Journal of the Korean Institute of Industrial Engineers
Vol. 49, No. 2, pp. 133-141, April 2023.
ISSN 1225-0988 | EISSN 2234-6457
https://doi.org/10.7232/JKIIE.2023.49.2.133

EDAD: 도메인 적응과 지식 증류를 통합한 효율적 도메인 적응 증류

서승원[1] · 황상흠[2†]

[1]서울과학기술대학교 데이터사이언스학과 / [2]서울과학기술대학교 산업공학과

EDAD: Efficient Domain Adaptive Distillation by Integrating Domain Adaptation and Knowledge Distillation

Seungwon Seo[1] · Sangheum Hwang[2]

[1]Department of Data Science, Seoul National University of Science and Technology
[2]Department of Industrial Engineering, Seoul National University of Science and Technology

In the field of natural language processing, a lot of progress has been made with the advent of Transformer having a self-attention mechanism. At the same time, the recently increasing model size causes difficulties in deploying the model for online serving that requires fast inference. To address this issue, one can employ model compression techniques when a target domain is coherent with the training corpus (i.e., a general domain) of pre-trained models such as BERT. However, the additional domain adaptation step is required along with model compression when we leverage such pre-trained models for special target domains such as medicine, law, finance, etc. In this paper, we propose an Efficient Domain Adaptive Distillation (EDAD) method to efficiently create a lightweight model capable of fast inference for a target domain by integrating knowledge distillation, which is one of the popular model compression methods, and domain adaptation processes. Experimental results demonstrate that EDAD can train a compact model for a target domain with much lower computational costs by integrating the two individual processes, adaptation and compression, into a single process and shows comparable performance with existing methods for named entity recognition (NER) tasks in the medical domain.

Keywords: Named Entity Recognition, Natural Language Processing, Domain Adaptation, Knowledge Distillation

1. 서 론

GPT(Radford *et al.*, 2018), BERT(Devlin *et al.*, 2019) 등 큰 규모의 텍스트 데이터로 사전학습된 언어 모델의 등장으로 자연어처리 분야에서 많은 발전이 이뤄지고 있다(Yao *et al.*, 2021). 특히 Transformer encoder 기반의 모델인 BERT가 다양한 자연어처리 태스크에서 대체로 좋은 성능을 보여 아직까지도 많은 자연어 태스크에 활용되고 있다. 하지만 BERT가 사전학습에 사용한 데이터는 Wikipedia, BookCorpus와 같은 일반적인 도메인의 데이터이기 때문에 의학(medicine)이나 금융(finance) 등 전문적인 도메인에 적용하기에는 최적의 선택이 아닐 수 있다. 이는 일반적인 도메인에서 사용되는 단어들의 분포와 전문적인 배경지식을 필요로 하는 특정 도메인에서 사용되는 단어들의 분포가 다를 뿐만 아니라 같은 단어라도 전혀 다른

이 논문은 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF- 2019R1A6A1A03032119)과 2022년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행되었음(P0017123, 2022년 산업혁신인재성장지원사업).
† 연락저자 : 황상흠 교수, 01811 서울시 노원구 공릉로 232 서울과학기술대학교 프론티어관 618호, Tel : 02-970-6462, E-mail : shwang@seoultech.ac.kr
2022년 11월 21 접수; 2023년 01월 08일 수정본 접수; 2023년 02월 13일 게재 확정.

의미를 가질 수 있기 때문이다. 따라서 사전학습된 BERT를 바로 활용하지 않고 타겟 도메인의 텍스트로 추가적인 학습을 진행하여 해당 도메인에 대해 최적의 모델을 얻기 위한 BioBERT(Lee *et al.*, 2019), FinBERT(Araci, 2019) 등의 연구가 진행되어 왔다. 또한 일반적인 텍스트로 사전학습된 BERT를 활용하지 않고 처음부터 타겟 도메인에 대한 대량의 텍스트 데이터로 학습을 진행하는 SciBERT(Beltagy *et al.*, 2019), PubmedBERT(Gu *et al.*, 2020) 등의 연구들도 존재한다. 그러나 앞서 말한 방법들은 타겟 도메인에 대한 거대한 텍스트 데이터셋과 이를 학습시키는데 많은 연산 자원을 필요로 하기 때문에 비용적인 측면에서 단점을 가지고 있다. 이에 따라 최근에는 적은 자원(GPU, 데이터 등)의 환경에서 효율적인 도메인 적응을 위해 vocabulary를 해당 도메인에 맞게 확장하는 등의 방법들도 연구되고 있다(Tai *et al.*, 2020, Yao *et al.*, 2021).

한편 자연어처리 분야를 포함한 딥러닝 분야에서는 더 큰 모델을 더 많은 데이터로 학습을 하는 것이 트렌드로 이어지고 있다. 최근에 발표된 언어 모델인 Megatron-NLG(Smith *et al.*, 2022), PaLM(Chowdhery *et al.*, 2022)은 무려 각각 530B개와 540B개의 파라미터를 가지며, 학습에 사용된 데이터셋은 각각 338.6B개의 토큰과 780B개의 토큰으로 구성된다. 모델과 데이터셋 규모의 확장은 분명 일반화 성능과 양의 상관관계를 보이고 있지만 학습 시간, 추론 시간 등 시간적인 비용 측면에서 해결해야 하는 문제점들이 많다. 또한 해당 모델과 데이터를 학습시키기 위해 필요한 컴퓨팅 자원 측면에서의 비용도 매우 크다는 단점이 있다. 따라서 빠른 모델 서빙과 실시간으로 수집되는 데이터에 빠른 추론을 요구하는 서비스에서나, IoT 디바이스에 모델을 이식해야 하는 상황에서는 BERT를 포함한 큰 언어 모델들은 한계를 가진다. 빠른 추론과 가벼운 모델을 만들기 위한 모델 압축 방법들은 이전부터 활발히 연구되고 있다. 가지치기(pruning), 지식 증류(knowledge distillation), 양자화(quantization) 등 다양한 모델 압축 방법이 존재하는데 흔히 사용하는 모델 압축 방법 중 하나는 상대적으로 큰 teacher 모델의 지식을 작은 student 모델로 전이(transfer)하는 지식 증류(knowledge distillation) 방식이다. 자연어처리 분야에서의 지식 증류 방법으로는 TinyBERT(Jiao *et al.*, 2020), MiniLM(Wang *et al.*, 2020) 등의 연구가 존재한다.

