JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 28 No. 9, pp. 35-46, September 2023
https://doi.org/10.9708/jksci.2023.28.09.035

Document Classification Methodology Using Autoencoder-based Keywords Embedding

Seobin Yoon*, Namgyu Kim*

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

[Abstract]

In this study, we propose a Dual Approach methodology to enhance the accuracy of document classifiers by utilizing both contextual and keyword information. Firstly, contextual information is extracted using Google's BERT, a pre-trained language model known for its outstanding performance in various natural language understanding tasks. Specifically, we employ KoBERT, a pre-trained model on the Korean corpus, to extract contextual information in the form of the CLS token. Secondly, keyword information is generated for each document by encoding the set of keywords into a single vector using an Autoencoder. We applied the proposed approach to 40,130 documents related to healthcare and medicine from the National R&D Projects database of the National Science and Technology Information Service (NTIS). The experimental results demonstrate that the proposed methodology outperforms existing methods that rely solely on document or word information in terms of accuracy for document classification.

▸ **Key words**: Deep Learning, Document Classification, Keyword Embedding, Document Embedding, Pre-Trained Language Model

[요 약]

본 연구에서는 문서 분류기의 정확도를 높이기 위해 문맥 정보와 키워드 정보를 모두 사용하는 이중 접근(Dual Approach) 방법론을 제안한다. 우선 문맥 정보는 다양한 자연어 이해 작업(Task)에서 뛰어난 성능을 나타내고 있는 사전학습언어모델인 Google의 BERT를 사용하여 추출한다. 구체적으로 한국어 말뭉치를 사전학습한 KoBERT를 사용하여 문맥 정보를 CLS 토큰 형태로 추출한다. 다음으로 키워드 정보는 문서별 키워드 집합을 Autoencoder의 잠재 벡터를 통해 하나의 벡터 값으로 생성하여 사용한다. 제안 방법을 국가과학기술정보서비스(NTIS)의 국가 R&D 과제 문서 중 보건 의료에 해당하는 40,130건의 문서에 적용하여 실험을 수행한 결과, 제안 방법이 문서 정보 또는 단어 정보만을 활용하여 문서 분류를 진행하는 기존 방법들에 비해 정확도 측면에서 우수한 성능을 나타냄을 확인하였다.

▸ **주제어**: 딥러닝, 문서 분류, 단어 임베딩, 문서 임베딩, 사전학습언어모델

• First Author: Seobin Yoon, Corresponding Author: Namgyu Kim
*Seobin Yoon (Shinebin0501@gmail.com), Graduate School of Business IT, Kookmin University
*Namgyu Kim (ngkim@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
• Received: 2023. 08. 09, Revised: 2023. 09. 15, Accepted: 2023. 09. 15.

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

I. Introduction

최근 컴퓨팅 자원(Computing Resource)의 발전과 대규모 비정형 데이터들의 공개로 딥러닝(Deep Learning) 기술이 다양한 분야에서 활발히 연구되고 있다. 딥러닝은 여러 층(Layer)을 쌓아 만든 인공신경망(Artificial Neural Network) 모델로, 대량의 데이터를 통해 학습을 수행하며 인간의 두뇌와 유사한 방식으로 작동하도록 모델링 된 기계 학습(Machine Learning)의 한 유형이다. 딥러닝의 발전에 힘입어 텍스트 데이터를 분석하는 자연어 처리(Natural Language Processing) 분야에서도 딥러닝 기술이 널리 활용되고 있다. 딥러닝을 활용한 자연어 처리의 대표적인 분야로는 문서의 핵심 내용을 추출하는 텍스트 요약(Text Summarization), 음성 대화 시스템과 같은 사용자의 질문에 답변을 생성하는 텍스트 생성(Text Generation), 주어진 문장(Context)을 읽고 주어진 문제(Question)에 대해 올바른 답(Answer)을 생성하는 질의응답(Question Answering) 그리고 뉴스 카테고리(Category), 사용자 감성(Sentiment) 등 사전에 정의된 클래스로 문서를 분류하는 텍스트 분류(Text Classification) 등이 있으며, 이러한 딥러닝을 활용한 다양한 텍스트 분석 응용 중 텍스트 분류는 학계와 업계에서 가장 널리 활용되고 많이 연구되는 분야로 인식되고 있다.

전통적 텍스트 분류는 사람이 직접 구축한 유의어 사전을 기반으로 단어들의 관계를 그래프로 표현하는 워드넷(WordNet)과 같은 시소러스(Thesaurus) 기법과 그리고 대량의 말뭉치(Corpus)와 통계 기법을 활용한 나이브 베이즈(Naive Bayes), K-최근접 이웃 분류(K-Nearest Neighbor Classifier)와 같은 통계 기반 기법[1]을 통해 수행되어왔다. 한편, 2018년 사전학습언어모델(Pre-Trained Language Model)인 Google의 BERT가 발표[2]되어 텍스트 분류뿐만 아니라 여러 가지 자연어처리 분야에서 뛰어난 성능을 나타내면서, 최근에는 사전학습언어모델 등 딥러닝을 활용한 텍스트 분류가 활발히 연구되고 있다.

BERT는 대량의 말뭉치를 사전학습한 딥러닝 모델로, 인공신경망 구조를 통해 문서의 특징을 추출하고 이에 대한 특징을 벡터값으로 표현하여 나타낸다. BERT는 기존의 언어모델들과 달리 트랜스포머(Transformer) 모듈[3]을 기반으로 양방향 문장 학습을 진행하여, 문장의 맥락(Context)을 고려한 자연어처리 작업을 할 수 있다는 장점을 갖는다. 이러한 장점으로 인해 BERT는 발표된 이후부터 최근까지 다양한 연구의 기반 기술로 널리 활용되고 활발히 수행되고 있다[2, 5-8].

하지만, BERT 등 문서 임베딩(Document Embedding)을 통한 문서 분류(Text Document Classification) 기법은 문서 전체의 맥락을 이해하여 분류를 수행한다는 장점을 갖지만, 저자 또는 다른 전문가에 의해 부여된 정형 또는 반정형 정보를 활용하지 못한다는 한계를 갖는다. 예를 들어 특허나 논문 등 전문 문서의 경우 저자가 제목, 초록뿐 아니라 문서의 내용을 가장 잘 나타낼 수 있는 키워드를 함께 제시하게 된다. 즉 키워드의 경우 문서의 내용을 가장 압축하여 표현하고 있는 중요한 정보인데, 초록 전체를 요약하는 문서 임베딩은 키워드 정보를 고려하지 않는 경우가 일반적이다. 이에 키워드 정보만 활용하는 문서 분류는 문서 전체의 내용을 충분히 반영하지 못하고, 문서 임베딩을 활용하는 문서 분류는 매우 유용한 정보인 키워드 정보를 활용하지 못한다는 한계를 갖는다. 따라서 키워드 정보와 문서 임베딩을 동시에 활용할 수 있는 방안에 대한 모색이 이루어지고 있지만[9], 두 정보를 활용하는 방식의 본질적인 차이로 인해 이러한 시도가 충분히 이루어지지 못했다.

이에 본 연구에서는 키워드 정보와 문서, 즉 초록 정보를 동시에 활용하여 문서 분류의 성능을 높이기 위해 신경망 모델 중 하나인 오토인코더(Autoencoder) 모델[10]을 활용한 방법을 제안하고자 한다. 구체적으로는 오토인코더를 통해 각 문서에 사용된 키워드를 벡터화하고, 한국어 말뭉치를 사전학습한 KoBERT를 사용하여 문서 전체의 내용을 하나의 벡터로 표현한 뒤, 이들 두 벡터를 동시에 활용하는 이중 접근 방법을 통해 문서 분류의 정확도(Accuracy)를 향상시킬 수 있는 방법론을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 선행 연구인 문서 분류, 단어 임베딩, 그리고 문서 임베딩을 소개하고, 3장에서는 본 연구에서 제안하는 오토인코더 기반 키워드 임베딩을 통한 문서 분류 방법론을 소개한다. 4장에서는 실제 데이터에 대한 실험을 통해 제안 방법론을 통한 성능 향상 정도를 측정하고, 마지막 5장에서는 본 연구의 기여와 한계를 정리한다.

