용 데이터의 각 벡터값과 사전학습용 데이터의 중심점 간 코사인 유사도를 기준으로 난이도를 계산하는 과정(단계 4)을 설명한다.

본 연구에서 대상으로 하는 분석 과제는 사전학습이 이루어진 모델을 특정 도메인의 문서 분류를 통해 미세조정하는 것이다. 이때 해당 도메인의 고유한 특질(Feature)을 뚜렷하게 나타내고 있는 데이터는 분류가 상대적으로 용이하고, 이와 반대로 고유한 특질을 충분히 포함하지 못한 데이터는 분류의 난이도가 높은 것으로 인식할 수 있다. 즉 데이터의 도메인 고유성이 높을수록 분류 난이도가 낮은 데이터로 간주하고자 하며, 도메인 고유성은 사전학습 모델의 학습에 사용된 데이터와의 차별성으로 측정하고자 한다. 즉 본 연구에서는 사전학습에 사용된 데이터와의 이질성이 클수록 도메인 고유성이 높고, 따라서 분류 난이도가 낮은 데이터로 간주한다.

<Fig. 5>는 사전학습에 사용된 문서 벡터와 미세조정에 사용할 문서 벡터를 평면에 도식화한 예이다. <Fig. 5>의 좌측에 있는 파란색 원은 사전학습에 사용된 문서 벡터를, 우측에 있는 회색 원은 분석할 도메인의 문서 벡터를 나타내는 것으로 가정한다. 그림에서 벡터①은 사전학습용 문서의 중심 벡터를 나타내며, <Fig. 6(a)>과 같이 사전학습에 사용된 각 문서 벡터의 평균을 통해 산출한다. 다음으로 <Fig. 5>의 벡터②와 벡터③은 미세조정용 데이터의 일부를 나타내며, <Fig. 6(b)>와 같이 이들 각각에 대해 중심 벡터인 벡터①과의 코사인 유사도를 계산하여 각 데이터의 유사도 값을 산출한다. 전술한 바에 따라 코사인 유사도가 클수록 해당 데이터의 난이도가 높은 것으로 해석되며, 이 정보는 이후 단계에서 모델 학습 시 데이터의 입력 순서 조정에 사용된다.

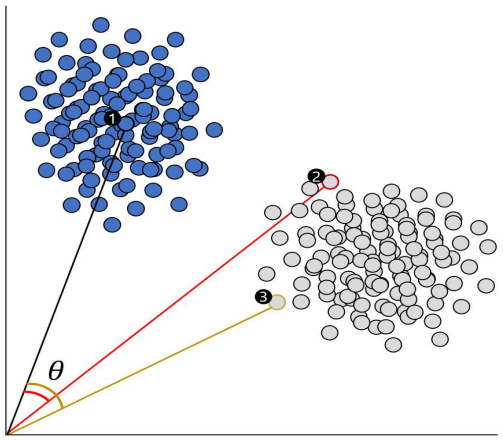


Fig. 5. Example of Calculating the Center Vector and the Cosine Similarity

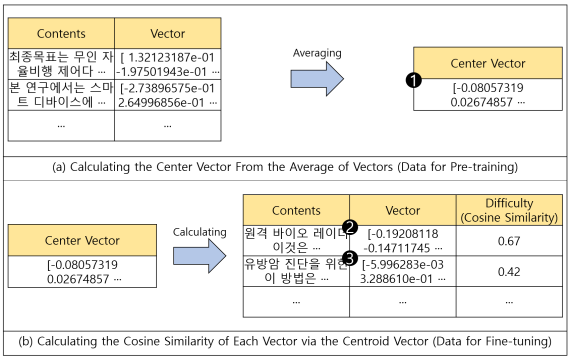


Fig. 6. Process of Calculating the Center Vector and the Cosine Similarity

## 5. Curriculum Learning-based Fine-tuning

본 절에서는 각 데이터의 난이도에 따라 미세조정용 데이터의 순서를 재정렬하는 과정(단계 5)과, 이 순서에 따라 커리큘럼 러닝 기반 LoRA 미세조정(단계 6)을 수행하는 과정을 설명한다. 제안하는 방법론의 구체적인 동작 예는 <Fig. 7>과 같다.

