ISSN 2288-4866 (Print)
ISSN 2288-4882 (Online)
https://www.jiisonline.org
J Intell Inform Syst 2023 December: 29(4): 165~184 http://dx.doi.org/10.13088/jiis.2023.29.4.165

프라이버시 보호를 위한 오프사이트 튜닝 기반 언어모델 미세 조정 방법론*

정진명
국민대학교 비즈니스IT전문대학원
(cbml0225@kookmin.ac.kr)

김남규
국민대학교 비즈니스IT전문대학원
(ngkim@kookmin.ac.kr)

최근 구글의 BERT, OpenAI의 GPT 등, 언어모델(Language Model)을 사용한 비정형 텍스트 데이터에 대한 딥러닝(Deep Learning) 분석이 다양한 응용에서 괄목할 성과를 나타내고 있다. 대부분의 언어모델은 사전학습 데이터로부터 범용적인 언어정보를 학습하고, 이후 미세 조정(Fine-Tuning) 과정을 통해 다운스트림 태스크(Downstream Task)에 맞추어 갱신되는 방식으로 사용되고 있다. 하지만 최근 이러한 언어모델을 사용하는 과정에서 프라이버시가 침해될 수 있다는 우려가 제기되고 있다. 즉 데이터 소유자가 언어모델의 미세 조정을 수행하기 위해 다량의 데이터를 모델 소유자에게 제공하는 과정에서 데이터의 프라이버시가 침해될 수 있으며, 반대로 모델 소유자가 모델 전체를 데이터 소유자에게 공개하면 모델의 구조 및 가중치가 공개되어 모델의 프라이버시가 침해될 수 있다는 것이다. 이러한 상황에서 프라이버시를 보호하며 언어모델의 미세 조정을 수행하기 위해 최근 오프사이트 튜닝(Offsite Tuning)의 개념이 제안되었으나, 해당 연구는 제안 방법론을 텍스트 분류 모델에 적용하는 구체적인 방안을 제시하지 못했다는 한계를 갖는다. 이에 본 연구에서는 한글 문서에 대한 다중 분류 미세 조정 수행 시, 모델과 데이터의 프라이버시를 보호하기 위해 분류기를 추가한 오프사이트 튜닝을 적용하는 구체적인 방법을 제시한다. 제안 방법론의 성능을 평가하기 위해 AIHub에서 제공하는 ICT, 전기, 전자, 기계, 그리고 의학 총 5개의 대분야로 구성된 약 20만건의 한글 데이터에 대해 실험을 수행한 결과, 제안하는 플러그인 모델이 제로 샷 모델 및 오프사이트 모델에 비해 분류 정확도 측면에서 우수한 성능을 나타냄을 확인하였다.

주제어 : 딥러닝, 언어모델, 언어모델 프라이버시, 미세 조정, 오프사이트 튜닝, 텍스트 분류

논문접수일 : 2023년 10월 24일 논문수정일 : 2023년 11월 27일 게재확정일 : 2023년 11월 29일
원고유형 : 학술대회 우수논문 교신저자 : 김남규

1. 서론

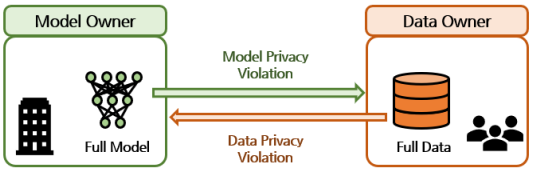
최근 심층 신경망 구조를 이용하여 데이터의 특성을 추출하는 딥러닝(Deep Learning)을 통해 방대한 양의 비정형 데이터의 분석에서 성과를 거두는 사례가 급증하고 있다. 특히 대규모 학습 데이터의 수집과 GPU의 사용이 용이해지고 딥러닝 알고리즘이 발전함에 따라, 딥러닝은 컴퓨터 비전, 음성 인식 등 다양한 분야에서 성과를 보이고 있다. 이렇게 다양한 유형의 데이터 중 텍스트 데이터를 다루는 자연어 처리 분야에서 딥러닝을 활용한 성과가 두드러지게 나타나고 있다. 대표적으로 감성 분석(Sentiment Analysis), 텍스트 분류(Text Classification), 자연어 추론

* 본 연구는 2023학년도 국민대학교 우수연구센터 사업비를 지원받아 수행된 연구임.

(Natural Language Inference)과 같은 자연어 이해 기술, 그리고 요약, 번역, 질의응답과 같은 자연어 생성 기술이 딥러닝에 힘입어 큰 발전을 이루었다.

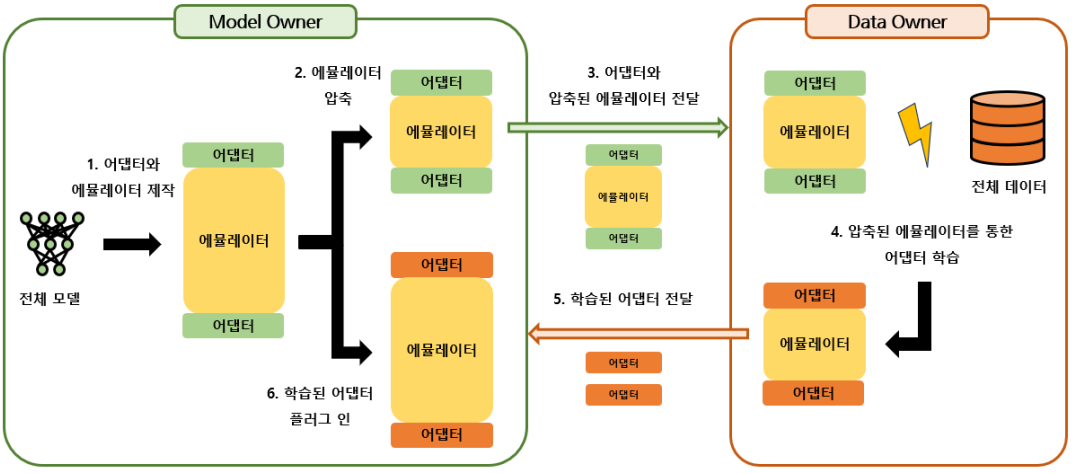
이러한 텍스트 딥러닝의 기술은 언어모델(Language Model)의 개념이 등장하면서 급성장하였다. 언어모델이란 인간의 언어를 이해하고 생성할 수 있도록 학습된 수리적 모형이며, 대표적으로 구글에서 발표한 BERT(Devlin et al., 2018), 그리고 OpenAI에서 발표한 GPT(Radford et al., 2018) 등이 있다. 대부분의 언어모델은 셀프 어텐션(Self-Attention) 메커니즘을 활용한 트랜스포머(Transformer) 모듈을 기반으로 설계되었으며, 임베딩(Embedding) 과정을 통해 입력 문장에 대한 밀집 벡터(Dense Vector)를 도출한다. 구체적으로 언어모델은 사전학습 데이터로부터 밀집 벡터를 도출하여 범용적인 언어정보를 학습하고, 학습된 모델 가중치는 미세 조정(Fine-Tuning) 과정을 통해 다운스트림 태스크(Downstream Task)에 맞추어 갱신된다.

이처럼 언어모델은 미세 조정 과정을 통해 다양한 분야의 텍스트 분석에 활발하게 사용되고 있으나, 최근 이러한 언어모델을 사용하는 과정에서 데이터 프라이버시가 침해될 수 있다는 우려가 제기되고 있다. 데이터 프라이버시란 모든 데이터는 권한을 가진 조직이나 개인에 의해서만 접근되어야 한다는 개념이며 데이터 소유자가 언어모델의 미세 조정을 수행하기 위해 다량의 기밀 데이터 전체를 모델 소유자에게 제공하는 과정에서 데이터의 프라이버시가 침해될 수 있다. 한편 프라이버시에 대한 우려는 모델 소유자 관점에서도 제기되고 있다. 즉, 모델 소유자가 해당 모델을 특정 도메인에 맞추어 갱신하기 위해 모델 전체를 데이터 소유자에게 공개하게 되면, 모델의 구조 및 가중치가 공개되어 모델의 프라이버시가 침해될 수 있다. 현실 세계에서는 모델을 개발한 모델 소유자와 이를 사용하고자 하는 데이터 소유자가 서로 다른 집단인 경우가 대부분이므로, 이러한 프라이버시의 문제는 언어모델의 활용에 큰 장애 요소로 작용하게 된다. 즉, 데이터 소유자는 데이터를 모델 소유자에게 제공할 수 없고 모델 소유자는 모델을 데이터 소유자에게 공개할 수 없기 때문에 언어모델의 성능 향상을 위한 미세 조정이 이루어지기 어려우며, 이러한 상황은 <그림 1>을 통해 묘사될 수 있다.



