JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 5, pp. 187-197, May 2020
https://doi.org/10.9708/jksci.2020.25.05.187

Self-Supervised Document Representation Method

Yeoil Yun*, Namgyu Kim**

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

[Abstract]

Recently, various methods of text embedding using deep learning algorithms have been proposed. Especially, the way of using pre-trained language model which uses tremendous amount of text data in training is mainly applied for embedding new text data. However, traditional pre-trained language model has some limitations that it is hard to understand unique context of new text data when the text has too many tokens. In this paper, we propose self-supervised learning-based fine tuning method for pre-trained language model to infer vectors of long-text. Also, we applied our method to news articles and classified them into categories and compared classification accuracy with traditional models. As a result, it was confirmed that the vector generated by the proposed model more accurately expresses the inherent characteristics of the document than the vectors generated by the traditional models.

▸ **Key words**: Deep Learning, Document Embedding, Pre-Trained Language Model, Self-Supervised Learning, Text Mining

[요 약]

최근 신경망 기반의 학습 알고리즘인 딥 러닝 기술의 발전으로 인해 텍스트의 문맥을 고려한 문서 임베딩 모델이 다양하게 고안되었으며, 특히 대량의 텍스트 데이터를 사용하여 학습을 수행한 사전 학습 언어 모델을 사용하여 분석 문서의 벡터를 추론하는 방식의 임베딩이 활발하게 연구되고 있다. 하지만 기존의 사전 학습 언어 모델을 사용하여 새로운 텍스트에 대한 임베딩을 수행할 경우 해당 텍스트가 가진 고유한 정보를 충분히 활용하지 못한다는 한계를 가지며, 이는 특히 텍스트가 가진 토큰의 수에 큰 영향을 받는 것으로 알려져 있다. 이에 본 연구에서는 다수의 토큰을 포함한 장문 텍스트의 정보를 최대한 활용하여 해당 텍스트의 벡터를 도출할 수 있는 자기 지도 학습 기반의 사전 학습 언어 모델 미세 조정 방법을 제안한다. 또한, 제안 방법론을 실제 뉴스 기사에 적용하여 문서 벡터를 도출하고 이를 활용하여 뉴스의 카테고리 분류 실험을 수행하는 외부적인 임베딩 평가를 수행함으로써, 제안 방법론과 기존 문서 임베딩 모델과의 성능을 평가하였다. 그 결과 제안 방법론을 통해 도출된 벡터가 텍스트의 고유 정보를 충분히 활용함으로써, 문서의 특성을 더욱 정확하게 표현할 수 있음을 확인하였다.

▸ **주제어**: 딥 러닝, 문서 임베딩, 사전 학습 언어 모델, 자기 지도 학습, 텍스트 마이닝

• First Author: Yeoil Yun, Corresponding Author: Namgyu Kim
*Yeoil Yun (yunyi94@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
**Namgyu Kim (ngkim@kookmin.ac.kr), School of Management Information Systems, Kookmin University
• Received: 2020. 04. 07, Revised: 2020. 04. 29, Accepted: 2020. 05. 05.

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

I. Introduction

최근 다양한 분야에서 텍스트 데이터를 분석에 활용하려는 시도가 증가함에 따라 자연어 처리(Natural Language Processing) 및 텍스트 마이닝(Text Mining)과 관련된 연구가 활발하게 수행되고 있다. 텍스트 마이닝의 전통적인 연구 분야로는 텍스트를 사전에 정의한 집단적인 특성에 따라 분류하는 텍스트 분류(Text Classification), 텍스트 내부에 존재하는 긍정, 부정과 같은 감성의 표현 및 정도를 예측하는 감성 분석(Sentiment Analysis), 텍스트 집합 내에 잠재된 공통의 주제를 도출하는 토픽 모델링(Topic Modeling) 등이 대표적이다. 최근에는 신경망(Neural Network) 기반의 딥 러닝(Deep Learning) 기술이 발전함에 따라 텍스트 데이터에 대해 다양한 딥 러닝 알고리즘을 적용하려는 시도가 증가하고 있다. 딥 러닝을 활용한 텍스트 분석의 예로, 텍스트로부터 핵심적인 내용을 추출하거나 새롭게 구성하는 텍스트 요약(Text Summarization), 텍스트 내에서 인명이나 기관명, 또는 지명과 같은 고유한 개체를 탐색하고 추출하는 개체명 인식(Named Entity Recognition), 주어진 질문에 대한 적절한 응답을 생성하거나 이미지나 동영상, 음성 파일과 같은 상이한 유형의 데이터로부터 특정 목적에 부합하는 문장 혹은 문단 등을 새롭게 구축하는 텍스트 생성(Text Generation) 등이 활발하게 연구되고 있다.

텍스트 데이터를 활용한 다양한 분석을 수행하기 위해서는 비정형의 텍스트를 정형적인 값으로 변환하는 구조화가 사전에 반드시 수행되어야 한다. 이처럼 텍스트를 특정 차원 공간 내의 벡터(Vector)로 정형화하는 작업을 임베딩(Embedding)이라 일컬으며, 임베딩을 수행하기 위해 통계학이나 기계 학습(Machine Learning)에 기반한 다양한 방법들이 전통적으로 활용되고 있다. 통계적인 기법에 기반한 기초적인 임베딩 방법으로는, 텍스트 집합 내에 존재하는 단어들의 등장 빈도나 등장 여부를 파악하고 이를 이용하여 개별 텍스트에 대한 벡터를 도출하는 단어 주머니 모델(Bag-of-Words Model)이 가장 대표적이다. 단어 주머니 모델은 단어의 등장 빈도나 등장 여부를 바탕으로 텍스트를 구조화할 수 있다는 장점을 갖지만, 개별 텍스트를 벡터로 표현하기 위해 텍스트 집합 내에 존재하는 모든 단어 수 만큼의 차원이 필요하다는 점에서 희소성(Sparsity)의 한계가 발생한다.

이러한 희소성 문제를 해결하기 위해 기계학습 기반의 밀집 벡터(Dense Vector) 생성을 위한 연구가 이루어졌으며, 이와 같은 방법들은 대체로 차원 축소(Dimensionality Reduction)를 통해 텍스트의 잠재 벡터(Latent Vector)를 생성하는 방법으로 구현되었다. 차원 축소를 통해 잠재 벡터를 도출하기 위해 사용되는 기법으로는 주성분 분석(Principal Component Analysis)이나 특이값 분해(Singular Value Decomposition), 음수 미포함 행렬 분해(Non-Negative Matrix Factorization) 등이 있으며, 주로 텍스트와 단어의 빈도 행렬을 다수의 행렬의 곱으로 표현한 뒤 잠재 행렬의 벡터를 추출하는 방식으로 차원 축소가 이루어진다. 차원 축소를 통해 생성된 밀집 벡터는 통계적 기법을 통해 추출된 희소 벡터에 비해 벡터 공간을 효율적으로 사용하며 개별 텍스트의 고유한 특성을 표현할 수 있다는 장점을 가진다. 하지만 차원 축소와 같은 기계 학습 기법을 적용한 임베딩 방법은 고정된 말뭉치(Corpus)로부터 텍스트와 단어의 빈도 행렬을 구축하고 잠재 행렬을 도출하기 때문에, 말뭉치에 존재하지 않는 새로운 텍스트에 대한 벡터를 도출하기 위해선 행렬 구축과 행렬 분해 연산을 다시 수행해야 한다는 한계가 존재한다.