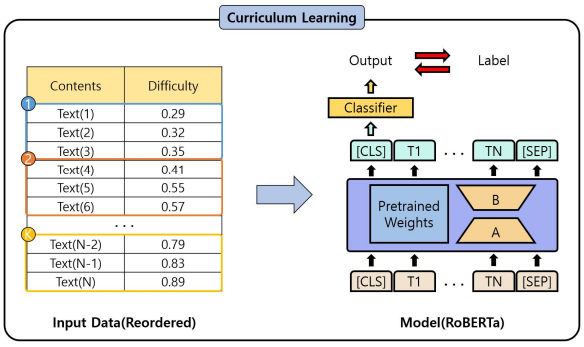


Fig. 7. Process of Curriculum Learning-based LoRA Fine-tuning

<Fig. 7>의 우측은 전통적인 RoBERTa 모델의 구조로 해당 모델을 분류 과제에 미세조정 시키는 과정을 나타내며, 좌측은 모델의 학습에 사용되는 데이터의 구조를 나타낸다. 일반적인 경우라면 입력 데이터의 순서는 크게 중요하지 않겠지만, 본 연구에서 다루는 커리큘럼 러닝의 경우 입력 데이터의 순서가 매우 중요하다. 따라서 본 연구에서는 입력 데이터를 난이도 오름차순으로, 즉 코사인 유사도가 낮은 데이터부터 학습에 사용하며, 이러한 내용이 <Fig. 7>의 좌측에 표현되어 있다.

하지만, 딥 러닝 모델은 일반적으로 데이터를 무작위로 섞는 작업을 통해 학습이 이루어지기 때문에, 일반적인 학습 방법을 적용할 경우 난이도에 따른 학습 순서가 뒤섞이게 된다. 반대로 데이터를 섞는 과정을 생략하고 반복 학습을 진행할 경우, 모델이 학습되면서 데이터의 순서를 기억하게 되어 순서 정보에 의존하는 경향이 생길 수 있다. 순서 정보에 의존하는 학습은 결국 과적합 현상으로 이어져, 모델의 일반화 능력이 떨어지는 부작용이 발생한다.

즉 전통적인 학습 데이터 구성 방법으로는 본 연구에서 제안하는 방법론을 구현할 수 없으므로, 이를 해결하기 위한 새로운 학습 데이터 구성 방안이 함께 제시되어야 한다. 이에 본 연구에서는 매 에폭(Epoch)마다 데이터를 섞되, 순서 정보를 가진 슬롯 내에서만 데이터를 섞는 Inner Slot Shuffling 방법을 새롭게 제안한다(Fig. 8).

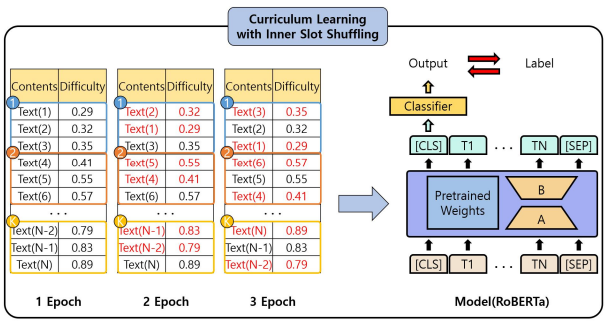


Fig. 8. Example of Inner Slot Shuffling

<Fig. 8>에서 Inner Slot Shuffling이 이루어지는 과정은 다음과 같다. 우선 학습할 데이터를 난이도에 따라 오름차순으로 정렬한다. 정렬이 완료된 데이터를 K개의 슬롯으로 묶으며, 본 예에서는 한 슬롯당 3개의 데이터가 배정된 경우를 보이고 있다. 이렇게 정렬된 순서대로 한 에폭의 학습을 수행한 후 데이터의 재정렬이 이루어지는데, 이때 각 데이터는 자신이 속한 슬롯 내에서만 순서가 변경된다. 이러한 방법을 통해 에폭마다 매번 새로운 순서의 데이터를 학습에 사용하면서도, 전반적으로 난이도가 낮은 데이터를 학습 초기에 사용하는 전략을 구현할 수 있다.