〈그림 1〉 언어모델 미세 조정 과정의 프라이버시 침해 유형

이러한 상황에서 발생할 수 있는 프라이버시 침해 문제를 막기 위해 최근 오프사이트 튜닝(Offsite Tuning)(Xiao et al., 2023) 이 제안되었다. 이는 모델 소유자와 데이터 소유자 양측의 프라이버시를 지키기 위한 방법으로, 대략의 구조는 <그림 2>와 같다. 먼저 모델 소유자가 자신의 모델로 어댑터(Adapter)와 에뮬레이터(Emulator)를 제작한 후, 어댑터는 그대로, 에뮬레이터는 압축하여 이를 데이터 소유자에게 전달한다. 데이터 소유자는 압축 에뮬레이터의 도움을 받아 자신의 데이터로 어댑터를 학습시킨 후, 학습된 어댑터를 모델 소유자에게 반환한다. 모델 소유자는 원본 모델의 어댑터를 데이터 소유자가 제공한 어댑터로 교체한다. 즉, 전체 모델 또는 전체 데이터의 직접적인 교환 없이 어댑터 교환을 통해 미세 조정이 이루어지며, 이를 통해 모델 프라이



〈그림 2〉 오프사이트 튜닝 개요

버시와 데이터 프라이버시를 지킬 수 있다.

하지만 해당 연구는 질의응답 데이터셋을 사용한 실험 과정 및 평가 결과만을 공개했다는 한계를 갖는다. 예를 들어 감성 분석과 같은 텍스트 분류 태스크의 경우, 언어모델의 출력을 분류 레이어(Classification Head)에 통과시켜 선형 변환을 수행한다. 하지만 해당 논문에서는 오프사이트 튜닝 시, 이러한 분류 레이어를 학습시키는 방안을 충분히 구체적으로 제시하지 않았다. 또한, 해당 연구에서는 영어 기반 언어모델 및 데이터셋을 사용한 실험 결과만을 소개하였으며, 한국어 기반의 언어모델 및 데이터셋의 적용 가능성을 다루지 않았다.

따라서 본 연구에서는 한글 문서에 대한 다중 분류 미세 조정 수행 시, 모델과 데이터의 프라이버시를 보호하기 위해 분류기를 추가한 언어모델에서 오프사이트 튜닝을 적용하는 구체적인 방법을 제안한다. 전체 과정은 <그림 2>의 구조에 따라 이루어지며, 분류 레이어는 모델 소유자와 데이터 소유자 각각 가장 우수한 성능을 나타낼 수 있는 방식으로 학습하는 상황을 가정한다. 실험을 통해 데이터 소유자가 모델 소유자로부터 전달받는 압축 모델인 오프사이트 모델과 학습된 어댑터를 장착한 원본 모델인 플러그인(Plug-in) 모델의 성능을 비교하여, 제안 방법론의 우수성을 보이고자 한다.

본 논문의 이후 구성은 다음과 같다. 2장에서는 현재까지 보고된 프라이버시 위협과 보호 방법을 소개한다. 3장에서는 오프사이트 튜닝을 적용한 다중 분류 미세 조정 방법을 제안하며, 4장에서는 제안 방법론을 기반으로 한 다중 분류 실험의 과정과 결과를 소개한다. 마지막으로, 5장에서는 본 연구의 기여와 한계를 정리한다.

2. 관련 연구

2.1. 프라이버시 위협 연구 동향

머신 러닝(Machine Learning)과 딥러닝의 목적은 데이터로부터 일반화된 패턴을 찾아 학습하는

것으로, 이 과정은 크게 데이터 수집, 모델 학습, 그리고 모델 평가 단계로 구성된다. 이러한 전체 단계를 동일한 조직이 수행한다면 프라이버시 침해 문제가 발생할 가능성은 크지 않겠지만, 각 단계를 서로 다른 조직이 수행하거나 그 과정이 외부에 드러날 경우 프라이버시 침해의 가능성이 높아지게 된다. 이러한 프라이버시 침해는 주로 적대적 공격(Adversarial Attack)을 통해 발생하며, 대표적인 유형으로 중독 공격(Poisoning Attack)(Baracaldo et al., 2017; Tian et al., 2022), 추론 공격(Inference Attack)(Fredrikson et al., 2015; Shokri et al., 2017), 그리고 회피 공격(Evasion Attack)(Biggio et al., 2013; Goodfellow et al., 2014) 등이 알려져 있다.

중독 공격은 수집된 학습 데이터에 악의적으로 조작된 데이터를 포함시켜 모델 학습을 방해하는 방법이다. 공격자는 조작된 데이터로 학습된 모델이 왜곡된 결과를 도출하거나, 성능이 하락하거나, 또는 특정 결과를 도출하도록 유도하는 것을 목표로 한다. 따라서 중독 공격을 수행하기 위해서는 공격자가 모델이 학습할 학습 데이터셋에 접근할 수 있어야 한다. 그런데 최근 딥러닝은 공개된 데이터셋을 다양하게 수집하여 학습을 수행하므로, 공격자는 이러한 상황을 악용하여 학습 데이터셋에 접근한다. 중독 공격의 대표적인 피해 사례로는 2016년에 마이크로소프트에서 공개한 인공지능 챗봇 테이(Chat Bot Tay)의 사례를 들 수 있다. 해당 챗봇은 트위터 이용자들로부터 공격적인 발언을 하도록 학습되어, 서비스 시작 16시간 만에 운영이 종료되었다.

추론 공격의 대표적인 유형으로는 특정 개인에 대한 정보가 모델 학습에 활용되었는지를 학습된 모델로부터 확인할 수 있는 멤버십 추론 공격(Membership Inference Attacks), 그리고 모델이 학습한 학습 데이터를 역으로 추출하는 모델 전도 공격(Model Inversion Attacks) 등이 있다. 멤버십 추론 공격은 알아내고자 하는 정보를 학습한 모델과 그렇지 않은 모델이 예측하는 값이 차이를 보인다는 사실을 바탕으로 한다. 이러한 공격은 특정 환자의 데이터가 질병 판별 모델의 학습에 사용되었는지 여부를 판별하여, 해당 환자가 어떤 질병을 앓았는지를 식별하는 등의 시도로 악용될 수 있다. 모델 전도 공격은 모델이 입력을 받아 출력하는 분류 결과와 신뢰도(Confidence)를 분석하여 역으로 학습 데이터를 복원하는 방식이며, 해당 공격을 통해 얼굴 인식 알고리즘을 위해 학습된 분류기로부터 학습에 사용된 얼굴 이미지를 재구성해 낼 수 있는 것으로 알려졌다.

다음으로 회피 공격은 모델의 일반화 성능을 해치지 않으면서, 추론 단계에서 잘못된 예측 결과를 출력하도록 입력 데이터에 변조를 가하는 방식이다. 이러한 공격은 특히 이미지 분류 모델에서 심각한 교란을 일으킬 수 있어 치명적인 것으로 알려져 있다. 이러한 노이즈는 사람의 눈으로는 구분할 수 없을 만큼 최소한으로 추가되며, 공격자는 노이즈를 추가함으로써 모델 출력 결과의 거짓 음성(False Negative) 비율을 높이거나 적대적 이미지가 특정 클래스로 분류되도록 시도한다. 실제로 구글의 연구원들은 물리적 스티커만을 사용하여 VGG16 모델에 대해 제시된 이미지를 잘못 분류하도록 만드는 데모를 공개하였다(Brown, 2018).

2.2. 프라이버시 보호 기술 연구 동향

프라이버시 침해 위협이 부상하면서 이를 예방할 수 있는 보호 기술들도 발전하고 있다. 프라이버시 보호 대상으로는 크게 데이터, 모델, 그리고 모델 출력 결과의 세 가지를 들 수 있다.

그중 데이터 프라이버시는 모델 학습에 사용되는 학습 데이터와 모델에 입력으로 주어지는 데이터에 포함된 개인정보의 보호를 의미한다. 이러한 데이터는 대부분 권한을 가진 조직이나 개인이 소유하며 독점적으로 관리한다. 또한 웹 스크래핑을 통해 수집된 공개 데이터에는 의도치 않게 개인정보가 포함되어 있을 수 있으므로, 이러한 정보들을 제거하거나 비식별화하는 작업이 필요하다. 한편 모델 프라이버시는 모델 구현체나 매개변수와 같은 직접적인 모델에 관한 정보의 비공개나 내부 연산의 암호화를 통한 모방 방지를 의미한다. 이를 위해 지식 증류(Knowledge Distillation)나 양자화(Quantization) 같은 모델 압축 기법이 적용될 수 있으며(Papernot et al., 2016) 최근 발표되는 거대 언어모델은 아예 처음부터 모델의 핵심 기술을 공개하지 않는 경우도 있다. 마지막으로 모델 출력 결과에 대해서도 프라이버시를 고려해야 한다. 학습된 모델은 원 데이터의 특징 정보들을 가중치 내에 저장하고 있으므로, 학습된 모델을 통해 원 데이터에 포함된 정보의 패턴을 추론할 수 있는 것으로 알려졌다(Wang & Liu, 2011). 이러한 악의적인 추론 공격을 통해 모델의 결괏값으로부터 정보를 추출하거나 모델과 유사한 결과를 출력하는 공격 모델을 만들 수 있으므로, 모델 출력 또한 프라이버시 보호의 대상이 된다.

