한국빅데이터학회지
제7권 제2호, 2022, pp. 11-29 https://doi.org/10.36498/kbigdt.2022.7.2.11

딥러닝 기반 사전학습 언어모델에 대한 이해와 현황

A Survey on Deep Learning-based Pre-Trained Language Models

박상언†

경기대학교 소프트웨어경영대학 ICT융합학부 경영정보전공

요 약

사전학습 언어모델은 자연어 처리 작업에서 가장 중요하고 많이 활용되는 도구로, 대량의 말뭉치를 대상으로 사전학습이 되어있어 적은 수의 데이터를 이용한 미세조정학습으로도 높은 성능을 기대할 수 있으며, 사전학습된 토크나이저과 딥러닝 모형 등 구현에 필요한 요소들이 함께 배포되기 때문에 자연어 처리 작업에 소요되는 비용과 기간을 크게 단축시켰다. 트랜스포머 변형 모형은 이와 같은 장점을 제공하는 사전학습 언어모델 중에서 최근 가장 많이 사용되고 있는 모형으로, 번역을 비롯하여 문서 요약, 챗봇과 같은 질의 응답, 자연스러운 문장의 생성 및 문서의 분류 등 다양한 자연어 처리 작업에 활용되고 있으며 컴퓨터 비전 분야와 오디오 관련 분야 등 다른 분야에서도 활발하게 활용되고 있다. 본 논문은 연구자들이 보다 쉽게 사전학습 언어모델에 대해 이해하고 자연어 처리 작업에 활용할 수 있도록 하기 위해, 언어모델과 사전학습 언어모델의 정의로부터 시작하여 사전학습 언어모델의 발전과정과 다양한 트랜스포머 변형 모형에 대해 조사하고 정리하였다.

■ 중심어 : 자연어 처리, 딥러닝, 언어모델, 트랜스포머, BERT, GPT

Abstract

Pre-trained language models are the most important and widely used tools in natural language processing tasks. Since those have been pre-trained for a large amount of corpus, high performance can be expected even with fine-tuning learning using a small number of data. Since the elements necessary for implementation, such as a pre-trained tokenizer and a deep learning model including pre-trained weights, are distributed together, the cost and period of natural language processing has been greatly reduced. Transformer variants are the most representative pre-trained language models that provide these advantages. Those are being actively used in other fields such as computer vision and audio applications. In order to make it easier for researchers to understand the pre-trained language model and apply it to natural language processing tasks, this paper describes the definition of the language model and the pre-learning language model, and discusses the development process of the pre-trained language model and especially representative Transformer variants.

■ Keyword : NLP, deep learning, language model, Transformer, BERT, GPT

2022년 11월 14일 접수; 2022년 12월 04일 수정본 접수; 2022년 12월 05일 게재 확정.
* 이 논문은 2021학년도 경기대학교 연구년 수혜로 연구되었음.
† 교신저자 (supark@kgu.ac.kr)

Ⅰ. 서 론

현재 자연어 처리 대부분의 작업에서 가장 많이 사용되고 있는 모형은 트랜스포머의 변형 모형이다[1]. 이러한 변형 모형들은 트랜스포머 구성 요소의 일부를 사용하는 다양한 인공신경망 모형과 사전학습 방법을 새롭게 제안했는데, 각 모형이 발표될 때마다 그때까지의 기존 벤치마크 성능을 갱신하는 놀라운 모습을 보여줬다. 이러한 발전은 2018년 GPT(Generative Pre-trained Transformer)와 BERT(Bidirectional Encoder Representations from Transformers)가 발표된 이후부터 급격하게 이루어졌으며, 번역을 비롯하여 문서 요약, 챗봇과 같은 질의 응답, 자연스러운 문장의 생성 및 문서의 분류 등 다양한 자연어 처리 작업에 활용되면서 놀라울 정도의 빠른 자연어 처리 분야의 성장에 기여하고 있다.

자연어 처리뿐만 아니라 컴퓨터 비전 분야에서도, 트랜스포머 모형을 사용함으로써 이미지 인식 성능의 최고 성능을 기록한 비전 트랜스포머[2] 그리고 GPT-2를 이미지 사전학습에 이용한 연구[3] 등과 같이 사전학습 언어모델에 대한 연구가 활발하게 진행되고 있다. 오디오 관련 분야에서도 음성 분리를 위해 트랜스포머 모형을 이용한 SepFormer[4], 더 나아가 비디오, 오디오, 텍스트 등 복합적인 다중모드(multimodal) 데이터를 다루기 위한 트랜스포머 모형을 제안한 VATT[5]와 같은 연구들이 있다.

이러한 트랜스포머 변형 모형의 발전과 다양한 활용 뒤에는 사전학습 언어모델의 전이학습이라는 공통적인 요소가 있다. 따라서 최근 자연어 처리의 발전 현황을 이해하기 위해서는 언어모델과 언어모델에 대한 사전학습 그리고 전이학습 및 트랜스포머 모형에 대해 필수적으로 이해할 필요가 있다.

이를 위해 본 논문에서는 언어모델의 정의로부터 시작하여, 사전학습 언어모델의 의미와 전이학습을 이용한 사전학습 언어모델의 활용에 대해 정리하고자 한다. 또한, 사전학습 언어모델의 발전 과정에 대해 살펴보고 발전의 마지막에 있는 트랜스포머 모형에 대해 간략하게 기술한다. 그 후 BERT, GPT, BART와 같은 다양한 트랜스포머 변형 모형의 현황에 대해 모형 별로 상세하게 정리한다. 다음으로 다국어 사전학습 언어모델의 한계를 극복하기 위한 한국어 트랜스포머 변형 모형의 개발 현황에 대해 정리하고자 한다.

Ⅱ. 사전학습 언어모델의 발전과 트랜스포머

이 장에서 기본적인 언어모델에 대해 먼저 기술하고 어떻게 자연어 처리의 다양한 분야로 언어모델이 확장되어 사용될 수 있는지와 사전학습 언어모델의 원리에 대해 정리한다. 그리고 사전학습 언어모델의 발전 과정에 대해 살펴보고 나서 트랜스포머 모형에 대해 간략히 기술하고자 한다.

2.1 언어모델과 사전학습

2.1.1 언어모델의 정의

언어모델(language model)은 단어의 시퀀스에 대해 확률을 할당하는 모델을 말하며, 이 확률은 식 (1)과 같이 계산할 수 있다[6]. 식 (1)에서 $w_1, w_2, ..., w_N$이 단어의 시퀀스라고 가정하면, 이 시퀀스가 나타날 확률은 각 단어들의 결합확률로 표현되며 식에서와 같이 조건부 확률의 곱으로 계산된다. 이 때 $p(w_i \mid w_1, w_2, ..., w_{i-1})$은 언어모델에 따라 다양한 방법으로 계산될 수 있다. 언어모델링(language modeling)은 이러한 언어모델을 학습하고 사용하는 프로세스로 정의된다[6]. 이와 같은 방식을 확장하면 $p(w_{n-k}, ..., w_n \mid w_1, ..., w_{n-k-1})$와 같이 정의되는 확률을 추정하는 언어모델도 구현할 수 있으며, 이는 주어진 입력 단어 시퀀스에 대해 대

상으로 하는 출력 단어 시퀀스의 확률을 추정하는 것으로 하나의 단어가 아닌 단어 시퀀스 즉 문장에 대해 확률을 할당하는 언어모델이 된다. 이를 기반으로 하면 번역, 문서 요약, 질의 응답과 같은 복잡한 자연어 관련 작업을 수행할 수 있다[7].

$$p(w_1, w_2, ..., w_N) = \prod_{i=1}^{N} p(w_i \mid w_1, w_2, ..., w_{i-1}) \quad (1)$$

단어의 시퀀스에 확률을 할당한다는 것은 주어진 문장이 해당 언어 관점에서 얼마나 자연스러운 문장인지를 판단한다고 할 수 있다. 예를 들어 “나는 학교에 간다”와 “나는 학교에 먹었다” 중 전자가 더 자연스러운 문장으로, 한국어 언어모델에서 후자에 비해 더 높은 확률을 갖게 된다. 이 때 ‘더 자연스러운’은 ‘일반적으로 더 많이 사용되는’으로 이해할 수 있으며 이와 같은 확률을 계산하기 위해서는 수많은 한국어 문장들을 학습함으로써 더 많이 사용되는 표현에 더 높은 확률을 할당할 수 있게 학습해야 한다. 결과적으로 언어모델은 해당 언어에 대한 이해를 높이는 학습이라고 할 수 있다.

이때 식 (1)에서 설명한 조건부 확률을 자연어처리 작업 관점에서 보면 $p(output \mid input)$의 형태로 볼 수 있다[7]. 여기서 주어진 자연어처리 작업의 input은 어떤 단어의 시퀀스이고, output은 주어진 문제에 따라 하나의 값이나 단어가 되거나 혹은 다른 단어의 시퀀스일 수 있다. 만일 주어진 문서를 정해진 클래스로 분류하는 문제라면 output은 클래스가 될 것이고 번역이나 요약 문제라면 output은 번역할 문장이거나 요약된 문장이 될 것이다. 즉 언어모델을 학습한다는 것은 문서 분류, 번역, 문서 요약, 질의 응답과 같은 다양한 자연어 처리 문제를 해결할 수 있는 모델로 자연스럽게 확장할 수 있다. 트랜스포머(Transformers), GPT, BERT와 같은 사전학습 언어모델은 이와 같은 방식으로 지난 몇 년간 눈부신 속도로 발전해 왔다.

2.1.2 사전학습 언어모델

사전학습 언어모델은 미리 학습된 언어모델이라는 의미이며, 일반적으로 언어모델의 학습은 비지도 학습으로 이루어진다. 지도학습은 주어진 입력에 대해 예측해야 할 출력을 명시적으로 할당하는 방식으로 학습하는 반면, 언어모델에서는 자연어 문장을 이용하여 식 (1)에서와 같이 주어진 단어 시퀀스에 대해 다음 단어를 예측하도록 데이터셋을 자동으로 생성함으로써 학습을 수행한다. 이와 같은 언어모델에서의 학습방법은 트랜스포머와 GPT, BERT 등을 거치면서 언어를 보다 더 효과적으로 학습할 수 있도록 개선되었다.