II. Preliminaries

1. Document Classification

문서 분류는 주어진 텍스트 문서가 어떤 카테고리에 속하는지를 구분하는 과업을 뜻한다. 일반적으로 텍스트 분류는 구분해야 하는 범주의 개수에 따라 이진 분류와 다중

분류로 누어지는데, 이진 분류(Binary Classification)는 분류해야 하는 범주의 개수가 2개일 경우를 뜻하며 다중 분류(Multi-Class Classification)는 분류해야 하는 범주가 3개 이상일 때를 말한다[11]. 텍스트 분류는 사람이 직접 구축한 유의어 사전을 기반으로 단어들의 관계를 그래프로 표현하는 시소러스 혹은 대량의 말뭉치와 통계 기법을 활용한 통계 기반 기법에 기반을 두어 주로 연구되었는데, 최근에는 기계 학습을 적용하여 자동으로 분류하는 연구가 높은 성과를 보이면서 인공신경망을 활용한 추론 기반 기법이 많이 사용되고 있다.

기계 학습은 인공 지능의 하위 범주에 포함되는 분야로, 컴퓨터가 수행할 작업을 데이터로부터 스스로 학습하도록 하는 방법을 연구하는 분야이다. 기계 학습의 학습 방법으로는 입력값(Input Data)과 입력값에 대한 정답(Label)으로 구성된 데이터를 활용해 학습시키는 지도학습(Supervised Learning), 정답이 없는 데이터를 비슷한 특징끼리 군집화하여 새로운 데이터에 관한 결과를 예측하는 비지도학습(Unsupervised Learning), 그리고 지도학습과 비슷하지만 완전한 정답을 제공하지 않고 정답에 가까운 결과를 나타낼 때마다 점수(Reward)를 부여하여 답을 찾아가도록 학습하는 강화학습(Reinforcement Learning) 등이 있다. 대표적인 지도학습 알고리즘으로는 회귀(Regression)와 분류(Classification)가 있으며[12], 비지도학습에는 군집화(Clustering), 밀도추정(Density Estimation), 그리고 차원축소(Dimensionality Reduction) 등이 있다[13]. 그리고 강화학습에는 Q-러닝(Q-Learning)과 A3C(Asynchronous Advantage Actor-Critic) 등이 널리 사용되고 있다[14].

인공신경망을 활용하여 문서 분류를 진행한 연구로는 합성곱 신경망(Convolution Neural Network)을 활용하여 국가과학기술정보서비스(NTIS)에 5년간의 연구보고서를 국가과학기술표준분류 체계에 맞게 문서 분류한 연구[15], 사전학습언어모델인 BERT를 활용하여 국가과학기술정보서비스의 국가 R&D 문서를 국가과학기술표준분류 체계에 맞추어 분류하는 전문 언어모델을 구축하는 연구[5], 그리고 금융 특화 말뭉치를 사용하여 금융 특화 한국어 사전학습언어모델을 만들어 문서 분류에 적용한 연구[4] 등을 들 수 있다. 최근에는 높은 분류 정확도만을 추구하는 것이 아니라 문서의 전문성까지 이해할 수 있는 도메인 특화 언어모델을 만드는 연구가 활발하게 이루어지고 있다.

2. Word Embedding

단어 임베딩(Word Embedding)은 단어를 수치화하여 벡터 공간으로 표현하는 과정을 의미하며, 이는 컴퓨터가 자연어를 이해하기 위해 진행되는 필수적인 자연어 처리 과정 중 하나이다. 일반적으로 단어 임베딩은 관련성 높은 주변 단어를 통해 의미적 정보를 계산하고, 이를 통해 개별 단어의 특징을 표현하는 벡터를 추출하는 과정으로 진행된다. 대표적인 단어 임베딩 방법으로는 인공신경망을 활용한 추론 기반 기법인 Word2Vec[16], GloVe[17], 그리고 FastText[18] 등이 있다.

Word2Vec은 2013년 Google에서 제안한 모델로 CBOW(Continuous Bag-Of-Words)와 Skip-Gram 두 가지 방식이 있다(Fig. 1).

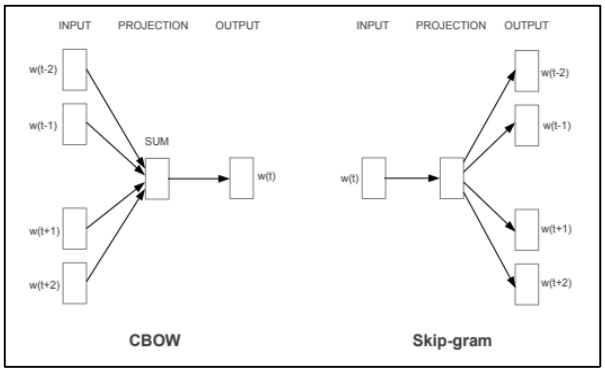


Fig. 1. Structure of CBOW & Skip-gram

둘 다 대상 단어와 주변 단어의 관계를 이용하여 단어를 저차원 벡터 공간에 임베딩하는 모델이지만, CBOW 방식은 주변 단어를 이용하여 대상 단어를 예측하는 반면, Skip-gram은 대상 단어 벡터를 이용하여 주변 단어를 예측한다는 점에서 차이가 있다. Skip-gram은 여러 단어를 예측하기 때문에 CBOW에 비해 비효율적일 수 있지만, 특정 조건에서 CBOW에 비해 우수한 성능을 나타낼 수 있음이 알려져 있다. 하지만 Word2Vec은 지정한 주변 단어 및 대상 단어인 윈도우(Window) 내에서만 학습되기 때문에, 말뭉치 전체의 맥락은 반영하지 못한다는 단점이 있다. 이에 2014년 Word2Vec의 단점을 보완하고자 등장한 임베딩 방법이 GloVe이며, 이는 임베딩 된 단어 간 벡터 유사도의 측정을 편리하게 하면서도 말뭉치 전체의 통계 정보를 반영할 수 있게 하여 기존 임베딩의 단점을 개선하였다. 하지만 위의 두 가지의 임베딩 방법론은 학습에 등장하지 않은 단어는 벡터값을 얻을 수 없다는 한계(Out Of Vocabulary, OOV)를 공통적으로 가지고 있다. FastText는 이러한 한계를 극복하고자 2016년 Facebook

에서 제안한 방법으로, 단어 단위로 임베딩하던 기존 방법론들과 달리 각 단어를 문자 단위인 N-gram으로 표현하여 학습하는 보조 단어(Sub-word) 기반 방법론이다. 단어 단위가 아닌 문자 단위로 학습하기 때문에, 학습에 사용되지 않았던 단어가 등장하더라도 단어를 구성하는 문자에 대한 학습 결과를 활용하여 해당 단어에 대해서도 벡터값을 구할 수 있다.