데이터 프라이버시를 보호하는 방법으로는 데이터 비식별화가 오랫동안 사용되고 있으며, 각국의 가이드라인도 제정된 바 있다. 우리나라의 경우 개인정보 비식별 조치 가이드라인을 통해 비식별화 방법으로 k-익명성, l-다양성, t-근접성 등의 기법들을 소개하고 있다. 하지만 이러한 보호모델은 공격자가 특정 정보에 대해 모델이 가정하고 있는 범위를 넘어선 배경지식을 가지고 있는 경우, 해당 정보에 대한 추론을 막기 어렵다는 한계가 있다. 이와 다르게 차분 프라이버시(Differential Privacy)(Arachchige et al., 2019)는 특정 정보를 포함하는 데이터셋과 그렇지 않은 데이터셋의 통계적 차이가 최대한 작아지도록 질의에 대한 응답 값을 변조한다. 따라서 공격자가 특정 정보에 대해 배경지식을 가지고 있다 할지라도 질의에 대해 동일한 통계 결과가 도출되므로 데이터 프라이버시를 효과적으로 보호할 수 있는 것으로 알려졌다. 이러한 장점으로 인해 차분 프라이버시는 머신 러닝뿐만 아니라 딥러닝 분야에서도 사용되고 있다.

모델 프라이버시를 보호하는 가장 대표적인 방법은 암호화 기반의 모델 학습이다. 하지만 해당 방법은 모델의 학습 데이터를 추출하는 모델 전도 공격을 예방할 수 있지만, 학습 데이터를 암호화하고 다시 복호화하는 과정에서 데이터가 유출될 수 있다는 한계를 갖는다. 이러한 문제를 예방하기 위해 최근에는 동형 암호(Homomorphic Encryption)(Martins et al., 2017) 방식이 주로 사용된다. 동형 암호는 평문을 연산한 값과 암호문을 연산한 값이 동일하게 나타나는 암호화 기법으로, 암호문을 복호화하지 않아도 데이터에 대한 연산이 가능하다는 장점이 있다. 동형 암호 알고리즘을 딥러닝에 적용한 대표적인 연구로는 마이크로소프트의 CryptoNet(Gilad-Bachrach et al., 2016)이 있으며, 알려진 라이브러리로는 서울대학교의 HEAAN(Cheon et al., 2017)이 있다.

모델 출력 결과의 프라이버시를 보호하는 방법으로는 모델과 관련한 배경 정보를 제공하지 않는 방법이 대표적이다. 먼저 모델에 대한 질의 횟수를 제한하는 방법이 있다(Xu et al., 2021). 이는 모델 엑세스를 제한함으로써 공격자가 모델 출력값으로 얻을 수 있는 정보량을 제한한다. 더 나아가 모델을 아예 비공개한다면 이러한 통계

정보의 유출을 최대한 예방할 수 있다. 비슷한 원리로 모델 압축 기법을 통해 모델의 일부를 가리는 방법도 존재한다. 이를 통해 모델에 저장된 자세한 정보를 가릴 수 있고 공격자가 모델 출력 결과를 역이용해 모델에 포함된 특정 정보를 추출하는 것을 방지하는 효과를 거둘 수 있다.

2.3. 딥러닝에서의 프라이버시 보호

앞서 소개된 방법들은 각각의 장단점을 지니고 있다. 즉, 프라이버시 보호는 하나의 방법만을 적용해서는 완전한 보호를 보장할 수 없으며, 보편화된 해결책도 아직 논의되지 않은 상황이다. 따라서 여러 방법을 적절히 조합하거나 응용하는 것이 필요하다. 이 밖에도 다양한 딥러닝 학습 방법을 적용하는 과정에서 프라이버시 보호 효과를 발견하여 발전시킨 경우도 존재한다. 주로 모델을 경량화하는 과정에서 이러한 효과를 볼 수 있는데, 그 이유는 경량화 기법이 학습된 정보를 포함하고 있는 노드들을 삭제하거나 변조하여 모델 전도와 같은 공격을 어렵게 만들기 때문이다(Huang et al., 2020; Papernot et al., 2018). 예를 들어 Private Aggregation of Teacher Ensembles (PATE) 기법의 경우 기본적으로 지식 증류를 활용하지만, 증류 과정에서 노이즈를 추가함으로써 기존 지식과 노이즈가 추가된 지식의 차이를 통해 프라이버시를 보호할 수 있음을 보였다. 또 다른 경량화 기법인 가지치기(Pruning) 역시 불필요한 노드의 삭제를 통해 프라이버시를 보호하는 효과를 보이는 것으로 나타났다. 본 논문에서 다루고자 하는 오프사이트 튜닝 역시 모델 압축을 통해 모델이 학습한 정보를 일부 삭제함으로써 프라이버시를 보호하는 방법이다.

프라이버시 보호 효과를 갖는 또 다른 딥러닝 학습 방법으로는 2017년 구글에서 발표한 연합 학습(Federated Learning)(McMahan et al., 2017)이 있다. 이는 데이터를 중앙 서버에 모아 처리할 때 학습 비용이 크다는 문제를 해결하기 위해 처음 고안되었다. 구체적으로 연합 학습은 각 사용자가 서버로부터 동일한 중앙 모델의 가중치를 전달받아 로컬 기기에 존재하는 데이터를 개별적으로 학습하며, 학습된 가중치를 서버로 모아 다시 중앙 모델을 학습하는 방식으로 동작한다. 이러한 방식을 통해 다양한 학습 데이터로 모델을 학습시키면서도 중앙 모델을 학습시키는 비용을 절감할 수 있을 뿐 아니라, 사용자 데이터에 직접 접근할 필요가 없으므로 데이터 프라이버시를 보장할 수 있다는 장점이 있다. 이러한 장점으로 인해 연합 학습은 민감한 데이터를 다루는 헬스케어, 의료 등의 분야에서 널리 사용되고 있다.

3. 제안 방법론

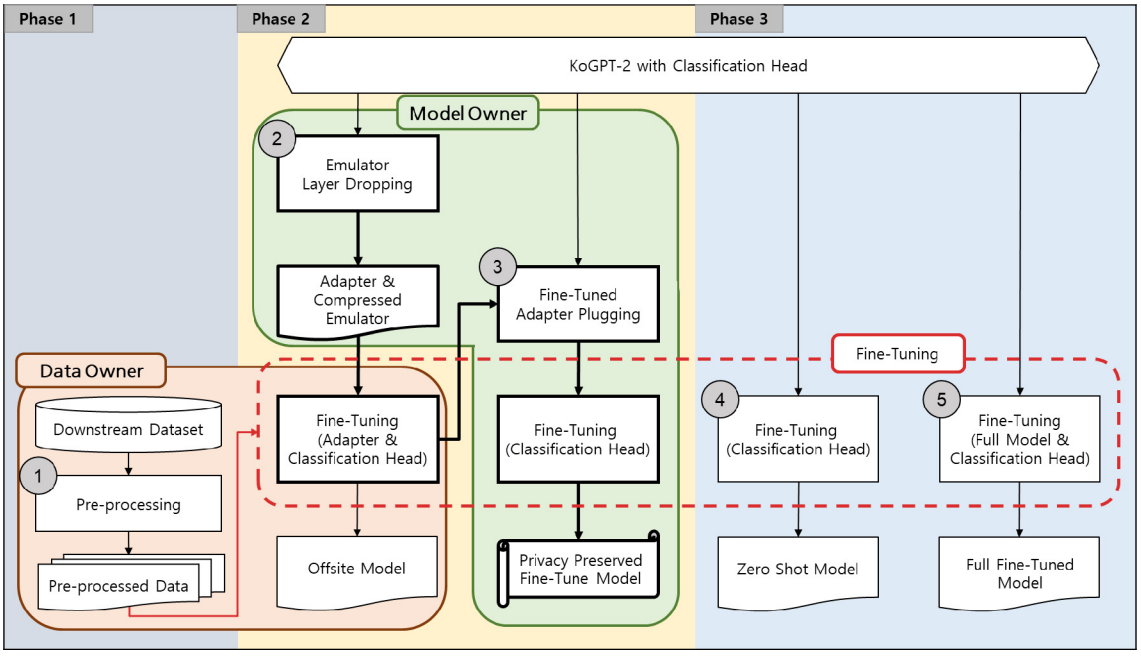
3.1. 제안 방법론 개요

본 장에서는 오프사이트 튜닝을 적용한 프라이버시 보호 다중 분류 미세 조정 방법론을 소개한다. 제안 방법론의 전체적인 과정은 <그림 3>을 따른다. 제안 방법론은 세 단계에 걸쳐 수행되며, 각각 모델 입력에 맞게 텍스트 데이터를 전처리하는 Phase 1, 오프사이트 튜닝을 적용한 다중 분류 미세 조정을 수행하는 Phase 2, 그리고 제안 방법론의 상대적 성능 평가를 위한 비교 모델을 제작하는 Phase 3로 구성된다. 또한 제안 방법론은 수행 주체에 따라 데이터 소유자 측, 혹은 모델 소유자 측에서 진행되는 단계로 구분할 수 있다. 이때 데이터 소유자의 역할은 미세

조정에 사용할 학습 데이터를 전처리하고 모델 소유자로부터 오프사이트 모델을 전달받아 어댑터를 학습시키는 것이다. 그리고 모델 소유자의 역할은 원본 모델을 어댑터와 에뮬레이터로 구분하고 에뮬레이터를 압축한 모델을 데이터 소유자에게 전달한 후, 학습된 어댑터를 돌려받아 플러그인 모델을 제작하는 것이다.