이와 같은 사전학습은 문서 분류, 번역 등과 같은 목표 작업에 대해 직접적으로 학습하지 않기 때문에 바로 적용하기에는 한계가 있다. 그럼에도 불구하고 사전학습이 의미를 갖는 이유는 2.1.1에서 설명한 바와 같이 언어모델의 기본적인 형태가 다양한 자연어 처리 문제로 쉽게 확장될 수 있는 구조이기 때문이다. 즉 단어의 시퀀스로부터 어떤 출력을 생성한다는 점에서 공통점을 갖고 있다고 할 수 있다. 풀어서 설명하자면 언어에 대한 이해가 높다면 다양한 자연어 처리 작업에 보다 쉽게 적응할 수 있다는 장점을 갖게 된다.

사전학습 언어모델의 장점을 다음과 같은 두 가지 관점에서 설명될 수 있다[6]. 첫째 사전학습 언어모델은 다양한 자연어 처리 작업의 정확도를 크게 높일 수 있다. 실제로 BERT를 기반으로 미세조정을 수행한 모델이 자연어 처리에서 인간보다도 더 높은 정확도를 보였으며[8], GPT-3은 자연어 생성에서 놀라운 수준의 유창함을 보였다[9]. 둘째 사전학습 언어모델은 머신러닝 기반의 자연어처리 모형 학습의 부담을 크게 줄였다. 전통적으로 자연어처리 모형을 학습하기 위해서는

매우 많은 수의 라벨이 있는 데이터를 생성해야 했으나 사전학습 언어모델을 사용하면 적은 수의 데이터로도 효과적인 학습을 하는 것이 가능해졌다.

이상의 두 가지 장점은 모두 사전학습 언어모델을 이용한 전이학습(transfer learning)이 가능하다는 점에서 발생한다. 전이학습[10]은 다른 분야에서 학습한 결과를 재사용함으로써 학습의 속도와 성능을 모두 향상시킬 수 있는 방법을 말한다. 다만 이 때 두 분야의 지식과 모형의 유사성이 높아야 학습된 지식의 전이(transfer)가 큰 효과를 낼 수 있다. 사전학습 언어모델은 2.1.1에서 설명한 바와 같이 언어의 이해라는 점에서 지식의 유사성이 높고 단어의 시퀀스를 입력으로 하여 주어진 출력의 확률을 계산한다는 점에서 모형의 유사성도 매우 높다. 사전학습 언어모델을 활용할 때에는 일반적으로 학습된 모형을 가져와서 주어진 문제에 대해 미세조정(fine tuning)으로 지도학습을 수행함으로써 대상 작업에 맞는 성능을 갖추게 된다.

2.2 사전학습 언어모델의 발전

사전학습 언어모델은 인공신경망에 기반한 언어모델이 등장하면서 급격히 발전하게 된다. 인공신경망을 이용한 자연어 처리를 위해 워드 임베딩이 사용되고, Word2Vec에 의해 범용적인 워드 임베딩 벡터의 전이학습이 다양한 분야에 활용되기 시작했다. ELMo(Embeddings from Language Model)는 워드 임베딩뿐만이 아니라 학습된 양방향 LSTM 모형 자체를 전이학습에 사용하게 되었으며, 이는 셀프 어텐션 기반의 트랜스포머 모형으로 이어진다. 이 절에서는 인공신경망 언어모델과 Word2Vec, ELMo에 대해 기술하고 다음 절에서 트랜스포머에 대해 설명한다.

2.2.1 인공신경망 언어모델의 등장

Bengio 등은 2000년에 최초의 인공신경망을 이용한 언어모델을 발표했다[11]. 이전의 언어모델은 n-gram에 기반한 모델이었으며 다음 단어가 n-1 개의 이전 단어에 의해 결정된다는 가정 하에 마코프체인과 같은 다양한 방법을 사용하여 확률을 계산하였다. 그러나 n-gram 언어모델은 학습해야 할 매개변수의 수가 n의 크기에 따라 기하급수적으로 증가한다는 점과 n이 커질수록 n-gram의 희소성이 매우 커지기 때문에 학습이 제대로 이루어지지 않는다는 문제를 갖고 있었다. Bengio 등이 제안한 모델은 인공신경망을 사용함으로써 이상과 같은 문제점을 해결했는데 이를 언어모델 관점에서 보면 식 (2)와 같이 인공신경망을 통해 조건부 확률을 추정하게 된다.

$$p(w_i \mid w_{i-n+1}, w_{i-n-2}, ..., w_{i-1}) = f_\theta(v_{i-n+1}, v_{i-n-2}, ..., v_{i-1}) \quad (2)$$

식 (2)에서 $v_{i-n+1}, v_{i-n-2}, ..., v_{i-1}$은 원래 단어인 $w_{i-n+1}, w_{i-n-2}, ..., w_{i-1}$의 워드 임베딩 벡터이며 $f(\bullet)$는 인공신경망을, 그리고 θ는 인공신경망의 가중치 즉 파라미터를 나타낸다. 인공신경망 기반 언어모델로 인한 개선 내용은 다음 두 가지로 요약될 수 있다[6]. 첫째 워드 임베딩을 통해 단어를 실수 벡터로 표현했다는 점이다. 워드 임베딩은 원핫인코딩에 비해 단어를 보다 효율적으로 표현할 수 있고, 일반화 성능, 견고성, 확장성에서 더 뛰어나다는 장점이 있다. 둘째 인공신경망 언어모델은 필요한 매개변수의 수를 크게 감소시켰다. 그 결과, 보다 효과적이고 효율적인 학습이 가능해졌다. 이 연구 이후에 워드 임베딩 기반의 인공신경망을 이용한 언어모델에 대한 연구가 크게 증가했으며 이로 인해 자연어 처리 분야에 딥러닝에 기반한 새로운 패러다임을 가져오게 되었다.

2.2.2 Word2Vec

Word2Vec[12]은 가장 널리 알려진 사전학습 워드 임베딩 기법이라고 할 수 있다. Word2Vec이 기존의 언어모델과 차별화되는 점은 단어에

의미적인 정보를 함축함으로써 단어 간의 유사도를 계산하거나 연산을 수행하고, 더 나아가서 학습된 결과를 다른 작업에서도 사용할 수 있는 전이학습이 가능하도록 한 것이다.

기존 언어모델이 앞의 단어 시퀀스를 이용해 다음 단어에 대한 확률을 계산한 것과 달리, Word2Vec은 주변의 단어를 이용해 중심에 있는 단어를 예측(CBOW, Continuous Bag of Words)하거나 반대로 중심의 단어를 이용해 주변 단어들을 예측(Skip-gram)하도록 학습함으로써 앞에 있는 단어들뿐 아니라 뒤에 있는 단어들로 인해 만들어지는 문맥까지 학습하도록 했다.

Word2Vec은 단어의 확률을 예측하는 언어모델 본래의 목적을 위한 모형이라기보다는 이를 이용하여 단어를 의미적으로 임베딩하는 것에 더 초점을 맞췄다고 볼 수 있다. 이와 같은 워드 임베딩 결과는 문서 분류와 같은 자연어 처리 작업의 워드 임베딩 벡터로 재사용함으로써 작업의 학습 속도와 정확도를 개선하기 위해 활용되었다. Word2Vec의 워드 임베딩 벡터는 대용량의 텍스트 데이터를 이용해 학습이 되었기 때문에, 이를 자연어 처리 작업을 위한 인공신경망의 초기 워드 임베딩 벡터로 사용하면 학습의 속도를 빠르게 할 뿐 아니라 작업의 성능도 높일 수 있었다. Word2Vec으로 인해 사전학습 언어모델의 전이학습이 가속화되었다고 할 수 있다.

2.2.3 ELMo

동음이의어는 동일한 형태의 단어가 글에서 다른 의미로 쓰이는 경우를 말하는데, Word2Vec의 가장 큰 한계점 중 하나로 지적된 점이 바로 이러한 동음이의어들이 Word2Vec에서 동일한 벡터로 임베딩된다는 것이었다. Word2Vec이 단어의 의미를 함축적으로 표현하고자 했다는 점에서 이것은 심각한 문제가 될 수 있다. ELMo [13]는 두 개의 레이어로 구성된 양방향 LSTM (Long Short-Term Memory)으로 언어모델을 구축하고 정방향과 역방향을 모두 학습시켰다. 즉, 정방향 LSTM에서는 앞에 있는 단어의 시퀀스로 다음 단어를 예측하도록 학습하고, 역방향 LSTM에서는 뒤에 있는 단어의 시퀀스로 앞 단어를 예측하도록 학습함으로써 문맥을 완전히 학습하고자 했다. Word2Vec의 워드 임베딩이 매우 직관적인 것에 비해 ELMo에서는 초기의 워드 임베딩 벡터, 첫 LSTM 레이어의 정방향과 역방향 출력 임베딩 벡터, 그리고 둘째 LSTM 레이어의 출력을 모두 결합하여 최종 임베딩 벡터를 생성했다. Word2Vec과의 가장 큰 차이점은, Word2Vec은 워드 임베딩 벡터가 학습 이후에 고정되는 것에 비해 ELMo는 학습된 인공신경망 모형에 의해 주어진 문장에 대해 가변적으로 임베딩 벡터를 생성한다는 점이다. 따라서 동일한 단어라 하더라도 문장에서의 위치에 따라 다른 벡터로 임베딩될 수 있다. Word2Vec에서는 사전학습으로 생성된 워드 임베딩 벡터를 재사용했으나, ELMo에서는 학습된 모형 자체, 즉 모형의 구조와 가중치 전체를 재사용한다.