이와 같은 기계 학습 기반의 임베딩 방법이 가진 문제를 해결하기 위해, 딥 러닝을 활용한 임베딩 방법이 최근 주목을 받고 있다. 딥 러닝 기반 임베딩 방법의 핵심은 신경망 구조를 구성하는 각 층(Layer) 간의 연결 가중치(Weight)를 학습하여 하나의 모델을 구축하는 것이며, 이와 같은 특징으로 인하여 모델 구축에 사용되지 않았던 새로운 텍스트에 대해서도 해당 텍스트가 가진 고유한 문맥(Context)을 파악하고 그 벡터를 추론(Inference)할 수 있다. 딥 러닝 기반의 텍스트 임베딩은 구조화의 단위에 따라 단어 임베딩(Word Embedding)이나 문서 임베딩(Document Embedding) 등으로 구분될 수 있다. 단어 임베딩은 특정 단어의 주변에 등장한 단어들을 바탕으로 해당 단어의 고유한 벡터를 도출하는 모델로서, 대표적인 단어 임베딩 모델로는 word2vec이나 glove, fasttext 등이 존재한다[1-3]. 문서 임베딩은 문서 내에 존재하는 단어들의 문맥을 파악하고 이를 종합하여 문서를 하나의 벡터로 종합하는 모델로서, doc2vec과 같은 모델이 대표적으로 사용되고 있다[4]. 하지만 이러한 모델 역시 모델 학습에 사용된 텍스트 내에 등장하지 않은 어휘(Out-of-Vocabulary)에 대한 의미를 명확하게 추론하기 어렵다는 한계가 존재한다.

이에 최근에는 대량의 텍스트 데이터를 학습에 사용한 사전 학습 언어 모델(Pre-Trained Language Model)을 활용하여 분석에 사용할 텍스트의 벡터를 추론하는 방식의 임베딩이 주로 이루어지고 있다. 사전 학습 언어 모델은 기존의 딥 러닝 기반 임베딩 모델과는 달리, 모델을 통해 방대한 학습 데이터로부터 추론된 벡터를 분석 목적에 부합하도록 조정하는 추가적인 학습이 가능하다는 특징을

가진다. 이러한 추가적인 학습을 미세 조정(Fine-Tuning)이라고 하며, 사전 학습 모델을 통한 추론과 그 결과에 대한 미세 조정을 수행하는 전반적인 분석 과정을 통틀어 전이 학습(Transfer Learning)이라고 일컫는다. 전이 학습의 개념은 사전 학습 모델이 딥 러닝 알고리즘의 난제인 학습을 위해 필요한 충분한 데이터의 확보와 방대한 학습 시간의 문제에 대한 해결 방안을 제시하였으며, 이에 따라 텍스트 데이터를 활용하는 다양한 분야에서도 사전 학습 언어 모델을 활용한 분석 시도가 활발하게 이루어지고 있다. 사전 학습 언어 모델의 대표적인 예로는 ELMo(Embeddings from Language Model)와 BERT(Bidirectional Encoder Representations from Transformer), XLNet(Extra Long Network) 등이 있으며[5-7], 이러한 사전 학습 언어 모델에 대해 미세 조정을 수행한 RoBERTa(Robustly Optimized BERT)와 ALBERT(A Lite BERT), DistilBERT(Distilled BERT) 등이 널리 사용된다[8-10]. 하지만 사전 학습 언어 모델의 추론은 학습에 사용되지 않았던 새로운 텍스트가 보유한 정보를 충분히 활용하지 못하는 경우, 해당 텍스트의 고유한 특성을 온전히 추출할 수 없다는 한계가 존재한다. 이러한 한계는 추론하고자 하는 텍스트가 포함하고 있는 토큰(Token)의 수에 따라 크게 영향을 받으며, 토큰의 수가 증가할수록 벡터를 추론하는 과정에서 나타나는 정보의 손실은 증가하게 된다(Fig. 1).

Fig. 1. Limitations of Traditional BERT-base Document Embedding

<Fig. 1>은 k개의 토큰을 가진 가상의 텍스트에 대한 벡터를 추론할 때, 세 개의 대표적 모델 각각이 사용하는 토큰의 정보량을 표현한 그림이다. <Fig. 1>의 (a) Original BERT는 대표적인 사전 학습 언어 모델인 BERT를 사용하여 벡터를 추론하는 과정으로, BERT는 추론을 위해 한정된 개수의 토큰만을 사용함을 알 수 있다. 그림에서 (a) 모델은 전체 k개의 토큰을 가진 텍스트 내에서 512번째 토큰인 'has'까지의 정보만을 사용하여 해당 텍스트의 벡터를 추론하며, 이는 텍스트의 길이가 길어질수록 손실되는 정보량이 증가함을 의미한다. 이러한 한계를 극복하고 텍스트의 모든 토큰을 사용하여 텍스트의 시퀀스(Sequence)를 참조한 벡터를 추론하기 위해, (b) Traditional Fine-Tuned BERT와 같이 기존의 BERT 모델에 순환신경망(Recurrent Neural Network) 기반의 딥 러닝 알고리즘을 사용하여 미세 조정을 수행한 모델이 활용되고 있다. 하지만 (b)와 같은 순환신경망 기반의 미세 조정 모델은 순환신경망 모델의 가중치 손실(Gradient Vanishing) 문제로 인하여 텍스트의 시퀀스가

길어질수록 이전에 등장한 토큰의 정보량이 감소하는 경향을 보인다. 즉 <Fig. 1>의 (b)에서 'BERT', 'has', 'its'와 같이 앞부분에 등장한 토큰이 'attention', 'weights'와 같이 마지막에 등장한 토큰에 비해 벡터를 추론하는 과정에서 낮은 수준의 정보를 제공함을 확인할 수 있다. 한편 <Fig. 1>의 (c) Self-Supervised Representation은 텍스트의 벡터 추론을 위해 k개의 모든 토큰을 균등하게 활용하여 해당 텍스트의 특성을 충분히 반영한 벡터를 추출할 수 있음을 보여준다. 이러한 방식은 특히 다수의 토큰을 포함한 장문의 텍스트를 고정된 공간에 벡터로 표현할 때, 텍스트 전체에 걸친 고유의 정보를 고르게 포함하여 정확하게 나타낼 수 있음을 의미한다.

