Journal of the Korea Academia-Industrial cooperation Society
Vol. 21, No. 1 pp. 169-177, 2020

https://doi.org/10.5762/KAIS.2020.21.1.169
ISSN 1975-4701 / eISSN 2288-4688

국가 과학기술 표준분류 체계 기반 연구보고서 문서의 자동 분류 연구

최종윤[1], 한혁[2], 정유철[1*]
[1]금오공과대학교 컴퓨터공학과, [2]한국과학기술정보연구원

Research on Text Classification of Research Reports using Korea National Science and Technology Standards Classification Codes

Jong-Yun Choi[1], Hyuk Hahn[2], Yuchul Jung[1*]
[1]Department of Computer Engineering, Kumoh National Institute of Technology
[2]Korea Institute of Science and Technology Information

요 약 과학기술 분야의 연구·개발 결과는 연구보고서 형태로 국가과학기술정보서비스(NTIS)에 제출된다. 각 연구보고서는 국가과학기술 표준 분류체계 (K-NSCC)에 따른 분류코드를 가지고 있는데, 보고서 작성자가 제출 시에 수동으로 입력하게끔 되어있다. 하지만 2000여 개가 넘는 세분류를 가지고 있기에, 분류체계에 대한 정확한 이해가 없이는 부정확한 분류코드를 선택하기 십상이다. 새로이 수집되는 연구보고서의 양과 다양성을 고려해 볼 때, 이들을 기계적으로 보다 정확하게 분류할 수 있다면 보고서 제출자의 수고를 덜어줄 수 있을 뿐만 아니라, 다른 부가 가치적인 분석 서비스들과의 연계가 수월할 것이다. 하지만, 국내에서 과학기술표준 분류체계에 기반을 둔 문서 자동 분류 연구 사례는 거의 없으며 공개된 학습데이터도 전무하다. 본 연구는 KISTI가 보유하고 있는 최근 5년간 (2013년~2017년) NTIS 연구보고서 메타정보를 활용한 최초의 시도로써, 방대한 과학기술표준 분류체계를 기반으로 하는 국내 연구보고서들을 대상으로 높은 성능을 보이는 문서 자동 분류기법을 도출하는 연구를 진행하였다. 이를 위해, 과학기술 표준분류 체계에서 과학기술 분야의 연구보고서를 분류하기에 적합한 중분류 210여 개를 선별하였으며, 연구보고서 메타 데이터의 특성을 고려한 전처리를 진행하였다. 특히, 가장 영향력 있는 필드인 과제명(제목)과 키워드만을 이용한 TK_CNN 기반의 딥러닝 기법을 제안한다. 제안 모델은 텍스트 분류에서 좋은 성능을 보이고 있는 기계학습법들 (예, Linear SVC, CNN, GRU등)과 비교하였으며, Top-3 F1점수 기준으로 1~7%에 이르는 성능 우위를 확인하였다.

Abstract In South Korea, the results of R&D in science and technology are submitted to the National Science and Technology Information Service (NTIS) in reports that have Korea national science and technology standard classification codes (K-NSCC). However, considering there are more than 2000 sub-categories, it is non-trivial to choose correct classification codes without a clear understanding of the K-NSCC. In addition, there are few cases of automatic document classification research based on the K-NSCC, and there are no training data in the public domain. To the best of our knowledge, this study is the first attempt to build a highly performing K-NSCC classification system based on NTIS report meta-information from the last five years (2013-2017). To this end, about 210 mid-level categories were selected, and we conducted preprocessing considering the characteristics of research report metadata. More specifically, we propose a convolutional neural network (CNN) technique using only task names and keywords, which are the most influential fields. The proposed model is compared with several machine learning methods (e.g., the linear support vector classifier, CNN, gated recurrent unit, etc.) that show good performance in text classification, and that have a performance advantage of 1% to 7% based on a top-three F1 score.

Keywords : Deep Learning, Text Classification, Research Report, Preprocessing, NTIS

이 연구는 금오공과대학교 학술연구비로 지원되었음. (2018-104-147)
*Corresponding Author : Yuchul Jung(Kumoh National Institute of Technology)
email: jyc@kumoh.ac.kr
Received September 5, 2019 Revised November 18, 2019
Accepted January 3, 2020 Published January 31, 2020

1. 서론

우리나라의 국가 연구개발(R&D) 과제들의 결과물은 연구보고서 형태로 국가과학기술지식정보서비스(NTIS) 시스템에 제출되고 있으며, 과학기술 분야의 경우 그 주제 범위가 33개의 대분류, 371개의 중분류, 2,898개의 소분류로 이뤄져 있을 만큼 매우 다양하다. 이러한 연구보고서는 제출 당시 저자들은 제출시스템에서 제공되는 분류체계를 참고하여 가장 관련 있는 분류코드를 정하여 제출하게 된다. 하지만, 제출시스템 차원에서 연구보고서에 적합한 분류코드들을 추천해 줄 수 있다면 연구자는 복잡한 분류체계를 모두 이해하지 않고서도 매우 적절한 분류코드를 제출 시에 결정할 수 있을 것이다. 그런 측면에서, 연구보고서의 자동 분류성능에 관한 연구는 그 의미가 크다고 하겠다.