# IV. Experiment

## 1. Experiment Overview

본 장에서는 3장에서 소개한 제안 방법론인 유사도 기반 커리큘럼 러닝을 실제 데이터에 적용한 실험 결과와 성능 분석 결과를 소개한다. 실험 데이터는 도메인 간 이질성을 강조하기 위해, 도메인 전문성을 중요하게 다룬 기존 연구의 결과를 참고하여 2011년부터 2020년까지 수행된 국가 R&D 과제 전문 문서를 사용했다[40]. 구체적으로 사전학습에는 '정보통신' 분야의 전문 문서 5,000건, 미세조정에는 '보건의료' 분야의 전문 문서 4,917건을 사용하였다. 실험 환경에 사용된 언어는 Python이며 자세한 하드웨어 및 소프트웨어 환경은 <Table 1>과 같다. 커리큘럼 러닝 기반 학습 모델의 성능을 평가하는 실험의 전체 개요는 <Fig. 9>와 같다.

Table 1. Experimental Environments

| | | |
|---|---|---|
| HW | GPU | Tesla V100 |
| | CPU | 32 Cores |
| | Memory | 320GB |
| SW | OS | Linux Ubuntu 20.04.5 LTS |
| | Python | 3.8.10 |
| | Pytorch | 1.14.0a0+410ce96 |
| | Transformers | 4.33.2 |
| | PEFT | 0.5.0 |

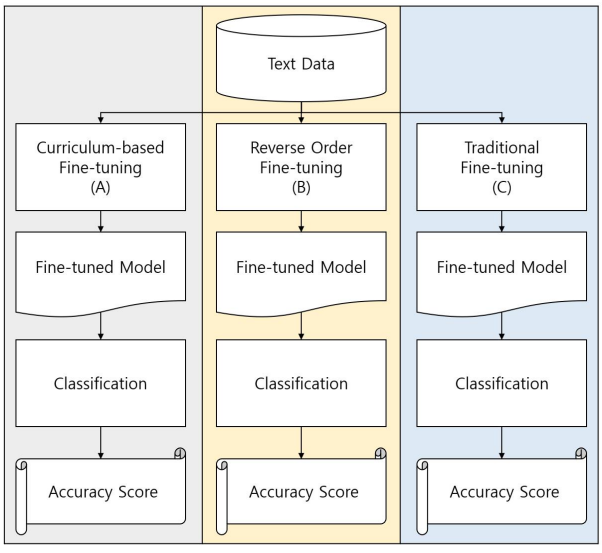


Fig. 9. Overall Process of Performance Evaluation

<Fig. 9>의 (A)는 본 연구에서 제안하는 방법론을 적용한 모델로, 코사인 유사도 값을 오름차순으로 정렬한 미세조정용 데이터에 대해 커리큘럼 러닝을 적용하여 모델을 학습한다. <Fig. 9>의 (B)와 (C)는 제안 방법론과의 성능 비교를 위해 수행한 실험이다. 제안 방법론인 (A)가 오름차순으로 정렬한 데이터를 통해 커리큘럼 러닝을 적용한 실험이라면, (B)는 이와 반대로 미세조정용 데이터를 코사인 유사도 값의 내림차순으로 정렬하여 커리큘럼 러닝을 적용한다. 마지막으로 (C)는 전통적인 미세조정 방법으로, 데이터의 순서를 무작위로 섞어 모델을 학습시킨 뒤 성능을 평가하는 과정이다.

본 실험에 앞서 실험 데이터에 대한 전처리, 임베딩, 그리고 레이블링을 수행하였으며, <Table 2>는 전처리 및 임베딩이 완료된 사전학습용 데이터의 예시이다.