이에 본 연구에서는 문서의 특성을 최대한 정확하게 반영하여 벡터를 추출할 수 있는 자기 지도 학습(Self-Supervised Learning) 기반의 BERT 미세 조정 모델을 제안한다. 자기 지도 학습은 모델이 분석 데이터에 대한 레이블(Label)을 자체적으로 설정하고 학습을 진행하는 비지도 학습(Unsupervised Learning)의 일종으로, 본 연구에선 자기지도 학습의 대표적인 모델인 오토인코더(AutoEncoder)를 활용하여 문서의 벡터를 추출하는 방법을 소개한다. 구체적으로 본 논문에서 제안하는 방법론은 (i) 문서를 고정된 수의 토큰을 가진 복수 개의 세그먼트(Segment)로 분할하고, (ii) 사전 학습 언어 모델인 BERT를 사용하여 분할된 세그먼트에 대한 벡터를 추론한 뒤 단일 문서의 모든 세그먼트 벡터를 나열하여 (세그먼트) × (벡터)로 구성된 2차원 행렬로 재구축한 뒤, (iii) 오토인코더를 활용하여 각 문서의 특성을 최대한 반영한 잠재 문서 벡터(Latent Document Vector)를 도출하는 과정으로 구성된다. 또한 본 연구에서 제안한 방법을 통해 도출한 문서 벡터와 기존의 문서 임베딩 방법 및 BERT의 미세 조정 모델을 통해 도출한 문서 벡터를 사용한 문서 분류 실험의 정확도를 비교하여 제안 방법의 우수성을 평가한다.

II. Related Research

1. Pre-Trained Language Models and its Fine-Tuned Model

사전 학습 언어 모델은 대량의 말뭉치를 사용하여 개별 텍스트의 문맥을 파악하는 신경망 기반의 학습 모델을 의미한다. 사전 학습 언어 모델은 기존의 임베딩 모델이 데이터의 부족으로 인해 단어나 문장 혹은 문서에 대한 의미를 충분히 파악하지 못한다는 한계를 극복하기 위해 등장하였다. 주로 뉴스 기사나 위키피디아(Wikipedia)의 텍스트 등이 모델 구축에 사용되며, 이를 통해 모델은 텍스트 내에 포함된 단어나 문장 개체에 대한 일반적인 의미를 파악할 수 있다. 이러한 과정을 통해 사전 학습 언어 모델은 학습에 사용되지 않은 새로운 텍스트에 대한 의미를 기존의 신경망 기반 임베딩 모델보다 정확하게 추론하게 된다.

사전 학습 언어 모델의 대표적인 예로는 ELMo, BERT 그리고 XLNet 등이 존재한다. 우선 ELMo는 단어의 의미를 추론하는 학습을 진행하는 사전 학습 언어 모델로, 양방향 순환신경망 학습을 진행하여 텍스트의 문맥을 파악하고 텍스트 내에 존재하는 단어의 벡터를 추론한다. 구체적으로 ELMo는 음절(Character) 기반 합성곱 신경망(Convolutional Neural Network)을 활용하여 개별 단어의 특성을 우선적으로 추출하고, 이후 순환신경망의 일종인 양방향 장·단기 메모리(Long-Short Term Memory) 모델을 활용하여 텍스트 전체에 대한 문맥을 파악하는 학습을 수행한다. 하지만 ELMo는 순환신경망의 가중치 손실 문제로 인하여 텍스트가 길어질수록 그 문맥을 명확하게 추론하기 어렵다는 한계가 존재한다.

이러한 한계를 극복하기 위해 어텐션 메커니즘(Attention Mechanism) 기반의 BERT가 고안이 되었다. BERT는 셀프 어텐션(Self-Attention) 메커니즘을 활용한 트랜스포머(Transformer) 기반의 학습을 수행한다[11]. 트랜스포머는 시퀀스가 존재하는 데이터에 대해 목적 데이터(Target Data)와 입력 데이터(Input Data)를 동일하게 설정하고, 데이터 내에 존재하는 개별 시퀀스 데이터 간의 연결 정도를 파악하는 학습을 수행하는 모델이다. BERT에 사용된 트랜스포머는 텍스트 내부에 존재하는 모든 토큰 간의 연결 정도를 측정하고, 이를 통해 전체 텍스트에 대한 문맥을 파악하는 학습을 수행한다. 또한 BERT 내부의 트랜스포머는 '<CLS>'라는 특별한 토큰을 가지고 있으며, 이 토큰은 트랜스포머 학습 과정에서 입력 텍스트에 대한 전반적인 문맥을 파악하는 역할을 담당한다[12]. BERT는 이러한 학습 과정으로 인해 문장 단위의 텍스트 벡터도 추론이 가능하다는 특징을 갖지만, 동시에 학습에 사용되는 토큰의 수에 제한을 두고 있다는 한계를 갖는다. BERT의 이러한 한계를 해결하기 위해 Transformer-XL(Extra Long)과 같이 BERT의 개량된 모델이 제안된 바 있으며[13], 이를 다시 개선한 XLNet이 등장하게 되었다. XLNet은 BERT와 동일하게 트랜스포머 기반의 학습을 수행하지만, 텍스트를 구간으로 분할하여 독립적인 트랜스포머 학습을 진행한다는 점과 입력 텍스트 내 토큰의 순서를 임의로 조정하는 퍼뮤테이션(Permutation) 연산을 수행한다는 점에서 길이가 긴 텍스트에 대한 문맥을 파악하기 용이하다.

한편, 사전 학습 언어 모델을 활용하여 분석 의도에 부합한 텍스트 벡터를 새롭게 추론하려는 시도가 활발하게 이루어지고 있으며, 이러한 시도는 주로 기존의 사전 학습 언어 모델을 미세 조정하는 방식으로 이루어지고 있다. 사전 학습 언어 모델의 미세 조정 방법을 제안한 연구로는 사전 학습 언어 모델을 활용한 텍스트 분류의 정확도 향상 방안 연구[14], BERT를 활용하여 문서 벡터를 추론하고 카테고리 분류를 수행한 연구[15, 16] 등이 존재한다. 또한, 임베딩의 관점에서 텍스트의 고유한 특성을 더욱 잘 반영할 수 있도록 사전 학습 언어 모델을 미세 조정을 수행한 연구도 존재한다. 구체적 예로는 문장 단위의 텍스트에 대한 문맥을 더욱 잘 파악하기 위해 미세 조정된 BERT 모델을 제안한 연구, 문서가 내포한 정보를 최대한 반영하여 벡터를 추론할 수 있는 미세 조정 모델을 제안한 연구 등을 들 수 있다[17, 18].

2. Self-Supervised Learning with Text Data

자기 지도 학습은 목적 데이터의 값을 자체적으로 설정하고 학습을 진행하는 비지도 학습 방법의 일종이다. 자기 지도 학습은 데이터에 대한 레이블이 존재하지 않을 경우나 이를 할당하기 어려운 경우에도 활용될 수 있다는 점에서 최근 주목을 받고 있다. 초기의 자기 지도 학습은 합성곱 신경망과 같이 레이블을 필요로 하는 지도 학습 모델에 대해, 입력 데이터로부터 의사 레이블(Pseudo Label)을 자체적으로 생성하고 학습을 진행하는 방식으로 이루어졌다[19, 20]. 의사 레이블을 생성하는 방식의 자기 지도 학습은 주로 입력 데이터에 대한 차원 축소를 수행하여 추출된 벡터를 모델의 목적 데이터로 설정하는 방식으로 진행되었다[21].