지식 증류 방법을 통해 타겟 도메인에 대해 가볍고 빠른 모델을 만드는 방법은 크게 두 가지 접근으로 나누어 볼 수 있다. 첫 번째는 타겟 도메인에 적응된 큰 모델을 작은 모델로 지식 증류를 진행하는 것, 두 번째는 일반적인 도메인의 큰 데이터셋으로 사전학습된 BERT로부터 지식 증류를 통해 작은 모델을 학습한 뒤, 학습된 작은 모델을 타겟 도메인에 적응시키는 것이다. 그러나 이 두 가지 방법 모두 순서에 관계없이 도메인 적응과 지식 증류, 두 개의 단계를 독립적으로 거쳐야 하는 단점이 존재한다. 따라서 본 논문에서는 독립적인 두 개의 단계를 하나로 통합하여 타겟 도메인을 위한 가볍고 빠른 모델을 효율적으로 만들어내는 것을 목적으로 하는 프레임워크인 효율적 도메인 적응 증류(Efficient Domain Adaptive Distillation, 이하 EDAD) 방법을 제안한다. 이 때, 도메인 적응 방식으로는 적은 데이터로 빠르게 타겟 도메인에 대해 적응할 수 있는 exBERT(Tai *et al.*, 2020) 방식을 채택하여 EDAD 프레임워크를 구성했고 지식 증류 방식으로는 가장 널리 사용되는 방법 중 하나인 TinyBERT를 활용한다. EDAD는 도메인 적응과 지식 증류를 단순히 동시에 진행하는 것이 아니라 α라는 파라미터를 도입하여 학습 과정에서 두 목적 함수의 반영 정도를 조절한다. 학습 초반에는 지식 증류에 관한 반영 비율을 높여 student 모델이 대규모의 데이터셋으로 학습된 teacher 모델의 지식을 전이받도록 하고, 후반으로 갈수록 도메인 적응에 관한 반영 비율을 높여 최종적으로 student 모델이 타겟 도메인에 대한 지식을 학습하도록 한다. 이전 문단에서 언급한 바와 같이, 타겟 도메인에 대한 큰 규모의 데이터셋은 학습하기에도 비용이 클 뿐 아니라 수집마저 어려울 수 있기 때문에 본 연구는 다른 도메인 적응 방법(e.g., BioBERT, SciBERT)에서 사용한 데이터에 비해 상대적으로 적은 데이터를 사용하여 도메인 적응을 진행한다.

본 연구에서는 자연어처리 분야에서 도메인 적응 토픽이 가장 많이 연구되는 분야인 의학 도메인을 타겟 도메인으로 설정했고, named entity recognition (NER) 태스크에 해당되는 BC5CDR, BioNLP09, NCBI-disease 세 종류의 데이터셋을 활용하여 제안한 방법의 평가를 진행했다. 실험 결과, 두 개의 단계를 하나로 통합한 EDAD는 시간 비용을 줄일 뿐만 아니라, 도입한 α를 통해 도메인 적응과 지식 증류의 반영 비율을 조절함으로써 학습 시간 대비 우수한 성능을 보임을 확인했다.

정리하자면, 본 논문의 contribution은 다음과 같다.

- 기존에는 전문적인 타겟 도메인에 대한 가벼운 모델을 만들기 위해서 도메인 적응과 지식 증류 단계가 독립적으로 진행되어 왔으나, 본 논문에서는 효율적인 타겟 도메인 모델 학습을 위해 두 단계를 하나로 통합한 EDAD 방법을 제안한다.
- 제안하는 방법의 효과를 검증하기 위해 세 종류의 NER 데이터셋에 대한 비교실험을 진행했고, 그 결과 EDAD가 효율적으로 타겟 도메인에 대한 가벼운 모델을 만들 수 있음을 보였다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 제2장에서는 도메인 적응과 지식 증류에 대한 선행 연구들을 소개한다. 제3장에서는 제안된 EDAD 방법론에 대해 자세히 설명한다. 제4장은 성능 검증을 위한 실험 환경과 결과를 포함하고 있고 마지막으로 제5장에서 결론을 서술한다.

2. 선행 연구

2.1 자연어처리 분야의 도메인 적응

도메인 적응이라는 토픽은 자연어처리 분야에서 중요하게

다뤄지는 문제 중 하나이다. 최근, 대량의 텍스트 데이터로 사전 학습된 모델이 다수 등장하고 있다. 그러나 해당 모델들이 학습에 사용한 데이터는 일반적인 도메인의 텍스트이기 때문에 의학, 금융과 같은 전문적인 도메인에서 해당 모델들을 바로 활용하기에는 타겟 도메인에 관련된 전문적인 지식이 부족할 수 있다. 이를 해결하고자 타겟 도메인에 특화된 모델을 만들기 위한 여러 연구가 진행되어 왔다. 특히 대량의 데이터가 공개되어 있는 의학 도메인에 대해 도메인 적응에 관련한 연구가 활발하게 이루어지고 있다.

