http://www.jsebs.org
ISSN: 2288-3908

The Journal of Society for e-Business Studies
Vol.25, No.1, February 2020, pp.203-214
https://doi.org/10.7838/jsebs.2020.25.1.203

한국어 기술문서 분석을 위한 BERT 기반의 분류모델

BERT-based Classification Model for Korean Documents

황상흠(Sangheum Hwang)*, 김도현(Dohyun Kim)**

초 록

최근 들어 기술개발 현황, 신규기술 분야 출현, 기술융합과 학제 공동연구, 기술의 트렌드 변화 등을 파악하기 위해 R&D 과제정보, 특허와 같은 기술문서의 분류정보가 많이 활용되고 있다. 이러한 기술문서를 분류하기 위해 주로 텍스트마이닝 기법들이 활용되어 왔다. 그러나 기존 텍스트마이닝 방법들로 기술문서를 분류하기 위해서는 기술문서들을 대표하는 특징들을 직접 추출해야 하는 한계점이 있다. 따라서 본 연구에서는 딥러닝 기반의 BERT모델을 활용하여 기술문서들로부터 자동적으로 문서 특징들을 추출한 후, 이를 문서 분류에 직접 활용하는 모델을 제안하고, 이에 대한 성능을 검증하고자 한다. 이를 위해 텍스트 기반의 국가 R&D 과제 정보를 활용하여 BERT 기반 국가 R&D 과제의 중분류코드 예측 모델을 생성하고 이에 대한 성능을 평가한다.

ABSTRACT

It is necessary to classify technical documents such as patents, R&D project reports in order to understand the trends of technology convergence and interdisciplinary joint research, technology development and so on. Text mining techniques have been mainly used to classify these technical documents. However, in the case of classifying technical documents by text mining algorithms, there is a disadvantage that the features representing technical documents must be directly extracted. In this study, we propose a BERT-based document classification model to automatically extract document features from text information of national R&D projects and to classify them. Then, we verify the applicability and performance of the proposed model for classifying documents.

키워드 : 문서분류, 딥러닝, 단어 임베딩
BERT, Document Classification, Deep Learning

This study was supported by the Research Program funded by the SeoulTech(Seoul National University of Science and Technology).

* First Author, Assistant Professor, Department of Industrial & Information Systems Engineering, Seoul National University of Science and Technology(shwang@seoultech.ac.kr)
** Corresponding Author, Associate Professor, Department of Industrial and Management Engineering, Myongji University(ftgog@mju.ac.kr)

Received: 2020-01-10, Review completed: 2020-02-20, Accepted: 2020-02-25

1. 서 론

국가 R&D의 투자 효율성을 높이고 연구생산성에 기여하기 위해 국가 R&D 사업, 과제, 인력, 연구시설·장비, 성과 등 국가연구개발 사업에 대한 정보를 한 곳에서 관리하는 국가과학기술종합정보시스템(NTIS)이 구축·운영되고 있다[5]. 또한 NTIS에서는 국가 R&D 과제 현황을 효과적으로 관리하고 효율적인 R&D 정보 처리를 위해 분류정보를 제공하고 있다. 연구자들은 분류정보를 통해 기술개발 현황, 신규기술 분야의 출현, 기술융합과 학제 공동연구, 기술의 변화 등을 파악할 수 있다. 일반적으로 NTIS의 분류정보는 연구를 진행한 연구자들로부터 정보를 얻게 되는데 연구자들도 본인의 연구가 정확하게 어떤 분류에 해당되는지에 대한 내용을 정확하게 파악하기 어렵다. 그 이유는 연구내용에 대해서는 잘 이해하고 있지만, 분류 체계에 대한 이해가 부족하기 때문이다. 따라서 데이터 기반의 R&D 과제 분류의 필요성이 대두되고 있다. 데이터 기반의 R&D 과제 분류를 통해 연구자들의 주관적인 의견이 배제된, 보다 객관적인 분류가 가능하다.

R&D 과제정보와 같은 기술문서의 분류를 위한 대표적인 방법으로는 텍스트마이닝을 이용한 문서 분류 방법이 있다. 텍스트마이닝은 자연어로 구성된 비정형 텍스트 데이터에 숨겨진 패턴 또는 관계를 추출하여 의미 있고 가치 있는 정보를 찾아내는 마이닝 기법이다. 텍스트마이닝을 적용하기 위해서는 먼저 각 기술문서들의 내용을 정확하게 요약하는 특징을 추출하고, 이를 특징값의 벡터로 표현하여야 한다. 이를 위해 빈도, 카이제곱 정보량(chi-square statistics), 상호정보량(mutual information) 등의 다양한 측도를 통해 자동으로 문서상의 중요한 요소(키워드, 구문, 문장 등)를 추출하고, 이를 바탕으로 SVM(support vector machine), KNN(k-nearest neighborhood) 등의 데이터마이닝 알고리즘을 통해 기술문서를 분류하게 된다.