이러한 단어 임베딩 모델들은 단어의 의미를 고려하는 특성 벡터값을 만들어내기 때문에, 문서 검색 성능을 올리는 연구[19], 단어 임베딩 모델만을 활용하여 문서 분류를 진행하는 연구[20], 그리고 문서 분류를 위한 신경망 모델에 적용되어 성능을 개선하기 위한 연구[21] 등에 활용되고 있다. 이처럼 단어 임베딩 방법론들을 사용하여 도출한 단어 벡터는 다양한 텍스트 분석에서 유용하게 사용되고 있다. 하지만 기존의 단어 임베딩 방법론들은 같은 형태를 지녔으나 여러 다른 뜻으로 사용되는 동음이의어 혹은 다의어의 처리 과정에서, 단어의 형태가 같다면 의미가 상이하더라도 동일한 벡터값으로 표현한다는 공통적인 한계를 갖는다. 이를 해결하기 위해서는 단순히 단어 자체의 의미만을 고려하지 않고, 단어의 문맥적 의미를 파악하기 위해 문장 또는 문서 전체를 살펴보는 접근이 필요하다.

3. Document Embedding

문서 임베딩은 문장 임베딩(Sentence Embedding)의 분석 단위인 문장을 문서 전체로 확장 시켜 하나의 벡터값으로 임베딩하는 방법이다. 문장 임베딩은 단어 임베딩의 한계를 극복하고 문장의 의미를 반영하는 벡터를 도출하기 위해 고안되었다.

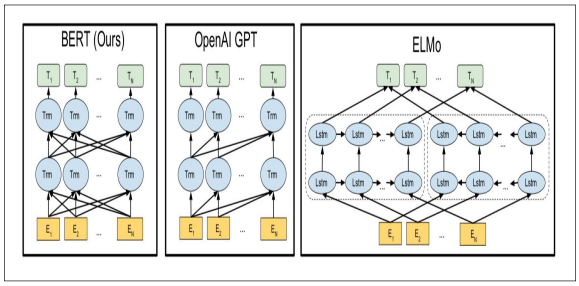


Fig. 2. Representative Pre-trained Language Model

문장 임베딩은 문장 하나를 하나의 벡터값으로 임베딩하는 방법이다. 문장 임베딩 연구가 활발히 이루어지게 된 것은 2014년 Word2Vec을 기반으로 Doc2Vec[22]이 등장하고, 이후 2018년 FastText를 기반으로 만든 ELMo[23], BERT[2], 그리고 GPT[24] 등이 등장하면서이다(Fig. 2).

특히, 딥러닝을 활용한 사전학습언어모델 BERT는 단방향 학습을 하는 기존의 언어모델들과 달리 트랜스포머(Fig. 3)의 인코더(Encoder) 모듈을 기반으로 문장의 양방향 학습을 진행하기 때문에, 문장의 맥락을 고려해야 하는 다양한 자연어 처리 분야에서 높은 성능을 나타내고 있다. BERT의 이러한 장점으로 인해 BERT는 2018년 처음 제안된 이후부터 현재에 이르기까지 다양한 자연어 처리 분야의 연구에서 활발하게 사용되고 있다.

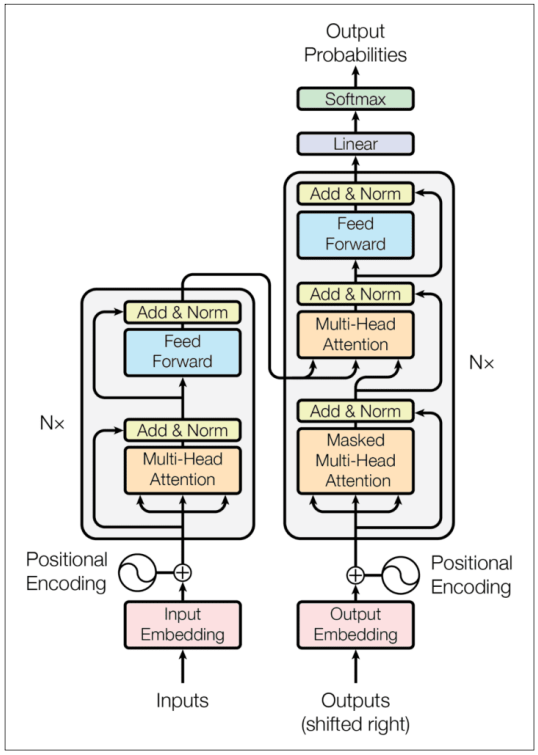


Fig. 3. Architecture of Transformer

BERT는 Book Corpus와 Wikipedia Data를 활용한 총 3.3억 단어의 대량의 말뭉치를 사전 학습시킨 모델로, 원하는 자연어 처리 작업에 맞추어 미세조정(Fine-Tuning)하여 사용한다. 이미 대량의 데이터를 사전에 학습한 모델을 사용하기 때문에, 하위 작업에서 사용할 수 있는 데이터의 양이 적더라도 높은 성능의 결과를 얻을 수 있다. 또한, 범용적인 모델이기 때문에 분야와 상관없이 활용할 수 있어, 학계와 업계를 막론하고 다양하게 사용되고 있다. 하지만, 문서와 함께 해당 문서의 내용을 잘 나타내는 키워드가 동시에 제공된 경우, BERT는 이 문서의 벡터를 도출하는 과정에서 키워드를 활용할 수 있는 방법을 제시하지 않는다. 물론 최근 연구들은 BERT의 토크나이저(Subword Tokenizer)에 주어진 키워드를 추가하는 방식으로, 주어진 키워드를 더욱 잘 이해하기 위한 방법을 고

안하였다[2, 4]. 하지만 이러한 시도들은 주어진 키워드의 정보를 문서 임베딩에 직접적으로 활용하는 것이 아니라 단지 키워드로 주어진 단어가 하위 단어로 분절되는 것을 방지하는 역할만 수행한다는 점에서 한계를 가지고 있다. 이와 달리 본 연구에서는 문서의 전체 내용을 표현하는 문서 벡터뿐 아니라 별도로 주어진 키워드 정보를 함께 활용하여, 문서에 대한 이해를 높이고 궁극적으로 문서 분류의 정확도를 향상시킬 수 있는 방법을 제안하고자 한다.

III. Proposed Method

1. Research Process

본 장에서는 문서의 전체 내용을 표현하는 문서 벡터뿐 아니라, 별도로 주어진 키워드 정보를 함께 활용하여 문서 분류의 정확도를 향상시킬 수 있는 방법을 제안한다. 제안 방법론의 전체 과정은 <Fig. 4>와 같다. 먼저 Phase 1은 문서를 전처리하고(단계 1), 사전학습언어모델인 KoBERT를 통해 문서별 벡터를 추출한다(단계 2). 다음으로 Phase 2는 각 문서별 키워드 포함 정보를 활용하여 각 키워드를 멀티-핫 벡터(Multi-hot Vector)로 표현한다(단계 3). 다음으로 이처럼 멀티-핫 벡터로 구성된 키워드 벡터를 오토인코더를 통해 잠재 공간으로 압축하여 표현한 뒤, 압축된 잠재 벡터(Latent Vector)를 키워드 벡터로 추출한다(단계 4). 다음으로 Phase 3은 Phase 1에서 도출한 문서별 벡터와 Phase 2에서 도출한 키워드 벡터를 모두 활용하여 분류기를 학습한다(단계 5). 각 과정에 대한 구체적인 내용은 본 장의 이후 각 절에서 설명하며, 제안 방법론의 성능 평가 결과는 4장에서 소개한다.

2. Text Preprocessing

본 절에서는 <Fig. 4>의 데이터 전처리 및 키워드 추출(단계 1) 과정을 소개한다. 먼저 주어진 문서에서 키워드를 추출하기 위해, 문서 내에서 특정 빈도 이상 사용된 고빈도 단어를 추출하여 키워드 집합을 구성한다. 이후 전체 문서에 대해, 키워드 집합에 수록된 키워드 중 단 하나의 키워드도 포함하지 않는 문서들을 식별하여 분석에서 제외한다. 이후 각 문서에 대해 외국어 및 특수 기호를 제거하고, 둘 이상의 연속된 공백을 하나의 공백으로 치환하는 등의 전처리 작업을 수행한다(단계 1).