제안 방법론 중 Phase 1에서는 텍스트 데이터셋의 전처리를 거친다(①). Phase 2에서는 오프사이트 튜닝을 적용한 다중 분류 미세 조정을 수행한다. 이를 위해 모델 소유자는 소유하고 있는 원본 모델을 어댑터와 에뮬레이터로 구분한 후, 에뮬레이터에 대한 레이어 삭제(Layer Drop)를 통해 원본 모델을 압축시켜 압축 에뮬레이터를 생성한다(②). 다음으로 모델 소유자는 생성한 어댑터와 압축 에뮬레이터를 데이터 소유자에게 전달하고, 데이터 소유자는 압축 에뮬레이터와 학습 데이터를 사용하여 어댑터를 학습시킴으로써 오프사이트 모델을 제작한다. 학습이 완료되면 데이터 소유자는 학습된 어댑터를 모델 소유자에게 전달하고, 모델 소유자는 학습된 어댑터를 원본 모델의 어댑터와 교체하여 플러그인 모델을 제작한다(③). 이러한 과정을 통해 모델과 데이터의 프라이버시를 유지하면서 미세 조정 과정을 수행할 수 있다.

Phase 3에서는 제안 모델의 상대적 성능을 평가하기 위한 두 가지 비교 모델을 제작한다. 첫 번째로는 모델의 미세 조정은 전혀 진행하지 않고 분류기만 학습한 제로 샷 모델을 제작한다(④). 두 번째로는 프라이버시는 전혀 고려하지 않았지만 가장 높은 정확도를 나타낼 수 있는 모델, 즉 전체 모델과 분류기가 전체 데이터에 대해 미세 조정을 수행하는 모델을 제작한다(⑤).



〈그림 3〉 제안 방법론 전체 개요

이때 기대하는 결과는 제안하는 플러그인 모델의 성능이 전체 미세 조정 모델의 성능에 비해 크게 떨어지지 않으면서, 제로 샷 모델이나 데이터 소유자가 제작한 오프사이트 모델의 성능보다 우수하게 나타나는 것이다. 각 단계에 대한 구체적인 과정은 다음 절에서 설명하며, 제안 방법론의 성능 평가의 과정 및 결과는 4장에서 소개한다.

3.2. 텍스트 데이터 전처리

본 절에서는 모델에서 사용하는 텍스트 데이터의 전처리 과정을 간략히 설명한다(①). 오프사이트 튜닝 시 데이터 소유자가 가지고 있는 데이터는 기밀이다. 마찬가지로, 본 연구에서도 오프사이트 튜닝이 요구되는 상황을 가정하기 위해 프라이버시 보호가 필요하다고 판단되는 도메인의 데이터를 수집한다. 이때 텍스트 분류 태스크를 수행함에 있어 텍스트의 품질은 모델 성능에 직접적인 영향을 미치므로, 본 연구에서는 품질이 보장된 데이터를 일차적으로 수집한다. 이후 전처리 수행 시, 문서에 포함된 규격이나 단위와 같은 정보는 그 내용을 유지하여 전처리에 수반되는 정보 손실을 최소화한다. 또한 Null 값 제거와 중복 제거를 수행하여, 학습 과정에서 문제를 야기하거나 학습 데이터가 추론 데이터에 동일하게 등장하는 문제를 예방한다. 그리고 각종 링크는 의미 정보가 모호하다고 판단하여 정규화를 통해 제거한다.

3.3. 오프사이트 모델 및 플러그인 모델 제작

본 절에서는 레이어 삭제를 적용하여 모델을 압축하고(②) 어댑터만을 학습하여 오프사이트 모델을 제작하는 과정, 그리고 오프사이트 모델로부터 학습된 어댑터를 전달받아 플러그인 모델을 제작하는 과정(③)을 소개한다.

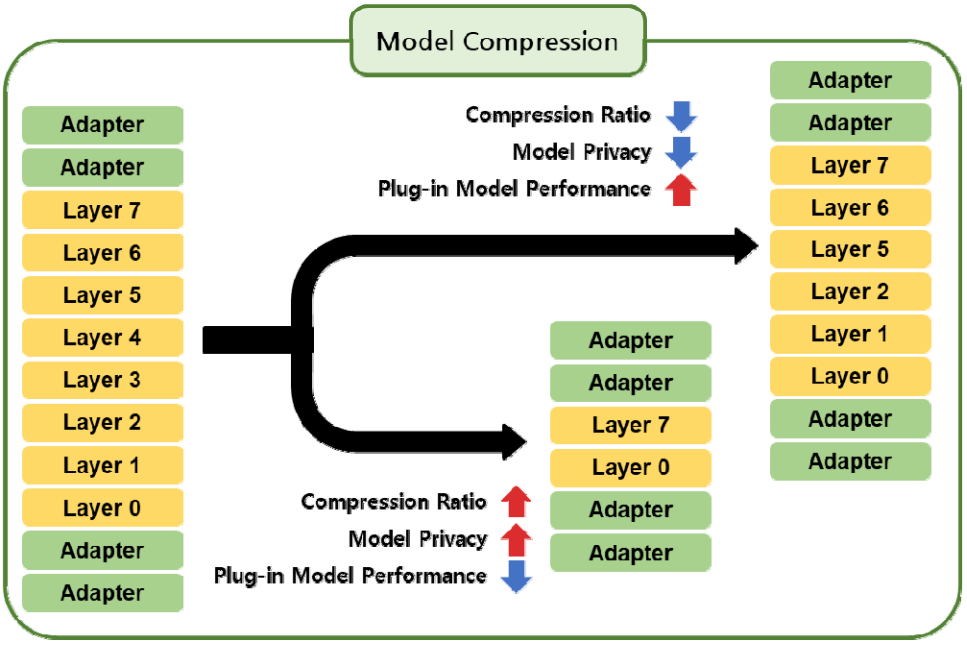
우선 오프사이트 모델을 만들기 위해 먼저 모델 소유자는 가지고 있는 원본 모델을 어댑터와 에뮬레이터로 나눈다. 이때 어댑터는 가중치가 업데이트되며 실제 학습이 이루어지는 부분으로, 모델 소유자와 데이터 소유자 사이의 학습 정보 교환을 매개하는 역할을 수행한다. 한편 에뮬레이터는 가중치가 고정되어 있으면서, 원본 모델의 연산을 모방하여 어댑터의 학습을 돕는 역할을 한다. 여기서 어댑터는 전체 모델에서 작은 부분을 차지하며 나머지 부분을 에뮬레이터로 설정한다.

어댑터 레이어를 선택하는 것은 곧 미세 조정이 이루어질 레이어를 선택하는 것과 같으며, 미세 조정되는 레이어의 수에 따라 미세 조정되는 모델의 성능도 달라진다(Lee et al., 2022). 어댑터 레이어의 수, 즉 학습되는 레이어 수가 증가하면 성능 또한 증가할 것으로 예상할 수 있지만, 동시에 모델 소유자의 원본 모델을 그만큼 많이 공개하게 되므로 모델 프라이버시는 약해진다. 본 연구에서는 오프사이트 튜닝을 제안한 논문에서 제시한 설정에 따라 모델의 위, 아래 각 2개씩, 총 4개 레이어를 어댑터 레이어로 선택하여 해당 레이어의 가중치만 갱신한다. 이때 어댑터의 학습이 얼마나 잘 이루어지는지 여부는 오프사이트 모델뿐 아니라 향후 제작될 플러그인 모델의 성능에도 영향을 미친다. 그러므로 어댑터의 품질을 평가할 수 있는 방안이 반드시 필요하며, 본 연구에서는 이를 위해 어댑터의 학습을 담당하는 오프사이트 모델의 다중 분류 미세 조정 성능을 확인한다. 구체적으로 본 논문에서는 오프사이트 모델 학습 시 학습률 스케줄러(Learning Rate Scheduler)를 적용하여 오프사이트 모델의 학습 최적화를 수행한다.