문서를 요약하는 중요한 특징을 추출하는 방법은 크게 추출요약(extraction)과 생성요약(abstraction)으로 나누어진다. 추출요약 방법은 존재하는 단어, 구문, 문장 중에서 중요도를 바탕으로 의미있는 요소를 선별하는 작업이며, 생성요약 방법은 시스템이 각 요소들의 내재된 의미를 이해하고 자연어 처리 기술을 바탕으로 문서를 요약하는 것이다. 그러나 지금까지 대부분의 연구는 자연어 처리 기술의 한계로 추출요약 방법 위주로 연구되어 왔다. 대표적인 방법으로 문장의 특성을 통계적으로 분석하여 주제문과 거리가 먼 문장들을 제거해가는 방법, 문서내 단어의 빈도수를 바탕으로 단어별 중요도를 계산하고, 단어 중요도를 바탕으로 문장의 중요도를 결정하는 TF-IDF 방법, 문장들의 유사도를 바탕으로 그린 네트워크상에서 노드(문장)의 중요성을 계산하여 문장의 중요도를 결정하는 그래프 기반 랭킹 방법 등이 있다. 그런데 추출요약 방법에는 문제점이 존재한다. 첫째로 핵심 구문 특성 요인 추출의 어려움이 있다. 즉 기존의 문서 요약 알고리즘의 경우, 키워드 동시 발생빈도 혹은 네트워크 분석 등의 방법을 활용하여 문장/키워드의 중요도를 판단하기 때문에, 구문 패턴 혹은 위치 등과 같은 핵심 문장/키워드들이 지닌 특징들을 밝혀내지 못한다. 둘째로 신규 문서에 대한 적용에 어려움이 있다. 기존 문서 요약 알고리즘은 보유하고 있는 문서에서의 키워드 발생 빈도 혹은 키워드 동시 발생 네트워크 등을 바탕으로

문장/키워드의 중요도를 판단하다보니, 분석에 사용되지 않은 신규 문서에 대한 핵심 문장/키워드 추출에 어려움이 있다. 따라서 기술 문서의 특징을 추출하는 방법으로 생성요약 방법에 대한 요구가 증가되고 있다.

최근 들어 자연어 및 시계열 데이터 분석 영역 특히 문서에 내재된 의미를 이해하고, 요약하는 자연어 처리 분야에서 딥러닝 기반의 모델들이 괄목할 만한 성과를 보이고 있다[7, 11, 12]. 딥러닝은 텍스트 데이터로부터 자동적으로 문서 특징들을 추출하고 및 이에 대한 표현을 가능하게 한다는 특징이 있다. 이에 따라 딥러닝을 기술문서 분류에도 활용하는 연구들이 많이 진행되고 있다[2, 3, 4, 10].

딥러닝 모델 중에서 특히, 2018년 발표된 BERT(Bidirectional Encoder Representations from Transformers) 모델[1]은 기존의 언어 표현 모델들과는 달리 Transformer 모듈[9]을 기반으로 양방향으로 입력 데이터의 맥락(context)을 인코딩한다. 이를 통해 BERT 모델은 자연어 처리와 관련된 많은 과제에 적용 가능한 자연어의 범용적인 수치 표현을 제공한다. 이는 자연어 처리 영역에서 매우 큰 의미를 가진다. 컴퓨터 비전 영역에서 객체 탐지, 객체 분할 등의 다양한 과제에서 딥러닝 모델이 빠른 시간 내에 큰 성과를 거둔 것은 이미지넷으로 사전 학습된 모델이 이미지에 대한 범용적인 특징 벡터를 추출해주기 때문이다. 이렇게 사전 학습된 모델을 기반으로 주어진 과제에 관한 데이터로 fine-tuning을 하게 되면 비교적 적은 수의 데이터로 좋은 결과를 얻을 수 있다. 자연어 처리 영역에서는 이와 같은 사전 학습된 모델을 범용적으로 활용하기 어려워 발전 속도가 더뎠는데 BERT 모델이 그러한 역할을 할 수 있게 되었다. BERT 모델은 대용량의 데이터로부터 주변 의미 정보를 반영한 특징 벡터를 학습하기 때문에 이러한 특징 벡터를 이용하여 다양한 자연어 처리 과제들에 적용 가능하다. Devlin et al.[1]는 BERT 모델을 문장 분류, 자연어 기반 질의 응답 등에 관한 벤치마크 데이터로 fine-tuning하여 가장 좋은 벤치마크 테스트 성능을 보였다. 본 연구에서는 한국어 데이터로 학습된 BERT 모델을 기반으로 인공지능 분과, 지능형로봇 분과에 해당되는 국가과제의 중분류기술명을 예측하는 딥러닝 모델의 예측 성능을 확인하고 이를 바탕으로 한국어 BERT 모델의 적용가능성을 살펴보고자 한다. 이를 위해 제2장에서는 관련연구로 Transformer와 BERT 모델에 대해서 살펴보고, 제3장에서는 한국어 BERT 모델을 기반으로 학습된 기술문서 분류 모델을 소개한다. 또한 제4장에서는 실험 구성과 결과를, 마지막 제5장에서는 본 연구를 통해 얻은 시사점에 대해서 논의하고자 한다.