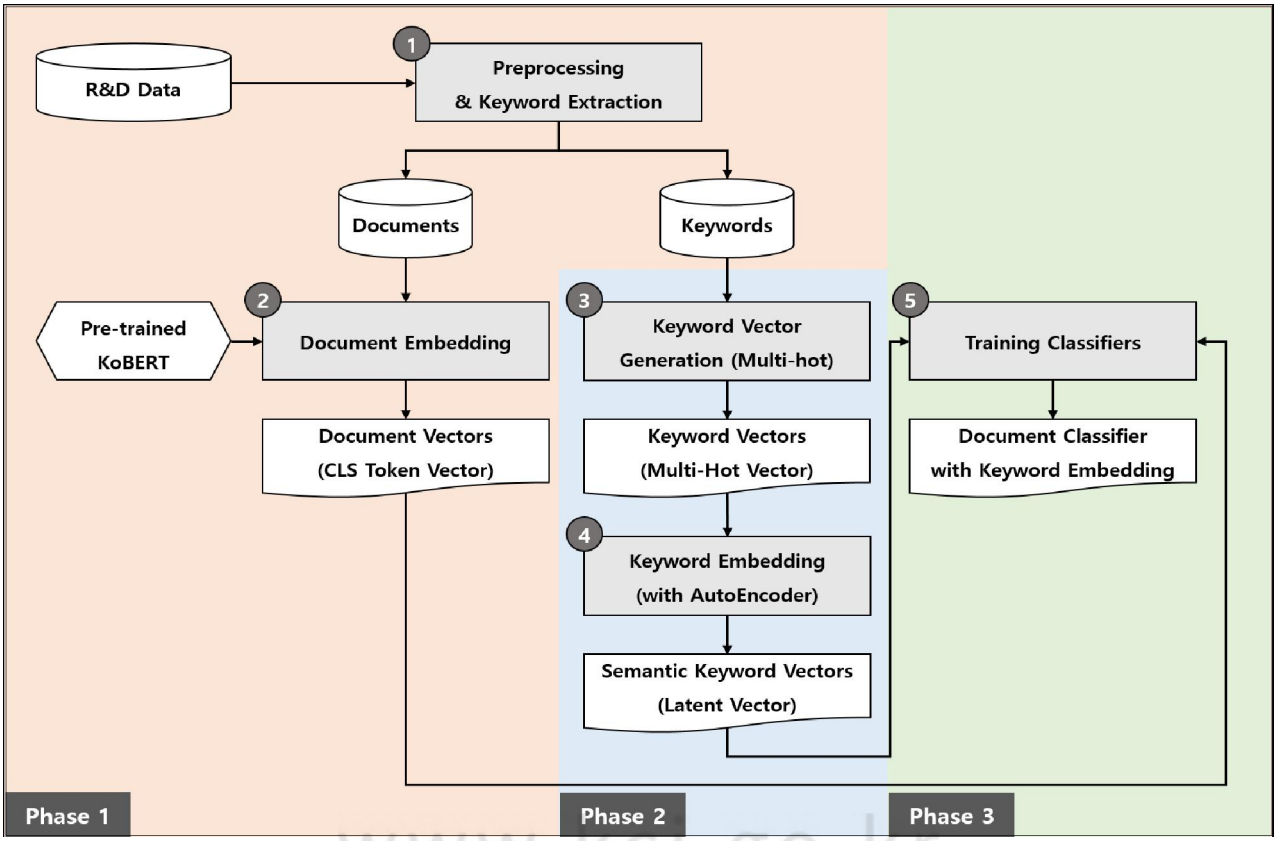


Fig. 4. Overall Process of the Proposed Methodology

3. Document Embedding

본 절에서는 <Fig. 4>의 문서 임베딩 과정(단계 2)을 소개한다. BERT는 보조 단어 토크나이저를 사용하여 문장 내 단어를 단어보다 더 작은 단위로 분절한다. 이때 분절된 문장의 첫 번째에는 입력받은 문장의 시작을 나타내는 특수 토큰인 'CLS'가 위치하며, 문장과 문장 사이에는 두 문장을 구분하기 위해 'SEP'라는 특수 토큰을 사용한다. 이때 특수 토큰인 CLS 토큰은 학습을 거치는 동안 입력된 문장 전체의 맥락을 학습하며, 학습이 완료된 후에는 각 문장을 대표하는 문장 임베딩 벡터로 사용된다.

단계 2의 문서 임베딩 과정에서는 단계 1에서 전처리가 이루어진 문서에 대해 한국어 말뭉치를 추가 학습한 KoBERT 모델을 사용하여 문서 임베딩 벡터를 추출한다. KoBERT 모델은 문서 데이터를 입력으로 받아서 문서 전체에 대한 벡터값을 출력한다. 이때 출력되는 벡터는 KoBERT 내부의 12개의 트랜스포머 인코더 레이어에서 출력되는 숨겨진 상태(Hidden Status) 값을 풀링 레이어(Pooling Layer)를 통해 평균한 값이다. 구체적으로 제안 방법론은 KoBERT의 12개 레이어 중 마지막 4개 레이어의 CLS 토큰값을 추출한 뒤, 이에 대한 평균을 계산하여 문서 임베딩 값으로 사용한다(Fig. 5).

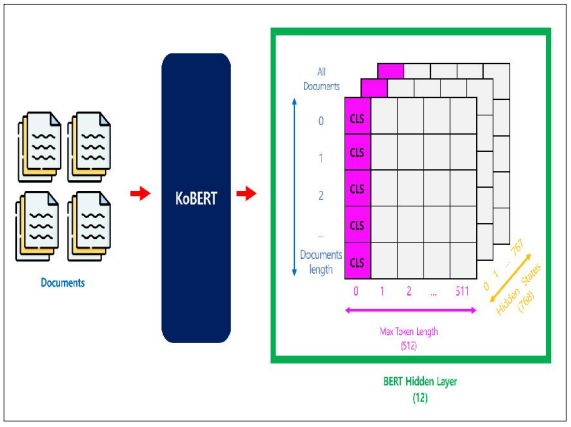
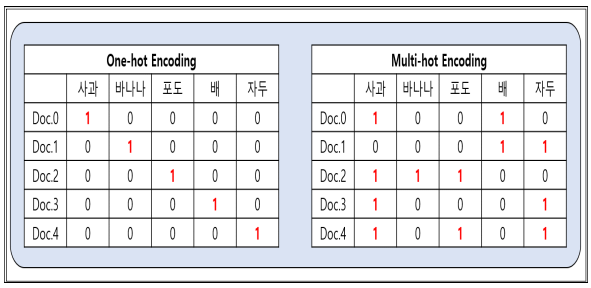


Fig. 5. Extraction of Document Vector

4. Keywords Embedding

본 절에서는 <Fig. 4>의 단계 1에서 도출된 키워드 데이터를 활용하여 문서별 키워드의 멀티-핫 벡터를 생성하고(단계 3), 오토인코더 모델과 문서별 키워드 멀티-핫 벡터를 사용하여 키워드 잠재 벡터를 추출하는 과정(단계 4)을 소개한다. 우선 오토인코더를 활용하여 잠재 벡터를 추출하기 위해서는 키워드 데이터를 벡터값으로 변환해야 한다. 이를 위해 단계 1에서 도출된 문서와 키워드에 대해 문서별 키워드의 출현 여부를 확인하고, 키워드가 한 번 이상 출현하는 경우 '1', 그렇지 않은 경우 '0'의 값을 부여하여 각 키워드를 문서 수만큼의 차원을 갖는 벡터로 변환한다. 이때 하나의 문서에 여러 키워드가 존재할 수 있으므로, 문서 벡터는 한 개의 값만 '1'로 표기하는 원-핫 벡터가 아닌 여러 개의 값이 '1'로 표기될 수 있는 멀티-핫 벡터의 형태로 나타난다(Fig. 6).