어댑터 레이어를 제외한 모델의 중간 레이어인 에뮬레이터는 가중치를 고정한다. 즉, 에뮬레이터

자체는 학습이 이루어지지 않고, 어댑터의 학습이 잘 이루어지도록 순전파(Forward Propagation) 연산만을 수행한다. 어댑터와 함께 데이터 소유자에게 전달되는 에뮬레이터는 <그림 4>에 나타난 바와 같이 모델 프라이버시를 위한 압축 과정을 거친다. 이때 압축률은 모델의 성능과 프라이버시에 영향을 미치는 중요한 변수로 작용한다. 압축률이 낮으면 어댑터의 학습이 충분히 이루어질 수 있지만, 그만큼 모델이 많이 공개되어 모델 프라이버시 보호가 덜 이루어질 수 있다. 반대로 압축률이 높으면 모델을 많이 숨기게 되어 프라이버시는 효과적으로 보호할 수 있지만, 어댑터 학습이 충분히 이루어지지 않아 오프사이트 모델 및 플러그인 모델 제작 시 성능이 낮아지게 된다. 따라서 성능과 프라이버시의 두 가지 요소를 고려하여 적절한 수준에서 모델 압축이 이루어져야 한다. 모델 압축 방법으로는 가지치기, 양자화, 지식 증류와 같은 다양한 딥러닝 모델 경량화 기법들을 사용할 수 있으나, 본 논문에서는 오프사이트 튜닝을 제안한 논문에서 제시된 바와 같이 레이어 삭제를 통한 모델 압축을 진행한다(Sajjad et al., 2023). 또한 모델 프라이버시와 어댑터의 품질의 균형을 고려하여 에뮬레이터의 압축률을 50%로 설정한다.

모델 소유자는 어댑터와 압축 에뮬레이터를 데이터 소유자에게 전달하고, 데이터 소유자는 ①에서 전처리한 데이터셋으로 미세 조정을 진행한다. 이때 다중 분류를 수행하기 위해 분류기가 추가되며, 해당 분류기의 가중치는 무작위로 초기화한다. 추가된 분류기는 언어모델이 입력 문장에 대한 임베딩을 통해 도출하는 벡터 값을 받아서 분류하고자 하는 클래스 수의 차원으로



〈그림 4〉 레이어 삭제를 통한 모델 압축 예시

선형 변환을 수행한다. 이후 미세 조정을 통해 어댑터의 가중치를 갱신함과 동시에 분류기의 가중치도 학습하여 오프사이트 모델을 제작한다. 학습이 종료되면 추후 모델 소유자에게 전달할 학습된 어댑터를 저장한다.

앞서 언급한 바와 같이 에뮬레이터의 압축은 오프사이트 모델 및 플러그인 모델의 성능에 영향을 미치게 된다. 이때 각 모델의 개별 성능뿐 아니라, 두 모델의 상대적 성능 차이도 매우 중요한 요소로 함께 고려되어야 한다. 오프사이트 모델을 제작하는 이유는 데이터 소유자가 모델 소유자의 원본 모델에 접근하지 않으면서 원본 모델의 미세 조정을 가능하게 하기 위함이다. 그런데 만약 오프사이트 모델만으로도 충분히 만족스러운 성능을 달성할 수 있게 되면, 데이터 소유자는 굳이 어댑터를 반환하여 플러그인 모델을 만들 필요가 없다. 즉, 더 이상 데이터 소유자가 모델 소유자에게 의존하여 이후 과정을 진행할 필요가 없게 되며, 이는 모델 소유자에게 일방적으로 불리한 상황이 된다. 반대로, 오프사이트 모델의 성능이 너무 낮으면 해당 모델을 통해 제작한 어댑터의 품질 또한 보장할 수 없으며, 결국 이러한 어댑터를 사용한 최종 플러그인 모델의 성능 또한 낮게 나타나게 된다. 따라서 제안 모델이 기술적 측면뿐 아니라 현실적 측면에서도 활용 가능성을 인정받기 위해서는, 어댑터와 에뮬레이터의 균형 있는 조합을 찾아 오프사이트 모델을 제작하여 오프사이트 모델과 플러그인 모델 간의 적절한 성능 균형이 이루어지는 지점을 찾아야 한다.

오프사이트 모델의 학습이 마무리되면, 데이터 소유자는 학습된 오프사이트 모델 중 어댑터 부분만을 모델 소유자에게 전달하고, 모델 소유자는 전달받은 어댑터를 사용하여 플러그인 모델을 제작한다. 구체적으로, 플러그인 모델은 원본 모델의 에뮬레이터를 그대로 유지하고, 원본 모델의 어댑터만 학습된 어댑터로 교체함으로써 완성된다. 이때 학습된 어댑터는 오프사이트 모델을 만드는 과정에서 가중치가 갱신되어 새로운 데이터에 대한 정보를 담고 있으므로, 어댑터가 양질의 정보를 얼마나 잘 학습하였는지에 따라 플러그인 모델의 성능이 결정된다.

플러그인 모델은 오프사이트 모델에 비해 상대적으로 많은 층과 매개변수로 구성되어 있으므로, 일반적으로 플러그인 모델의 성능이 오프사이트 모델에 비해 우수하게 나타날 것으로 기대할 수 있다. 하지만 이러한 성능 차이가 항상 일관되게 나타나는 것은 아니며, 이는 가벼운 오프사이트 모델과 무거운 플러그인 모델이 각각 최적의 성능을 나타낼 수 있는 태스크의 난이도가 서로 다른 것에서 그 이유를 찾을 수 있다. 만약 얕은 신경망을 갖는 모델로도 충분히 해결할 수 있는 낮은 난이도의 태스크라면, 지나치게 복잡한 구조의 플러그인 모델의 성능이 오프사이트 모델의 성능보다 오히려 낮게 나타나는 경우도 발생할 수 있다. 따라서 모델 소유자의 프라이버시가 보호되는 상황, 즉 오프사이트 모델의 성능이 플러그인 모델을 능가하지 않는 상황을 유지하기 위해서는 일정 수준 이상의 난이도를 갖는 다운스트림 태스크가 필요하며, 본 논문에서는 이를 위해 전문적인 도메인의 데이터셋을 다양한 클래스로 분류하는 다중 분류 태스크를 수행하고자 한다.

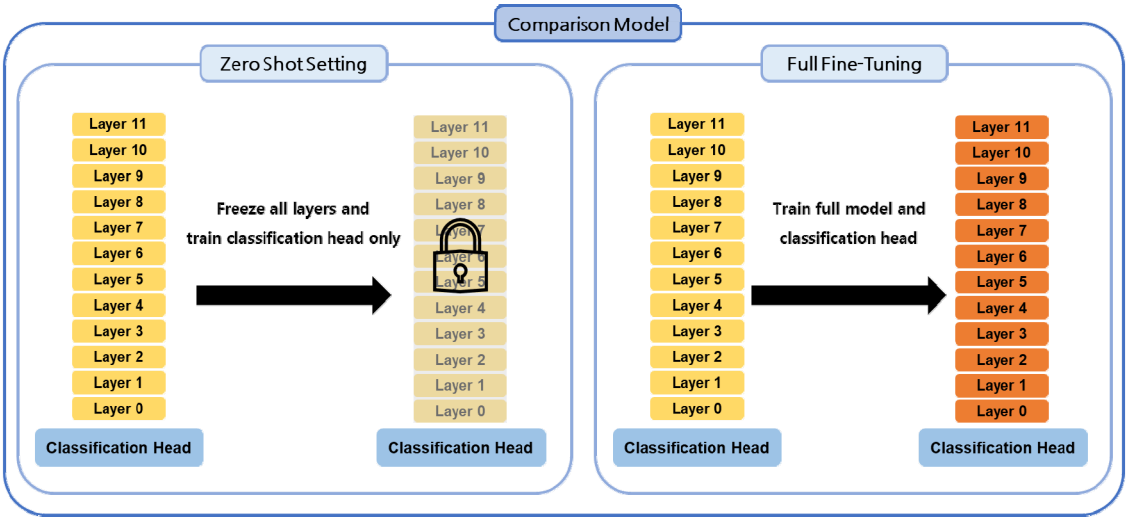
3.4. 비교 모델 제작

본 절에서는 제안 방법론의 상대적 성능평가를 위한 비교 모델 제작 과정을 소개한다. 이를

위해 모델의 모든 가중치는 고정하고 분류기만을 학습하는 제로 샷 모델(④)과 전체 가중치와 분류기가 미세 조정되는 모델(⑤)을 제작한다. 다만 이들 모델들은 현실에서의 사용을 가정한 것이 아닌 제안 방법론과의 상대적 성능 비교를 위해 도입한 모델이므로, 데이터나 모델 프라이버시 침해의 고려 대상이 아니다. 제로 샷 모델의 경우 다운스트림 태스크에 대한 정보를 전혀 학습하지 않은 상태에서 분류기만을 학습하므로, 모든 모델 중 가장 낮은 성능을 보일 것으로 예상한다. 반면 전체 미세 조정 모델의 경우 프라이버시 침해를 고려하지 않고 전체 데이터에 대해 전체 모델의 미세 조정이 이루어지므로, 모든 모델 중 가장 높은 성능을 보일 것으로 예상한다. 두 모델의 가중치 갱신 범위는 <그림 5>를 통해 확인할 수 있다.