2.3 트랜스포머 모형

트랜스포머[14]는 현재 사전학습 언어모델에서 가장 중요한 딥러닝 모형이며 자연어 처리뿐만 아니라 컴퓨터 비전, 음성 처리와 같은 다양한 분야에서 활발하게 사용되고 있다. 원래는 번역을 위해 개발된 시퀀스-투-시퀀스(Sequence-to-Sequence) 모형이었으나 GPT, BERT와 같은 다양한 트랜스포머 변형 모형(X-formers)이 개발되면서 자연어 처리의 거의 모든 분야에서 SOTA(State-of-the-art) 즉 최고 성능을 기록하는 모형이 되었다. 최근에는 컴퓨터 비전에 활용되면서 비전과 관련한 작업에서도 최고 성능을 갱신하고 있다.

트랜스포머 변형 모형은 다음과 같은 측면에서 기본 트랜스포머 모형을 개선하고 있다[1]. 첫

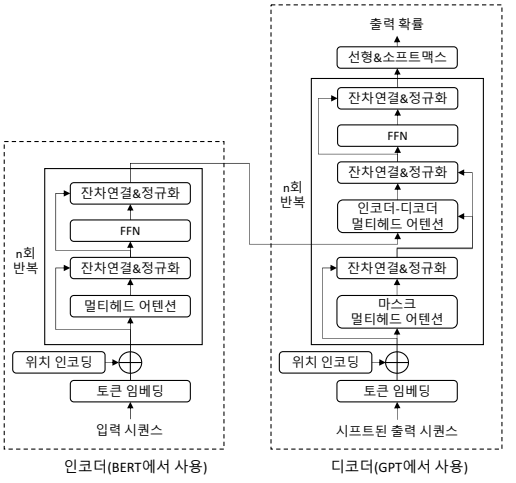
째 모형의 성능이다. 트랜스포머는 셀프 어텐션(self-attention) 모듈의 높은 연산과 메모리 복잡도로 인해 긴 시퀀스를 처리하는 데 있어 비효율적으로 알려져 있다. 이를 개선하기 위해 트랜스포머 변형 모형들은 보다 가벼운 어텐션 모듈을 설계하거나 분할을 통해 복잡도를 낮추고 있다. 둘째는 모형의 일반화이다. 트랜스포머는 데이터의 구조적 편향에 대한 가정이 거의 없어 작은 규모의 데이터를 학습하기에는 적합하지 않다. 따라서 변형 모형들은 구조적 편향에 대한 가정을 추가하거나 정규화와 사전학습과 같은 방법을 이용해 이를 개선하고 있다. 셋째는 모형의 적응력에 대한 부분이다. 다양한 자연어 처리 작업에 대해 트랜스포머를 적용하기 위한 많은 변형 모형이 개발되고 있다. 여기서는 트랜스포머의 기본 구조에 대해 기술하고, 다음 장에서 다양한 변형 방식과 대표적인 변형 모형에 대해 정리보고자 한다.

2.3.1 셀프 어텐션과 기본 트랜스포머 모형의 구조

식(1)의 언어모델을 확장하면 한 단어의 확률을 계산하는 대신 단어 시퀀스의 확률을 계산하는 문제로 확장될 수 있으며 이것을 조건부 언어모델(Conditional Langauge Model)이라고 한다[6]. 번역과 같은 작업이 대표적인 조건부 언어모델의 예로, 예를 들어 영어를 한국어로 번역하는 경우에는 주어진 영어 문장의 단어 시퀀스를 조건으로 하여 한글 문장의 단어 시퀀스에 대한 확률 계산을 함으로써, 가장 확률이 높은 한글 문장을 찾는 작업으로 해석할 수 있다. 이와 같이 주어진 입력 시퀀스로부터 출력 시퀀스를 생성하는 모형을 시퀀스-투-시퀀스 모형이라고 한다. 초기의 시퀀스-투-시퀀스 모형은 RNN (Recurrent Neural Networks)으로 구현하였으며 인코더와 디코더로 이루어져 있다. 인코더는 입력 단어 시퀀스의 정보를 압축하고 디코더는 압축된 정보로부터 출력 단어 시퀀스를 생성하는 역할을 한다.

트랜스포머가 발표된 논문의 제목 “Attention is all you need”에서 알 수 있듯이 트랜스포머의 핵심은 셀프 어텐션에 있다. 이 논문 이전에 어텐션은 시퀀스-투-시퀀스 모형의 인코더에서 디코더로 연결되는 어텐션을 의미했다. 시퀀스-투-시퀀스를 RNN 모형으로 구현하면 입력 단어 시퀀스는 순차적으로 축적된 노드를 통해서만 출력 단어 시퀀스에 영향을 미칠 수 있었으나, 어텐션은 입력 단어들이 출력 단어에 직접적으로 영향을 미칠 수 있도록 고안되었다. 셀프 어텐션은 이러한 어텐션이 입력과 출력 사이에만 존재하는 것이 아니라 동일한 시퀀스 내에서 각 단어들 사이에 존재하도록 설계하였다.

이 때 인코더에서는 입력 단어 시퀀스의 모든 단어들 사이 즉 양방향으로 셀프 어텐션이 가능하나, 디코더의 출력 단어 시퀀스에서는 순방향으로만 어텐션이 가능하다. 이것은 디코더가 작동할 때 단어를 하나씩 생성하는 작업을 반복하여 전체 단어 시퀀스를 생성하기 때문이다. 따라서 아직 생성되지 않은 단어가 역방향으로 앞선 단어에 영향을 미칠 수는 없다. 이로 인해 인코더가 셀프 어텐션을 사용하는 것과는 달리 디코더에서는 마스크 셀프 어텐션을 사용하는데, 이것



〈그림 1〉 기본 트랜스포머 모형 구조 [14]

은 어텐션이 발생하면 안되는 위치의 어텐션 행렬 값을 가림으로써 역방향 어텐션을 방지하는 기법을 말한다.

트랜스포머 모형은 이러한 셀프 어텐션을 구현하는 것에 초점이 맞춰져 있다. 설명한 바와 같이 인코더는 완전한 셀프 어텐션을 사용하는 것에 비해 디코더는 마스크 셀프 어텐션을 사용한다. 여기에 기존의 시퀀스-투-시퀀스 모형의 어텐션과 마찬가지로, 인코더에서 디코더로 연결되는 어텐션이 추가되며 이것을 인코더-디코더 어텐션이라고 한다.

그림 1은 트랜스포머의 전체적인 구조를 보여준다. 트랜스포머의 인코더에서 입력 단어 시퀀스는 먼저 토큰 임베딩 과정을 거치고 여기에 위치 인코딩이 더해진다. 인코더의 한 레이어는 멀티헤드 어텐션, 잔차연결과 정규화, FFN (Feed-orward Network) 다시 잔차연결과 정규화의 네 단계로 이루어진다. 인코더는 설계자의 의도에 따라 여러 레이어로 구성될 수 있다.

디코더는 출력 단어 시퀀스의 첫 토큰(주로 문장의 시작을 알리는 토큰)을 시작으로 하여, 하나씩 생성되는 단어가 순차적으로 추가되는 시퀀스를 입력으로 받는다. 인코더와 동일하게 토큰 임베딩 과정과 위치 인코딩을 거친 후에 레이어의 입력으로 사용된다. 디코더는 인코더와 달리 마스크 멀티헤드 어텐션, 잔차연결과 정규화의 두 단계 이후 인코더로부터 연결되는 인코더-디코더 어텐션, 잔차연결과 정규화의 두 단계가 추가된다. 여기에 인코더의 마지막 두 단계와 같은 FFN, 잔차연결과 정규화가 추가되어 한 레이어가 총 6단계로 구성된다. 인코더와 마찬가지로 여러 레이어로 구성될 수 있으며 최종적으로 디코더의 출력은 선형 혹은 소프트맥스 층을 거쳐서 원래 모형이 목표로 하는 확률을 추정한다.

아래부터는 트랜스포머 논문[14]의 내용을 기반으로 트랜스포머의 구성요소를 설명한다.

2.3.2 위치 인코딩

트랜스포머는 기존의 시퀀스-투-시퀀스 모형에서 주로 사용한 RNN을 사용하지 않고 구현되기 때문에 토큰 간의 순서에 대한 정보가 없다. 따라서 위치에 대한 정보를 토큰에 추가하는 작업을 일반적인 토큰화 후에 수행한다.

2.3.3 어텐션 모듈

셀프 어텐션 모듈은 트랜스포머 모형의 핵심이라고 할 수 있다. 식 (3)과 같이 어텐션 값은 Q(Query), K(Key), V(Value)의 함수로 표현된다. 이 셋은 토큰의 임베딩 벡터로부터 계산되기 때문에 토큰의 원래 정보를 담고 있다고 생각하면 된다. Q는 자신을 제외한 다른 토큰들에게 어텐션의 정도를 묻는 질문으로 생각할 수 있고 K는 이에 대해 다른 토큰들이 주는 답으로 생각할 수 있다. 따라서 QK^T는 다른 토큰들로부터 자신에게 오는 어텐션의 정도를 표현한 것으로 해석할 수 있다. 여기에 Softmax 함수를 적용한 값을 어텐션 행렬이라고 한다. 여기에 어텐션을 주는 각 토큰의 정보를 담고 있는 V를 곱해서 어텐션 벡터를 완성한다.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

멀티헤드 어텐션은 이상의 어텐션을 다양한 관점에서 해석하고 이를 결합하기 위해 사용한다. 즉 식 (4)와 같이 적절한 수(논문에서는 8개를 사용)의 관점만큼 각각의 어텐션을 계산하고 그 결과를 결합해서 멀티헤드를 계산한다. 식 (4)에서 각 토큰에 대한 Q, K, V는 각각의 헤드 별로 가중치(W_i^Q, W_i^K, W_i^V)를 적용해서 헤드마다 새로운 값 즉 관점이 될 수 있도록 한다.

$$\begin{aligned}&\mathrm{MultiHead}\ (Q,K,V)\\&\quad=\mathrm{Concat}(\mathrm{head}_1,...,\mathrm{head}_h)\,W^O\\&\text{where}\,\mathrm{head}_i=\mathrm{Attention}\ (QW_i^Q,KW_i^K,VW_i^V)\end{aligned}$$

(4)

트랜스포머에서는 서로 다른 세 종류의 어텐션이 사용된다. 첫째는 지금까지 설명된 셀프 어텐션이다. 셀프 어텐션은 양방향의 어텐션을 모두 학습하며 인코더에서 사용된다. 둘째는 마스크 셀프 어텐션(Masked Self-attention)으로 디코더에서 사용되며 한 방향으로만 어텐션을 학습한다. 디코더에서는 순차적으로 토큰이 생성되기 때문에 역방향의 어텐션을 이용할 수 없기 때문에 마스킹을 이용해 이것을 구현한다. 셋째는 인코더 디코더 어텐션(encoder-decoder attention)이다. 전통적인 시퀀스-투-시퀀스 모형과 같이 트랜스포머에서도 인코더로부터 디코더로 연결되는 어텐션을 사용한다. 이때 위 설명에 따라 어텐션 계산방식을 생각해보면 Q는 디코더로부터 오고 K와 V는 인코더로부터 오는 것을 알 수 있다.