One-hot Encoding

	사과	바나나	포도	배	자두
Doc.0	1	0	0	0	0
Doc.1	0	1	0	0	0
Doc.2	0	0	1	0	0
Doc.3	0	0	0	1	0
Doc.4	0	0	0	0	1

Multi-hot Encoding

	사과	바나나	포도	배	자두
Doc.0	1	0	0	1	0
Doc.1	0	0	0	1	1
Doc.2	1	1	1	0	0
Doc.3	1	0	0	0	1
Doc.4	1	0	1	0	1

Fig. 6. Examples of One-hot and Multi-hot Vectors

도출된 키워드 멀티-핫 벡터는 잠재 벡터를 추출하기 위해 오토인코더의 입력 데이터로 사용한다. <Fig. 7>은 오토인코더의 구조를 나타내며, 5차원의 키워드 멀티-핫 벡터를 3차원 잠재 벡터로 압축한 후 다시 복원하는 예를 보인다.

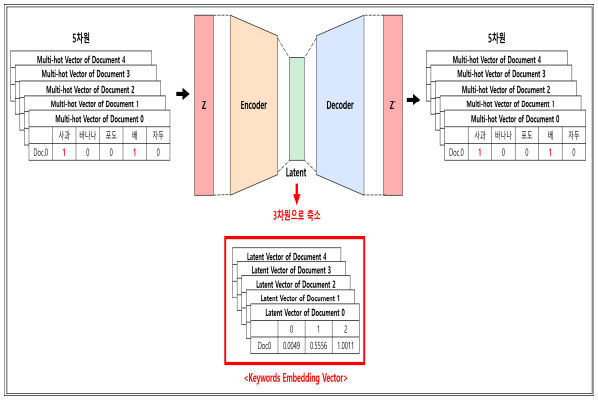


Fig. 7. Example of Latent Vector Generation

이때, 인코더를 통해 압축된 3차원의 잠재 벡터는 5차원으로 주어진 입력 데이터 내용을 최대한 잘 표현하도록 압축된 벡터이다. 따라서 키워드를 압축한 잠재 벡터는 각 문서의 키워드 구성 정보를 잘 나타내는 임베딩 값으로 볼 수 있다. 본 제안 방법에서는 이러한 잠재 벡터를 문서의 키워드 임베딩 벡터값으로 사용한다(단계 4).

5. Training Classifier by Hybrid Approach

본 절에서는 앞에서 도출된 문서 임베딩 벡터와 키워드 임베딩 벡터를 모두 사용하여 분류기 학습(단계 5)을 진행

하는 과정을 소개한다. <Fig. 8>과 같이 문서 분류기는 기본적으로 Dense Layer를 여러 개 쌓은 구조이며, 입력 데이터의 문서 임베딩 벡터와 키워드 임베딩 벡터를 모두 사용하는 하이브리드(Hybrid) 접근 방식으로 학습을 진행한다. 이렇게 학습된 분류기는 문서 정보와 키워드 정보를 모두 활용하여 분류를 수행하게 되며, 기존의 문서 임베딩 또는 단어 임베딩만을 사용하여 문서 분류를 진행하는 경우에 비해 정확도 측면에서 우수한 성능을 보일 수 있을 것으로 기대한다. 이때, 분류 성능 평가를 위해 사용되는 정확도는 실제 데이터와 예측 데이터가 얼마나 같은지를 판단하는 지표로, 해당 지표의 산출 식은 다음 (식 1)과 같다.

$$(식\ 1)\ Accuracy(정확도) = \frac{정답을\ 맞춘\ 건수}{전체\ 예측\ 건수}$$

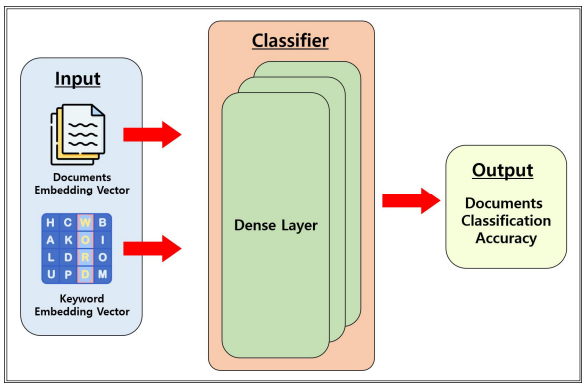


Fig. 8. Training Classifiers Using Hybrid Approach

IV. Experiment

1. Overview

본 장에서는 제안 방법론의 성능 평가를 위해 사용된 실험 데이터를 소개하고, 제안 방법론을 적용한 실험 결과에 대한 분석을 통해 제안 방법론의 성능을 평가한다. 실험에는 국가과학기술정보서비스에서 제공하는 2011년부터 2020년까지의 국가 R&D 과제 문서 중 보건 의료 분야의 문서 40,130건을 추출하여 사용하였다. 키워드 데이터의 경우, 국가 R&D 과제 보건 의료 문서에서 특정 빈도 이상 사용된 고빈도 단어 927개를 선정하였으며, 문서의 분류 대상(Target)은 과학기술표준분류 체계에서 보건 의료 분류의 하위 분류에 해당하는 15개의 중분류 코드를 사용하였다. 실험 환경은 <Table 1>과 같이 Python 3.7과 Tensorflow-gpu 2.4.0, 그리고 Pytorch 1.7.1, Keras 2.4.3과 Transformer 3.0.1을 기반으로 구축하였고, GPU는 Tesla V100 16GB를 사용하였다.

Table 1. Experimental Environments

S/W	
Python	3.7
Tensorflow-gpu	2.4.0
Keras	2.4.3
Pytorch	1.7.1
Transformer	3.0.1
H/W	
GPU	Tesla V100 16GB

2. Preprocessing and Keywords Extraction

실험에 사용된 문서 데이터는 국가과학기술정보서비스의 국가 R&D 데이터로, 해당 데이터는 2019년 과학기술표준분류체계의 대분류를 기준으로 보건 의료, 정보 통신, 그리고 생명 과학 등 33개의 과제 분야로 분류된다. 먼저, 해당 문서에서 키워드를 추출하기 위해 국가 R&D 과제 분야별로 특정 빈도 이상 사용된 고빈도 단어들을 도출하였다. 도출된 키워드는 총 4,932개로, 이중 보건의료 분야에 해당하는 키워드만을 추출하여 총 927개의 키워드 집합을 구성하였다.

이후, 33개의 과제 분야 중 보건 의료 분야에 해당하는 문서들을 추출하였으며, 문서의 핵심 내용을 표현하고 있는 “과제명”과 “연구목표” 부분을 추출하고 연결(Concatenation)하여 분석 단위 문서를 구성하였다. 또한 이렇게 구성된 문서들에 대해 상기 927개의 키워드 중 단 하나의 키워드도 포함하고 있지 않은 문서를 찾아 제거하여, 그 결과 총 40,130개의 문서를 분석 대상으로 설정하였다. 또한 3장에서 제시한 절차에 따라 전처리를 수행한 후, 과학기술표준분류체계 중분류 15개를 0부터 14까지의 숫자로 코드화하여 각 문서의 레이블로 부여하였다.

3. Generating Keyword Multi-hot Vectors

오토인코더의 잠재 벡터값을 추출하여 키워드 임베딩값을 얻기 위해서는, 먼저 키워드 데이터를 벡터 구조로 변환해야 한다. 이를 위해 전처리한 국가 R&D 보건 의료 문서를 대상으로, 각 문서마다 전체 키워드에 대해 출현 빈도를 측정하였다. 문서 내에 포함된 키워드에 대해서는 ‘1’의 값을 부여하고, 포함되지 않은 키워드에 대해서는 ‘0’의 값을 부여하였다(Fig. 9).