<그림 5>의 좌측은 가장 낮은 성능을 나타낼 것으로 예상하는 기본(Baseline) 모델의 가중치 갱신 범위를 나타낸다. 이때 모델은 사전학습한 정보만을 가지고 태스크를 수행하므로 제로 샷 모델이라고 한다. 즉 해당 모델은 추론만을 수행하므로 모델 자체의 가중치 역시 갱신되지 않도록 모두 고정한다. 하지만 제안 모델과의 분류 태스크 성능 비교를 위해 분류기를 학습해야 하며, 이때 분류기 학습을 위해 사용되는 데이터는 제안 모델의 분류기 학습에 사용된 데이터를 동일하게 사용한다.

<그림 5>의 우측은 가장 이상적인 성능을 나타낼 것으로 예상하는 전체 미세 조정 모델의 가중치 갱신 범위를 나타낸다. 해당 모델은 모델 소유자와 데이터 소유자가 동일한 상황, 즉 전체 데이터에 대해 전체 모델의 미세 조정을 수행할 수 있는 상황을 가정한다. 마찬가지로 본 모델 역시 제안 모델과의 분류 태스크 성능 비교를 위해 분류기를 학습해야 하며, 이때 분류기 학습을 위해 사용되는 데이터는 제안 모델의 분류기 학습에 사용된 데이터를 동일하게 사용한다. 물론 본 모델은 데이터와 모델의 프라이버시 침해가



〈그림 5〉 두 가지 비교 모델의 가중치 갱신 범위

문제되지 않는 상황을 가정하므로, 본 연구에서 제안하는 플러그인 모델 및 오프사이트 모델과의 성능 비교 대상이 아니다. 다만 프라이버시를 고려하는 제안 모델이 성능만을 고려한 모델에 비해 어떤 수준의 상대적 성능을 나타내는지 확인하기 위해 본 모델을 비교 모델로 설정하였다.

4. 실험

4.1. 실험 개요

본 장에서는 3장에서 소개한 오프사이트 튜닝을 적용한 언어모델 미세 조정 방법론을 실제 데이터에 적용한 실험 과정과 그 결과를 소개한다. 실험에는 프라이버시 보호가 필요할 것이라고 예상되는 기술과학 도메인의 데이터셋을 사용하였으며, 구체적으로 AIHub에서 제공하는 한국어-영어 번역 말뭉치 데이터셋을 활용하였다. 해당 데이터셋은 ICT, 전기, 전자, 기계, 그리고 의학 총 5개의 대분야로 구성되어 있으며 세부적으로 총 18개의 소분야를 갖는다. 해당 데이터셋의 각 문장은 하나의 소분야에 속해 있으며 소분야별 데이터 수는 상이하다. 따라서 소분야별 균형을 맞추기 위해 각 소분야별로 학습 데이터 10,000건, 평가 및 추론 데이터 각 1,000건씩의 데이터를 추출하여, 총 180,000건의 학습 데이터셋, 18,000건의 평가 데이터셋, 그리고 18,000건의 추론 데이터셋을 구축하였다. 즉, 본 실험에서 다운스트림 태스크란 텍스트 데이터를 입력으로 받아서 이를 18개의 소분야 중 하나로 분류하는 작업을 수행하는 것을 의미한다.

실험을 위해 SKT에서 공개한 한국어 사전학습 언어모델인 KoGPT-2 Ver 2.0을 사용하였다. KoGPT-2는 GPT-2의 공개된 버전 중 가장 작은 GPT-2 small을 기반으로 한국어 데이터셋에 대해 미세 조정한 모델이다. 해당 모델을 선택한 이유는 오프사이트 튜닝을 제안한 논문에서 거대 언어모델에 대한 오프사이트 튜닝의 활용 가능성을 언급하고 있으며, 실험 또한 트랜스포머 디코더 기반의 언어모델을 활용한 결과를 공개하였기 때문이다. 이에 본 실험에서도 디코더 기반의 언어모델을 사용하고자 하며, 동시에 한국어 데이터셋의 활용 가능성을 보이기 위해 한국어 데이터를 학습한 KoGPT-2 언어모델을 HuggingFace에서 다운로드하여 사용하였다. 이후 분류 태스크를 수행하기 위해 해당 모델에 대해 무작위로 가중치가 초기화된 분류기를 추가하였다. 모델과 분류기에 대한 구체적인 정보는 <표 1>에 나타내었다. 이후 실험 모델에 대해 오프사이트 튜닝과 비교 모델 제작을 위한 미세 조정을 수행하여 각각 오프사이트 모델, 플러그인 모델, 제로 샷 모델, 그리고 전체 미세 조정 모델을 만들고 성능을 비교하였다. 이때 미세 조정이란 모델의 가중치를 학

〈표 1〉 언어모델 및 분류 모델 개요

Language Model (SKT KoGPT-2 Ver 2.0)				Classification Head	
Num. of parameters	125M	Dropout	0.1	Type	Linear
Num. of layers	12	Hidden size	768	Num. of in features	98,304
Num. of head	12	Max sequence length	128	Num. of out features	18

〈표 2〉 실험 환경

Hardware		Software	
CPU	Intel(R) Xeon(R) Gold 5220 CPU 4 cores	OS	Ubuntu 20.04.1 LTS
GPU	NVIDIA Tesla V100	Python	3.8.5
Memory	16 GB	PyTorch	2.0.1
		Transformers	4.32.1

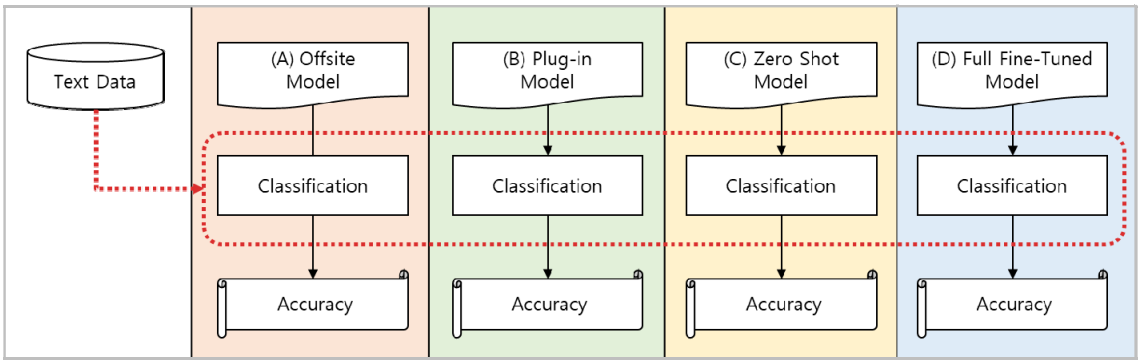
습 데이터를 통해 갱신하는 작업을 의미한다. 각 모델별로 가중치가 갱신되는 부분은 3장에서 언급한 바와 같이 서로 다르며, 구체적인 제작 방법은 이후 절에서 소개한다. 모든 실험은 Python으로 구현하였으며 실험 환경은 <표 2>와 같다.

각 모델에 대한 추론 성능 비교 실험 개요는 <그림 6>과 같다. 이때 모델 A와 모델 B는 오프사이트 튜닝 단계를 거쳐 산출되는 모델이다. 모델 C와 모델 D는 오프사이트 튜닝 결과를 비교하기 위한 목적으로 만들어진다. 그 후 각 모델에 대해 동일한 추론 데이터셋으로 분류 실험을 수행하였으며, 분류 정확도(Classification Accuracy)를 측정하여 분류 성능을 비교하였다.

$$\textit{분류 정확도}\,(\%) = \frac{\textit{올바른 클래스로 예측한 건수}}{\textit{전체 데이터셋 건수}}$$

4.2. 텍스트 데이터 전처리 결과

본 절에서는 수집한 텍스트 데이터셋에 대한 전처리 과정을 소개한다. 본 실험을 위해 AIHub에서 다운로드 받은 텍스트 데이터는 특허정보원, 한국학술정보 등에서 제공한 원천데이터를 일련의 과정을 통해 정제한 후 공개한 것이다. 제공된 데이터는 학습 데이터 1,195,228 건, 평가 데이터 149,403 건으로 구성되어 있으며, 실험을 위해 먼저 두 데이터셋을 합쳐 총 1,344,631 건의 데이터를 확보하였다. 이후 해당 데이터셋에 대해 Null 값 제거와 중복 제거를 수행하고, 정규화를 통해 링



〈그림 6〉 성능 비교 실험 프로세스

크를 제거하였다. 다음으로 KoGPT-2 에서 제공하는 토크나이저를 활용하여 텍스트 데이터를 토큰화하였다. 이러한 과정을 거쳐 각 소분야별로 학습 데이터 10,000 건, 평가 및 추론 데이터 각 1,000 씩을 추출하여 학습, 평가, 그리고 추론 데이터셋을 구축하였다.