2. 관련연구

2.1 Transformer

Transformer[9]는 시퀀스 데이터를 모델링함에 있어 attention 기법을 활용하는 기계 번역 모델이다. 기계 번역 영역에서 가장 좋은 성능을 보였던 딥러닝 모델들은 순환 신경망(recurrent neural network) 기반의 방법들이었는데, Transformer는 그보다 더 나은 번역 성능을 보였다. 하지만 더욱 중요한 의의는 데이터 처리 과정의 병렬화에 있다. 순환 신경망 계열의 모델들은 필연적으로 입력 데이터를 순차적으로 처리

해야 하기 때문에 길이가 긴 시퀀스의 경우 학습 시간이 오래 걸리는 단점이 있다. 하지만 Transformer는 주어진 시퀀스를 한 번에 입력으로 받아 처리하도록 설계되어 이러한 단점을 극복하였다.

Transformer는 다른 자연어 처리 모델들과 마찬가지로 크게 인코더-디코더 구조를 가진다. 각각의 인코더, 디코더는 여러 개의 인코더 계층, 디코더 계층이 쌓여 있는 구조이다. Transformer의 인코더 계층은 self-attention 계층과 완전 연결 계층으로 구성되어 있다. Self-attention 계층은 입력 문장에서 각 위치에 해당되는 단어의 의미를 파악할 때 다른 위치에 존재하는 단어들의 의미를 참조하도록 구성되어 있다. 이러한 self-attention 계층을 여러 개 동시에 활용하여 multi-head attention을 구현한다. 이렇게 생성된 여러 self-attention으로부터의 인코딩 벡터는 완전 연결 계층을 거치면서 다시 한 번 인코딩된다.

2.2 BERT

BERT 모델[1]은 Transformer의 인코더를 여러 개 쌓아 구성된 모델로 주어진 맥락에 맞게 특정 단어의 임베딩 벡터를 출력하도록 학습된 모델이다. 대량의 문서 뭉치로 효과적인 임베딩이 학습되도록 사전 학습한 후 이렇게 학습된 모델을 다양한 하위 과제에 fine-tuning 하여 좋은 성능을 달성했다. BERT 모델의 중요한 의의는 사전 학습한 모델의 구조를 유지한 채 다양한 하위 과제들에 대해 좋은 성능을 보였다는 것이다.

BERT 모델의 핵심은 사전 학습 과정에서 단어들에 대한 좋은 임베딩을 학습하는 것이다. 주어진 문장들의 앞뒤 맥락을 모두 고려하기 위해 Transformer의 인코더를 여러 층 쌓았고, 특정 단어를 예측할 때 간접적인 방식으로 스스로 참조하는 것을 막기 위해 입력 단어의 일정 비율을 제거한다. 이렇게 제거된 입력 단어를 출력 계층에서 예측하게 함으로써 단어의 임베딩을 학습한다. 또한, 단어뿐만 아니라 주어진 문장 사이의 관계도 학습하기 위해 두 개의 문장을 주고 각 문장의 관계를 예측하도록 한다. 예를 들어, A와 B 두 개의 문장을 주고 B 문장이 A 문장 이후에 나타나는 문장인지를 묻는 방식이다. BERT 모델의 첫 번째 입력으로 특별히 할당되어 있는 [CLS] 토큰이 들어가기 때문에 이러한 종류의 예측은 첫 번째 위치하는 출력 벡터로 수행하게 된다. 즉, [CLS] 토큰에 해당되는 출력 벡터를 이용하는 것이다. 이 출력 벡터는 본 연구에서 문서 단위의 분류 모델을 만들 때에 활용된다.

이러한 사전 학습 과정을 거치게 되면 BERT 모델은 주어진 입력 문장에 대해 각 단어별로 맥락에 맞는 임베딩 벡터를 출력하게 된다. 이러한 출력 벡터들은 다양한 하위 과제들에 활용될 수 있다. 앞서 설명한 바와 같이 분류 문제의 경우 [CLS] 토큰에 해당되는 출력 벡터를 이용하여 fine-tuning 할 수 있고, 질의응답 과제의 경우 입력으로 질문과 상황에 대한 문단을 주고 출력으로 정답이 나오도록 fine-tuning 할 수 있다. 즉, 사전 학습된 BERT 모델의 구조는 변화시키지 않고 출력 벡터들을 입력으로 받는 몇 개의 계층만 추가하게 되면 해당 과제를 수행하는 모델을 만들 수 있게 된다.