Table 2. Example of Preprocessing and Embedding

| Contents | Clean Contents | Vector | Label |
|---|---|---|---|
| ? 소셜 및 정보 네트워크 원천 … | 소셜 및 정보 네트워크 원천 … | [3.753447e-02 -5.59128e-01 … | 3 |
| ○ 무장투하 비행시험 수행 … | 무장투하 비행시험 수행 … | [0.10200754 -0.13753854 … | 0 |
| - 의료, 생명과학, 비즈니스, … | 의료 생명과학 비즈니스 … | [-6.06542e-01 -2.545919e-01 … | 2 |

구체적으로는 문서의 내용 중 한국어의 자음과 모음만 존재하는 데이터 제거, 문자나 공백이 아닌 특수 문자 제거, HTML 태그 제거, 연속된 공백을 단일 공백으로 대체, 그리고 텍스트의 맨 앞 혹은 맨 뒤의 공백을 제거하는 전처리를 수행했다. Clean Contents 열은 전처리가 완료된 데이터의 결과 예시이다. 전처리를 완료한 데이터는 코사인 유사도 계산을 위해 임베딩을 수행하여 벡터 형태로 변환하였다. 고품질의 문장 임베딩을 위해 KR-SBERT를 사용하여 문서를 토큰 단위로 분리 후, 분리된 각 토큰에 대해 임베딩을 수행하여 하나의 벡터 형태로 표현했다. Vector 열은 전처리가 완료된 문서 데이터에 대한 임베딩 벡터 결과 예시이다.

본 실험에서는 정보통신 분야의 전문 문서에 대한 분류 미세조정을 수행하고 이를 해당 모델의 사전학습으로 간주했는데, 이는 적은 양의 데이터만으로 이후 미세조정에 사용될 데이터와의 이질성이 크게 나타나는 사전학습 모델을 획득하기 위함이다. 구체적으로는 정보통신 분야의 전문 문서에 대해 과학기술표준분류 체계에 따라 총 5개의 영역으로 나누어진 문서에 5개 레이블을 적용하여 분류 미세조정을 수행하였다.

이후 실제 미세조정은 보건의료 분야 문서에 대한 분류 작업을 통해 수행하였으며, 이 과정에서 컨티뉴얼 러닝(Continual Learning)의 Class-Incremental Learning[4] 시나리오를 적용하여 해당 실험 환경을 구성하였다. 구체적으로 사전학습용 데이터는 '0 ~ 4'의 레이블로, 미세조정용 데이터는 '5 ~ 9'의 레이블로 서로 겹치지 않게 부여하였다. 레이블 부여 작업까지 완료된 사전학습용 데이터는 최종적으로 훈련용 3,000개, 검증용 1,000개, 그리고 평가용 1,000개를 사용하였다.

정보통신 분야의 데이터를 통해 사전학습을 수행한 실험 설정은 <Table 3>과 같다.

Table 3. Hyperparameters for Pre-training

| Model | KLUE/RoBERTa-large |
|---|---|
| Task | Classifcation (5 labels) |
| Batch Sizes | 16 |
| Epochs | 10 |
| Optimizer | AdamW |
| Learning Rate | 0.0004 |
| Scheduler | Linear |
| Warmup Ratio | 0.1 |
| Max Length | 128 |

사전학습에 사용된 모델은 한국어 도메인에서 우수한 성능을 보이는 KLUE/RoBERTa-large 모델이며, 과학기술표준분류 체계에 따라 나누어진 5개의 레이블을 분류하는 미세조정을 진행했다. 또한 분류기의 각 클래스에 고유한 출력 단위를 부여하고자, 분류기의 출력 차원은 10으로 설정하여 총 10가지의 레이블을 분류하는 학습을 진행했다. 학습에 필요한 하이퍼파라미터(Hyperparameter) 중 학습률(Learnnig Rate)은 1e-3에서 1e-6 사이의 값 중 가장 높은 성능을 나타낸 학습률인 4e-4로 설정하였으며, 검증용 데이터 분류 정확도가 가장 우수하게 나타난 에폭 5의 모델을 최종 미세조정 모델로 선정하였다.

## 2. Results of Curriculum-based Fine-tuning

본 절에서는 사전학습이 완료된 모델을 통해 유사도 기반 커리큘럼 러닝을 적용한 과정을 소개한다. <Table 4>는 난이도에 따라 정렬한 미세조정용 데이터의 예시이다.