4.3. 오프사이트 튜닝 및 플러그인 모델 제작 결과

본 절에서는 전처리한 데이터를 사용하여 오프사이트 튜닝을 수행하는 전체 과정과 결과를 소개한다. 오프사이트 모델 제작을 위해 HuggingFace 에서 KoGPT-2 모델을 다운로드 받은 후, 전체 12 개 레이어 중 위, 아래 각각 2 개씩의 총 4 개 레이어를 어댑터로, 나머지 8 개 레이어를 에뮬레이터로 구분하였다. 여기에 분류 태스크 수행을 위해 (98,304, 18) 차원의 분류기를 추가하였다. 이후 모델을 압축하기 위해 에뮬레이터 8 개 레이어에 대해 50% 압축률로 레이어 삭제를 수행하여 최종적으로 0 번째, 2 번째, 5 번째, 7 번째 레이어가 남도록 하였다. 레이어 삭제 알고리즘은 오프사이트 튜닝을 제안한 논문에서 제시하는 알고리즘을 그대로 사용하였다. 그 후 남은 4 개 레이어를 압축 에뮬레이터로 하여 가중치를 고정시킨 다음, 어댑터와 분류기만 가중치가 갱신되게 하는 미세 조정을 실시하였다.

오프사이트 모델의 어댑터와 분류기 학습을 완료한 후, 오프사이트 모델 학습 후 평가 성능이 가장 좋았던 에폭에서 저장된 어댑터를 저장했다. 오프사이트 모델은 에폭 2 에서 정확도 0.426 의 가장 우수한 성능을 나타냈으며, 이때의 모델을 구성하고 있는 어댑터를 오프사이트 모델의 어댑터로 확정하였다. 그 후 원본 KoGPT-2 모델의 기존 어댑터를 학습된 어댑터로 교체하고 분류기를 새로 추가하여 플러그인 모델을 제작하였다. 플러그인 모델에 대해 전체 레이어의 가중치를 고정하고 분류기에 대한 미세 조정을 실시하였다. 오프사이트 튜닝에 적용된 구체적인 하이퍼파라미터 정보는 <표 3>과 같다.

4.4. 비교 모델 제작 결과

본 절에서는 제안 방법론의 상대적인 성능을 비교를 위한 비교 모델의 제작 과정 및 결과를 간략히 소개한다. 먼저 제로 샷 모델을 만들기 위해 오프사이트 튜닝과 마찬가지로 KoGPT-2 모델을 불러온 후 분류기를 추가하였다. 이후 분류기를 제외한 언어모델의 가중치는 모두 고정하고, 오프사이트 튜닝에서 사용한 데이터셋으로 분류기의 학습을 진행하였다. 한편 정확도 측면에서 가장 이상적인 성능을 나타낼 것으로 예상하는 전체 미세 조정 모델은 언어모델 및 분류기의 모든 가중치를 갱신하기 때문에, 학습 가능한 파라미터의 수가 가장

〈표 3〉 모델별 학습 하이퍼파라미터 요약

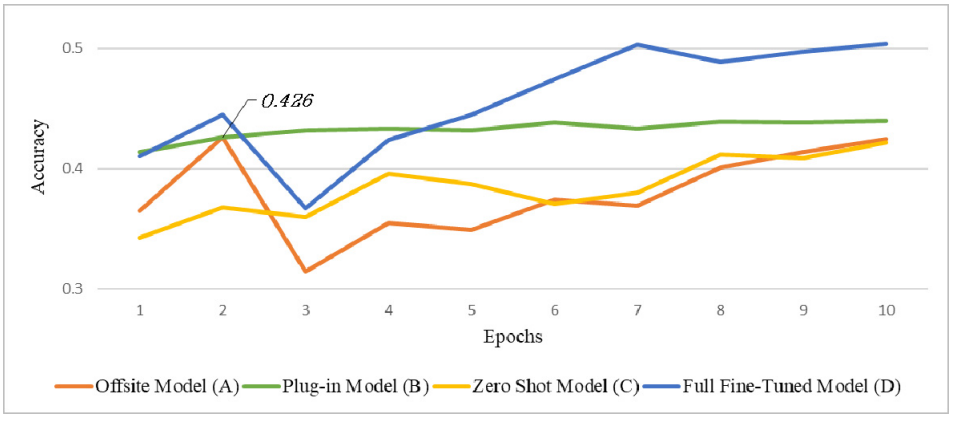
Model	Common Parameters	Learning Rate	Num. of Trainable Parameters
Offsite Model	**Epochs**=10, **Batch Size**=8 **Optimizer**=AdamW **Loss Function**=Cross Entropy **Learning Rate Scheduler**=Linear	0.00005	30,120,978
Plug-in Model		0.00001	1,769,490
Zero Shot Model		0.00001	1,769,490
Full Fine-Tuned Model		0.00001	126,934,290

많음을 <표 3>에서 확인할 수 있다. 전체 미세 조정 모델의 분류기 학습에도 역시 오프사이트 튜닝에서 사용한 동일한 데이터셋을 사용하였다.

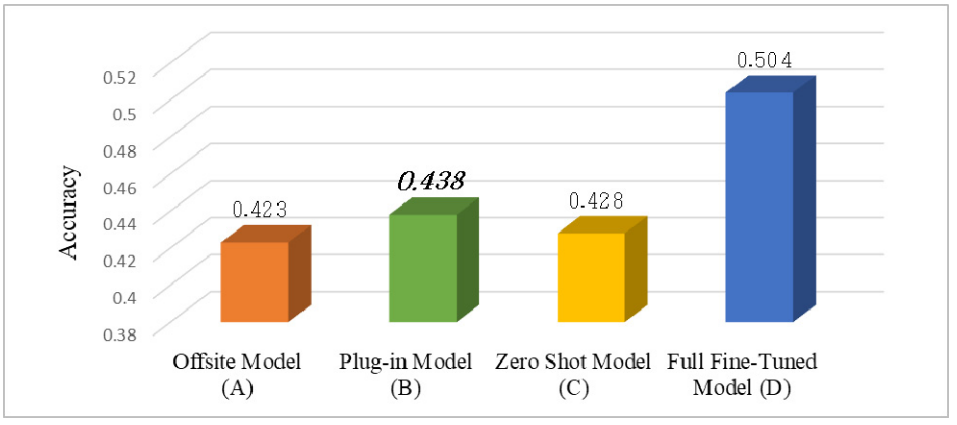
4.5. 네 가지 모델의 분류 성능 분석

본 절에서는 제안 모델과 비교 모델을 포함한 총 4가지 모델의 성능 분석 결과를 소개한다. 우선 앞의 두 절에서 소개한 4개 모델의 제작 과정에서 확인된 각 모델의 에폭별 평가 정확도는 <그림 7>과 같다. <그림 7>에서 각 모델별로 가장 우수한 성능을 보이는 에폭은 서로 다르게 나타남을 확인하였으며, 해당 지점, 즉 각 모델별로 정확도가 가장 높게 나타나는 지점에서 모델을 결정하여 분류 성능 비교에 사용하였다. 동일한 추론 데이터셋에 대해 각 모델의 분류 성능을 측정한 결과는 <그림 8>과 같다.

<그림 8>에서 플러그인 모델의 성능이 오프사이트 모델의 성능보다 우수하게 나타났으며, 이는 데이터의 소유자가 오프사이트 모델에 만족



〈그림 7〉 각 모델의 에폭별 정확도 비교



〈그림 8〉 각 모델의 분류 성능 비교

하지 않고 모델 소유자가 소유한 플러그인 모델의 사용에 대한 니즈를 유지할 수 있음을 의미한다. 또한 플러그인 모델이 제로 샷 모델에 비해 우수한 성능을 보이고 전체 미세 조정 모델에 비해 낮은 성능을 보인 것은 지극히 당연한 결과로 이해할 수 있다. 다만 전체 미세 조정 모델과 플러그인 모델의 성능 차이가 비교적 크게 나타난 것은 제안 방법론의 한계로 지적될 수 있다. 물론 전체 미세 조정 모델은 프라이버시 보호를 전혀 고려하지 않은 모델이기 때문에 두 모델의 직접적인 성능 비교는 큰 의미를 갖지는 않는다. 다만 본 실험 결과는, 궁극적으로는 프라이버시를 보호하면서도 전체 미세 조정 모델과 유사한 성능을 나타내기 위한 방안이 지속적으로 모색될 필요성을 강조하는 것으로 해석될 수 있다.