단어 임베딩 방법으로 자주 활용되는 Word2Vec 모델과의 차이점은 BERT는 문맥을 고려한 모델이라는 점이다. 즉, Word2Vec와는 달리, BERT는

문장 형태와 위치에 따라 같은 단어라도 다른 임베딩 벡터값을 갖게 되어 단어의 중의성 문제를 해결할 수 있다. 예를 들어 “bank account”와 “bank of the river”에서 bank의 임베딩 벡터값은 Word2Vec 모델에서는 동일한 반면, BERT에서는 문장에 따라 다른 임베딩 벡터값을 갖게 된다.

3. BERT 기반의 한국어 기술문서 분류 모델

본 장에서는 사전 학습된 한국어 BERT 모델과 국가과제 기술문서 데이터에 대한 설명, 그리고 한국어 BERT 모델을 기반으로 학습된 기술문서 분류 모델에 대해 소개한다.

3.1 한국어 BERT와 기술 문서 데이터

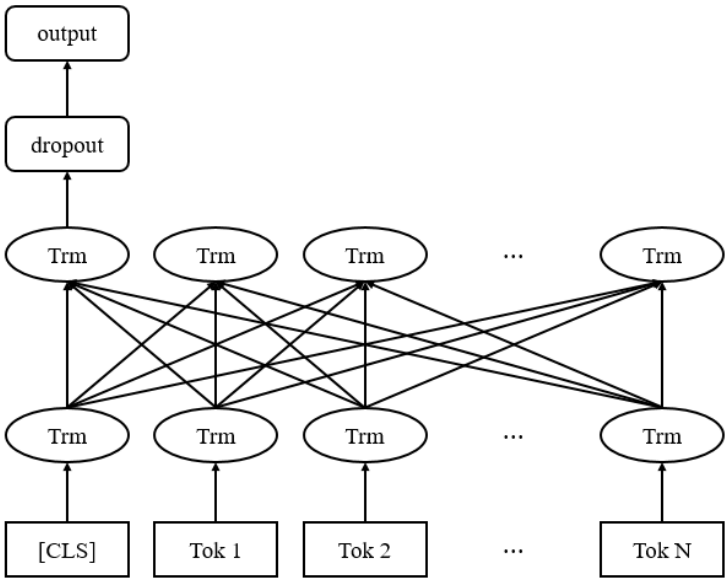
본 연구에서는 SK TBrain에서 공개한 한국어 BERT 모델(https://github.com/SKTBrain/KoBERT)을 이용하여 분류 모델을 학습하였다. 이 BERT 모델은 구글에서 개발한 원래의 BERT 모델과 같은 구조를 가지고 있다. 구글에서 공개한 다국어 지원 BERT 모델도 활용할 수 있지만 한국어에 특화되어 있지 않아 한국어 데이터에 대해 최적의 성능을 보이지 않는다. 본 연구에서 사용한 BERT 모델은 500만 개 이상의 문장으로 구성된 한국어 위키와 2천만 개 이상의 문장으로 구성된 한국어 뉴스 데이터로 학습되었다. 이 BERT 모델의 사전의 크기는 8,002이고 한국어 텍스트를 토큰화하기 위해 한글 위키와 뉴스 텍스트를 기반으로 문장 단위 토크나이저(SentencePiece tokenizer)를 따로 학습하여 제공한다.

본 연구에서 활용한 데이터는 국가과학기술지식정보서비스(NTIS)에 등록되어 있는 인공지능 분과와 지능형로봇 분과의 국가과제 데이터이다. 인공지능 분과에 해당되는 기술문서는 총 4,159개이고 지능형로봇 분과에 해당되는 기술문서는 총 2,959개이다. 두 분야의 기술문서는 공통으로 33개의 중분류기술명로 분류되기 때문에 본 연구에서는 두 분야의 기술문서를 모두 합쳐 총 7,118개의 데이터를 학습 대상으로 했다. 개별 기술문서는 국문과제명, 연구목표요약, 연구내용요약, 과제의 한글키워드, 과제의 영문키워드 정보를 가지고 있는데 이 중에서 과제의 영문키워드는 제외했고, 나머지 정보들은 모두 순서대로 나열하여 통합했다. 즉, 개별 기술문서는 하나의 문장으로 이루어진 것으로 취급했다.

3.2 BERT 기반의 기술문서 분류 모델

본 연구에서의 문서 분류 모델은 <Figure 1>과 같다. <Figure 1>에서 [CLS]는 모든 입력 문장의 시작을 나타내는 특별한 토큰이고, Tok 1부터 Tok N은 토큰화 과정을 거친 입력 문장의 각 토큰들을 의미한다. 토큰화 과정은 한글 위키와 뉴스 텍스트 데이터를 기반으로 학습된 토크나이저를 활용한다(https://github.com/SKTBrain). 또한 기술 문서의 과제명, 과제요약, 기대효과 등 다양한 속성의 텍스트들은 그 속성을 무시하고 하나의 문장으로 취급하여 입력으로 사용한다.