Table 4. Example of Train Data

| Clean Contents | Difficulty | Label |
|---|---|---|
| 보청기 시스템을 임상평가하고 이에 따른 결과를 통해 단위 개발 기술들 … | 0.828 | 5 |
| 최첨단 차원 스캐너를 사용하여 각종 사건사고에서 현장 증거 및 인체증거물 대한 영상증거물을 … | 0.760 | 8 |
| 조류인플루엔자 감염에 따른 장 상피세포 막 조직의 손상 및 조절 기전 분석 … | 0.356 | 6 |

전술한 바와 같이 본 연구에서는 사전학습용 데이터와의 이질성이 클수록 도메인 고유성이 높으며, 데이터 도메인 고유성이 높을수록 분류 난이도가 낮은 것으로 간주한다. 따라서 <Table 4>의 Difficulty 열은 사전학습용 데이터의 중심과 각 미세조정용 데이터와의 코사인 유사도로 측정되며, 제안 방법론인 모델 (A)는 쉬운 난이도의 데이터부터 순차적으로 모델이 학습할 수 있도록 Difficulty의 값을 오름차순으로 정렬하여 학습을 수행한다. 한편 비교 모델인 (B)는 Difficulty를 내림차순으로 정렬하여 학습을 수행하고, 모델 (C)는 Difficulty에 따른 순서를 고려하지 않는다. 가공이 완료된 미세조정용 데이터는 최종적으로 훈련용 2,950개, 검증용 983개, 그리고 평가용 984개를 사용하였다. 이후 보건의료 분야의 훈련용 데이터를 통해 5개 레이블에 대한 분류 미세조정이 이루어졌으며 구체적인 실험 개요는 <Table 5>와 같다.

Table 5. Hyperparameters of Fine-tuned Models

| | Full Fine-tuning | | | LoRA Fine-tuning | | |
|---|---|---|---|---|---|---|
| | (A) | (B) | (C) | (A) | (B) | (C) |
| Learning Rate | 3e-6 | 5e-6 | 5e-6 | 7e-5 | 5e-4 | 4e-4 |
| Model | KLUE/RoBERTa-large | | | | | |
| Task | Classification (5 labels) | | | | | |
| Batch Sizes | 16 | | | | | |
| Epochs | 10 | | | | | |
| Optimizer | AdamW | | | | | |
| Scheduler | Linear | | | | | |
| Warmup Ratio | 0 | | | | | |
| Max Length | 128 | | | | | |

본 연구에서는 제안 방법론을 일반 모델에 적용했을 때와 LoRA 적용 모델에 적용했을 때의 성능 변화 양상을 비교하기 위해, 총 6가지의 모델에 대한 실험을 설계했다. 사전학습 모델과 마찬가지로 학습률은 1e-3에서 1e-6 사이의 값 중 가장 높은 성능을 보이는 학습률을 선정하였고, 결정된 학습률로 각 모델의 최종 학습을 수행했다.

### 3. Performance Evaluation

본 절에서는 제안하는 방법론인 유사도 기반 커리큘럼 러닝을 완료한 모델의 성능을 다른 모델들의 성능과 함께 비교한 결과를 소개한다. 성능 비교 측정 척도는 분류 정확도를 사용하였으며, 평가에는 미세조정용 데이터 중 평가용 데이터 984개를 사용하였다. 방법론을 적용한 모델과 비교 모델과의 성능 평가 결과는 <Table 6> 및 <Fig 10>과 같다.

Table 6. Performance Comparison Result

| | Curriculum - based Fine-tuning (A) | Reverse Order Fine-tuning (B) | Traditional Fine-tuning (C) |
|---|---|---|---|
| Full Fine-tuning | 74.60% | 72.28% | 73.99% |
| LoRA Fine-tuning | 72.68% | 71.98% | 72.58% |