현재까지 연구된 의학 도메인에 대한 적응 방식은 크게 세 가지 범주로 나눌 수 있다. 첫 번째는 BioBERT의 유형이다. BioBERT는 사전학습된 BERT를 온전히 활용하여 대량의 의학 도메인 데이터에 대해 추가로 학습을 진행한다. 따라서 BioBERT는 BERT와 동일한 vocabulary를 갖게 되는데 여전히 의학 도메인에서만 사용되는 일부 단어들을 모델링하기 어렵다는 한계가 존재한다. 이러한 단점을 지적하며 등장한 것이 두 번째 유형인 SciBERT의 방식이다. SciBERT는 사전학습된 BERT를 활용하지 않고 새로운 vocabulary와 대량의 타겟 도메인 데이터(의학 도메인 82%, 컴퓨터 과학 18%)에 대해 학습을 진행한다. 새로운 vocabulary는 타겟 도메인의 데이터로부터 만들어지기 때문에 충분히 전문적인 단어들을 잘 모델링할 수 있다는 장점이 있다. 두 방식 모두 의학 도메인에서 사전학습된 BERT보다 유의미한 차이를 보이며 좋은 성능을 낼 수 있었다. 하지만 모두 대량의 타겟 도메인 데이터를 필요로 하기 때문에 이를 학습시키는 데에 높은 비용의 시간 및 컴퓨팅 자원이 요구된다는 단점이 있다. 이를 보완하기 위해 최근 등장한 세 번째 도메인 적응 유형이 exBERT의 방식이다. 해당 방식은 첫 번째 유형과 동일하게 사전학습된 BERT를 그대로 활용하되 vocabulary와 모델의 확장을 통해 타겟 도메인에서만 사용되는 단어도 잘 모델링할 수 있도록 하는 접근을 취한다. 비록 의학 도메인이라도 일반적인 도메인에서 쓰이는 단어들과 겹치는 부분이 다수 존재하기 때문에 대량의 일반적인 도메인 데이터로 사전학습된 BERT의 지식을 활용할 수 있다는 장점과 확장된 vocabulary와 모듈을 통해 타겟 도메인의 지식도 학습할 수 있다는 이점 덕분에 BioBERT나 SciBERT 보다 효율적으로 도메인 적응이 가능하다. Tai *et al.*(2020)에서 같은 학습 시간 대비 BioBERT보다 exBERT가 여러 downstream 태스크에서 높은 성능을 보이는 것을 실험적으로 확인했으며, 해당 연구 이후 vocabulary를 확장하여 타겟 도메인에 적응하는 방법들이 등장하고 있다(Hong *et al.*, 2020; Yao *et al.*, 2021). 본 연구에서는 효율적인 도메인 적응을 위해 exBERT의 방식을 차용한다.

2.2 자연어처리 분야의 지식 증류

자연어처리 분야에서 도메인 적응과 더불어 또 다른 중요한 토픽은 바로 모델 압축이다. IoT와 빅데이터 관련 기술의 발전을 계기로 점점 더 많은 데이터들이 쌓여가고 있고 이를 실시간으로 처리해야 하는 서비스도 늘어나고 있다. 자연어처리 분야에서도 마찬가지로 모델의 실시간 추론, 혹은 온 디바이스(on device) AI 등을 요구하는 서비스들이 많아지고 있다. 하지만 BERT를 포함한 최신의 모델들은 점점 크기가 커져가고 있기 때문에 위와 같은 서비스에서 활용하기에 적합하지 않다. 따라서 해당 모델들의 성능을 최대한 유지할 수 있는 가벼운 모델을 만들기 위한 연구가 이전부터 활발하게 진행 중이다. 그 방법 중 하나가 지식 증류(knowledge distillation)이다. 이는 Hinton *et al.*(2015)에 의해 처음 등장한 개념으로, 상대적으로 큰 teacher 모델과 작은 student 모델을 두고, student 모델이 최대한 teacher 모델을 모방할 수 있도록 유도하는 방법이다. 자연어처리 분야에서는 MiniLM 등의 다양한 지식 증류 방식들이 연구되고 있다. 그 중 가장 흔히 사용되는 지식 증류 방법인 MiniLM과 TinyBERT를 간략히 설명하자면, MiniLM은 teacher 모델과 student 모델의 마지막 layer 간의 주의 행렬(attention matrix)과 주의 모듈(attention module)의 value에 대한 correlation matrix가 유사해지도록 학습하는 방식이고 TinyBERT는 teacher 모델의 일부 layer와 student 모델의 모든 layer의 잠재 상태(hidden state)와 주의 행렬이 유사해지도록 유도하는 방식이다. 두 방식에서 볼 수 있듯이 자연어처리 분야에서의 지식 증류는 주로 언어 모델에서 가장 중요한 메커니즘인 주의 메커니즘(attention mechanism)을 모방하려 한다는 특징이 있으며 본 논문에서 제안하는 EDAD에서는 TinyBERT 방식을 활용한다.

3. 방법론

3.1 도메인 적응을 위한 모델 확장

본 논문에서 제안하는 EDAD의 프레임워크는 <Figure 1>과 같다. 먼저 exBERT의 구조를 차용하여 teacher 모델과 student 모델에 extension module을 각각 추가함으로써 확장된다. Original module은 기존 BERT의 block에 대응하며 구조는 <Figure 1>-B와 같다. Extension module은 original module과 동일한 구조이며 exBERT의 실험 결과에 따라 extension module의 크기를 original module 크기의 약 33%로 설정하였다. Teacher 모델은 $\text{BERT}_{\text{Base}}$를 확장한 모델이며 student 모델은 $\text{BERT}_{\text{Tiny}}$를 확장한 모델이다. Teacher 모델은 original module에 사전학습된 BERT의 파라미터를 불러온 뒤 해당 부분을 freeze하고 학습을 진행한다. 일반적인 도메인의 텍스트로 학습된 지식들은 original module에 그대로 보존하면서 extension module에 타겟 도메인의 지식을 학습함으로써 효율적인 도메인 적응을 할 수 있게 된다. Student 모델 역시 original module과 extension module을 가지지만 teacher와 달리 모든 파라미터들이 타겟 도메인에 대한 데이터셋으로 학습을 진행하게 된다. 또한 student 모델에만 도메인 적응 loss와 지식 증류 loss가

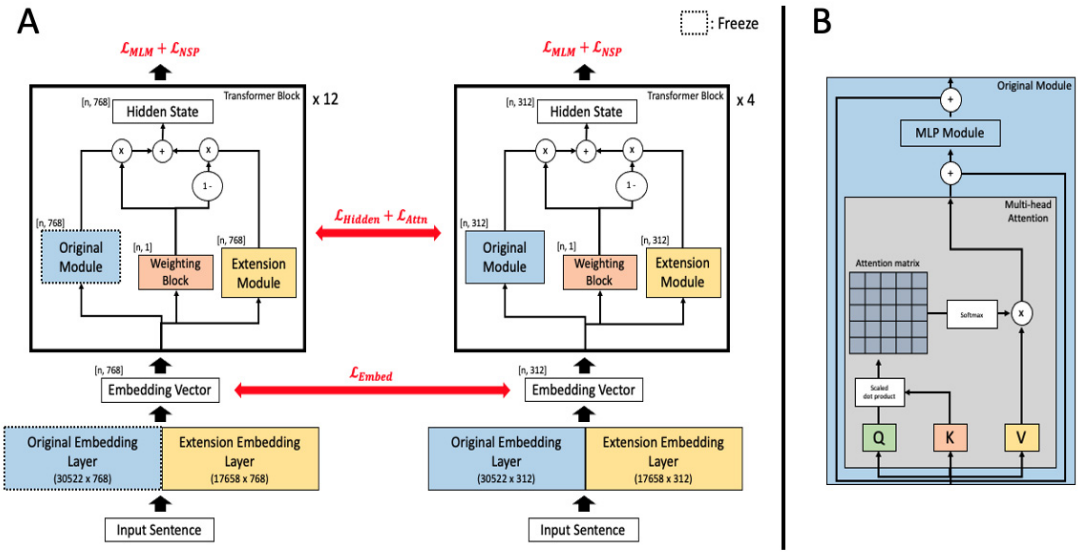

Figure 1. EDAD Framework. (A-left) Teacher model, (A-right) student model and (B) original module