입력된 문장의 임베딩 벡터는 여러 층으로 쌓인 Transformer(<Figure 1>에서 Trm)들로부터 도출된다. 구체적으로, 사전 학습된 BERT 모델의 풀링된 출력 벡터, 즉 <Figure 1>에서



〈Figure 3〉 BERT-based Document Classification Model

[CLS] 토큰에 해당되는 출력을 입력으로 사용하여 분류기를 구성한다. 분류기는 과적합을 막기 위한 dropout[8] 계층과 완전 연결 계층으로 구성된다. 본 분류 모델에서는 dropout 비율로 0.1을 사용했다. [CLS] 토큰의 출력에 해당되는 768차원의 임베딩 벡터는 dropout 계층과 완전 연결 계층을 거쳐 총 33차원의 벡터를 출력하고 이 벡터의 요소들은 각각 특정 기술분야에 속할 확률을 나타낸다.

4. 실험 구성 및 결과

4.1 실험 구성 및 학습 방법

실험을 위해 주어진 한국어 기술문서 데이터는 최소한의 전처리 과정을 거쳤다. 먼저 문자가 아닌 특별한 기호 등은 모두 제거하였고 각 문서의 중분류기술명이 숫자가 아닌 경우는 학습 데이터에서 제외하였다. 이렇게 해서 추려진 데이터의 총 개수는 7,108개이다. 이 중에서 무작위로 추출된 약 30%의 데이터는 학습된 분류 모델의 성능 평가에 활용했다. 따라서 학습에 활용된 데이터는 총 4,976개, 테스트에 활용된 데이터는 총 2,132개이다.

설계된 BERT 모델은 fine-tuning 방식으로 학습되었다. 모델을 학습하기 위해 Adam optimizer[6]를 사용했고 초기 학습률은 0.00005로 했다. 학습은 총 50 epoch 수행했고, minibatch의 크기는 32로 했다.

주어진 문제는 개별 기술문서에 대해 총 33개의 중분류기술명 중에서 각각의 기술명에 속하는지를 분류하는 문제이기 때문에 개별 중분류 기술명에 속하는지 여부를 손실함수로 모델링했다. 따라서 개별 중분류 기술명에 속하는지 여부가 이진 분류 문제가 되고 모든 기술

분류에 대해 평균적인 손실을 계산하여 개별 문서의 손실값을 계산한다.

각각의 기술문서는 소수의 기술분야에 속하기 때문에 주어진 데이터는 클래스 불균형 문제를 안고 있다. 이에 대한 영향을 줄이기 위해 positive 클래스에 대해 가중치를 부여하는 방식으로 최종 손실함수를 구성했다. 최적의 가중치는 3이었고 이는 반복 실험을 통해 결정되었다.

4.2 실험 결과

학습된 분류 모델의 성능은 테스트 데이터에 대해 개별요소 단위, 기술분류 단위, 문서 단위로 나누어 평가했다.

4.2.1 개별요소 단위 성능

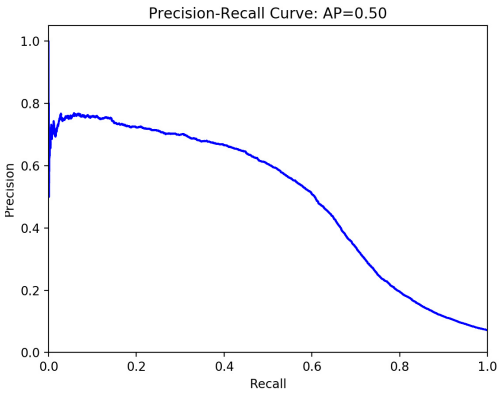
개별요소 단위의 성능은 개별 테스트 기술문서에 대한 33개의 예측 값을 모두 독립적으로 보고 평가한 것이다. 총 2,132개의 테스트용 기술문서가 있고 개별 문서는 33개의 독립적인 예측을 수행하기 때문에 개별요소 단위 평가의 대상이 되는 예측값은 총 70,356개이다. 이 중 실제로 positive 클래스에 해당되는 개별요소는 5,035개로 클래스 불균형이 매우 심한 것을 알 수 있다.