5. 결론

텍스트 딥러닝과 언어모델이 발전하면서 미세 조정을 통한 언어모델의 다양한 활용 방안들이 제안되었다. 하지만 언어모델을 미세 조정하는 과정에서 학습에 참여하는 모델 소유자와 데이터 소유자의 프라이버시가 침해될 수 있다는 우려가 제기되면서, 두 집단의 프라이버시를 모두 고려할 수 있는 방법으로 오프사이트 튜닝이 보고되었다. 하지만 해당 연구는 분류기를 통해 다중 분류를 수행하는 시나리오에 대해서는 구체적인 방안을 제시하지 않았다. 따라서 본 연구에서는 분류기가 추가된 언어모델을 대상으로 오프사이트 튜닝을 통해 플러그인 모델을 제작하는 방안을 제시하였다. 구체적으로 언어모델에 분류기를 추가한 후, 총 180,000건의 학습 데이터로 텍스트 분류 오프사이트 튜닝을 수행하였다. 이를 통해 오프사이트 모델로부터 학습된 어댑터를 사용하여 플러그인 모델을 제작한 후, 이들 모델의 성능을 제로 샷 모델 및 전체 미세 조정 모델과 비교하였다. 그 결과 제안 모델의 성능이 제로 샷 모델과 오프사이트 모델의 성능보다 우수하게 나타나는 것을 확인하였다.

본 연구는 최근 그 중요성이 점차 강조되고 있는 언어모델의 프라이버시 이슈를 다루고 있으며, 구체적으로 모델과 데이터의 프라이버시를 보호하면서 언어모델의 미세 조정을 수행하는 방안을 제시하고 있다. 본 연구에서는 양질의 오프사이트 모델 및 어댑터를 제작하기 위해, 오프사이트 모델 학습 시 에폭 최적화를 수행하는 방안을 사용하였다. 오프사이트 모델의 성능이 너무 낮은 경우에는 플러그인 모델의 성능도 함께 저하될 수 있고, 반대로 오프사이트 모델의 성능이 너무 높아서 플러그인 모델과의 성능 격차가 작은 경우에는 플러그인 모델에 대한 니즈 자체가 사라질 수 있다. 따라서 이러한 조건을 모두 만족시키면서 궁극적으로 플러그인 모델의 성능을 향상시키기 위해 에뮬레이터의 압축률 조절, 오프사이트 튜닝 시의 학습 파라미터 조정 등의 다양한 시도가 후속 연구에서 다루어질 것으로 기대하며, 이는 본 연구의 학술적 기여로 인정받을 수 있다. 또한 본 연구는 디코더 기반의 한국어 언어모델에서 오프사이트 튜닝의 적용 가능성을 보였으며, 이는 추후 한국어 기반의 거대 언어모델 미세 조정 시 해당 기술을 적용하고자 할 때 도움을 줄 수 있다는 점에서 실무적 기여를 인정받을 수 있다.

하지만 본 연구는 실제로 프라이버시 보호가 요구되는 민감한 모델 및 데이터를 사용한 실험 대신, 공개된 모델과 데이터에 대해 제한적인 실험을 수행하였다는 한계가 있다. 향후 다양한 모

델 및 데이터에 대한 실험을 통해 제안 방법론의 효용성을 확인할 필요가 있다. 또한 플러그인 모델의 학습 과정에서 데이터 전체에 대한 직접적인 접근은 발생하지 않았지만, 플러그인 모델의 분류기 학습 과정에는 전체 데이터가 사용되었다는 점에 유의해야 한다. 이러한 설정은 엄밀한 측면에서 프라이버시 침해의 요인으로 작용할 수 있으므로, 향후 분류기 학습 역시 오프사이트 모델에서 학습한 분류기를 플러그인 모델이 재사용하는 방식으로 개선이 이루어져야 한다. 마지막으로 약간의 학습 파라미터 변화로도 오프사이트 모델과 플러그인 모델의 성능 격차가 역전될 수 있으므로, 모델 프라이버시와 어댑터의 품질을 고려하여 학습 파라미터의 다양한 조합을 시도하는 방식으로 더욱 엄밀한 실험이 수행되어야 한다.

참고문헌(References)

Arachchige, P. C. M., Bertok, P., Khalil, I., Liu, D., Camtepe, S., & Atiquzzaman, M. (2019). *Local differential privacy for deep learning. IEEE Internet of Things Journal, 7*(7), 5827-5842.

Baracaldo, N., Chen, B., Ludwig, H., & Safavi, J. A. (2017). Mitigating poisoning attacks on machine learning models: A data provenance based approach. *Proceedings of the 10th ACM workshop on artificial intelligence and security,* 103-110.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., & Roli, F. (2013). Evasion attacks against machine learning at test time. *Machine Learning and Knowledge Discovery in Databases,* 387-402.

Brown, T. B., (2018, January 23). Adversarial Patch. youtube. Retrieved September 5, 2023, from https://www.youtube.com/watch?v=i1sp4X57TL4

Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. *International Conference on the Theory and Applications of Cryptology and Information Security,* 409-437.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805,* 1 - 16.

Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1322-1333.

Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. *International conference on machine learning*, 201-210.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572,* 1-11.

Huang, Y., Su, Y., Ravi, S., Song, Z., Arora, S., & Li, K. (2020). Privacy-preserving learning via deep net pruning. *arXiv preprint arXiv:2003.01876,* 1-43.

Lee, Y., Chen, A. S., Tajwar, F., Kumar, A., Yao, H., Liang, P., & Finn, C. (2022). Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466,* 1-25.

Martins, P., Sousa, L., & Mariano, A. (2017). A survey

on fully homomorphic encryption: An engineering perspective. *ACM Computing Surveys (CSUR), 50*(6), 1-33.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics, PMLR*, 1273-1282.

Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016). Distillation as a defense to adversarial perturbations against deep neural networks. *2016 IEEE symposium on security and privacy (SP),* 582-597.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., & Erlingsson, Ú. (2018). Scalable private learning with pate. *arXiv preprint arXiv: 1802.08908,* 1-34.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training, *Preprint*, 1–12.

Sajjad, H., Dalvi, F., Durrani, N., & Nakov, P. (2023). On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language, 77,* 1-12.

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. *IEEE symposium on security and privacy (SP),* 3-18.

Tian, Z., Cui, L., Liang, J., & Yu, S. (2022). A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys, 55*(8), 1-35.

Wang, T., & Liu, L. (2011). Output privacy in data mining. *ACM Transactions on Database Systems (TODS), 36*(1), 1-34.

Xiao, G., Lin, J., & Han, S. (2023). Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 1–12.

Xu, R., Baracaldo, N., & Joshi, J. (2021). Privacy-preserving machine learning: Methods, challenges and directions. *arXiv preprint arXiv:2108.04417,* 1-40.

Abstract

Privacy-Preserving Language Model Fine-Tuning Using Offsite Tuning

Jinmyung Jeong* · Namgyu Kim**

Recently, Deep learning analysis of unstructured text data using language models, such as Google's BERT and OpenAI's GPT has shown remarkable results in various applications. Most language models are used to learn generalized linguistic information from pre-training data and then update their weights for downstream tasks through a fine-tuning process. However, some concerns have been raised that privacy may be violated in the process of using these language models, i.e., data privacy may be violated when data owner provides large amounts of data to the model owner to perform fine-tuning of the language model. Conversely, when the model owner discloses the entire model to the data owner, the structure and weights of the model are disclosed, which may violate the privacy of the model. The concept of offsite tuning has been recently proposed to perform fine-tuning of language models while protecting privacy in such situations. But the study has a limitation that it does not provide a concrete way to apply the proposed methodology to text classification models. In this study, we propose a concrete method to apply offsite tuning with an additional classifier to protect the privacy of the model and data when performing multi-classification fine-tuning on Korean documents. To evaluate the performance of the proposed methodology, we conducted experiments on about 200,000 Korean documents from five major fields, ICT, electrical, electronic, mechanical, and medical, provided by AIHub, and found that the proposed plug-in model outperforms the zero-shot model and the offsite model in terms of classification accuracy.

Key Words : Deep Learning, Language Model, Language Model Privacy, Fine-Tuning, Offsite-Tuning, Text Classification

Received : October 24, 2023 Revised : November 27, 2023 Accepted : November 29, 2023
Corresponding Author : Namgyu Kim

* Graduate School of Business IT, Kookmin University
** Corresponding Author: Namgyu Kim
Graduate School of Business IT, Kookmin University
77 Jeongneung-ro, Seongbuk-gu, Seoul 136-702, Korea
Tel: +82-2-910-5425, Fax: +82-2-910-4017, E-mail: ngkim@kookmin.ac.kr

저 자 소 개

정진명
현재 국민대학교 비즈니스IT전문대학원 석사과정에 재학 중이다. 건국대학교 글로컬 캠퍼스 생명공학전공으로 학사 학위를 취득하였으며, 한국지능정보시스템학회 학술대회 최우수 논문상을 수상하였다. 주요 관심분야는 자연어 처리, 딥러닝 등이다.

김남규
현재 국민대학교 비즈니스IT전문대학원 및 경영정보학부 교수로 재직 중이다. 서울대학교 컴퓨터공학과에서 학사 학위를 취득하고, KAIST 테크노경영대학원에서 Database와 MIS를 전공하여 경영공학 석사 및 박사학위를 취득하였다. 한국지능정보시스템학회 부회장, 한국정보기술응용학회 부회장, 한국경영학회 상임이사, 한국경영정보학회 이사, 한국인터넷정보학회 이사 등을 역임하였으며, 주요 관심분야는 텍스트 마이닝, 딥러닝, 데이터 모델링 등이다.