함께 적용되는데 이는 다음 장에서 설명한다. 한편, teacher 모델과 student 모델 모두 확장된 vocabulary인 exBERT의 vocabulary를 사용한다. exBERT의 vocabulary는 original BERT의 vocabulary에 의학 도메인의 텍스트로부터 WordPiece (Wu *et al.*, 2016) 방식으로 만들어진 도메인 특화된(domain-specific) vocabulary를 추가로 확장하여 만들어진다. 이 때 domain-specific vocabulary에서 original BERT의 vocabulary와 겹치는 token들은 모두 제외하게 된다. 이렇게 만들어진 vocabulary의 크기는 original BERT vocabulary의 크기인 30,522에 domain-specific vocabulary의 크기인 17,658이 더해져 총 48,180이다.

3.2 지식 증류

도메인 적응을 위한 방법에 이어, 우리는 지식 증류의 방법으로 TinyBERT를 채택한다. TinyBERT는 상대적으로 작은 student 모델의 모든 layer의 잠재 상태(hidden state)와 주의 행렬(attention matrix)을 teacher의 일부 layer의 잠재 상태와 주의 행렬과 유사해지도록 유도하는 방법이다. 예를 들어, teacher 모델의 layer가 12개이고 student 모델의 layer가 4개인 경우, teacher 모델의 $3\times k(k\in\{1,2,3,4\})$번째 layer와 student 모델의 k번째 layer 간의 지식 전이가 진행된다. 구체적으로, (a) input이 embedding layer를 통해 나온 두 모델의 embedded 벡터 간의 mean squared error(이하 MSE), (b) multi-head attention에서 계산되는 두 모델의 주의 행렬 간의 MSE, (c) BERT block을 통해 나온 두 모델의 잠재 상태 간의 MSE를 최소화하는 방향으로 학습이 진행된다. 이 때 teacher 모델과 student 모델의 잠재 차원(hidden dimension)이 다르므로 (a)와 (c)를 계산할 때 linear layer 한 개를 추가하여 student 모델의 embedded 벡터와 잠재 상태의 차원을 teacher 모델의 잠재 차원으로 맞춰준 뒤 MSE 계산을 하게 된다. (a), (b), (c) 과정을 수식으로 나타내면 각각 식 (1), (2), (3)과 같다. E는 embedded 벡터, A는 주의 행렬, H는 잠재 상태, M은 잠재 차원을 맞추어 주기 위한 linear layer의 파라미터, l은 input의 길이, d와 d'는 각각 teacher와 student 모델의 잠재 차원, h는 head 개수를 의미한다.

$$L_{Embed} = MSE(E^S M,\ E^T), \quad \text{where } E^S \in R^{l\times d'},\ E^T \in R^{l\times d},\ M \in R^{d'\times d} \tag{1}$$

$$L_{Attn} = \frac{1}{h}\sum_{i=1}^{h} MSE(A_i^S,\ A_i^T), \quad \text{where } A_i \in R^{l\times l} \tag{2}$$

$$L_{Hidden} = MSE(H^S M,\ H^T), \quad \text{where } H^S \in R^{l\times d'},\ H^T \in R^{l\times d},\ M \in R^{d'\times d} \tag{3}$$

TinyBERT의 과정은 크게 general distillation과 task-specific distillation으로 이루어져 있는데 본 연구에서는 특정 downstream 태스크에 특화된 모델이 아니라 특정 도메인에 특화된 작은 모델을 만드는 것이 목적이므로 general distillation만 고려한다. 따라서 최종적인 loss L_{Tiny}는 식 (4)와 같다.

$$L_{Tiny} = L_{Embed} + L_{Hidden} + L_{Attn} \tag{4}$$

3.3 EDAD

EDAD는 도메인 적응 과정과 지식 증류 과정을 통합한다. 학습에 활용되는 loss는 크게 두 가지이다. 첫 번째는 도메인 적응 과정의 loss인 기존의 BERT loss(식 (5)), 두 번째는 지식 증류 과정의 loss인 L_{Tiny}이다(식 (4)). 식 (5)는 BERT의 학습

방식인 masked language modeling(MLM)과 next sentence prediction(NSP)로 이루어져 있으며(Devlin *et al.*, 2019), 식 (4)는 3.2장에서 언급한 바와 같이 teacher와 student 모델 간의 잠재 상태와 주의 행렬에 대한 MSE loss이다. EDAD는 두 가지 목적함수를 단순하게 합하지 않고 α라는 파라미터를 도입하여 학습 과정 동안 도메인 적응과 지식 증류의 반영 비율을 조절한다(식 (6)). α는 선형적으로 감소하게 되는데, 학습 초반에는 α=1부터 시작하여 식 (4)를 주로 반영하여 학습이 진행되고 후반으로 갈수록 식 (5)에 비중을 두어 학습하게 된다. Teacher 모델은 식 (5)에 의해서만 학습되고 student 모델은 식 (4)와 식 (5)가 결합된 식 (6)에 의해 학습함으로써 효율적으로 타겟 도메인에 대한 작은 모델을 만들어 낼 수 있게 된다. α에 대한 분석은 4.4.4절에서 진행한다.