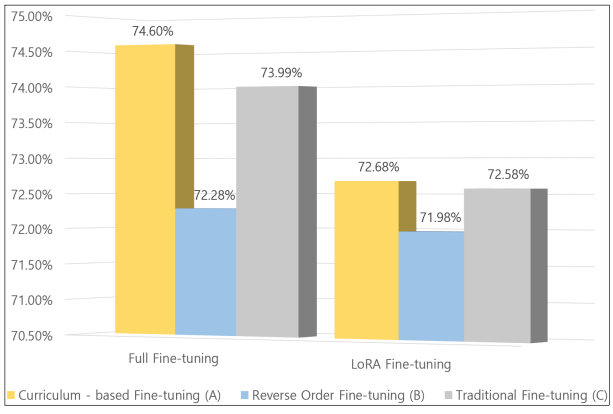


Fig. 10. Performance Comparison

실험 결과 제안 방법론인 오름차순 기반 커리큘럼 러닝을 통해 학습한 모델 (A)가 LoRA를 적용하지 않았을 때와 적용했을 때 모두 가장 우수한 성능을 나타내는 것을 확인하였다. 그리고 본 방법론을 반대로 적용한 모델 (B)는 전통적인 미세조정 방법을 적용한 모델 (C)보다도 분류 정확도가 낮게 나타나는 것을 확인할 수 있다. 이러한 결과는 일반 모델과 LoRA를 적용한 모델 모두, 제안 방법론을 통한 학습 방법이 효과적임을 나타낸다.

한편 <Table 6>에서 세 가지 모델 모두 LoRA 미세조정이 전체 미세조정에 비해 다소 낮은 정확도를 보이는 것으로 나타났다. 하지만 이미 알려진 바와 같이 LoRA 미세조정은 정확도가 다소 낮아지더라도 학습 효율을 높이기 위한 목적으로 사용되므로, 학습 효율성을 확인하기 위해 제안 모델 (A)가 두 가지 미세조정에서 가장 높은 성능에 달성하는 시점을 비교하는 실험을 수행하여 <Fig 11>에 그 결과를 요약하였다. 본 실험에서 전체 미세조정은 최적 성능에 도달하기 위해 10 에폭의 학습이 필요한 반면, LoRA 미세조정의 경우 5 에폭만에 빠르게 최적 성능에 도달함을 확인하였다.

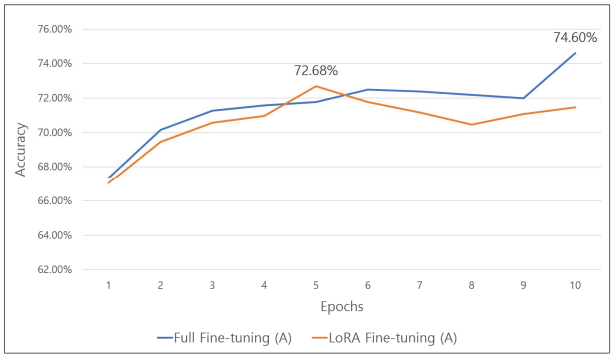


Fig. 11. Accuracy in Each Epoch of the Proposed Model

이상의 실험을 통해 LoRA 미세조정은 전체 미세조정에 비해 정확도는 다소 낮지만 효율적인 학습이 가능하며, 커리큘럼 러닝 기반 미세조정이 일반 미세조정에 비해 LoRA 미세조정과 전체 미세조정 모두에서 우수한 성능을 보임을 확인하였다.

## V. Conclusions

최근 거대 언어모델은 다양한 과제에 활용되어 우수한 성과를 거두고 있다. 하지만 실제 현장에서는 거대 언어모델을 학습시키기에 필요한 자원이 제한적이라는 한계로 인해, 자원 내에서 효과적이고 효율적으로 모델을 활용하는 방법에 주목하고 있다. 이러한 수요에 따라 본 연구는 모델을 효과적으로 학습시키기 위해, 데이터의 도메인 이질성에 기반을 두어 난이도를 정의하고 이를 학습 순서에 반영하는 커리큘럼 러닝 방법론을 제시하였다. 제안 방법론을 언어모델에 적용하여 전문 문서 분류 실험을 수행한 결과, 제안 방법론이 학습 효과 측면에서 LoRA 미세조정과 전체 미세조정 모두에서 전통적인 미세조정을 적용한 모델에 비해 우수한 성능을 달성함을 확인하였다.