Fig. 9. Example of Keyword Multi-hot Vector

4. Document Embedding with KoBERT

KoBERT는 Google에서 제안한 사전학습언어모델인 BERT를 활용하여 한국어 말뭉치를 사전 학습한 한국어 언어모델이다. 본 실험에서는 한국어 문서를 다루기 때문에 해당 모델을 활용하여 CLS 토큰 추출을 진행하였다. BERT의 CLS 토큰은 특별 토큰으로, 모델을 통해 적용할 수 있는 다양한 하위 작업 중 분류 작업을 할 때 주로 사용된다. 본 실험에서는 <Fig. 10>과 같이 KoBERT의 12개의 레이어 중 마지막 4개 레이어의 CLS 토큰값의 평균을 계산하여 문서 임베딩값으로 활용하였다.

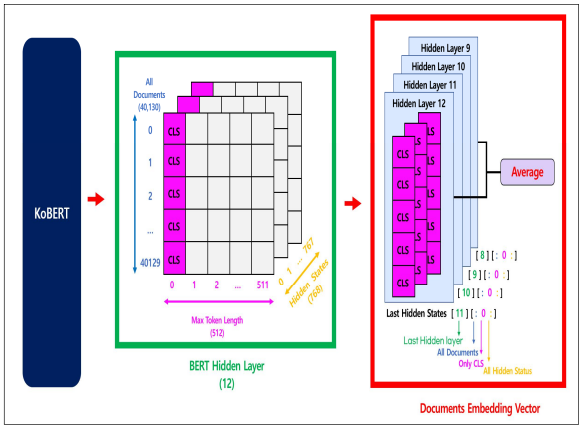


Fig. 10. Document Vector Generation Using KoBERT

5. Generating Keyword Latent Vectors Using Autoencoder

국가 R&D 보건 의료 문서로부터 키워드 임베딩 벡터를 추출하기 위해, 키워드 멀티-핫 벡터를 활용하여 오토인코더의 학습을 수행해야 한다. 오토인코더는 데이터를 압축하는 인코더와 이를 다시 복원하는 디코더(Decoder)의 쌍으로 구성되어있다. 오토인코더는 <Fig. 11>과 같이 인코더를 통해 입력된 키워드 멀티-핫 벡터 데이터를 압축하는 과정과 디코더를 통해 압축된 데이터를 원래 데이터로 복구하는 학습을 반복하여 학습을 진행하였다. 이후, 학습이 완료된 오토인코더의 인코더 부분을 사용하여 전체 키워드 멀티-핫 벡터를 입력으로 잠재 벡터값을 추론하여 키워드 임베딩 벡터값으로 사용하였다.

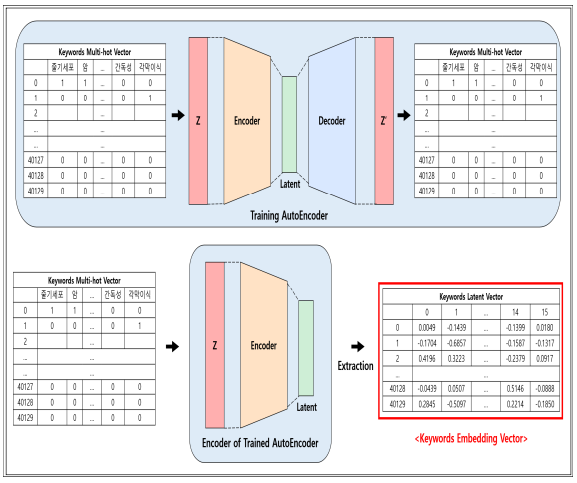


Fig. 11. Keyword Latent Vector Extraction

6. Performance Evaluation

본 절에서는 문서 정보와 단어 정보를 모두 활용하여 문서 분류를 진행하는 제안 방법론과 문서 정보 혹은 단어 정보 한 가지만을 사용하여 문서 분류를 진행하는 기존 방법론의 분류 정확도를 분석한다. 실험 과정은 <Fig. 12>와 같다. <Fig. 12-①>과 <Fig. 12-③>은 문서 정보와 단어 정보만을 사용하여 분류기 학습을 진행한 후, 분류 결과를 평가하는 과정이다. <Fig. 12-②>는 문서 정보인 CLS 토큰값을 오토인코더로 압축하여 분류기 학습을 진행한 실험이며, <Fig. 12-①+③>과 <Fig. 12-②+③>은 본 논문에서 제안하는 방법으로 문서 정보와 단어 정보를 이중 접근 방식을 통하여 분류기 학습을 진행한 후, 분류 결과를 평가하는 과정이다. <Fig. 12-①+②>는 제안하는 방법인 문서 정보와 단어 정보 조합으로 인해 발생하는 데이터의 차원 증가가 성능에 영향을 준 것이 아닌지 평가하기 위해 진행한 과정이다.

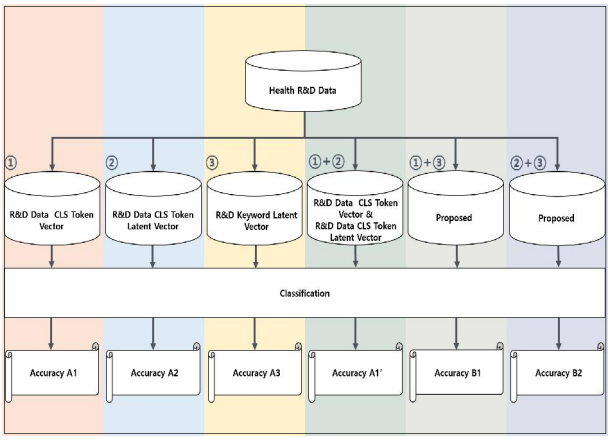


Fig. 12. Performance Evaluation Experiment Overview

학습에 사용된 데이터와 하이퍼 파라미터 구성은 <Table 2>와 같다. 우선 전체 데이터를 학습용 25,683건, 검증용 8,026건, 그리고 평가용 8,026으로 분할하여 학습을 진행하였다. 하이퍼 파라미터는 에폭(Epoch) 30, 배치 크기(Batch Size) 128로 설정하였고, 활성화 함수(Activate Function)와 옵티마이저(Optimizer)는 각각 'Relu', 'Adam'을 사용하였다.

Table 2. Data Partition and Hyperparameters

Data Partition	
Train	24,078
Validation	8,026
Test	8,026
Hyperparameters	
Epoch	30
Batch	64/128/256/512
Activate Function	Relu
Optimizer	Adam

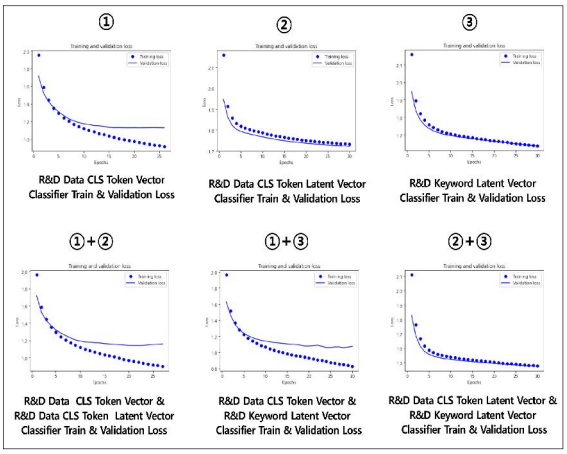


Fig. 13. Classifier Model Per Experiment Loss Value Eer Epoch

실험 결과는 <Fig. 14>와 <Table 3>에 요약되어있다. 먼저, <Table 3>의 Baseline A2, A3은 문서 정보를 압축한 CLS 잠재 벡터 벡터값과 단어 정보를 가진 키워드 잠재 벡터값만을 사용하여 학습한 모델로 40.32%와 46.08%의 정확도를 나타냈다. <Table 3>의 Proposed B1은 문서 정보와 단어 정보 두 가지를 이중 접근 방식으로 입력 후 분류기 학습을 진행한 제안 모델로 49.51%의 정확도를 나타내어 문서 정보나 단어 정보만을 사용하여 문서 분류를 진행하였을 때보다 성능이 더 우수한 것을 확인하였다.