최근에는 이러한 의사 레이블 생성 과정 자체를 생략하려는 다양한 노력이 이루어지고 있으며, 그 결과 오토인코더라는 방법을 적용한 연구가 활발히 이루어지고 있다[22]. 오토인코더는 목적 데이터를 입력 데이터와 동일하게 설정하여 학습을 진행하는 신경망 기반의 모델이다. 오토인코더는 입력층(Input Layer)과 출력층(Output Layer) 사이에 존재하는 모든 은닉층(Hidden Layer)이 병목층(Bottleneck Layer)이라 불리는 은닉층을 기준으로 대칭의 형태를 이루는 신경망 모델이다. 병목층 이전의 은닉층에서는 입력 데이터에 대한 특성을 추출하고 병목층 이후의 은닉층에서는 추출한 특성을 바탕으로 다시 입력 데이터와 최대한 유사하게 출력 데이터를 생성하도록 구성되어있다. 오토인코더를 텍스트 데이터에 활용한 연구는 주로 학습 과정에서 병목층에 남게 되는 입력 데이터의 특성을 추출하여, 이를 입력 데이터에 대한 임베딩 결과로 활용한다. 구체적인 예로 텍스트의 임베딩을 위해 합성곱 신경망 또는 순환신경망 기반의 은닉층으로 오토인코더를 구성한 뒤, 병목층의 벡터를 추출하는 연구[23, 24], 다수의 오토인코더를 구성하여 텍스트의 특성을 추출하는 연구[25, 26] 등을 들 수 있다.

3. Evaluation of Embedding Models

한편 텍스트 데이터에 대한 임베딩 모델을 제안하는 연구가 지속적으로 제안됨에 따라, 임베딩 모델 혹은 그 결과에 대한 성능을 평가하고자 하는 시도 역시 지속적으로 이루어지고 있다. 임베딩 모델에 대한 평가는 크게 내부적인(Intrinsic) 방법과 외부적인(Extrinsic) 방법으로 나뉠 수 있다[27]. 우선, 내부적인 방법은 분석가의 판단에 따라 임베딩 모델의 우수성을 평가하는 방식을 의미하며, 외부적인 방법은 임베딩 결과로 도출된 텍스트의 벡터를 다양한 분석 모델에 적용하여 모델의 성능을 평가하는 방식으로 수행된다. 구체적으로 내부적인 방법은 임베딩 모델을 통해 도출된 단어나 문서의 벡터 간 코사인 유사도(Cosine Similarity) 측정이나 벡터 연산(Vector Analogy) 등을 수행하며, 이를 통해 분석가가 실제로 유사하다고 판단하는 텍스트 개체 간의 관계가 모델을 통해 도출된 벡터 간에도 유사한 관계로 파악되는지 확인함으로써 모델의 성능을 평가한다[28, 29]. 이와 같은 평가 방법은 소수 인원으로 구성된 분석가 집단 내부에서 평가를 진행하는 인-하우스(In-House) 방법과[30, 31], 크라우드소싱(Crowdsourcing) 기반의 다수의 의견을 반영한 평가를 진행하는 방식으로 다시 구분될 수 있다[32].

반면 외부적인 평가 방법은 모델의 활용성에 초점을 맞추고 있으며, 주로 도출된 텍스트 벡터를 입력 데이터로 사용한 텍스트 분류 모델, 개체명 인식 모델, 정보 검색(Information Retrieval) 성능, 감성 분석 모델의 성능을 평가하는 방식으로 진행한다[33, 34]. 외부적인 평가 방법을 지향하는 연구는 텍스트 개체 간의 유사도 비교나 벡터 연산과 같은 내부적인 방법의 경우 단일 모델 내에서만 평가가 이루어지고 있기 때문에, 다양한 임베딩 모델 간의 비교가 어렵다는 점을 한계로 지적하고 있다[35].

III. Proposed Method

1. Research Process

3장에서는 본 논문에서 제안하는 자기 지도 학습 기반 사전 학습 언어 모델의 미세 조정 방법 원리를 가상의 예

와 함께 설명한다. 제안 방법론의 전체적인 과정은 <Fig. 2>와 같다.

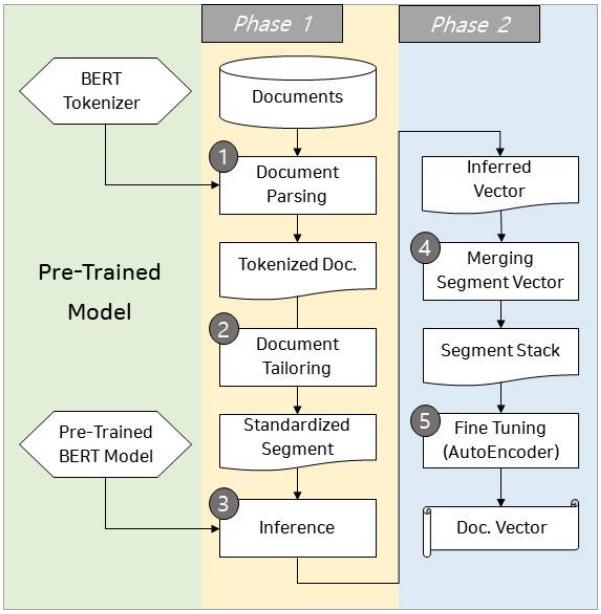

Fig. 2. Overall Research Process

<Fig. 2>는 문서 임베딩 과정에서 문서 내에 존재하는 토큰의 정보를 충분히 활용하는 제안 방법론의 개요를 나타낸다. 제안 방법론은 문서를 복수의 세그먼트로 분리한 뒤 사전 학습 언어 모델을 사용하여 세그먼트의 벡터를 추론하는 Phase 1, 그리고 도출된 벡터들을 문서 단위로 종합하여 미세 조정을 수행하는 Phase 2의 두 단계로 구성된다. 구체적으로 Phase 1은 임베딩의 대상이 되는 문서를 (1) 사전 학습 언어 모델의 전용 토크나이저(Tokenizer)를 사용하여 파싱(Parsing)하고, (2) 이를 동일한 규격의 세그먼트(Standardized Segment), 즉 고정된 수의 토큰을 가진 복수의 세그먼트로 분할한 뒤, (3) 사전 학습 언어 모델을 사용하여 각 세그먼트에 대한 벡터를 추론하는 작업을 수행한다. 이후 Phase 2에서는 (4) 추론을 통해 도출된 세그먼트 벡터의 집합을 문서 단위로 종합하여 시퀀스를 가진 행렬로 구성하고, (5) 자기 지도 학습의 일환인 오토인코더를 활용하여 미세 조정을 수행하는 과정을 통해 최종적으로 문서의 고유한 특성을 충실하게 반영한 문서 벡터를 도출하게 된다.