본 연구는 커리큘럼 러닝의 새로운 난이도 측정 방법, 즉 도메인 고유성이 높은 데이터부터 학습하면 더 효과적이라는 새로운 통찰을 제시했다는 점에서 학술적 기여를 인정받을 수 있다. 즉, 신뢰할 수 있는 난이도 사전 정보를 용이하게 정의하는 하나의 방식을 제안하였으며, 향후 난이도 사전 정보 정의를 위한 다양한 후속 연구가 이어질 것으로 기대한다. 또한, 현장의 특성에 따라 LoRA 등 변형 모델이 사용되는 경우에도 제안 방법론이 우수한 성능을 보임을 확인하였으며, 이러한 성능 향상 측면에서 본 연구의 실무적 기여를 찾을 수 있을 것이다. 특히, 해당 방법론은 데이터 이질성 측정에 기반을 두고 있으므로, 다양한 유형의 데이터에 대한 난이도 정의를 통해 여러 모델 성능 향상에 기여할 수 있을 것으로 기대한다.

다만 본 연구는 데이터 도메인 간 이질성을 측정하기 위해, 특정 도메인에 대한 미세조정을 일종의 사전학습으로 가정한 뒤 실험을 진행하였다. 추후 연구에서는 충분한 양의 데이터를 확보하여 사전학습을 수행하고, 새로운 데이터를 통해 미세조정을 수행하여 제안 방법론의 견고성을 확인해야 한다. 또한 본 연구에서는 제안 방법론의 성능을 분류 과제에 한정하여 평가하였다. 향후 연구에서 다른 다양한 과제들에 본 방법론을 적용한 뒤 성능 차이를 평가하는 확장된 실험이 수행될 필요가 있다.

## ACKNOWLEDGEMENT

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2023S1A5A2A01077252)

## REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv:1810.04805, Oct. 2018. DOI: 10.48550/arXiv.1810.04805

[2] W. X. Zhao et al., “A Survey of Large Language Models,” arXiv:2303.18223, Mar. 2023. DOI: 10.48550/arXiv.2303.18223.

[3] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained Models for Natural Language Processing: A Survey,” Science China Technological Sciences, Vol. 63, No. 10, pp. 1872–1897, Sep. 2020. DOI: 10.1007/s11431-020-1647-3

[4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-Rank Adaptation of Large Language Models,” arXiv:2106.09685, Oct. 2021. DOI: 10.48550/arXiv.2106.09685

[5] G. M. van de Ven and A. S. Tolias, “Three Scenarios for Continual Learning,” arXiv:1904.07734, 2019. Apr. DOI: 10.48550/arXiv.1904.07734

[6] J. Kirkpatrick et al., “Overcoming Catastrophic Forgetting in Neural Networks,” Proceedings of the National Academy of Sciences, vol. 114, no. 13. Proceedings of the National Academy of Sciences, pp. 3521–3526, Mar. 14, 2017. DOI: 10.1073/pnas.1611835114

[7] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive Neural Networks,” arXiv:1606.04671, Jun. 2016. DOI: 10.48550/arXiv.1606.04671

[8] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” Proceedings of the 26th Annual International Conference on Machine Learning. ACM, Jun. 2009. DOI: 10.1145/1553374.1553380

[9] J. Hoffmann et al., “Training Compute-Optimal Large Language Models,” arXiv:2203.15556, Mar. 2022. DOI: 10.48550/arXiv.2203.15556

[10] M. Kumar, B. Packer, and D. Koller, “Self-Paced Learning for Latent Variable Models”, Advances in neural information processing systems, 2010.

[11] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. Hauptmann, “Self-Paced Curriculum Learning,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, no. 1. Association for the Advancement of Artificial Intelligence (AAAI), Feb. 21, 2015. DOI: 10.1609/aaai.v29i1.9608

[12] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly., “Parameter-Efficient Transfer Learning for NLP,” arXiv:1902.00751, Feb. 2019. DOI: 10.48550/arXiv.1902.00751

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” Advances in Neural Information Processing Systems, Vol. 30, 2017.

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” arXiv:1910.01108, Oct. 2019. DOI: 10.48550/arXiv.1910.01108

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” arXiv:1907.11692, Jul. 2019. DOI: 10.48550/arXiv.1907.11692

[16] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention.” arXiv.2006.03654, Jun. 2020. DOI: 10.48550/arXiv.2006.03654

[17] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.” arXiv.1909.11942, Sep. 2019. DOI: 10.48550/arXiv.1909.11942

[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," OpenAI blog, Vol. 1, No. 8, pp. 9, 2019.