2.3.4 FFN(Feed-forward Network)

어텐션 모듈로부터 생성된 벡터는 각 토큰 위치 별로 독립적으로 완전연결 FFN을 통과하여 다음 출력으로 변환된다. 이 과정은 셀프 어텐션 결과를 최종 목적에 맞게 다시 학습하는 과정으로 생각할 수 있다. 식 (5)는 두 개의 선형 변환으로 이루어진 FFN을 보여주며 두 변환 사이에 ReLU 활성화 함수가 사용된다.

$$\mathrm{FFN}(x)=\max(0,x\,W_1+b_1)\,W_2+b_2 \qquad (5)$$

2.3.5 잔차 연결과 정규화

잔차연결(residual connection)은 셀프 어텐션 과정을 거치면서 원래 임베딩 벡터의 정보가 지나치게 손실 혹은 변형되는 것을 막기 위해 추가한다. 레이어 정규화(layer normalization)는 레이어의 출력 데이터에 대해 평균과 분산을 이용하여 정규화를 함으로써 데이터를 안정화하고 학습 속도를 개선하는 효과를 가져온다[15].

Ⅲ. 트랜스포머 변형 모형 현황

트랜스포머 변형 모형은 기존의 트랜스포머 모형의 일부만 사용하거나 새로운 언어모델의 사전학습 방법을 제안함으로써 모형의 학습 방법을 변경한 모형을 말한다. 구조 관점에서 볼 때 먼저 트랜스포머의 디코더만을 이용하여 생성 모델에 장점을 보이는 GPT 계열의 변형 모형이 있고, 둘째로 트랜스포머의 인코더만을 이용하여 범용 사전학습 언어모델을 제안한 BERT 계열의 변형 모형이 있다. 마지막으로는 인코더와 디코더를 모두 사용하는 시퀀스-투-시퀀스 모형이지만 노이즈제거 오토인코더 형태로 사전학습 방법을 제안한 BART(Bidirectional Auto-Regressive Transformer)가 있다. 이 장에서는 먼저 트랜스포머 변형 모형에서 사용되는 토크나이저에 대해 알아보고 GPT 계열, BERT 계열 그리고 BART의 순서로 트랜스포머 변형 모형의 현황을 살펴보고자 한다.

3.1 트랜스포머 변형 모형과 토크나이저

토크나이저는 주어진 입력 텍스트를 사전학습 언어모델에서 사용하는 토큰으로 분할하는 기능 즉 토큰화를 수행한다. 트랜스포머 변형 모형 이전에는 단어 기반 토큰화가 일반적으로 사용되었다. 단어 기반 토큰화는 가장 직관적이고 기본적인 토큰화 방식으로, 주어진 텍스트를 단어 단위로 분할하는 것이다. 한국어에서는 주로 형태소 분석을 통해 최소 의미 단위인 형태소 단위로 분리한다. 특정 말뭉치에 대해 자연어 처리를 하게 될 경우, 먼저 말뭉치 전체에서 사용된 단어로 어휘사전을 만들고 각 단어에 대해 숫자로 된 식별자를 할당한다. 그리고 이 어휘사전을 이용하여 분리한 입력 텍스트를 숫자로 된 식별자의 시퀀

스로 변환한다. 단어 기반 토큰화를 할 때는 모형의 성능과 공간 효율성 등을 고려하여 적절한 수의 어휘를 결정하게 되는데, 이때 사전에 없는 단어는 일괄적으로 [UNK] 즉 unknown 토큰으로 변환된다. 입력 텍스트에서 [UNK] 토큰의 비중이 커지면 모형의 성능에 결정적인 영향을 미치게 되는데 이와 같은 문제를 OOV(Out-Of-Vocabulary) 문제라고 한다. OOV 문제를 해결하기 위해서는 [UNK] 토큰의 수를 줄이는 것이 중요한데, 단어 기반 토큰화에서는 거의 필연적으로 수가 커지게 된다.

이러한 문제점을 극복하기 위해 제안된 방식이 서브워드 분리(subword segmentation)이다. 서브워드 분리는 하나의 단어를 의미가 있는 더 작은 서브워드로 분리하는 것을 말하며, 서브워드의 조합으로 다양한 단어를 표현할 수 있게 되므로 [UNK] 토큰의 수를 줄이는 것이 가능하다.

BPE(Byte Pair Encoding)은 원래 데이터 압축 알고리즘이지만 자연어 처리 분야에서 대표적인 서브워드 분리 알고리즘이 되었다. BPE에서는 초기 어휘사전을 글자 단위로 구성하고 말뭉치에서 빈도가 높은 쌍을 단계적으로 통합하는 방식으로 서브워드를 만들어간다. 이 과정을 반복하면 어휘 사전은 빈도가 높은 서브워드로 구성되고 토큰화 과정에서 텍스트를 이러한 서브워드 단위로 토큰화하므로 [UNK] 토큰의 수를 줄일 수 있다.

구글은 구글 번역기에서 BPE를 변형한 워드피스 토크나이저를 이용해 토큰화를 수행했다[16]. 위에서 설명한 바와 같이 BPE가 빈도수에 기반하여 서브워드를 병합하는 것과 달리 워드피스 토크나이저는 말뭉치의 우도를 높이는 쌍을 선택하여 병합한다. 워드피스 토크나이저는 트랜스포머 변형 모형인 BERT에서도 토크나이저로 사용되었으며, 이후 다양한 트랜스포머 변형 모형은 각기 모형의 의도에 맞게 서브워드 토큰화를 수행하는 토크나이저를 사전학습하여 모형에서 사용했다.

3.2 GPT 모형 현황

3.2.1 GPT-1

트랜스포머의 변형 모형 중에서 가장 널리 알려진 것은 BERT와 GPT라고 할 수 있다. 둘 다 2018년에 발표되었는데, GPT가 6월에 발표된 반면 BERT는 10월에 발표되어 시기적으로 GPT가 살짝 앞서있다. GPT가 BERT와 다른 점은 트랜스포머의 디코더로 언어모델 사전학습을 수행한다는 점이다[17]. 이로 인해 문서 생성에서 더 장점을 갖는다고 볼 수 있다. 일반적으로 문서 생성은 앞 단어부터 순차적으로 생성하기 때문이다.

GPT는 언어모델 사전학습을 대량으로 수행하는 방안을 제시하고 이를 이용해 다양한 자연어 처리 작업에 전이학습을 할 수 있는 기반을 마련했다는 점에서 큰 의의가 있다. GPT 이전의 일반적인 자연어 처리 작업은 Word2Vec 계열의 워드 임베딩 정도를 전이학습으로 사용하고 신경망 모형 자체를 전이학습으로 사용하는 경우는 많지 않았다. 따라서 신경망에 대한 학습은 특정 작업을 대상으로 이루어졌고 대부분 지도학습이었기 때문에 라벨이 있는 대규모의 데이터를 확보하는 것에 많은 어려움이 있었다. GPT는 라벨이 없는 대규모의 텍스트를 대상으로 비지도 언어모델 사전학습을 통해 자연어에 대한 이해가 높은 신경망 모형을 학습하고, 이 모형을 소규모의 데이터를 가지는 자연어 처리 학습에 전이하여 지도학습을 함으로써 매우 효과적인 자연어 처리가 가능하도록 했다. 이후 이러한 2단계 학습모형은 거의 모든 자연어 처리 학습에서 활용된다. 언어모델로 사전학습된 GPT를 MNLI(Multi-Genre Natural Language Inference), SNLI(Stanford Natural Language Inference), QNLI(Question Natural Language Inference), RACE(Large-scale ReAding Comprehension Dataset From Examination) 등에 적용한 결과, 기존의 ELMo나

양방향 LSTM 기반의 모형을 능가하는 성능을 보였다.

3.2.2 GPT-2

처음 발표된 GPT(GPT-1)가 언어모델 사전학습과 이후 자연어처리 작업에서의 미세조정을 분리하여 두 단계로 학습하는 것을 가정한 반면 GPT-2는 사전학습만으로 바로 사용할 수 있는 모형을 제안했다는 점에서 의의가 있다[Radford, 2019]. 이와 같이 미세조정이 전혀 없이 수행하는 자연어 처리 작업을 논문에서는 제로샷 작업(Zero-shot task)이라고 하고, 그러한 환경을 제로샷 환경(Zero-shot setting)으로 부르고 있는데, 이는 메타 러닝의 개념을 가져왔다고 볼 수 있다. 논문의 목표는 다양한 데이터셋을 이용해 언어모델 비지도 사전학습만으로 일반화 성능을 최대한 높여서 다양한 작업에 바로 전이하여 사용할 수 있는 범용 모형을 만드는 것이다.

식(1)의 조건부 확률은 $p(w_{n-k}, ..., w_n \mid w_1, ..., w_{n-k-1})$의 형태 즉 주어진 단어 시퀀스로부터 새로운 단어 시퀀스의 확률을 추정하는 형태로 확장할 수 있으며, 트랜스포머는 이 확률을 셀프 어텐션 기반의 인공신경망으로 추정함으로써 탁월한 발전을 이루었다. 이와 같은 관점에서 볼 때 하나의 자연어 처리 작업은 $p(output \mid input)$을 추정하는 모형으로 표현할 수 있는데, 여기에 데이터 입력뿐 아니라 수행해야 할 작업을 함께 입력으로 넣어서 $p(outputvertinput, task)$를 추정하는 모형으로 확장이 가능하다. 즉 다양한 자연어 처리 작업을 수행할 수 있는 하나의 일반화된 모형을 개념적으로 제안하고 있다.