각 과정에 대한 구체적인 작동 원리는 본 장의 이후 절부터 가상의 예시와 함께 설명하며, 실제 데이터를 적용한 제안 방법론의 성능 평가 결과는 4장에서 소개한다.

2. Document Parsing and Tailoring

다음으로 <Fig. 2>의 BERT 전용 토크나이저를 사용한 문서의 토크나이징(단계 1), 그리고 문서를 고정된 수의 토큰을 가진 복수의 세그먼트로 분할하는 과정(단계 2)을 소개한다. 사전 학습 언어 모델을 활용하여 텍스트의 벡터를 추론하기 위해서는 텍스트를 모델에 입력 가능한 형태로 변환하는 작업이 필요하며, 이를 위해 BERT와 같은 사전 학습 언어 모델은 텍스트를 토큰의 집합으로 분할하는 전용 토크나이저를 제공한다. 토크나이저를 사용하여 문서에 대한 파싱을 수행한 결과의 예가 <Fig. 3>에 나타나있다.

Deep learning also known as deep structured learning or differential programming is ···
... and symbolic, while the biological brain of most living organisms is dynamic and analog"

deep	learn	_ing	also	know	_n
as	deep	structure	_ed	learn	_ing
or	different	_i	_al	programming	is
···					
and	symbol	_ic	while	the	biologic
_al	brain	of	most	live	_ing
organism	_s	is	dynamic	and	analog

Fig. 3. Example of Document Parsing

<Fig. 3>은 특정 문서를 BERT 토크나이저로 파싱한 가상의 결과를 나타낸다. 문서는 고유한 의미를 지닌 토큰들의 집합으로 표현되며, 사전 학습 언어 모델은 토큰의 집합으로 표현된 문서로부터 토큰의 등장 순서와 핵심 단어 등의 정보를 파악함으로써 문서의 문맥을 파악하고 이를 반영하여 해당 문서의 벡터를 추론할 수 있다.

하지만, 일반적으로 문서 내에 포함된 토큰의 수는 사전 학습 언어 모델이 처리 가능한 수준을 초과하기 때문에, 많은 경우 일부 토큰에 대한 정보를 활용하지 못한 채로 벡터의 추론이 이루어진다는 한계가 존재한다. 이에 본 연구에서는 문서 내 모든 토큰의 정보를 충분히 활용하여 문서 벡터를 도출하는 방안을 제안하며, 이를 위해 우선 문서를 재단하여 동일한 규격을 갖는 복수의 세그먼트로 분할하는 작업을 수행한다. <Fig. 4>는 토큰 단위로 표현된 단일 문서를 복수의 세그먼트로 분할한 가상의 결과를 나타낸다. <Fig. 4>에서 분할 결과로 생성된 N개의 세그먼트는 각각 동일한 수인 9개의 토큰으로 구성됨을 확인할 수 있다.

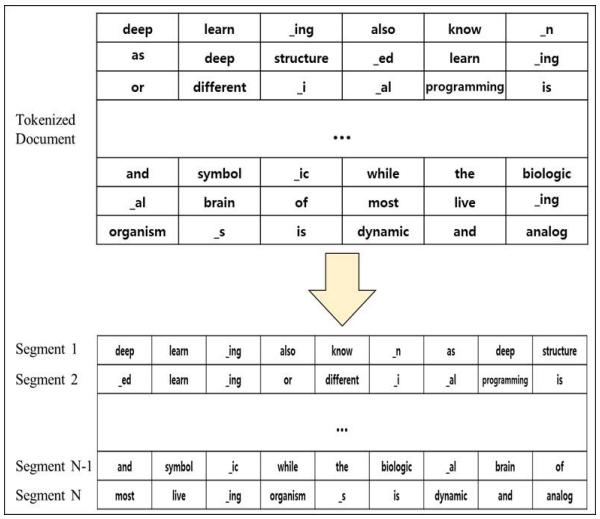

Fig. 4. Example of Document Tailing

3. Segment Inference and Stack Generation

이번 절에서는 <Fig. 2>의 (단계 3)과 (단계 4), 즉 BERT 모델을 통해 세그먼트 벡터를 추론하고 이를 문서 단위로 병합하여 문서의 시퀀스 행렬을 구축하는 과정을 소개한다. 우선, BERT 모델을 사용하여 분할된 세그먼트의 벡터를 추론하는 과정은 다음과 같다. BERT는 입력 데이터에 대해 도출되는 벡터의 차원 수를 768로 고정하여 구축하였으므로, 이미 학습된 BERT 모델에 세그먼트 단위로 분할한 데이터를 입력으로 사용하여 추론되는 각 세그먼트 벡터 역시 768차원으로 표현된다. 이후, 동일 문서에 대한 모든 세그먼트 벡터를 등장 순서에 따라 열 방향(Vertical)으로 쌓아, 문서에 대한 시퀀스 정보를 포함한 행렬을 구축한다(Fig. 5).

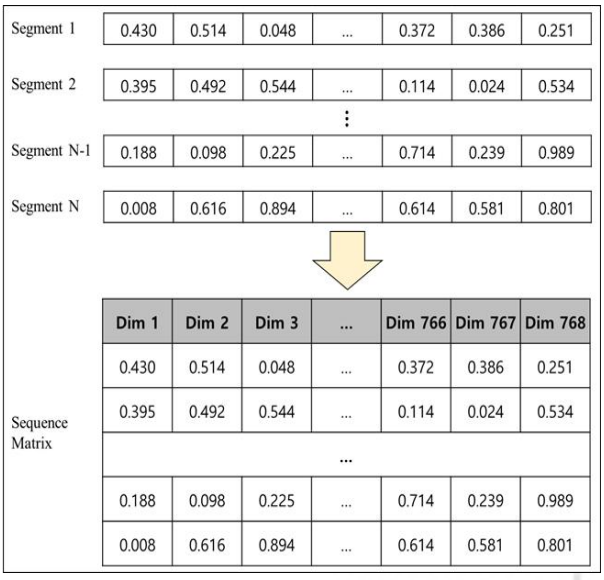

Fig. 5. Example of Sequence Matrix Generation

4. Fine-Tuning with Autoencoder

마지막으로 <Fig. 2>의 마지막 모듈인 (단계 5)에 해당하는 과정, 즉 오토인코더를 통해 미세 조정을 수행하고 그 결과로 문서의 벡터를 추출하는 과정을 소개한다. 본 과정에서는 문서 벡터를 도출하기 위해 2차원으로 구성된 문서의 시퀀스 행렬을 오토인코더 모델의 입력 데이터와 목적 데이터로 동시에 설정하여 학습을 진행한다.