$$L_{BERT} = L_{MLM} + L_{NSP} \tag{5}$$

$$L_{Total} = (1-\alpha) L_{BERT} + \alpha L_{Tiny} \tag{6}$$

4. 실 험

4.1 데이터셋

본 연구에서는 공개된 데이터가 많은 의학 도메인에 초점을 맞춰 실험을 진행했다. 의학 도메인에서의 대표적인 태스크 중 하나인 named entity recognition(NER) 데이터셋을 활용하여 다양한 방법들 간의 비교를 진행했다. NER 태스크는 문장 내에서 각 토큰들에 대해 사전에 정의된 명칭(i.e., named entity)을 분류하는 태스크로 품사 태깅이 하나의 예시이다. 성능 비교를 위해 Tai *et al.*(2020)와 동일하게 MTL-Bioinformatics-2016 (https://github.com/cambridgeltl/MTL-Bioinformatics-2016)에서 제공하는 BC5CDR, BioNLP09, NCBI-disease를 사용한다. BC5CDR은 “B-Chemical”, “B-Disease”, “I-Chemical”, “I-Disease” 총 네 개의 named entity가 존재하며 B는 시작 이름, I는 중간이름을 뜻한다. 예를 들어 “povidone-iodin”에서 povidone은 “B-Chemical”, -iodine은 “I-Chemical”로 명명된다. BioNLP09는 “B-Protein”와 “I-Protein”, 그리고 NCBI-disease는 “B-Disease”와 “I-Disease”의 named entity가 존재한다. 도메인 적응과 지식 증류 과정에 사용한 의학 도메인의 텍스트 데이터는 PubMed Central(PMC, https://www.ncbi.nlm.nih.gov/pmc/tools/ftp/)에 공개되어 있으며, 약 500M개의 토큰이 포함된 텍스트 데이터(이하 PMC500M)를 사용했다. 이는 BERT나 SciBERT, BioBERT에서 사용한 데이터셋에 비해 최소 6배에서 최대 9배 정도 작은 크기이다.

4.2 비교 방법론

본 연구에서는 teacher 모델의 도메인 적응 이후 지식 증류를 진행하여 만들어낸 student 모델(이하 Two-stage 방법)과 도메인 적응과 지식 증류 과정을 하나로 통합한 EDAD를 통해 만들어진 student 모델을 비교한다. Two-stage 방법에서 teacher 모델의 도메인 적응 방식은 2장에서 언급했던 모든 도메인 적응 유형(i.e., BioBERT, SciBERT, exBERT)을 활용한다. 대규모의 의료 데이터로 학습된 BioBERT v1(이하 BioBERT)과 SciBERT를 teacher로 활용했을 때와 상대적으로 작은 규모의 PMC500M 데이터셋을 활용해 각각의 방법을 따라 직접 구현한 BioBERT(our implemented BioBERT, 이하 oiBioBERT)와 SciBERT(our implemented SciBERT, 이하 oiSciBERT) 모델을 teacher로 활용했을 때 모두를 비교했다. exBERT 또한 PMC500M 데이터셋을 활용해 직접 구현하여 teacher 모델로 활용했다. 모든 student 모델은 TinyBERT의 general distillation(GD) 과정을 통해 학습되므로 각 도메인 적응 방식에 GD를 아래 첨자로 붙여 명명한다(e.g., $\text{BioBERT}_{\text{GD}}$). 한편 BioBERT와 SciBERT는 huggingface(https://huggingface.co/)에서 제공하는 모델을 사용하였는데, BioBERT는 cased vocabulary 버전만 제공되기 때문에 해당 버전을 사용했다. 따라서 BioBERT, $\text{BioBERT}_{\text{GD}}$를 제외한 모든 모델은 uncased vocabulary 버전이다.

4.3 실험 세팅

공정한 비교를 위해 모두 동일한 하드웨어 환경에서 실험을 진행하였다. 사용한 gpu는 RTX 3090 4개이며 데이터 전처리 시간을 제외한 오로지 학습에 걸리는 시간만을 측정하여 방법들간 비교를 진행했다. 먼저 Two-stage 방법의 경우, teacher 모델을 의학 도메인에 대해 적응시키는 과정을 거친 뒤 지식 증류 과정을 진행하였다. 이러한 Two-stage 방법은 도메인 적응과 지식 증류 과정 모두 3 epoch 동안 진행하여 총 6 epoch의 학습이 필요한 반면 EDAD는 오로지 3 epoch 동안만 학습하며 이때 도메인 적응과 지식 증류를 함께 진행한다. 하이퍼파라미터에 대한 세팅으로는 먼저 batch size와 input의 최대 길이는 128로 고정했으며, learning rate는 Two-stage 방법의 첫 번째 단계(도메인 적응)와 EDAD에서는 1e-4, Two-stage 방법의 두 번째 단계(지식 증류)에서는 1e-5를 사용했다. oiSciBERT의 경우에는 learning rate를 고정하지 않고 스케쥴링을 적용하였는데 랜덤 추출된 초기값으로부터 학습되는 oiSciBERT의 특성상 학습의 수렴 양상이 안정적이지 않았기 때문이다. oiSciBERT의 수렴을 위해 BERT의 학습 방식을 참조하여 전체 iteration의 10% 동안은 warm up을 수행하고 그 이후 선형적으로 감소시키는 방식으로 학습하였다. 추가로 EDAD에 도입된 식 (6)의 α는 3.3절에서 언급한 바와 같이 1부터 시작하여 0까지 선형적으로 감소한다. 이를 제외한 나머지 하이퍼파라미터는 모두 Devlin *et al.*(2019)와 동일하다. 마지막으로 모델 간의 비교는 downstream 태스크에 대해 teacher와 student 각각 3, 15 epoch동안 fine-tuning하여 검증한 결과이며 평가 척도는 F1-score이다. Epoch은 훈련 과정에서 loss가 충분히 수렴하는지를 고려하여 결정했다. 상대적으로 파라미터 개수가 많은

teacher 모델은 적은 epoch만으로도 충분히 loss가 수렴했고 student 모델은 상대적으로 더 오랜 학습이 필요했다. 한편, 보다 엄밀한 비교 검증을 위해 downstream 태스크에 대한 F1-score는 5번 반복실험 후 평균값을 취했고, 도메인 적응과 지식 증류 과정에서 걸린 총 시간을 의미하는 time cost도 3번 반복실험 후 평균값을 비교했다. 또한 F1-score와 time cost 모두 신뢰수준 95%에 대한 오차도 함께 기록했다.