이 때 중요한 점은 언어모델에 대한 비지도 학습에서는 입력에서 작업(task)에 대한 부분과 출력을 명시적으로 지정하지 않는다는 점이다. 다음 문장을 보면 이 개념을 이해할 수 있다. 문장에서 앞부분은 불어에 해당하고 중간의 “translated to English”는 작업에 해당하며 뒷부분은 앞부분이 번역된 영어 문장에 해당한다. 비지도 학습에서는 이 문장에서 작업과 번역된 답을 명시적으로 지정하지 않고 셀프 어텐션 모형에 의해 학습을 함으로써 자연스럽게 번역에 대한 학습이 이루어지도록 한다. 이는 인간이 수많은 책을 읽는 것만으로 다양한 사고와 판단을 할 수 있다는 것을 그대로 인공지능 학습에 이용했다는 점에서 많은 시사점을 준다고 할 수 있다.

“Brevet Sans Garantie Du Gouvernement”, translated to English: “Patented without government warranty”[7]

모형의 구조 관점에서 GPT-2는 트랜스포머의 디코더 부분을 사용한다는 점에서 GPT-1과 거의 동일하다고 볼 수 있다. 다만 각 레이어에서 정규화 블록을 입력 부분으로 옮기고 마지막 셀프 어텐션 블록 뒤에 정규화 블록을 추가했다는 차이가 있다. 또한 모형의 표현력을 증가시키기 위해 어휘 집합의 크기를 늘리고 한번에 입력가능한 토큰의 수를 512개에서 두 배인 1,024개로 늘렸다. GPT-2를 GPT-1과 구별시키는 중요한 차이점은 데이터셋의 구성이라고 할 수 있다. 위에서 보인 예와 같이 학습에 사용하는 문서들은 그 자체로 다양한 자연어처리 작업에 적용할 수 있는 내용들을 포함하고 있어야 한다. 이를 위해 웹 스크래핑을 이용해 40GB에 달하는 자체 데이터셋인 WebText를 구축했다.

GPT-2는 언어모델 사전학습만으로 다양한 자연어처리 작업을 수행하는 것을 목표로 했기 때문에 실험에서는 미세조정 없이 바로 벤치마크 테스트를 했다. 우선 LAMBADA를 비롯하여 언어모델의 성능을 측정하는 8개의 데이터셋에서는 7개의 데이터셋에서 가장 좋은 성능을 보였다. 참고로 LAMBADA는 문장을 완성하는 문제로 언어의 장기의존성을 평가한다. 상식추론 능력을 평가하는 위노그라드 스키마 챌린지

(Winograd Schema Challenge)에서도 70.7%로 기존 최고 성능을 7% 개선했다. 위노그라드 스키마 챌린지는 문장에서 대명사가 가리키는 것이 무엇인지를 맞추는 문제이다. 독해능력을 평가하는 CoQA(Conversation Question Answering) 데이터셋에서는 당시 BERT를 기반으로 미세조정을 한 모형이 0.89에 가까운 F1 성능을 보인 것에 비해 그보다 많이 떨어지는 0.55의 F1 성능을 보였다. 그럼에도 불구하고 미세조정이 없이 그 정도의 성능을 보인 것은 매우 고무적이라고 할 수 있다. 그 외 문서 요약 성능을 평가하는 CNN/DailyMail 데이터셋, 번역 성능을 평가하는 WMT-14 데이터셋에서는 기존 최고 성능보다는 못하지만 인상적인 성능을 보였다.

3.2.3 GPT-3

GPT-2가 메타러닝 관점에서 제로샷 러닝(zero-shot learning) 즉 미세조정이 전혀 없는 일반화 모형에 치중했다면 GPT-3은 퓨샷 러닝(few-shot learning)으로 인한 성능 향상에 더 초점을 맞췄다고 할 수 있다[9]. 미세조정이 자연어처리 작업에 대한 지도학습 과정에서 라벨이 있는 수천개 정도의 데이터를 사용하고 사전학습 모형의 가중치를 변경하는 것에 비해, 퓨샷 러닝에서는 훨씬 적은 수의 데이터를 사용하고 사전학습 모형의 가중치를 변경하지 않는다. 이것은 사람이 몇 개 안 되는 예제만으로도 새로운 문제에 쉽게 적응하는 것을 반영했다고 할 수 있다. 원샷 러닝(one-shot learning)은 예제를 하나만 사용하는 것을 말한다. 제로샷 러닝은 원샷 러닝에서 예제 대신 작업에 대한 자연어 설명을 사용한 것과 같다. 즉 사람에게 해야 할 작업이 무엇인지 알려주는 과정과 비슷하다. GPT-3에서는 제로샷, 원샷, 퓨샷의 세 모형을 학습하고 기존 최고 성능과 비교하였다.

위에서 설명한 바와 같이 보통 사람은 새로운 자연어처리 작업을 하기 위해 수천개에 달하는 예제 데이터를 필요로 하지는 않는다. 이는 언어에 대한 이해가 높고 유사한 경험을 갖고 있기 때문인데, 만일 언어모델이 사전학습을 통해 사람과 같이 높은 언어 이해도를 갖게 된다면 마찬가지로 몇 개의 예제 데이터만으로도 충분히 높은 성능을 보일 수 있을 것이다. GPT-3은 이와 같은 점에 착안하여 퓨샷 러닝의 가능성을 제시하고 있다. GPT-3가 개선된 다른 점은 파라미터의 크기이다. GPT-2가 15억개 정도의 파라미터를 사용한 것에 비해 GPT-3은 1,750억개 정도로 100배가 넘는 수의 파라미터를 사용했다. 훈련에 사용한 데이터셋도 크게 증가했는데, 우선 WebText를 확장한 WebText2가 22% 가량 사용되고 필터링을 거친 크롤링 데이터가 60% 사용되었다. 이로 인해 훈련에 어마어마한 양의 계산비용이 소모되었으리라고 짐작할 수 있다. 모형이 이렇게 커지게 되면 훈련은 당연히 감당하기 어렵지만 자연처처리 작업에서의 사용도 쉽지 않을 수 있다.

훨씬 커진 모형과 퓨샷 러닝으로 인해 우선 언어모델의 성능을 측정하는 LAMBADA에서 기존 최고 성능을 갱신했다. 퓨샷 모형에서 가장 높은 성능을 보였고, 흥미로운 점은 제로샷 모형의 성능이 원샷 모형보다 높게 나왔다는 것이다. GPT-2가 번역에서 그다지 높은 성능을 보이지 못한 것에 비해 GPT-3의 퓨샷 모형은 불어에서 영어로 혹은 독일어에서 영어로 번역하는 작업에서 기존 최고 성능을 능가하기도 했다. GLUE[18]를 확장한 SuperGLUE[19]에서는 SOTA에는 미치지 못했으나 미세조정을 한 BERT-Large보다는 높은 성능을 보였다. SuperGLUE는 GLUE의 일부 작업에 대해 사람에 근접하는 성능을 보이는 딥러닝 모형이 나옴에 따라, 작업을 더 어렵고 다양하게 변화시키는 작업을 했으며 8개의 벤치마크 데이터셋으로 구성되어 있다. 이 외에도 다양한 데이터셋에서 주목할 만한 성능을 보였다.

GPT-3은 가능성과 한계를 동시에 보였다. 모

형이 커서 많은 계산비용을 필요로 한다는 것 외에 대상 자연어 처리 작업에 따라 편차가 존재하는 것, 퓨샷 러닝의 모호함 등이 한계로 지적되었다. 이 외에도 사전학습된 언어모델이 성별이나 인종, 종교 등에 대한 편향을 가질 수 있는 가능성 등이 제시되었다.

3.3 BERT 모형 현황

3.3.1 기본 BERT 모형

BERT는 대표적인 변형 트랜스포머 모형으로 트랜스포커의 복잡도를 낮추기 위해 트랜스포머의 인코더만 사용하여 언어모델 사전학습을 수행한다[8]. 디코더가 마스크 셀프 어텐션을 사용하여 순방향의 셀프 어텐션만 학습되는 것에 비해 BERT는 완전한 셀프 어텐션을 사용함으로써 양방향의 완전한 문맥을 학습할 수 있다는 장점이 있다.

BERT는 언어모델 사전학습을 위해 두 개의 학습 방법을 제안하였다. 첫째는 MLM(Masked Language Model) 즉 마스크 언어모델이다. 전체 토큰에서 15%의 토큰을 무작위로 마스킹하여 입력 토큰에서 제외하고 마스킹 된 단어를 예측하도록 모형을 학습했다.

둘째는 NSP(Next Sentence Prediction) 즉 다음 문장 예측 학습이다. NSP를 위한 데이터 셋 구성을 위해 절반의 데이터는 동일한 문서의 연속된 두 개의 문장에 대해 'IsNext'라는 라벨을 붙였고, 나머지 절반은 임의의 문장과 전체 말뭉치에서 무작위로 가져온 문장의 쌍에 대해 'NotNext'라는 라벨을 붙였다. 학습은 주어진 두 문장에 대해 두 가지 라벨 중 맞는 것을 예측하도록 수행했다.

3.3.2 RoBERTa(Robustly Optimized BERT Pretraining Approach)

RoBERTa는 논문의 제목인 "Robustly Optimized BERT Pretraining Approach"의 약자로, 기존의 BERT가 언어모델 관점에서 충분히 학습되지 못했다고 보고 동일한 BERT 모형에 대해 보다 견고한 학습 방법을 제안했다[20]. 논문에서 밝히고 있듯이 이 연구는 BERT 사전학습 연구[8]의 복제 연구로, 새로운 모형을 제안하기보다는 기존의 BERT 모형의 사전학습을 보완하기 위한 방안을 제시하고 이를 RoBERTa로 명명하였으며, 실험을 통해 BERT 이후에 발표된 다른 모형에 비해 더 나은 성능을 보이는 것을 검증하였다. RoBERTa에서는 BERT에서 사용된 문자 단위의 BPE를 그대로 사용하지 않고 GPT-2에서 사용된 바이트 단위의 BPE를 사용하여 토큰화를 수행했다.