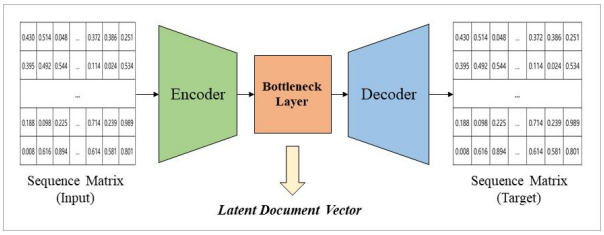

Fig. 6. Example of Autoencoder Training

<Fig. 6>은 오토인코더를 활용하여 시퀀스 행렬에 대한 미세 조정을 진행하는 학습 과정을 나타낸다. 오토인코더는 인코더(Encoder)-병목층-디코더(Decoder)의 구조로 이루어져 있으며, 인코더에서는 입력 시퀀스 행렬에 대한 특성을 학습하는 과정을, 그리고 디코더에서는 특성화된 벡터를 기반으로 다시 목적 시퀀스 행렬을 복원하는 과정을 수행한다. 병목층은 인코더와 디코더의 중간에 존재하는 층으로, 입력 데이터의 특성이 가장 높은 수준으로 응축되어 저장된다. 병목층에 저장된 벡터를 입력 데이터에 대한 잠재 벡터라고 일컬으며, 본 연구에서는 문서 임베딩의 최종 결과, 즉 입력 문서의 특성을 충분히 반영한 문서 벡터로 이 잠재 벡터를 채택한다.

IV. Experiment

1. Experiment Overview

4장에서는 앞서 소개한 제안 방법론을 실제 데이터에 적용한 결과 및 제안 방법론의 성능 분석 결과를 소개한다. 실험을 위해 2019.11.01.부터 2020.01.31.까지 작성된 'IT', '정치', '사회', '세계', '경제', '생활문화'의 6개 카테고리를 가진 뉴스 기사를 수집하였다. 이후, 분석에 사용될 기사가 보유한 세그먼트 수의 차이를 최소화하기 위해 길이가 3,000자에서 4,000자 이내에 존재하는 기사를 카테고리별로 2,000건씩 추출하여 총 12,000건의 기사를 사용하였으며, 실험 환경은 Python 3.7을 통해 구축하였다.

2. Parsing and Tailoring News Articles

우선, 12,000건의 뉴스 기사를 BERT 토크나이저를 사용하여 파싱한 뒤, 이를 동일 규격을 갖는 세그먼트 집합으로 분할한 결과를 제시한다. <Table 1>은 임의의 IT 뉴스 기사를 BERT 토크나이저를 사용하여 파싱한 결과의 예이며, 해당 기사는 총 1,588개의 토큰으로 구성됨을 확인할 수 있다.

Table 1. Example of Tokenized Article

Sequence	Token	Sequence	Token
1	PC	…	…
2	시장은	1584	가격이
3	전반	1585	달라
4	적으로	1586	질
5	감소	1587	수
6	하는	1588	있다

이후, 토큰 단위로 표현된 뉴스 기사를 동일한 토큰 수를 가진 세그먼트들의 집합으로 재구성하였으며, 본 연구에선 모델이 입력 세그먼트에 대한 문맥을 충분히 학습할 수 있도록 세그먼트 당 토큰 수를 250의 임의의 값으로 설정하였다. <Table 2>는 <Table 1>의 기사를 총 7개의 세그먼트로 분할한 결과를 보여준다.

Table 2. Example of Standardized Segment

Segment 1		Segment 2		…	Segment 7	
Seq.	Token	Seq.	Token	…	Seq.	Token
1	PC	251	제품	…	…	…
2	시장은	252	이다	…	1587	수
3	전반	253	17	…	1588	있다
…	…	…	…	…	1589	No-Op
249	어울리	499	때문에	…	…	…
250	는	500	주변	…	1750	No-Op

<Table 2>는 1,588개의 토큰을 보유한 뉴스 기사를 각각 250개의 토큰으로 구성된 7개의 세그먼트로 분할한 결과를 보여준다. 한편, BERT를 사용하여 벡터를 추론하기 위해서는 입력 시퀀스가 모두 동일한 개수의 토큰을 보유하고 있어야 하며, 본 연구에서는 이를 위해 뉴스 기사의 마지막 세그먼트의 일부를 No-Op(No-Operation) 토큰으로 할당하여 BERT 내부에서 입력 텍스트에 대한 문맥을 파악하는 연산에 사용하지 않도록 설정하였다. 이에 따라 <Table 2>에서 Segment 7의 Sequence 1,589부터의 토큰은 “No-Op” 값이 할당되었다.

3. Sequence Matrix Generation

다음으로 BERT를 사용하여 분할된 세그먼트에 대한 벡터를 추론하고 이를 다시 문서 단위로 병합하여, 시퀀스 행렬, 즉 세그먼트 스택을 구축한 결과를 소개한다.

Table 3. Example of Sequence Matrix

Dim 1	Dim 2	Dim 3	…	Dim 766	Dim 767	Dim 768
-0.050	-0.060	-0.558	…	0.037	0.015	-0.043
-0.046	-0.039	-0.602	…	0.002	-0.009	0.012
0.011	-0.054	-0.554	…	0.059	-0.045	0.018
-0.003	-0.062	-0.505	…	0.008	-0.048	0.004
-0.053	-0.061	-0.671	…	-0.016	-0.034	-0.033
-0.061	-0.040	-0.617	…	0.032	0.001	-0.051
0.064	0.052	0.057	…	0.069	-0.012	0.037

<Table 3>은 BERT를 사용하여 <Table 2>의 7개 입력 세그먼트에 대응되는 출력 세그먼트 벡터를 추론하고, 이를 등장 순서에 따라 열 방향으로 쌓아 생성한 시퀀스 행렬을 나타낸다. 각 세그먼트는 BERT를 통해 768차원의 벡터로 임베딩되며, 이를 쌓아 뉴스 기사에 대한 7 × 768의 시퀀스 행렬이 도출되였음을 확인할 수 있다.

4. Fine-Tuning and Performance Evaluation

4절에서는 앞서 도출한 시퀀스 행렬에 대한 미세 조정을 수행하여 각 뉴스 기사의 문서 벡터를 도출하고, 기존 문서 임베딩 방법과의 비교를 통해 제안 방법론의 성능을 분석한 결과를 소개한다. 성능 비교 실험의 전체 프로세스는 <Fig. 7>과 같다.