<Table 1>은 개별요소 예측의 성능을 평가한 것이다. 평가 측도로는 정확도, F-score, 정밀도, 재현율을 사용했다. 위에서 밝힌 바와 같이 클래스 불균형이 매우 심하기 때문에 정확도 성능 측도는 크게 의미가 없다. 예를 들어 모든 개별 예측이 negative 클래스라면 산술적으로 약 92.84%의 예측 정확도를 얻게 된다. 이러한 경우 유효한 성능 측도는 정밀도, 재현율, F-score 등이다. 이 성능 측도들의 값은 대략 0.5 이상으로 학습된 모델이 우수하게 예측을 수행하는 것으로 평가할 수 있다. 현재 학습된 분류 모델의 정밀도, 재현율 값이 대략 0.55 정도이다. 정밀도가 0.55라는 것의 의미는 모델이 예측한 positive 클래스들 중 실제 positive인 비율이 0.55라는 것이다. 즉, 모델에 의해 예측된 기술분류 2개 중 실제 기술분류에 1개 이상이 포함된다는 의미이다. 반면, 재현율이 0.55라는 의미는 주어진 실제 positive들 중에서 모델이 절반 정도는 맞췄다는 것이다. 이는 실제 기술분류 2개 중에서 모델이 정확하게 1개 이상은 예측한다는 의미를 가진다. 그리고 F-score는 정밀도와 재현율의 조화평균값이다. 본 연구에서 사용된 기술문서 데이터의 클래스 불균형이 심하다는 점, 개별 기술문서가 속하는 기술분류의 수가 상당히 적다는 점 등을 고려할 때, 우수한 예측 성능을 보인다고 판단할 수 있다. 이러한 데이터의 특성들은 뒤이어 기술된 분석 결과들을 통해 알아본다.

〈Table 1〉 Prediction Performances Averaged over all Outputs

Performance measures	Values
Accuracy	0.9368
F-score	0.5561
Precision	0.5591
Recall	0.5531

<Figure 2>는 임계치에 따른 정밀도와 재현율 값의 변화를 나타내는 정밀도-재현율 곡선이다. 이 곡선 아래 면적이 평균 정밀도를 나타내는 것으로 학습된 모델은 최종적으로 0.5 정도의 성능을 보였다.



〈Figure 2〉 Precision-Recall Curve on Categories

4.2.2 기술분류 단위 성능

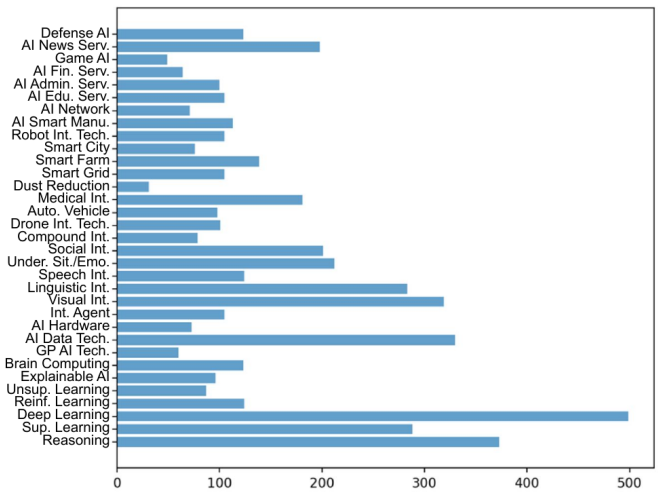
기술분류 단위의 성능을 보기 위해 총 33개의 기술분류들을 대상으로 각각의 기술분류에 대한 예측성능을 평가했다. 먼저 기술분류 단위로 테스트 데이터의 분포를 시각화하면 <Figure 3>과 같다. “딥러닝” 기술로 분류된 문서가 가장 많았고 “미세먼지저감” 기술로 분류된 문서가 가장 적었다.

성능을 평가하기 위해 구체적으로 개별 기술분류에 대해 각각 F-score, 정밀도, 재현율 값을 계산하고 이의 평균값을 산출하였다. 정확도의 평균값은 개별요소 단위의 예측 성능과 동일하기 때문에 생략하였다. <Table 2>는 기술분류 단위의 예측성능을 나타낸다. 개별 기술분류에 대해 평균적으로 0.54 정도의 F-score 값을 보였고, 정밀도와 재현율 모두 평균적으로 0.5 이상의 성능을 보였다.

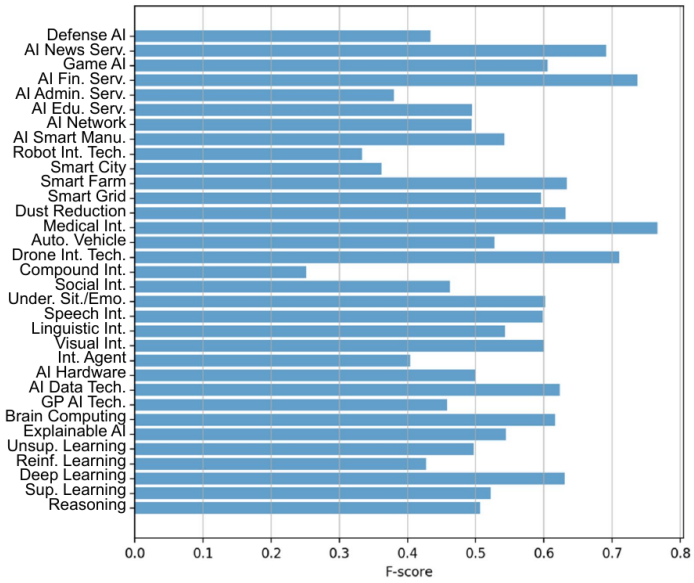
〈Table 2〉 Prediction Performances Averaged over Categories