4.4 결과

4.4.1 PMC500M으로 학습된 모델 간 비교(BERT, oiBioBERT, oiSciBERT, exBERT)

<Table 1>은 PMC500M을 활용하여 학습한 방법들의 downstream 태스크에 대한 결과이다. 여기서 Params는 모델의 파라미터 개수를 의미한다. 또한 Teacher 모델들의 Time Cost는 도메인 적응 과정에서 걸린 시간을, student 모델들의 Time Cost는 teacher의 도메인 적응 과정에서 걸린 시간에 지식 증류 과정에서 소요된 시간이 누적된 값을 의미한다. 즉, student 모델의 Time Cost는 타겟 도메인에 대해 작은 모델을 만드는 과정에 소요되는 총 시간을 의미한다. 결과를 보면, 먼저 BERT는 도메인 적응 과정이 없기 때문에 다른 방법들에 비해 성능이 낮고 마찬가지로 $BERT_{GD}$의 성능도 낮은 것을 볼 수 있다. 의학 도메인에 특화된 vocabulary를 가진 oiSciBERT와 exBERT는 일반적인 도메인에 대한 vocabulary만을 가진 oiBioBERT보다 조금 더 높은 성능을 기록하였다. 또한 대량의 데이터로 사전 학습된 BERT를 그대로 활용하면서 의학 도메인의 전문적인 단어들을 고려할 수 있는 exBERT가 oiSciBERT 보다 높은 성능을 나타낸 것을 미루어 보아, exBERT 방식이 적은 양의 타겟 데이터로 효율적인 도메인 적응이 가능하다는 것을 알 수 있다. Teacher 모델의 순위와 같이 student 모델 역시 $exBERT_{GD}$ > $oiSciBERT_{GD}$ > $oiBioBERT_{GD}$순으로 성능이 높았다. Teacher 모델의 성능이 높을수록 student의 성능도 높은 추세를 보였고, 의학 도메인에 특화된 단어들을 고려할 수 있는 exBERT와 oiSciBERT가 oiBioBERT보다 높은 성능을 보였으며 마지막으로 동일한 데이터셋, 동일한 epoch 만큼 학습했음에도 불구하고 exBERT가 oiSciBERT보다 높은 성능을 보였다. 이를 통해 타겟 도메인의 데이터로 모델을 처음부터 학습하는 것보다 일반적인 도메인의 대량 텍스트로 사전학습된 BERT를 활용하면서 vocabulary를 확장하는 것이 가장 효율적으로 타겟 도메인에 대한 teacher와 student 모델을 만들 수 있음을 확인했다.

4.4.2 exBERT와 EDAD의 비교

<Table 2>는 EDAD와 $exBERT_{GD}$의 비교 결과이다. 결과를 보면 EDAD 는 타겟 도메인에 대해 작은 모델을 만들 때 거치는 두 단계의 과정을 하나로 결합함으로써 teacher 모델과 student 모델이 각각 exBERT, $exBERT_{GD}$와 동일함에도 불구하고 약 4시간 정도 빠르게 student 모델을 만들 수 있음을 알 수 있다. 뿐만 아니라, 더 높은 성능을 내는 student 모델을 만들어 낼 수 있었다. 다시 말해, 도메인 적응과 지식 증류 과정을 각각 3 epoch씩 총 6 epoch의 학습과정을 거쳐 만들어지는 $exBERT_{GD}$에 비해 EDAD는 오로지 3 epoch 동안만 학습했기 때문에 시간

Table 1. Comparison Results of BERT, oiBioBERT, oiSciBERT, and exBERT

Model	Params	Time Cost (H)	BC5CDR	BioNLP09	NCBI-disease	Average
Teacher Model						
BERT	110M	-	$85.1754_{\pm 0.3096}$	$89.3804_{\pm 0.0767}$	$91.0645_{\pm 0.1287}$	$88.5401_{\pm 0.1717}$
oiBioBERT	110M	$9.7_{\pm 0.2}$	$85.5299_{\pm 0.3719}$	$90.0898_{\pm 0.1004}$	$91.1761_{\pm 0.2021}$	$88.9319_{\pm 0.2248}$
oiSciBERT	111M	$10.6_{\pm 0.1}$	$85.9016_{\pm 0.0397}$	$90.3897_{\pm 0.0695}$	$91.6535_{\pm 0.1086}$	$89.3150_{\pm 0.0726}$
exBERT	152M	$14.1_{\pm 0.2}$	**$86.1297_{\pm 0.1587}$**	**$91.4077_{\pm 0.0884}$**	**$91.8331_{\pm 0.1910}$**	**$89.7902_{\pm 0.1460}$**
Student Model						
$BERT_{GD}$	15M	$4.6_{\pm 0.0}$	$83.2744_{\pm 0.0968}$	$87.5442_{\pm 0.0750}$	$90.2027_{\pm 0.1293}$	$87.0071_{\pm 0.1004}$
$oiBioBERT_{GD}$	15M	$14.3_{\pm 0.2}$	$83.5326_{\pm 0.0946}$	$88.4730_{\pm 0.0570}$	$90.4914_{\pm 0.0559}$	$87.4990_{\pm 0.0692}$
$oiSciBERT_{GD}$	15M	$15.5_{\pm 0.2}$	$83.7249_{\pm 0.1055}$	$89.1764_{\pm 0.0656}$	$90.2115_{\pm 0.0979}$	$87.7043_{\pm 0.0897}$
$exBERT_{GD}$	22M	$21.6_{\pm 0.2}$	**$83.7393_{\pm 0.1623}$**	**$90.0453_{\pm 0.1308}$**	**$90.5952_{\pm 0.0669}$**	**$88.1266_{\pm 0.1200}$**

Table 2. Comparison with $exBERT_{GD}$ and EDAD

Model	Params	Time Cost	BC5CDR	BioNLP09	NCBI-disease	Average
Student Model						
$exBERT_{GD}$	22M	$21.6_{\pm 0.2}$	$83.7393_{\pm 0.1623}$	$90.0453_{\pm 0.1308}$	$90.5952_{\pm 0.0669}$	$88.1266_{\pm 0.1200}$
EDAD	22M	$17.4_{\pm 0.2}$	**$84.1471_{\pm 0.0592}$**	**$90.2555_{\pm 0.0595}$**	**$90.6800_{\pm 0.1383}$**	**$88.3609_{\pm 0.0857}$**

적인 비용이 줄어들었고, 동시에 α로 도메인 적응과 지식 증류 loss의 반영 비율을 조절함에 따라 더 높은 성능까지 얻을 수 있음을 보였다. 이 때 epoch이 절반으로 줄었음에도 학습시간이 절반이 되지 않은 것은 EDAD 과정에는 teacher 모델의 도메인 적응 loss와 teacher 모델과 student 모델 간의 지식증류 loss 뿐만 아니라 student 모델의 도메인 적응 loss가 학습과정에 추가되므로 상대적으로 연산량이 많기 때문이다.