[20] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, “Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 9. Institute of Electrical and Electronics Engineers (IEEE), pp. 2251–2265, Sep. 2019. DOI: 10.1109/tpami.2018.2857768

[21] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, “Generalizing from a Few Examples: A Survey on Few-Shot Learning,” arXiv:1904.05046, Apr. 2019. DOI: 10.48550/arXiv.1904.05046

[22] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a Few Examples,” ACM Computing Surveys, Vol. 53, No. 3, pp. 1–34, Jun. 2020. DOI: 10.1145/3386252

[23] T. B. Brown et al., “Language Models are Few-Shot Learners,” arXiv:2005.14165, May. 2020. DOI: 10.48550/arXiv.2005.14165

[24] H. Touvronet al., “LLaMA: Open and Efficient Foundation Language Models,” arXiv:2302.13971, Feb. 2023. DOI: 10.48550/arXiv.2302.13971

[25] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, Alpaca: A Strong, Replicable Instruction-Following Model, https://crfm.stanford.edu/2023/03/13/alpaca.html

[26] The Vicuna Team, Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality, https://lmsys.org/blog/2023-03-30-vicuna/

[27] H. Touvron et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models.” arXiv:2307.09288, Jul. 2023. DOI: 10.48550/arXiv.2307.09288

[28] J. B. Lee, KoLLaMA, https://huggingface.co/beomi/kollama-7b

[29] J. B. Lee, KoAlpaca, https://github.com/Beomi/KoAlpaca

[30] B. Kim et al., “What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers,” arXiv:2109.04650, Sep. 2021. DOI: 10.48550/arXiv.2109.04650

[31] KakaoBrain, KoGPT, https://github.com/kakaobrain/kogpt

[32] KT, Mi:dm, https://huggingface.co/KT-AI/midm-bitext-S-7B-inst-v1

[33] X. Wang, Y. Chen, and W. Zhu, “A Survey on Curriculum Learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence. Institute of Electrical and Electronics Engineers (IEEE), pp. 4555 - 4576, Mar. 2021. DOI: 10.1109/tpami.2021.3069908

[34] C. Li, H. Farkhoor, R. Liu, and J. Yosinski, “Measuring the Intrinsic Dimension of Objective Landscapes,” arXiv:1804.08838, Apr. 2018. DOI: 10.48550/arXiv.1804.08838

[35] A. Aghajanyan, L. Zettlemoyer, and S. Gupta, “Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning,” arXiv:2012.13255, Dec. 2020. DOI: 10.48550/arXiv.2012.13255

[36] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, “AdapterFusion: Non-Destructive Task Composition for Transfer Learning,” arXiv:2005.00247, May. 2020, DOI: 10.48550/arXiv.2005.00247

[37] S. Park et al., “KLUE: Korean Language Understanding Evaluation.” arXiv:2105.09680, May. 2021. DOI: 10.48550/arXiv.2105.09680

[38] A. Aghajanyan, L. Zettlemoyer, and S. Gupta, “Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning,” arXiv:2012.13255, Dec. 2020. DOI: 10.48550/arXiv.2012.13255

[39] SNU NLP Lab, KR-SBERT, https://github.com/snunlp/KR-SBERT

[40] E. Yu, S. Seo, and N. Kim.,"Building Specialized Language Model for National R&D through Knowledge Transfer Based on Further Pre-training," Knowledge Management Research, Vol. 22, No. 3, pp. 91-106, Sep. 2021.

## Authors

Daegeon Kim received the B.A. degree in Management Information Systems from Kookmin University in 2023 and currently enrolled in Graduate School of Business IT, Kookmin University.

He is interested in curriculum learning, deep learning, and natural language processing

Namgyu Kim received the B.S. in Computer Engineering from Seoul National University in 1998, M.S. and Ph.D. degrees in Management Engineering from KAIST, Korea, in 2000 and 2007, respectively.

Dr. Kim joined the faculty of the School of Management Information Systems at Kookmin University, Seoul, Korea, in 2007. He served as the Dean of the Graduate School of Business IT at Kookmin University and is currently a professor at the Business IT. He is interested in deep learning, text mining, and data modeling.