RoBERTa에서 개선한 내용의 첫째는 동적 마스킹의 사용이다. 기존 BERT는 마스킹을 데이터 전처리 과정에서 한번만 수행하기 때문에 입력 데이터의 마스킹이 고정된다. 따라서 항상 동일한 마스킹에 대해 학습하게 된다. RoBERTa에서는 이를 피하기 위해 학습 데이터를 10회 복사하고 매번 다르게 마스킹이 되도록 했다. 이는 동일한 텍스트에 대해서 다른 학습을 수행하도록 함으로써 보다 효율적으로 학습을 수행했다고 할 수 있다.

둘째, 앞서 설명한 바와 같이 BERT는 MLM과 NSP 학습을 수행한다. 그러나 NSP 학습의 필요성에 대한 의문이 BERT 발표 이후에 꾸준히 제기되었다. BoBERTa에서는 기존의 BERT가 NSP를 구현하기 위해 두 세크먼트의 쌍을 입력으로 사용한 것에 비해, NSP를 제외하고 길이가 512 토큰보다 작은 연속된 문장들로 하나의 입력을 구성했다. 그 결과 NSP를 제외한 것이 더 성능이 좋음을 보였다.

셋째, 기존의 BERT가 하나의 배치 크기를 256 시퀀스로 구성한 것에 비해 2K와 8K의 두 크기로 배치를 구성하여 실험한 결과, 배치 크기를 늘리는 경우 성능이 향상되는 것을 보였다.

GLUE(General Language Understanding Evaluation)는 자연어 처리 성능을 평가하기 위해 9개의 자연어 처리 작업에 대해 구성된 벤치마크

데이터셋이다[18]. RoBERTa는 이 GLUE 외에도 SQuAD 2.0과 RACE에서 발표 당시 가장 좋은 성능을 보이는 것을 입증했다.

3.3.3 ALBERT(A Lite BERT)

ALBERT는 기존의 BERT보다 큰 모형을 보다 효과적으로 학습하기 위해 제안되었다[21]. 모형이 커질 때 발생하는 가장 큰 문제는 매개변수 즉 파라미터의 수가 늘어난다는 것이다. ALBERT는 파라미터의 수를 줄이기 위한 두 가지 방안을 제시했는데 첫째는 토큰 임베딩 벡터의 크기를 줄이는 것이다. BERT에서는 토큰 임베딩과 인코더 레이어 내의 히든 벡터 임베딩 크기를 동일하게 했는데, 그 결과 모형이 커지면 임베딩 과정에서 매우 큰 수의 파라미터를 요구하게 된다. ALBERT에서는 토큰 임베딩을 레이어의 히든 벡터 임베딩 크기보다 작게 설정했는데, 그 이유는 토큰 임베딩은 토큰 자체의 정보만 담고 있는 것에 비해 히든 벡터는 셀프 어텐션을 반영하여 주변 단어와의 관계 정보를 담고 있어서 더 큰 벡터가 요구된다고 보았기 때문이다. 이렇게 하면 어휘 크기의 원핫벡터로부터 더 작은 크기의 초기 토큰 임베딩으로 변환하는 파라미터와 이로부터 다시 히든 벡터 임베딩으로 변환하는 파라미터로 분리되고 그 결과 파라미터 수를 BERT에서 요구하는 것보다 크게 줄일 수 있다. 둘째 방안은 인코더를 구성하는 레이어들의 멀티헤드 어텐션과 FFN 파라미터를 서로 공유하는 것이다. 이렇게 함으로써 필요한 파라미터의 수를 줄였을 뿐 아니라 파라미터를 보다 안정화하는 효과를 가져왔다.

파라미터의 수를 줄이는 것 외에도 ALBERT는 기존의 NSP를 더 향상시키기 위한 방안을 제시했다. 기존 BERT의 NSP는 둘째 문장을 말뭉치에서 임의로 가져왔기 때문에 엄밀하게 말하면 문장의 순서만 예측하기보다는 두 문장의 내용간 유사성을 함께 학습하게 된다. ALBERT는 연속한 두 문장의 순서를 바꿔서 'NotNext'에 해당하는 데이터셋을 구성함으로써 NSP를 SOP (Sentence Ordering Prediction)으로 변경하여 학습하도록 했다.

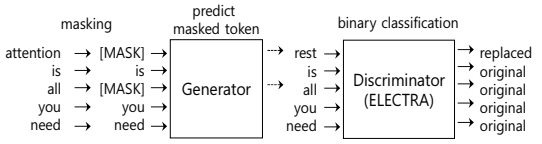
ALBERT는 GLUE, SQuAD, RACE 데이터셋의 대부분의 분야에서 RoBERTa를 능가하는 성능을 보였다.

3.3.4 ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately)

ALBERT가 큰 모형에서 학습의 효율성을 개선하고자 했다면 ELECTRA는 사전학습 자체의 효율성을 향상시키고자 한 연구라고 할 수 있다[22]. BERT 모형은 MLM 학습에서 전체 토큰의 15%를 무작위로 [MASK]이라는 새로운 토큰으로 변경하고 이 토큰의 원래 단어를 예측하는 방식으로 예측했다. 여기에는 두 가지 문제점이 있는데, 첫째는 이로 인해 입력 텍스트 전체 토큰의 15%에 대해서만 학습이 이루어진다는 것이고 이로 인해 학습 효율이 떨어질 수밖에 없다. 둘째 [MASK]라는 토큰은 원래 자연어 텍스트에는 존재하지 않는 토큰이기 때문에, 사전학습과 실제 자연어 처리 작업을 위한 미세조정 사이에 불일치가 존재하고 이것 역시 학습의 효율을 떨어뜨릴 수 있다.

ELECTRA는 이 문제를 해결하기 위해 그림 2와 같은 교체 토큰 탐지(Replaced Token Detection) 방식을 제안했다. ELECTRA는 두 개의 트랜스포머 인코더로 구성되는데, GAN(Generative Adversarial Network)과 유사하게 생성자(Generator) 인코더와 판별자(Discriminator) 인코더가 결합되어 사전학습을 수행하도록 했다. 그림 2와 같이 생성자는 BERT와 동일하게 먼저 전체 토큰의 15%에 대해 마스킹을 수행하고 마스킹된 위치의 토큰들을 예측해서 전체 입력을 재구성한다.이 때 어떤 토큰은 올바르게 예측하고 어떤 토큰은 틀릴 수 있는데, 올바르게 예측한 토큰과 마스킹되지 않은 토

큰에 대해서는 ‘original’이라는 라벨이, 잘못 예측한 토큰에 대해서는 ‘replaced’라는 라벨이 부여된다. 판별자는 생성자가 출력하는 모든 토큰에 대해 이진분류로 ‘original’과 ‘replaced’를 예측하도록 학습한다. 생성자가 잘못 예측한 단어는 ‘replaced’가 되고 판별자는 이 ‘replaced’를 탐지해야 하므로, 이 과정을 교체 토큰 감지(replaced token detection)라고 부른다.



〈그림 2〉 교체 토큰 감지[22]

생성자는 MLM과 동일한 손실함수로 학습되고 판별자는 이진 교차 엔트로피(Binary Cross Entropy)로 학습되며, GAN과는 달리 두 모형이 경쟁적 혹은 적대적으로 학습되지 않고 두 모형의 손실함수를 합한 함수를 최소화하도록 학습한다. GAN처럼 학습하기 위해서는 생성자가 판별자를 속이기 위해 학습해야 하나, 개념적으로 ELECTRA의 생성자는 가짜를 생성하기보다 원래 토큰을 최대한 정확하게 예측하여 진짜를 생성하기 위해 학습된다는 차이가 있고 생성자로부터 샘플링을 통한 역전파가 불가능했기 때문에, 생성자는 본래 목적에 맞게 최대우도추정법으로 학습했다. ELECTRA가 BERT에 비해 보다 효율적으로 학습되는 이유는 생성자가 마스킹된 토큰에 대해서만 예측하는 것이 아니라 모든 토큰에 대해 예측하는 방식으로 학습되기 때문이다. 따라서 사전학습이 완료된 후에 후속 자연어 처리 작업(downstream work)에서는 생성자를 버리고 판별자만을 사용한다. 생성자는 BERT와 동일하게 마스킹된 단어에 대해서만 예측하도록 학습되기 때문이다.

ELECTRA 모형의 설계에서는 가중치의 공유와 생성자의 크기에 대한 부분이 고려되었다. 먼저 가중치의 공유와 관련하여 생성자와 판별자는 동일한 트랜스포머 인코더 구조를 가지므로 가중치의 공유가 가능하다. 실험을 통해 토큰 임베딩과 레이어 내부의 히든 벡터 모두를 공유하는 것이 가장 좋은 성능을 보였으나, 문제는 이렇게 할 경우 생성자와 판별자가 동일한 가중치 즉 파라미터 수를 가지게 된다는 점이다. 즉 기존 BERT에 비해 두 배의 파라미터를 학습해야 하고 이것은 학습의 효율을 떨어뜨리게 된다.

또 한 가지 문제는 생성자가 잘 학습될수록 판별자의 예측이 어려워지고 이로 인해 판별자의 성능이 떨어지는 결과를 가져왔다. 또한 생성자가 ‘replace’하는 단어가 줄어들수록 판별자의 학습 대상 토큰의 수 역시 줄어들게 되고 이로 인해 학습 효율이 떨어질 것으로 생각할 수 있다. 이러한 문제들에 대한 해결책으로 판별자의 크기에 비해 생성자의 크기를 1/4 혹은 1/2로 줄임으로써 더 높은 성능을 얻고 학습 효율을 높일 수 있었다.