다음으로 <Table 3>의 문서 정보인 CLS 토큰값을 오토인코더로 축소하지 않은 768차원의 데이터를 활용하여 학습한 Baseline A1 모델은 62.50%의 정확도를 나타내었다. 이와 비교하기 위한 제안 모델인 Baseline B2는 768차원의 CLS 토큰값과 단어 정보인 16차원의 키워드 잠재 벡터를 이중 접근 방식으로 학습하였으며, 해당 모델은 63.69%의 정확도를 나타내어 문서 정보나 단어 정보만을 활용하였을 때보다 분류 정확도가 향상됨을 재확인할 수 있었다. 마지막으로, 단순히 문서 정보와 단어 정보를 조합하면서 입력 데이터 차원이 증가하는 것이 분류 성능에 영향을 주었는지 확인하기 위해 제안 모델과 분류 정확도를 비교하는 추가 실험을 진행하였다. 모델의 입력 데이터 크기를 문서 정보와 단어 정보를 조합했을 때의 차원수인 784차원으로 설정하기 위해 768차원의 CLS 토큰값과 16차원의 CLS 토큰 잠재 벡터값을 조합하여 784차원의 데이터를 생성한 뒤, 학습을 진행한 <Table 3>의 Baseline A1′모델과 같은 차원의 데이터를 사용하는 제안 모델인 Baseline B2 분류 정확도를 비교하였다. Baseline A1′의 분류 정확도는 62.38%로 오히려 입력 데이터의 차원이 더 작은 Baseline A1 모델보다도 성능이 낮은 것을 확인할 수 있었다. 결국, 문서 정보에 단어 정보를 더 주기 위해 불가피하게 발생한 차원 수 증가는 성능에는 영향이 없으며, 실제 유용한 정보인 단어 정보가 추가로 주어진 것이 분류 성능을 올렸다는 결과를 확인할 수 있었다.

Table 3. Classification Accuracy Results by Model

Model Type		Dim.	Batch Size			
			64	128	256	512
Baseline A1	CLS(Full)	768	**62.50**	60.93	60.59	60.42
Baseline A2	CLS(Latent)	16	**40.32**	40.03	39.75	39.68
Baseline A3	Keywords(Latent)	16	**46.08**	45.09	45.09	44.68
Baseline A1′	CLS(Full) + CLS(Latent)	784	**62.38**	61.35	61.43	59.67
Proposed B1	CLS(Full) + Keywords(Latent)	784	**63.69**	61.60	63.59	62.17
Proposed B2	CLS(Latent) + Keywords(Latent)	32	**49.51**	48.69	49.41	48.59

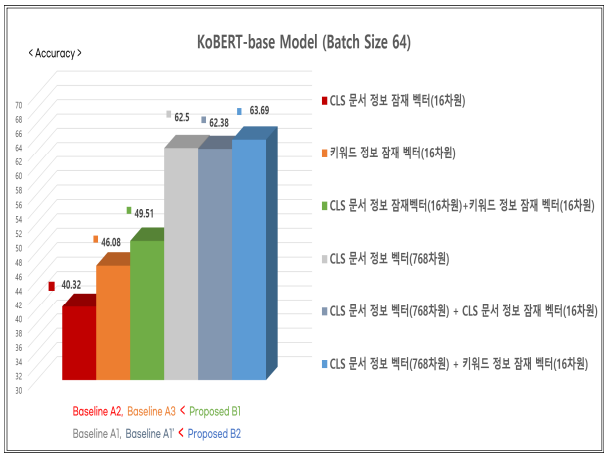


Fig. 14. Document Classification Accuracy by Experiment

추가로, 배치 사이즈 별로 성능 차이가 발생함에 따라 제안 모델과 비교모델의 성능을 이에 맞추어 비교해 보았다. <Fig. 15>과 <Fig. 16>은 이를 그래프로 나타낸 것이다. <Fig. 15>에서 확인할 수 있듯이 배치 사이즈 별로 성능 차이가 발생하지만, 그럼에도 제안 방법론이 모든 배치 사이즈에서 성능이 우수한 것을 확인하였다. <Fig. 16>은 동일한 차원의 크기로 비교모델과 제안 모델의 데이터의 크기를 맞춘 뒤 진행한 실험 그래프이다. 해당 그림에서도 배치 사이즈 별로 성능 차이는 발생하지만, 비교 모델보다 제안 방법론이 모든 배치 사이즈에서 성능이 우수한 것을 확인하였다.

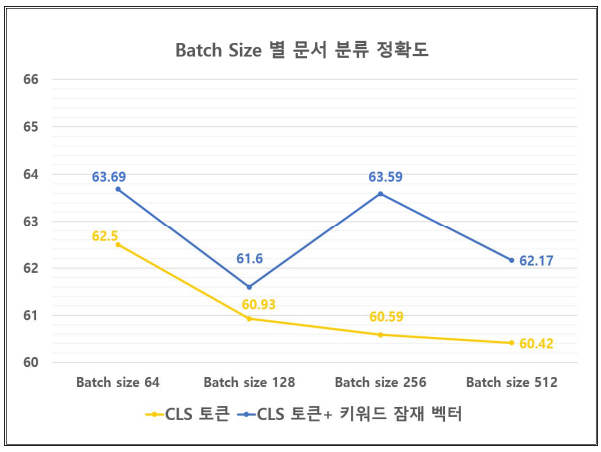


Fig. 15. By Batch Size of Comparative Model and Proposed Methodology Document Classification Accuracy

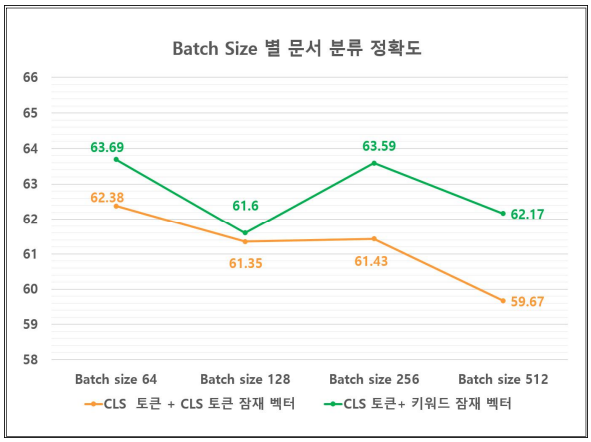


Fig. 16. Document Classification Accuracy by Batch Size

V. Conclusions

본 연구에서는 KoBERT와 오토인코더를 통해 문서의 전체 내용을 표현하는 문서 벡터뿐만 아니라 별도로 주어진 키워드 정보를 함께 활용하여, 문서에 대한 이해를 높이고 궁극적으로 문서 분류의 정확도를 향상시킬 수 있는 방법을 새롭게 제안하였다. 또한, 실제 국가 R&D 보건 의료 문서와 국가 R&D 보건 의료 키워드 데이터를 사용하여 실험을 수행한 결과, 제안 방법론이 문서 정보와 단어 정보를 따로 사용하여 분류를 진행했을 때 분류 정확도 측면에서 우수한 성능을 보임을 확인하였다.