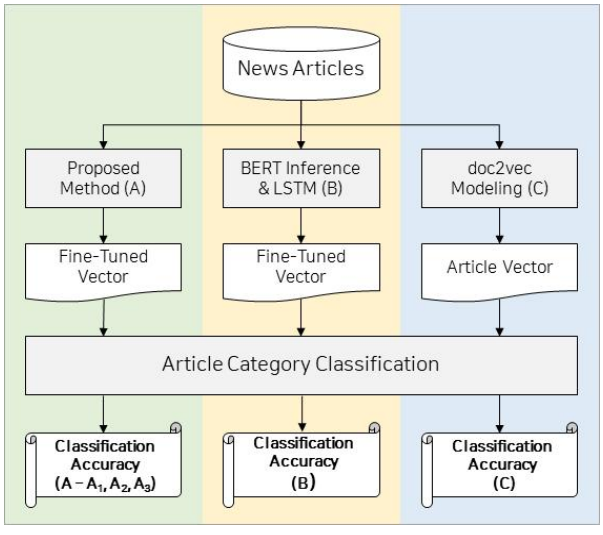

Fig. 7. Overall Process of Performance Evaluation

<Fig. 7>의 (A)는 제안 방법론을 통해 도출한 문서 벡터를 평가하는 과정이다. 구체적으로 오토인코더 기반 미세 조정을 수행하여 뉴스 기사에 대한 단일 벡터를 추출하고, 이를 사용하여 뉴스 기사의 카테고리를 분류한 후 분류 정확도(Classification Accuracy)를 측정한다. <Fig. 7>의

(B)와 (C)는 제안 방법론과의 비교를 위해 사용한 기존의 문서 임베딩 방법으로, (B)는 장·단기 메모리를 활용한 순환신경망 기반의 BERT 미세 조정을 통한 문서 벡터를 추출한 방법을, (C)는 doc2vec을 통해 문서 벡터를 추출한 방법을 의미한다. 한편 본 연구에서는 시퀀스 행렬의 미세 조정에 사용한 오토인코더 내부의 연결층을 각각 (A_1) 합성곱 신경망, (A_2) 양방향 장·단기 메모리, (A_3) 합성곱 신경망 + 양방향 장·단기 메모리로 구성하였으며, 임베딩 차원은 (A), (B), (C) 모두에 대해 100차원과 768차원으로 설정하여 모델별로 두 개의 독립적인 벡터를 추출하였다.

뉴스 기사의 카테고리 분류를 위해 전체 12,000건의 기사를 학습용 10,800건과 카테고리 예측용 1,200건으로 분할하였으며, 카테고리별 뉴스 기사의 비율은 카테고리별로 동일하게 설정하였다. 분류 기법은 (A), (B), (C) 세 가지 모델에 모두 동일하게 랜덤 포레스트(Random Forest)를 사용하였으며, 1,200건의 데이터에 대한 세 가지 모델 각각의 분류 정확도를 측정하였다. 실험 결과는 <Table 4>에 요약되어 있으며, 임베딩 차원을 768과 100으로 설정한 두 실험의 결과가 각각 <Fig. 8>과 <Fig. 9>에 나타나있다.

Table 4. Accuracy Comparison

#Dim	(A) Proposed Method			(B) BERT + LSTM	(C) doc2vec
	(A_1) AutoEncoder withCNN	(A_2) AutoEncoder withRNN	(A_3) AutoEncoder withCNN +RNN		
786	74.17%	72.08%	74.83%	60.08%	62.00%
100	72.25%	70.83%	72.75%	58.25%	60.00%

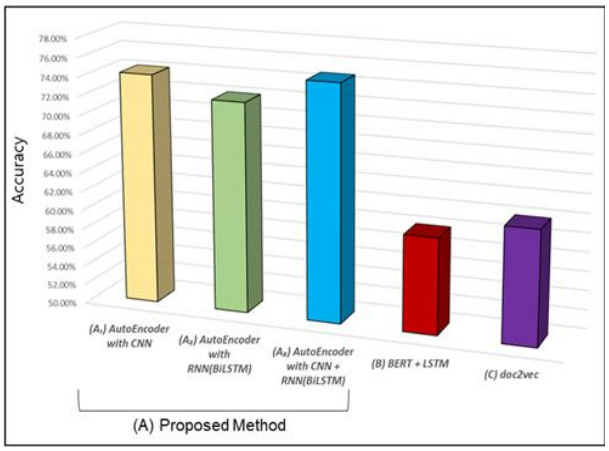

Fig. 8. Accuracy Comparison (#Dim = 768)

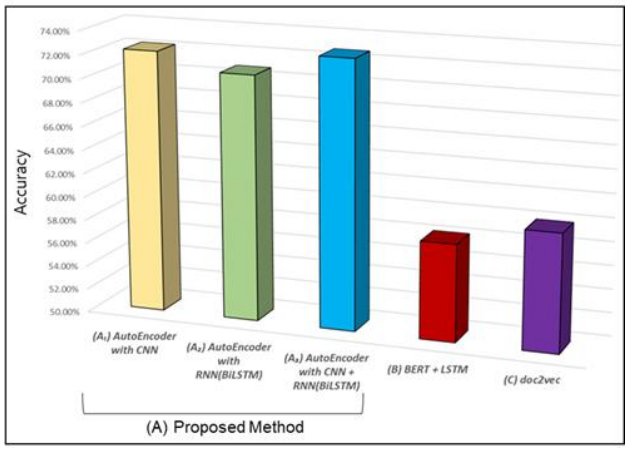

Fig. 9. Accuracy Comparison (#Dim = 100)

<Fig. 8>과 <Fig. 9>에서 제안 방법론인 A_1, A_2, 그리고 A_3의 분류 정확도가 기존의 문서 임베딩 방법인 BERT+LSTM이나 doc2vec에 비해 우수한 분류 정확도를 나타냄을 확인할 수 있다. 특히 제안 방법론의 세 가지 구현 중 오토인코더 내부의 연결층으로 합성곱 신경망과 순환 신경망을 모두 활용한 (A_3) 모델의 분류 정확도가 768차원의 경우 약 74.83%, 100차원의 경우에는 약 72.75%로 분류 정확도가 가장 높게 나타났다. 차원 수의 변화는 분류 정확도에 크게 영향을 주지는 않았지만, 768차원을 사용한 경우의 분류 정확도가 100차원을 사용한 경우에 비해 전체 모델에서 근소하게 높게 나타났다.

이상 본 연구에서 제안한 문서 임베딩 방법론의 우수성을 평가하기 위해 외부적인 방법, 즉 임베딩 결과로 도출된 문서 벡터를 문서 분류에 적용하여 임베딩 모델의 성능을 평가한 실험 결과를 요약하였다. 실험 결과 본 연구에서 제안한 자기 지도 학습 기반의 임베딩 방법이 기존의 임베딩 방법에 비해 문서 고유의 특성을 충분히 반영하는 벡터를 더욱 정확하게 생성함을 확인하였다.

V. Conclusions

텍스트 임베딩은 텍스트 분석을 위해 반드시 수행해야 하는 과정이며, 텍스트 데이터를 활용한 다양한 연구와 활용사례가 소개됨에 따라 텍스트 임베딩 기법을 제안하는 연구 역시 다양하게 수행되고 있다. 최근에는 신경망 학습 기반의 딥 러닝 알고리즘을 활용하여 텍스트의 고유한 문맥을 벡터로 표현하고자 하는 시도가 활발히 이루어지고 있으며, 대량의 말뭉치 데이터를 사용하여 학습을 진행한 사전 학습 언어 모델의 등장으로 텍스트의 벡터를 추론하는 임베

딩 방법이 주목을 받고 있다. 또한, 사전 학습 언어 모델을 통해 추론된 텍스트의 벡터를 분석 의도에 부합하게 변환하는 미세 조정 방법에 관한 연구 역시 활발히 진행되고 있다.