생성자가 전체 토큰에 대한 교체 토큰 탐지를 수행함으로써 사전학습 효율을 높인다는 것을 증명하기 위해 ELECTRA는 세 가지 실험을 수행했다. 첫째는 ELECTRA의 구조를 유지하면서 생성자에서 마스킹한 15%의 토큰에 대해서만 판별자의 손실함수를 계산했다. 즉 전체 토큰이 아닌 마스킹된 토큰에 대해서만 예측 손실을 계산했다. 이 실험(실험 1)의 목적은 BERT와 동일하게 15%를 유지하면서 교체 토큰 탐지를 수행하는 것이 원래 토큰을 예측하는 것에 비해 더 나은 효과를 보이는지 보기 위한 것으로 생각할 수 있다. 둘째는 판별자의 입력으로 마스킹된 토큰 대신 생성자가 생성한 토큰을 사용하고, 이 토큰들에 대해서 교체를 탐지하는 이진분류를 하는 대신 원래 토큰을 예측하는 MLM 학습을 수행했다. 이 실험(실험 2)의 목적은 [MASK]를 사용하지 않는 것만으로 BERT에 비해 나은 성과를 보이는지 보기 위해서라고 생각할 수 있다. 마지막 실험(실험

3)은 생성자가 출력한 모든 토큰에 대해 MLM 학습 즉 원래 토큰을 예측하도록 학습하는 것이다. 실험 결과 실험 1, 실험 2 모두 BERT에 비해서는 조금 높은 성능을 보였으나 ELECTRA보다는 많이 떨어졌다. 실험 3은 실험 1, 실험 2에 비해 월등한 성능을 보였으나 ELECTRA에는 미치지 못했다.

ELECTRA는 ALBERT가 큰 모형을 고려한 것에 비해 사전학습 자체의 효율성을 향상시키자 한 것으로 그 결과 크기가 작은 모형에서 매우 우수한 학습 효율을 보였다. 결과적으로 GLUE와 SQuAD에서 ALBERT가 갱신했던 많은 기록을 다시 갈아치우는 결과를 가져왔다.

3.4 BART 모형 현황

BART는 BERT와 GPT가 각각 갖는 문제점을 극복하고자 한 모형이라고 할 수 있다[23]. GPT는 트랜스포머의 디코더를 사용하기 때문에 생성 위주의 모형에 강점을 보이는 반면 순방향 셀프 어텐션만 학습되므로 양방향의 문맥정보는 반영하지 못한다는 단점이 있다. BERT는 트랜스포머의 인코더를 사용하기 때문에 양방향 셀프 어텐션을 이용한 양방향 문맥정보를 반영한다는 장점이 있지만 마스킹 방식의 학습은 순차적인 생성에는 적합하지 않다. BART는 표준적인 시퀀스-투-시퀀스 트랜스포머 모형을 사용하여 언어모델을 사전학습하는 새로운 방안을 제시한다는 점에서 의의가 있다.

BART는 트랜스포머 모형을 노이즈 제거 오토인코더(denoising autoencoder)로 사용하는데, 이 모형은 원래의 입력에 노이즈를 추가하여 오토인코더의 입력으로 사용하고 출력 즉 라벨은 원래의 입력을 사용하여 학습함으로써, 오토인코더 모형이 노이즈를 제거하고 원본을 복원하도록 한다. 일반적인 오토인코더에 비해 노이즈 제거 오토인코더는 더 강건하고 효과적인 학습이 가능한 것으로 알려져 있다. BART는 먼저 다양한 방식으로 원래의 토큰에 노이즈를 추가하고 이를 양방향 인코더로 인코딩한다. 인코딩한 결과는 트랜스포머 모형과 동일하게 단방향 디코더로 연결되고 디코더는 원래의 토큰을 예측한다. 학습은 디코더가 예측한 토큰과 원래의 토큰에 대한 손실함수를 사용하여 이루어진다.

BART에서는 노이즈를 추가하기 위해 다양한 방법을 제시했다. 첫째, BERT와 동일하게 무작위로 토큰을 선정하여 [MASK] 토큰으로 치환했다. 둘째, 무작위로 토큰을 선정해서 제거하고 모형이 제거된 토큰의 위치를 예측하도록 한다. 셋째, 임의의 길이의 연속된 토큰을 하나의 [MASK] 토큰으로 치환하고, 모형이 치환된 토큰의 수를 예측하도록 한다. 넷째, 하나의 문서를 문장으로 분리하고 문장들을 무작위로 뒤섞는다. 다섯째, 문서에서 하나의 토큰을 무작위로 선택하고 이 토큰이 시작점이 되도록 문서를 회전시킨다.

논문에서는 사전학습된 BART 언어모델을 이용해 문장 분류, 토큰 분류, 문장 생성, 번역의 네 가지 자연어처리 작업에 대해 미세조정하는 방법을 제시했다. 먼저 문장 분류에서는 동일한 임베딩 토큰을 인코더와 디코더에 입력하고 디코더 마지막 토큰의 최종 히든 벡터를 분류 토큰을 분류기의 입력으로 사용했다. 토큰 분류 문제에서는 전체 문서를 입력으로 사용하고 디코더의 최종 히든 벡터 전체를 각 토큰에 대한 최종 임베딩 표현으로 사용하였다. 질의 응답이나 문서 요약과 같은 문장 생성 작업에서는 트랜스포머 모형 자체를 그대로 사용하여 입력 시퀀스에 대해 순차적으로 출력 시퀀스를 생성하도록 학습했다. 번역은 기본적으로 문장 생성과 동일하게 동작하지만, 입력과 출력의 언어가 서로 다르면서 동시에 단어 간의 매핑이 이루어지기 때문에 이를 반영할 수 있도록 인코더의 임베딩 레이어를 수정하였다.

대형 언어모델에 대한 학습 효과를 보기 위해 BART를 RoBERTa와 동일한 수준으로 학습하고 분류 작업 중심의 벤치마크 데이터셋 들에 대해 비교한 결과, SST, QQP, QNLI, RTE 데이터셋에 대해서는 RoBERTa보다 좋은 성능을 보였으나, SQuAD 2.0, MNLI, STS-B, MRPC, CoLA에서는 뒤떨어지는 성능을 보였다. 다만 성능의 차이는 크지 않았다. 사실 BART에서 제안한 사전학습 방식은 문장 생성, 번역과 같은 시퀀스-투-시퀀스 작업에서 강점이 있기 때문에, 일반적인 분류 작업에서의 성능은 뒤처지지 않는다는 정도로도 의의가 있다고 볼 수 있다. 실제로 문장 요약 작업의 성능을 보기 위해 CNN/DailyMail과 Xsum 데이터셋에 대해 비교한 결과 BART는 전 분야에서 기존 최고 성능을 능가했다. 또한 대화 응답 성능을 측정하는 ConVAI2 데이터셋, 질의응답 성능을 측정하는 ELI5, 번역 성능을 측정하는 WMT16 루마니아어-영어 번역에서도 최고 성능을 갱신했다. 결과적으로 BART는 시퀀스-투-시퀀스 작업의 전 분야에서 장점을 보이는 사전학습 언어모델이라고 정리할 수 있다.

Ⅳ. 국내 트랜스포머 변형 모형 현황

4.1 KorBERT

KorBERT는 ETRI 엑소브레인 연구진에서 개발 및 배포하고 있는 한국어 BERT로, 기존의 구글 다국어 BERT(multilingual BERT)가 104개의 다국어를 통합하여 하나의 모형으로 제공하는 반면, KorBERT는 교착어인 한국어만의 특성을 반영하여 만든 형태소분석 기반의 언어모델을 제공하고 있다[24]. 이외에 추가로 다국어 BERT와 유사하게 만든 워드피스 기반 언어모델도 함께 제공하고 있는데 두 모델은 23GB 크기의 동일한 한국어 말뭉치로 학습되었으며 토크나이저가 다르기 때문에 사용하는 어휘집합이 다르고 이로 인해 성능에서도 차이를 보인다. 전반적으로 형태소 기반 언어모델이 더 우수한 성능을 보인다. 의미역 인식, 기계 독해, 단락 순위화, 문장 유사도 추론, 문서 주제분류의 다섯가지 항목에 대해 모형의 성능을 평가하였으며 구글 다국어 BERT에 비해 평균 4.5% 우수한 것으로 나타났다.

4.2 KoBERT

KoBERT는 KorBERT와 동일한 목적 즉 구글 다국어 BERT의 한국어 처리 한계를 극복하기 위해 SKT-AI에서 개발되어 유사한 시기에 발표된 모형으로, 한국어로 사전학습된 센텐스피스(sentencepiece) 토크나이저를 사용한다[25]. 센텐스피스 토크나이저는 사전 토큰화 작업을 하지 않고 토큰화를 수행하기 때문에 언어에 종속되지 않는다는 장점을 가지고 있어 한국어에도 활용이 가능하다. KorBERT에 비해 매우 적은 학습 데이터셋인 5M의 문장과 54M의 단어를 사용해 학습하였다. 어휘사전의 크기도 8,002로 KorBERT의 30,797개에 비해 훨씬 적다. 그럼에도 불구하고 네이버 감성 분석 데이터셋에 대해 다국어 BERT의 성능인 0.875보다 더 우수한 0.901의 성능을 보였다는 점에서 인상적이다.

4.3 KR-BERT

KR-BERT 역시 KorBERT, KoBERT와 동일한 목적으로 개발되었다[26]. 문자 관점에서 봤을 때, 한국어는 11,172개의 문자가 존재하는 반면 구글 다국어 BERT에는 이 중에서 1,187개만의 문자만 포함된 점을 지적했다. 또한 작은 규모의 언어모델 필요성을 강조하고 그에 부합하는 모형을 제안하고자 했다. 한국어는 하나의 음절이 자음과 모음으로 이루어져 있어 이러한 문자소 단위가 더 의미를 가질 것으로 보고 토크나이저의 단위를 음절 단위와 문자소 단위의 두가지로 구분했다. 이 둘에 대해 구글의 기본 워드피스 토크

나이저와 양방향 워드피스(Birectional WordPiece) 토크나이저를 각각 적용하여 총 네 개의 모형을 구현하고 성능을 비교했다. 그 결과 음절단위 양방향 워드피스 토크나이저를 사용한 모형이 언어모델이 마스크 언어모델 정확도에서 가장 우수한 성능을 보이는 것으로 나타났다. 음절단위 KR-BERT는 2.47GB의 말뭉치로 학습되었고 어휘사전의 크기는 16,424이다.