4.4.3 대규모의 타겟 도메인 데이터로 학습된 SciBERT, BioBERT와 EDAD의 비교

<Table 3>은 대량의 의학 도메인 데이터를 활용한 도메인 적응 방식들과 제안하는 EDAD 간의 비교 결과이다. BioBERT와 SciBERT에 대한 Time Cost는 Tai *et al.*(2020)를 참조하여 4개의 V100 gpu를 사용했을 때의 도메인 적응에 걸리는 시간을 기록했다. SciBERT는 3.17B, BioBERT는 4.5B개의 토큰으로 이루어진 대량의 텍스트 데이터로 사전학습이 진행되었기 때문에 downstream 태스크에 대해 가장 높은 성능들을 보였다. 4.4.1절에서와 같이 성능이 가장 높은 SciBERT의 student 모델인 SciBERT_{GD}가 가장 높은 성능을 보이는 것을 알 수 있다. 하지만 BioBERT_{GD}는 높은 teacher의 성능에도 불구하고 낮은 F1-score를 나타냈는데, 본 연구에서 지식 증류 단계에 사용한 데이터셋에 cased token의 수가 적기 때문에 유일하게 cased vocabulary를 사용하는 BioBERT의 student 모델이 제대로 학습되지 않았다고 판단된다. 실제로 사용한 데이터셋을 cased vocabulary와 uncased vocabulary로 tokenize해본 결과, cased vocabulary를 사용하여 tokenize했을 때의 unknown token의 수가 uncased vocabulary를 사용하여 tokenize했을 때보다 약 28% 더 많이 존재했다. 한편, SciBERT_{GD}는 student 모델 중에서 가장 높은 성능을 보이지만 Tai *et al.*(2020)에서 SciBERT를 4개의 V100 gpu로 학습할 때의 소요되는 시간으로 환산한 결과를 참조해보면 SciBERT_{GD} 모델을 만들기 위해 걸리는 시간이 600시간을 훨씬 넘는 것을 알 수 있다. 반면 EDAD는 약 39배 적은 시간으로 SciBERT_{GD}와 비교할 만한 성능을 보였다.

4.4.4 α에 대한 비교 실험

우리는 EDAD가 다양한 도메인 적응 방식들과 비교했을 때 가장 효율적으로 의학 도메인에 대한 작은 모델을 만들 수 있음을 앞서 실험을 통해 보였다. 본 장에서는 EDAD에서 도메인 적응과 지식 증류의 반영 정도를 조절하는 α에 대해 1) α의 스케줄링에 따른 비교와 2) loss 반영 순서에 따른 비교를 진행하였다. PMC500M 데이터셋으로 사전학습하는 과정에서 α를 조절해가며 student 모델을 학습하고, 학습된 student 모델을 downstream 데이터셋에 대해 fine-tuning하여 성능을 비교했다. 이에 대한 결과는 <Table 4>와 같다. 먼저 1)에 대해 분석해보면, 오로지 student 모델의 도메인 적응만 진행하였을 때(<Table 4>의 “Only DA”)보다는 teacher 모델과의 지식 증류 과정이 포함된 나머지 모든 경우들이 좋은 성능을 보였다. 이는 지식 증류가 상대적으로 작은 student 모델을 만드는 데에 있어 중요하다는 것을 의미한다. 또한 오로지 지식 증류를 통해 student 모델을 학습하는 경우(<Table 4>의 “Only DT”)나 동

Table 3. EDAD vs. Pretrained Model of Using a Large Dataset for Domain Adaptation

Model	Params	Time Cost	BC5CDR	BioNLP09	NCBI-disease	Average
Teacher Model						
BioBERT	109M	1104h	**88.5435**$_{\pm 0.2128}$	91.7610$_{\pm 0.2000}$	92.9517$_{\pm 0.1213}$	91.0854$_{\pm 0.1781}$
SciBERT	111M	672h	88.3215$_{\pm 0.1473}$	**92.0259**$_{\pm 0.1633}$	**93.0337**$_{\pm 0.1460}$	**91.1270**$_{\pm 0.1522}$
Student Model						
BioBERT_{GD}	14M	1108.1$_{\pm 0.0}$	83.9479$_{\pm 0.0863}$	88.6230$_{\pm 0.0664}$	90.0468$_{\pm 0.0849}$	87.5392$_{\pm 0.0792}$
SciBERT_{GD}	15M	676.3$_{\pm 0.1}$	**84.5567**$_{\pm 0.0980}$	89.7536$_{\pm 0.0468}$	**90.8911**$_{\pm 0.0718}$	**88.4005**$_{\pm 0.0722}$
EDAD	22M	17.4$_{\pm 0.2}$	84.1471$_{\pm 0.0592}$	**90.2555**$_{\pm 0.0595}$	90.6800$_{\pm 0.1383}$	88.3609$_{\pm 0.0857}$

Table 4. Results for the Effect of Scheduling of α

Model	BC5CDR	BioNLP09	NCBI-disease	Average
Only DA(α=1)	82.2035$_{\pm 0.1044}$	89.3904$_{\pm 0.0585}$	89.5085$_{\pm 0.0473}$	87.0341$_{\pm 0.0701}$
Only DT(α=2)	**84.1738**$_{\pm 0.1930}$	89.8971$_{\pm 0.1026}$	89.9760$_{\pm 0.0888}$	88.0156$_{\pm 0.1281}$
Same ratio(α=0.5)	84.0446$_{\pm 0.1031}$	89.8504$_{\pm 0.1429}$	90.0598$_{\pm 0.0177}$	87.9849$_{\pm 0.0879}$
Linear	84.1471$_{\pm 0.0592}$	**90.2555**$_{\pm 0.0595}$	**90.6800**$_{\pm 0.1383}$	**88.3609**$_{\pm 0.0857}$
Linear reverse	84.0738$_{\pm 0.1709}$	89.9140$_{\pm 0.0578}$	90.2483$_{\pm 0.1629}$	88.0787$_{\pm 0.1305}$
Cosine	84.1412$_{\pm 0.0805}$	89.8814$_{\pm 0.0509}$	90.3499$_{\pm 0.0942}$	88.1242$_{\pm 0.0752}$
Cosine reverse	83.7576$_{\pm 0.0840}$	89.8400$_{\pm 0.0616}$	90.3437$_{\pm 0.1039}$	87.9804$_{\pm 0.0832}$