Performance measures	Values
F-score	0.5372
Precision	0.5353
Recall	0.5454

기술분류 별로 평균 F-score 값을 그려보면 <Figure 4>와 같다. “정밀의료”, “AI기반금융” 등의 기술분류에 대해 예측 성능이 가장 높은데 이는 이러한 기술분야들은 특정 단어들을



〈Figure 3〉 Actual Category Histogram of Test Data



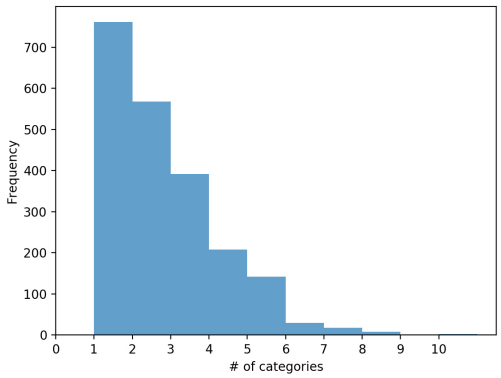
〈Figure 4〉 F-score of Technology Categories

이용하여 분야를 특징짓는 것이 상대적으로 용이하기 때문인 것으로 판단된다. 반면, “복합지능” 기술에 대해서는 예측 성능이 낮은데 이 기술분야 자체가 명확하게 특징짓기 어렵기 때문인 것으로 생각된다.

4.2.3 문서 단위 성능

마지막으로 문서 단위로 예측성능을 평가했다. 먼저 개별 문서들이 몇 개의 기술분야에 속하는지 그 분포를 시각화해보면 <Figure 5>와 같다. 전체 테스트 데이터의 79.6%인 1,698개의 기술문서들이 1개 내지 3개의 기술분야에 속하고, 8개 이상의 기술분야에 동시에 속하는 기술문서도 소수 존재하는 것을 알 수 있다.

여기서는 개별 문서 별로 F-score, 정밀도, 재현율을 계산하고 전체 문서에 대해 평균을 내어 성능을 평가하였다. 그 성능은 <Table 3>과 같다.



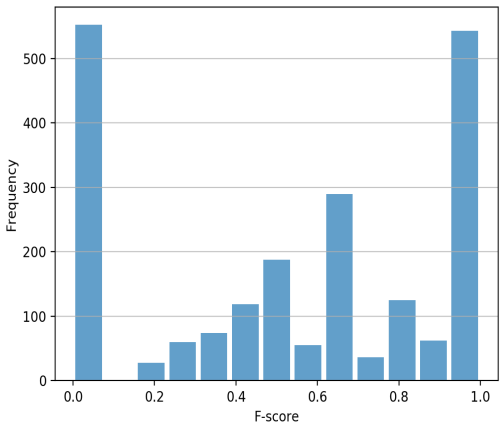
〈Figure 5〉 Document Frequency with Respect to the Number of Categories

〈Table 3〉 Prediction Performances Averaged over Documents

Performance measures	Values
F-score	0.5336
Precision	0.5625
Recall	0.5655

앞선 실험결과와 마찬가지로 대략 0.5 이상의 F-score, 정밀도, 재현율 값을 보였다. 문서 단위로 예측하는 관점에서 이러한 예측 성능을 가늠해보면 다음과 같다. 예를 들어, 실제 2개의 기술분야에 속하는 문서가 있다고 하자. 이 데이터에 대해 모델이 2개의 기술분야를 예측했는데 이 중 한 개만 실제 기술분야일 때 정밀도 값과 재현율 값 모두 0.5가 된다. 주어진 데이터의 대부분이 1개에서 3개의 기술분야에 속한다는 사실과 고려하는 기술분야가 총 33개라는 사실을 고려했을 때 학습된 모델이 일정 수준 이상의 예측력을 보인다고 말할 수 있다. 또한, 단지 1개의 기술분야에 속하는 문서들의 경우 이를 맞추지 못하면 정밀도와 재현율 값이 0이 된다는 사실을 고려하면 더욱 그렇다.

<Figure 6>은 개별 문서 단위의 F-score 값의 분포를 나타낸 것이다. 여기서 536개의 문서의 F-score 값이 0인 것을 알 수 있는데 이 중 334개의 문서가 단지 1개의 기술분야에 속하는 것들이다.