본 연구에서는 텍스트가 보유한 고유한 정보를 충분히 반영한 벡터를 추출하기 위해, 자기 지도 학습 기반의 사전 BERT 미세 조정 모델을 통한 문서 임베딩 방법론을 제안하였다. 또한, 제안 방법론을 통해 실제 뉴스 기사의 벡터를 도출하고 이를 사용하여 뉴스 기사의 카테고리 분류 실험을 수행한 결과, 제안 방법론을 통해 도출된 텍스트의 벡터가 기존의 임베딩 모델을 통해 생성된 벡터보다 우수한 성능을 보임을 확인하였다.

본 연구는 사전 학습 언어 모델의 미세 조정을 위해 별도의 목적 데이터 생성 과정이 불필요한 자기 지도 학습 방법을 적용했다는 점에서 학술적 기여를 인정받을 수 있을 것이다. 또한, 기존의 사전 학습 언어 모델이 장문의 텍스트에 대한 문맥을 충분히 고려하지 못한다는 한계를 극복하기 위한 방안을 본 연구에서 제시한 것 역시 기여로 인정받을 수 있을 것이다. 한편, 본 연구의 제안 방법에는 모델의 미세 조정을 위해 임의로 지정하는 하이퍼파라미터(Hyperparameter)가 일부 존재하며, 해당 값의 변화에 따른 제안 방법론의 결과에 대한 고려가 이루어져야 한다. 또한, 향후 본 연구에서 제안한 방법론에 대한 다양한 관점에서 보다 엄밀한 평가가 이루어져야 할 필요가 있으며 이를 위해 상이한 특성을 갖는 다양한 문서에 대한 추가 실험이 수행되어야 한다.

REFERENCES

[1] T. Mikolov, C. Kai, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, Jan, 2013.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532-1543, 2014.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," arXiv:1607.04606, Jul, 2016.

[4] T. Mikolov, I. Sutskever, C. Kai, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," Advances in Neural Information Processing Systems, Vol. 26, pp. 3111-3119, Dec, 2013.

[5] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," arXiv:1802.05365, Feb, 2018.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, Oct, 2018.

[7] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet : Generalized Autoregressive Pretraining for Language Understanding," Advances in Neural Information Processing Systems, Vol. 32, pp. 1-11, Dec, 2019.

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

[9] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations," arXiv:1909.11942, Sep, 2019.

[10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter," arXiv:1910.01108, Oct, 2019.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," Proceedings of the 31st Conference on Neural Information Processing Systems, pp. 1-11, 2017.

[12] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What Does BERT Looking At? An Analysis of BERT's Attention," arXiv:1906.04341, Jun, 2019.

[13] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformers-XL: Attentive Language Models Beyond a Fixed-Length Context," arXiv:1901.02860, Jan, 2019.

[14] C. Sun, X .Qiu, Y. Xu, and X. Huang, "How to Fine-Tune BERT for Text Classification?," Proceedings of the 18th China National Conference on Chinese Computational Linguistics, pp. 194-206, 2019.

[15] A. Adhikari, A. Ram, R. Tang, and J. Lin, "DocBERT: BERT for Document Classification," arXiv:1904.08398, Apr, 2019.

[16] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, "Hierarchical Transformers for Long Document Classification," arXiv:1910.10781, Oct, 2019.

[17] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv:1908.10084, Aug, 2019.

[18] R. Zhang, Z. Wei, Y. Shi, and Y. Chen, "BERT-AL: BERT for Arbitrarily Long Document Understandding," Proceedings of the International Conference on Learning Representations 2020, pp. 1-10, 2020.

[19] D. Lee, "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks," Proceedings of the International Conference on Machine Learning 2013 Workshop, pp. 1-6, 2013.

[20] M. S. Ahmed, L. Khan, and N. Oza, "Pseudo-Label Generation for Multi-Label Text Classfication," Proceedings of the 2011 Conference on Intelligent Data Understanding, pp. 60-74, 2011.
[21] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, J. Zhao, and B. Xu, "Self-Taught Convolutional Neural Networks for Short Text Clustering," Neural Networks, Vol. 88, pp. 22-31, Apr, 2017.
[22] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved Variational Autoencoders for Text Modeling using Detailed Convolutions," Proceedings of the 34th International Conference on Machine Learning, pp. 3881-3890, 2017.
[23] D. Yeo, G. Lee, and J. Lee, “Pipe Leak Detection System using Wireless Acoustic Sensor Module and Deep Auto-Encoder,” Journal of The Korea Society of Computer and Information, Vol. 25, No. 2, pp. 59-66, Feb, 2020.
[24] A. V. M. Barone, "Towards Cross-lingual Distributed Repre sentations without Parallel Text Trained with Adversarial Autoencoders," arXiv:1608.02996, Aug, 2016.
[25] L. Jiwei, L. Minh-Thang, and J. Dan, "A Hierarchical Neural Autoencoder for Paragraph and Documents," arXiv:1506.01057, Jun, 2015.
[26] Y. Chen and M. J. Zaki, "KATE: K-Competitive Autoencoder for Text," Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining, pp. 85-94, 2017.
[27] A. Bakarov, "A Survey of Word Embeddings Evaluation Methods," arXiv:1801.09536, Jan, 2018.
[28] Y. Tsvetkov, M. Faruqui, and C. Dyer, "Correlation-based Intrinsic Evaluation of Word Vector Representations," arXiv:1606.06710, Jun, 2016.
[29] J. Zhang and T. Baldwin, "Evaluating the Utility of Document Embedding Vector Difference for Relation Learning," arXiv:1907.08184, Jul, 2019.
[30] T. Baumel, R. Cohen, and M. Elhadad, "Sentence Embedding Evaluation using Pyramid Annotation," Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 145-149, 2016.
[31] J. H. Lau and T. Baldwin, "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation," arXiv:1607.05368, Jul, 2016.
[32] F. F. Liza and M. Grzes, "An Improved Crowdsourcing based Evaluation Technique for Word Embeddings Methods," Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 55-61, 2016.
[33] M. Batchkarov, T. Kober, J. Reffin, J. Weeds, and D. Weir, "A Critique of Word Similarity as a Method for Evaluating Distributional Semantic Models," Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP, pp. 7-12, 2016.
[34] G. Wang, S. Shin, and W. Lee, “A Text Sentiment Classification Method Based on LSTM-CNN,” Journal of The Korea Society of Computer and Information, Vol. 24, No. 12, pp. 1-7, Dec, 2019.
[35] M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer, "Problems with Evaluation of Word Embeddings using Word Similarity Task," arXiv:1605.02276, May, 2016.

Authors

Yeoil Yun received the B.A. degree in Management Information Systems from Kookmin University in 2019 and currently enrolled in Graduate School of Business IT, Kookmin University.

Yeoil Yun is interested in text mining, natural language processing, deep learning, and feature engineering.

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

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