일한 비율로 도메인 적응에 대한 loss와 지식 증류에 대한 loss를 반영하여 학습하는 것(<Table 4>의 “Same ratio”)에 비해 α를 스케줄링에 따라 조절하였을 때가 비슷하거나 더 높은 성능을 보였다. 한편, α의 스케줄링 방법들을 비교하면 선형적으로 감소시켰을 때(<Table 4>의 “Linear”)가 cosine annealing 스케줄에 따라 감소시키는 것(<Table 4>의 “Cosine”)보다 높은 성능을 기록했다. 이는 cosine annealing 스케줄처럼 급격히 두 요소의 반영정도를 바꾸는 것보다 선형적으로 바꾸는 것이 본 연구에서 고려한 데이터셋에 더 효과적인 것으로 판단된다. 2)에 대한 결과를 보면, 학습 초기에는 지식 증류에 더 큰 비중을 두고 후반에는 도메인 적응에 더 큰 비중을 두는 것이 결과적으로 가장 높은 성능을 보였다. 이는 초반에 teacher를 통해 빠르게 지식을 전이 받고 후반에는 student가 타겟 도메인에 대한 정보를 학습하는 것이 효과적이라는 것을 의미한다. 이러한 비교 실험 결과를 바탕으로 EDAD는 가장 높은 성능을 보인 linear 스케줄링 방식을 채택한다.

5. 결 론

본 연구는 자연어처리 분야에서 적은 데이터 및 컴퓨팅 자원을 가진 환경에서 타겟 도메인에 대해 작은 모델을 필요로 할 때 효율적으로 모델을 만들 수 있는 방법인 EDAD를 제안한다. 효율적으로 타겟 도메인에 대한 작은 모델을 만들기 위해 기존의 도메인 적응 방법인 exBERT를 활용하여 EDAD 프레임워크를 구성한다. 일반적으로 타겟 도메인에 대한 작은 모델을 학습하기 위해서는 도메인 적응과 지식 증류가 독립적으로 이루어져야 하지만 제안한 EDAD는 두 과정을 통합하여 학습 시간을 단축했다. 뿐만 아니라 학습과정 동안 도메인 적응과 지식 증류의 반영 비율을 조절하는 파라미터 α를 도입하여 학습 초반에는 teacher 모델의 지식을 student 모델에게 전이하는 것에 집중하고 학습 후반에는 student 모델이 타겟 도메인에 적응하는 것에 집중함으로써 타겟 도메인의 downstream 태스크에서 더 나은 일반화 성능을 보였다. EDAD의 효과를 확인하고자 의학 도메인의 주요 태스크들 중 하나인 NER에 해당되는 데이터셋을 사용하여 검증하였다. 향후에는 의학 도메인 뿐만 아니라 금융, 컴퓨터 과학 등 다양한 도메인에서 검증해 볼 필요가 있고, 도메인 적응과 지식 증류 loss의 반영 비율을 조절하는 최적의 스케줄링 방법의 탐색, vocabulary와 모델의 확장으로 인해 필연적으로 늘어나는 student 모델의 효율적 압축 등의 연구가 필요하다.

참고문헌

Araci, D. (2019), FinBERT: Financial sentiment analysis with pre-trained language models, arXiv preprint arXiv:1908.10063.

Beltagy, I., Lo, K., and Cohan, A. (2019), SciBERT: A Pretrained Language Model for Scientific. Text. *EMNLP-IJCNLP*, 3615-3620.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., … and Fiedel, N. (2022), PaLM: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019), BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. Conference North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 4171-4186.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021), Domain-specific language model pretraining for biomedical natural language processing, *ACM Transactions on Computing for Healthcare*, **3**(1), 1-23.

Hinton, G., Vinyals, O., and Dean, J. (2015), Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hong, J., Kim, T., Lim, H., and Choo, J. (2021), AVocaDo: Strategy for adapting vocabulary to downstream domain, In *Proceedings of EMNLP’21*, 4692-4700.

Huggingface, https://huggingface.co/.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. (2020), TinyBERT: Distilling BERT for natural language understanding, In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 4163-4174, Online. Association for Computational Linguistics.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019), BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining, *Bioinformatics*, **36**(4), 1234-1240.

MTL-Bioinformatics-2016, https://github.com/cambridgeltl/MTL-Bioinformatics-2016.

PMC, https://www.ncbi.nlm.nih.gov/pmc/tools/ftp/.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018), Improving language understanding with unsupervised learning, *Technical report*, OpenAI.

Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., … and Catanzaro, B. (2022), Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. arXiv preprint arXiv:2201.11990.

Tai, W., Kung, H. T., Dong, X., Comiter, M., and Kuo, C. (2020), exbert: Extending pretrained models with domain-specific vocabulary under constrained training resources. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 1433-1439.

Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., and Zhou, M. (2020), Minilm: Deep. selfattention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, NeurIPS 2020, Virtual.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., … and Dean, J. (2016), Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Yao, Y., Huang, S., Wang, W., Dong, L., and Wei, F. (2021), *Adapt-and-Distill: Developing Small, Fast and Effective Pretrained Language Models for Domains*, Findings of the Association for Computational Linguistics: ACL-IJCNLP.

저자소개

서승원 : 서울과학기술대학교 산업공학과에서 2022년 학사학위를 취득하고 서울과학기술대학교에서 데이터사이언스학과 학석사 연계과정에 재학 중이다. 연구 분야는 딥러닝 방법론 개발 및 응용이다.

황상흠 : KAIST 산업 및 시스템공학과에서 2005년 학사, 2012년 박사학위를 취득하였다. 삼성전자 종합기술원과 루닛에서 연구원으로 재직했고 2018년부터 서울과학기술대학교 산업공학과에 재직 중이다. 주요 연구 분야는 기계학습/딥러닝 방법론 개발 및 응용 등이다.