〈Figure 6〉 Document Frequency with Respect to F-score

5. 결론 및 시사점

본 연구에서는 한국어 BERT 기반 분류 모델의 기술문서 분류 예측 가능성을 확인했다. 이를 위해 사전 학습된 한국어 BERT 모델을 fine-tuning하여 분류 모델을 학습했다. 분류 모델의 학습을 위해 7,000건 이상의 국가과제 기술문서를 데이터로 활용했고 주어진 기술문서가 해당되는 총 33개의 기술분류를 각각 예측하도록 했다.

학습된 모델의 성능 평가 결과, 데이터와 주어진 과제의 특성을 고려했을 때 어느 정도 예측력을 보임을 확인했다. 학습된 모델은 평균적으로 0.5 이상의 F-score 및 평균 정밀도 값을 보였고, 문서단위의 분류에서는 이러한 성능이 실제로도 의미가 있음을 확인했다. 본 연구를 통해 한국어 BERT 기반 분류 모델이 기술문서 분류에 활용가능하다는 점을 확인하였으며, 향후에 한국어 BERT 기반 분류 모델이 다양한 한글 문서의 분류에 활용될 수 있을 것으로 기대한다.

본 연구에서는 주어진 기술문서에 대한 정보를 모두 하나의 문장으로 처리했다는 점에서 한계를 가진다. 추후 연구과제로는 과제명, 연구목표요약, 연구내용요약 등의 변수들을 개별적으로 처리하여 학습시키는 방식으로 분류 모델의 성능을 높이는 고도화 연구가 포함될 수 있다. 또한, 학습 데이터의 수를 더 확보하여 모델을 학습하면 사람의 문서 의미 파악에 실제로 근접한 분류 성능을 가지는 모델을 얻을 수 있을 것으로 생각한다.

References

[1] Devlin, J., Chang, M. W., and Lee, K. T., "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv:1810.04805, 2018.

[2] Jo, H., Kim, J. H., Yoon, S., Kim, K. M., and Zhang, B. T., "Large-scale text classification methodology with convolutional neural network," Proceedings of the 2015 Korean Information Science Society Conference, pp. 792-794, 2015.

[3] Kim, J. M. and Lee, J. H., "Text document classification based on recurrent neural network using word2vec," Journal of Korean Institute of Intelligent Systems, Vol. 27, No. 6, 2017.

[4] Kim, Y., "Convolutional neural network for sentence classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751, 2014.

[5] Kim, Y. J., Kim, T. H., Lim, C. S., and Kim, J. S., "A study on NTIS standard code and classification service development," Proceedings of the 2007 Korea Contents Association Conference, pp. 376-380, 2007.

[6] Kingma, D. and Ba, J., "Adam: A method for stochastic optimization," Proceedings of the 3rd International Conference on Learning Representations, 2015.

[7] Oh, S. W., Lee, H., Shin, J. Y., and Lee, J. H., "Antibiotics-resistant bacteria infection prediction based on deep learning," The Journal of Society for e-Business Studies, Vol. 24, No. 1, pp. 105-120, 2019.

[8] Srivastava, N., Hinton, G., krizhevsky, A., Sutskever, I., and Salakhutdinov, R., "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, Vol. 15, pp. 1929-1958, 2014.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., "Attention is all you need," Proceedings of the 31st Conference on Neural Information Processing Systems, 2017.

[10] Yang, Y. J., Lee, B. H., Kim, J. S., and Lee, K. Y., "Development of an automatic classification system for game reviews based on word embedding and vector similarity," The Journal of Society for e-Business Studies, Vol. 24, No. 2, pp. 1-14, 2019.

[11] Yoon, D., Kim, S., and Kim, D., "Clustering of time series data using deep learning," Journal of Applied Reliability, Vol. 19, No. 2, pp. 167-178, 2019.

[12] Young, T., Hazarika, D., Poria, S., and Cambria, E., "Recent trends in deep learning based natural language processing," arXiv:1708.02709, 2017.

저 자 소 개

황상흠	(E-mail: shwang@seoultech.ac.kr)
2005년	KAIST 산업공학 (학사)
2012년	KAIST 산업및시스템공학 (박사)
2012년~2014년	삼성전자 종합기술원 전문연구원
2015년~2018년	Lunit Inc. 선임연구원
2018년~현재	서울과학기술대학교 글로벌융합산업공학과 조교수
관심분야	데이터마이닝, 기계학습

김도현	(E-mail: ftgog@mju.ac.kr)
2000년	KAIST 산업공학 (학사)
2002년	KAIST 산업공학 (석사)
2007년	KAIST 산업공학 (박사)
2011년~2014년	한국과학기술정보연구원 선임연구원
2014년~현재	명지대학교 산업경영공학과 부교수
관심분야	데이터마이닝, 기계학습, 과학계량학