키워드는 1개가 아닌 하나의 문서 내에서 여러 개가 존재하고, 여러 개의 키워드 벡터를 하나로 만드는 과정은 키워드 개수와 순서가 정해져 있지 않다. 그렇기에 단순히 더하거나 평균을 내서 사용하기에는 각 단어들을 아우를 수 있는 하나의 벡터로 만들 수 없다. 이러한 한계를 극복하기 위해 본 연구에서는 오토인코더 모델을 사용한 키워드 임베딩 방법을 새롭게 제안하였고, 기존연구의 한계를 개선한 점에서 학술적 기여를 인정받을 수 있을 것이다. 또한, 사전학습언어모델의 등장 이후, 다양한 연구 기반 기술로 널리 활용되고 있는 가운데 문서 분류 문제에서 저자 또는 전문가에 의해 부여된 정형 또는 반정형 정보를 활용하지 못하던 기존연구 한계를 개선하여 문서 분류 성능 향상을 보였다는 점은 실무적인 측면의 기여로 인정받을 수 있을 것이다.

하지만 본 제안 방법은 문서 임베딩을 추출하기 위해 KoBERT를 사용해야 하고 단어 임베딩을 추출하기 위해 오토인코더 학습이 따로 진행되어야 하며 문서 분류를 위한 분류기 모델 학습까지 진행되어야 한다는 점에서

BERT와 같이 End-to-End 학습이 어렵다는 단점이 있다. 한편으로 단어 고유 정보를 살리기 위해 사용된 오토인코더의 잠재 벡터의 경우, 데이터를 압축하는 과정에서 정보 손실이 발생하는 한계가 있다. 추후, 이러한 한계를 극복하여 단어 정보 손실이 발생하지 않는 또 다른 단어 임베딩 방법을 적용하거나, 문서 임베딩을 추출한 BERT를 활용할 때 단어 고유의미를 보존하면서 문서 분류를 진행하는 연구가 추가로 수행되어야 한다. 또한 본 실험 과정에서 오토인코더의 잠재 벡터의 차원 수에 따라 잠재 벡터값이 분류 성능에 영향을 주는 현상을 확인하였다. 따라서 향후 연구에서는 오토인코더의 잠재 벡터 차원수와 정보 손실의 영향도를 고려한 세밀한 실험이 수행되어야 한다. 또한 성능 평가 측면에서는 정확도뿐 아니라, Precision, Recall, F1 score 등을 사용한 다양한 평가가 이루어질 필요가 있다.

REFERENCES

[1] Y. Saito, “Deep Learning From Scratch 2,” Hanbit Media, 2019.

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

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” Advances in Neural Information Processing Systems, December 2017. DOI: 10.48550/arXiv.1706.03762

[4] D. Kim, J. Park, D. Lee, S. Oh, S. Kwon, I. Lee, and D. Choi, “KB-BERT: Training and Application of Korean Pre-trained Language Model in Financial Domain,” Journal of Intelligence and Information Systems, Vol. 28, No. 2, pp. 191-206, June 2022. DOI: 10.13088/JIIS.2022.28.2.191

[5] E. Yu, N. Kim, and S. Seo, “Building a Specialized Language Model for National R&D through Knowledge Transfer Based on Further Pre-training,” Knowledge Management Review, vol. 22, no. 3, pp. 91-106, Sep 2021. DOI: 10.15813/KMR.2021.22.3.006

[6] H. Park, and K. Shin, “Aspect-Based Sentiment Analysis Using BERT: Developing Aspect Category Sentiment Classification Models,” Journal of Intelligence and Information Systems, vol. 26, no. 4, pp. 1-25, Dec 2020. DOI: 10.13088/JIIS.2020.26.4.001

[7] S. Yoon et al., “Using BERT for the Development of a Classification Model for Rejection Reasons of Trademark Opinion Notices," Management Information Systems Review, vol. 40, no. 3, pp. 41-58, Sep 2021. DOI: 10.29214/damis.2021.40.3.003

[8] S. Hwang, and D. Kim, “BERT-based Classification Model for Korean Documents,” The Journal of Society for e-Business Studies, vol. 25, no. 1, pp. 203-214, Feb 2020. DOI: 10.7838/jsebs.2020.25.1.203

[9] J. Joo et al., “Document Classification using Recurrent Neural Network with Word Sense and Contexts,” KIPS Transactions on Software and Data Engineering, vol. 7, no. 7, pp. 259-266, Jul 2018. DOI: 10.3745/KTSDE.2018.7.7.259

[10] B. Pierre, and K. Hornik, “Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima,” Neural Networks, vol. 2, no. 1, pp. 53-58, Jan 1989. DOI: 10.1016/0893-6080(89)90014-2

[11] S. Jang, and D. Ki, “A Study on the Relationship between Class Similarity and the Performance of Hierarchical Classification Method in a Text Document Classification Problem,” The Journal of Society for e-Business Studies, vol. 25, no. 3, pp. 77-93, Aug 2020. DOI: 10.7838/jsebs.2020.25.3.077

[12] S. Lee, “An Introduction to Machine Learning Focusing on Predictive Models Using Supervised Learning,” Educational Research Institute, College of Education, Ewha Womans University, vol. 53, no. 3, pp. 1-43, Sep 2022. DOI: 10.15854/jes.2022.09.53.3.1

[13] S. Jo, and S. Kang, “Industrial Applications of Machine Learning (Artificial Intelligence),” IE Magazine, vol. 23, no. 2, pp. 34-38, Jun 2016.

[14] J. Kim, “[Special Plan] Artificial Intelligence Technology and Chemical Engineering,” News & Information for Chemical Engineers, vol. 39, no. 2, pp. 179-195, April 2021.

[15] J. Choi, H. Hahn, and Y. Jung, “Research on Text Classification of Research Reports using Korea National Science and Technology Standards Classification Codes,” Journal of the Korea Academia-Industrial Cooperation Society, vol. 21, no. 1, pp. 169-177, Jan 2020. DOI: 10.5762/KAIS.2020.21.1.169

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” arXiv:1301.3781, Sep 2013. DOI: 10.48550/arXiv.1301.3781

[17] J. Pennington, R. Socher, and CD. Manning, “Glove: Global Vectors for Word Representation,” In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543, Oct 2014. DOI: 10.3115/v1/d14-1162

[18] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, Jun 2017. DOI: 10.1162/tacl_a_00051

[19] W. Kim, D. Kim, and H. Jang, “Semantic Extension Search for Documents Using Word2Vec,” The Journal of the Korea Contents Association, vol. 16, no. 10, pp. 687-692, Oct 2016. DOI: 10.5392/jkca.2016.16.10.687

[20] Y. Choi, and S. Choi, “A Study on Patent Literature Classification Using Distributed Representation of Technical Terms,” Journal of the Korean Society for Library and Information Science, vol.

53, no. 2, pp. 179-199, May 2019. DOI: 10.4275/KSLIS.2019.53.2.179

[21] Y. Kim, and S. Lee, “Combinations of Text Preprocessing and Word Embedding Suitable for Neural Network Models for Document Classification,” Journal of KIISE, vol. 45, no. 7, pp. 690-700, Jul 2018. DOI: 10.5626/JOK.2018.45.7.690

[22] Q. Le, and T. Mikolov, “Distributed Representations of Sentences and Documents,” In International Conference on Machine Learning (PMLR), vol. 32, no. 2, pp. 1188-1196, May 2014. DOI: 10.48550/arXiv.1405.4053

[23] ME. Peters, N. Mark, I. Mohit, G. Matt, C. Christopher, and L. Kenton, “Deep Contextualized Word Representations,” arXiv:1802.05365, Mar 2018. DOI: 10.48550/arXiv.1802.05365

[24] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-training,” 2018. URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Authors

Seobin Yoon received the B.E. in Computer Software Engineering, Digital Forensics from Wonkwang University in 2021, M.S. degree in Management Engineering from Graduate School of Business IT, Kookmin University in 2023. She is interested in deep learning, data mining, and natural language processing.

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

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