4.4 KoELECTRA

KoELECTRA는 ELECTRA를 한국어에 대해 학습한 모형으로, 버전 3의 경우 신문, 메신저, 웹 등의 한국어 데이터로 이루어진 20GB의 “모두의 말뭉치”를 이용하여 학습되었다[27]. 모형의 크기에 따라 KoELECTRA-Base와 KoELECTRA- Small의 두 모형을 제공한다. 토크나이저는 워드피스 토크나이저를 사용하는데 이는 트랜스포머 라이브러리만으로 사전학습 모형을 사용할 수 있도록 하기 위해서이다. 성능 평가 결과 NSMC, Naver NER, KorNLI, KorSTS, KorQuAD와 같은 대부분의 한국어 벤치마크 데이터셋에서 KoBERT와 XLM-Roberta-Base[28]보다 우수한 성능을 보였다.

4.5 KorSciBERT

KorSciBERT는 과학기술분야에 특화된 BERT 사전학습 언어모델이라는 점에서 차별점을 갖는다[29]. 한국과학기술정보연구원과 한국특허정보원이 함께 연구한 결과물로, 논문과 특허에 관련된 말뭉치 97GB를 대상으로 학습했으며 어휘사전의 크기는 15,330이다. 위의 다른 모형이 범용적인 목적으로 사용된다면 KorSciBERT는 주로 논문과 특허에 관련된 학술 목적으로 사용이 된다. 토크나이저는 Mecab-ko 토크나이저와 BERT의 워드피스 토크나이저를 결합하여 구현했다.

4.6 KoGPT2

SKT-AI에서 발표한 KoGPT2는 GPT-2를 한국어 위키백과, 청와대 국민청원 등을 포함한 40GB 이상의 한국어 말뭉치로 학습했으며, 이상의 모델과 마찬가지로 GPT-2의 한국어 성능을 개선하고자 하는 목적으로 개발되었다[30]. GPT-2가 생성에 많은 강점을 보이는 것과 마찬가지로 KoGPT2 역시 챗봇(Ko-GPT2-Chatbot)과 같은 문장 생성 분야에서 활용될 것으로 생각된다[31].

4.7 KoGPT

KoGPT는 카카오브레인이 공개한 GPT-3 모형의 한국어 특화 언어모델이다[32]. 학습에 사용된 토큰의 수는 2,000억개, 학습된 매개변수의 수는 60억개 정도로 비교적 작은 크기의 모형이지만 다양한 벤치마크에서 준수한 성능을 보였다. 연구목적으로 개발되었으며 거친 언어에 대한 전처리를 별도로 하지 않았다는 특징이 있다. 향후 모형의 크기를 100배 정도로 키울 계획이라고 한다. REST 기반 API를 제공하기 때문에 비교적 쉽게 활용이 가능하다.

4.8 KoBART

KoBART는 KoBERT, KoGPT2와 동일하게 SKT-AI에서 개발되었으며, BART를 40GB의 한국어 말뭉치에 대해 학습한 모형이다[33]. 학습 데이터는 한국어 위키 백과에서 5M개의 문장 그리고 뉴스, 책, 모두의 말뭉치 v1.0, 청와대 국민청원과 같은 다양한 말뭉치에서 0.27B개의 문장으로 구성했다. 토크나이저는 음절 BPE 토크나이저를 사용했고, 이모티콘과 이모지를 포함한 어휘사전의 크기는 30,000이다. BART의 목적이 생성 모형에 있다는 점을 생각하면 문서 요약, 질의 응답과 같은 분야가 주 활용분야가 될 것으로 생각된다.

Ⅴ. 결 론

사전학습 언어모델이 일반화되면서 자연어 처리의 활용도가 매우 높아졌다. 이전에는 자연어 처리 작업을 위해 해당 작업에 맞는 최소 수천 개 이상의 라벨이 있는 데이터를 확보해서 지도학습을 해야만 했다. 충분한 데이터를 확보하는 것도 어려운 일이었으나 거기에 지도학습을 위해 라벨을 붙이는 작업도 만만치 않았다. 또한 목적에 맞는 머신러닝 혹은 딥러닝 알고리즘을 구현하는 것도 어려운 일이었다. 그러나 사전학습 언어모델이 사용되면서 자연어 처리 작업이 훨씬 수월해졌다. 첫째 대량의 말뭉치를 대상으로 사전학습이 되어있기 때문에 적은 수의 데이터를 이용한 미세조정학습으로도 높은 성능을 기대할 수 있게 되었다. 둘째 사전학습 언어모델은 딥러닝 모형의 구조와 가중치 그리고 사전학습된 토크나이저 등 구현에 필요한 요소들이 함께 배포되기 때문에 라이브러리를 사용하여 쉽게 구현을 할 수 있다. 이로 인해 자연어 처리 작업에 소요되는 비용과 기간을 크게 단축시켰다. 셋째 사전학습 언어모델을 다양한 분야에 활용하는 예제와 벤치마크 데이터가 공개되어 있어, 목표로 하는 작업이 어떤 분야에 해당하는지 쉽게 식별하고 그에 맞게 데이터를 변환할 수 있게 되었으며, 최종적으로 목표에 맞는 적절한 모형을 구현할 수 있게 되었다.

그러나 사전학습 언어모델을 잘 활용하기 위해서는 언어모델의 동작 원리, 사전학습 언어모델을 구현한 인공신경망 모형에 대한 이해할 필요가 있으며, 나아가 다양한 트랜스포머 변형 모형에 대해 파악하고 각 모형의 장단점에 대해 알 필요가 있다. 본 논문은 이를 위해 언어모델과 사전학습 언어모델의 정의로부터 시작하여 사전학습 언어모델의 발전과정과 그 정점에 있는 트랜스포머 모형 그리고 다양한 트랜스포머 변형 모형에 대해 조사하고 정리하였다. 마지막으로 한국어 트랜스포머 변형 모형을 정리함으로써 연구자들이 보다 쉽게 사전학습 언어모델에 대해 이해하고 자연어 처리 작업에 활용할 수 있도록 돕고자 했다.

참 고 문 헌

[1] T. Lin et al., “A survey of transformers”, AI Open, 2022.

[2] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale.”, arXiv preprint arXiv:2010.11929, 2020.

[3] M. Chen, et al., “Generative pretraining from pixels”, International conference on machine learning. PMLR, 2020.

[4] C. Subakan et al., “Attention is all you need in speech separation”, ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021.

[5] H. Akbari et al., “Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text” Advances in Neural Information Processing Systems, Vol.34, pp.24206-24221, 2021.

[6] H. Li, “Language models: past, present, and future”, Communications of the ACM, Vol.65, No.7, pp56-63, 2022.

[7] A. Radford et al., “Language models are unsupervised multitask learners.”, OpenAI blog Vol.1, No.8, p.9, 2019.

[8] J. Devlin et al., “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805, 2018.

[9] T. Brown et al., “Language models are few-shot learners”, Advances in neural information processing systems, Vol.33, pp.1877-1901, 2020.

[10] S.J. Pan and Y. Qiang, “A survey on transfer learning”, IEEE Transactions on knowledge and data engineering, Vol.22, No.10, pp.1345-1359, 2009.

[11] Y. Bengio et al., “A neural probabilistic language model”, Advances in neural information processing systems, Vol.13, 2000.

[12] T. Mikolov et al., “Efficient estimation of word representations in vector space”, arXiv preprint arXiv:1301.3781. 2013.

[13] J. Sarzynska-Wawer et al., “Detecting formal thought disorder by deep contextualized word representations”, Psychiatry Research, Vol.304, p.114135, 2021.

[14] A. Vaswani et al., “Attention is all you need”, Advances in neural information processing systems, Vol.30, pp.5998-6008, 2017.

[15] J.L. Ba et al., “Layer normalization”, arXiv preprint arXiv:1607.06450, 2016.

[16] Y. Wu et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation”, arXiv preprint arXiv:1609.08144, 2016.

[17] A. Radford et al., “Improving language understanding by generative pre-training”, 2018.

[18] A. Wang et al., “GLUE: A multi-task benchmark and analysis platform for natural language understanding”, arXiv preprint arXiv:1804.07461, 2018.

[19] A. Wang et al., “Superglue: A stickier benchmark for general-purpose language understanding systems”, Advances in neural information processing systems, Vol.32, 2019.

[20] Y. Liu et al., “Roberta: A robustly optimized bert pretraining approach”, arXiv preprint arXiv:1907.11692, 2019.

[21] Z. Lan et al., “Albert: A lite bert for self-supervised learning of language representations”, arXiv preprint arXiv:1909.11942, 2019.

[22] K. Clark et al., “Electra: Pre-training text encoders as discriminators rather than generators”, arXiv preprint arXiv:2003.10555, 2020.

[23] M. Lewis et al., “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”, arXiv preprint arXiv:1910.13461, 2019.

[24] https://aiopen.etri.re.kr/service_dataset.php, 2019.

[25] https://github.com/SKTBrain/KoBERT, 2019.

[26] S. Lee et al., “Kr-bert: A small-scale korean-specific language model”, arXiv preprint arXiv:2008.03979, 2020.

[27] https://github.com/monologg/KoELECTRA, 2020.

[28] https://huggingface.co/xlm-roberta-base

[29] https://aida.kisti.re.kr/data/107ca6f3-ebcb-4a64-87d5-cea412b76daf, 2021.

[30] https://github.com/SKT-AI/KoGPT2, 2020.

[31] https://github.com/haven-jeon/kogpt2-chatbot, 2022.

[32] https://github.com/kakaobrain/kogpt, 2021.

[33] https://github.com/SKT-AI/KoBART, 2020.

저 자 소 개

박 상 언(Sangun Park)

・1992년 8월 : 한국과학기술원 전산학과(공학사)
・1999년 2월 : 한국과학기술원 경영공학과 (공학석사)
・2006년 8월 : 한국과학기술원 경영공학과 (공학박사)
・2007년 3월～현재 : 경기대학교 경영정보전공 교수

<관심분야> : 딥러닝, 텍스트마이닝, 머신러닝