NTIS 시스템에서 관리되는 연구보고서의 분류체계는 과학기술 정통부의 국가 과학기술 표준 분류체계 (National Science & Technology Standards Classification Codes)[1]를 기본으로 하고 있다. 연구보고서의 메타정보는 과제명, 연구목표 요약, 기대효과 요약과 같은 항목을 가지고 있기는 하지만, 작성자에 따라 그 내부 작성형태는 매우 다양하다. 예를 들어 과제명과 키워드에는 연구 분류에 대한 핵심단어를 가지고 있지만, 그 외의 필드 값들은 정해진 형식이 따로 존재하지 않아, 매우 자유롭게 기술되어 있다. 이 부분은 일반적인 문서의 형태와는 다소 다른 형태이다.

본 연구에서는 연구보고서 메타정보의 특성을 고려하여 가장 영향력 있는 자질인 과제명과 키워드만을 입력으로 하는 TK_CNN기법을 제안한다. 또한 제안 기법은 최근 수행되는 전처리 기법, 워드임베딩, 그리고 텍스트 분류에서 많이 사용되는 기계학습기법들과 조합 실험 후 비교를 진행하였다. 특히, 데이터 전처리에 의한 차이를 보기 위해 단어 레벨 및 문자 단위에서의 데이터 처리를 진행하였으며, Word2Vec 및 Glove와 같은 임베딩 기법도 실험도 채택하였다. 그리고, 텍스트 분류에서 널리 쓰이는 SVM 기법과 최근 각광받고 있는 딥러닝 (Deep Learning) 알고리즘들인 CNN 및 RNN 계열의 알고리즘을 채택하되, 메타정보의 필드별 특성을 고려한 실험을 진행하였다.

일련의 실험을 통해, 제안기법인 TK_CNN이 여러 조합의 실험들 중에서 가장 간단하면서도 좋은 성능을 보였다. (Top-3 F1점수 기준 86%). 이는 모든 자질을 사용하는 ALL_CNN대비 약 5%가량 높은 성능이며, 일반적으로 Feature engineering을 하지 않는 딥러닝 기법에서도 자질 선택이 매우 중요함을 확인시키고 있다. 본 연구에서 차별화되는 기여도는 다음과 같다.

(1) 최근 5년간 (2013-2017년) 국내 NTIS 연구보고서 20만 건을 대상으로 과학기술 표준분류 체계를 적용한 최초의 연구이다.
(2) NTIS 연구보고서 메타정보들을 대상으로 가장 영향력 있는 제목과 키워드 필드를 사용한 TK_CNN 기반의 자동분류 모델을 제안하였다.
(3) 최근 자동문서분류에서 좋은 성능을 보이는 알고리즘들과 제안 모델 TK_CNN을 비교한 실험결과를 토대로 성능 우위를 검증하였다.

본 논문에서는 2장에서 이전 연구를 소개하며, 3장에서 본 논문에서 사용한 알고리즘과 새롭게 제안하는 문서 분석을 위한 알고리즘을 소개하였다. 4장에서는 제안 모델을 이용한 실험을 진행하고 결과를 기술하였으며, 5장에 서는 본 연구의 결과에 대한 분석을 시도하였으며, 6장에서 연구의 결론 및 향후 연구 방향을 제시하고 있다.

2. 본론

텍스트 분류는 디지털 문서가 생겨난 이후로 계속적으로 중요도가 증가하고 있는 연구 주제이다. 텍스트 분류는 한정된 범주의 분류체계를 기준으로 자동화된 기법으로 처리하고자 하는 다양한 시도가 있었다. 텍스트 분류는 스팸 메일을 판단하거나 영화의 댓글을 통해 반응을 구별하는 것과 같은 2가지의 분류코드가 존재하는 경우에서부터, 20 Newsgroups와 본 연구와 같이 수십 ~ 수백 여개의 분류 중에서 적합한 분류코드를 찾는 다중 분류가 있다. 이러한 텍스트 분류에 쓰인 알고리즘으로는 SVM이 가장 대표적이다. 최근에는 딥러닝 계열의 알고리즘들인 CNN[2]과 RNN[3]이 많이 쓰인다.

2.1 Support Vector Machine (SVM)

계속적으로 증가하는 디지털 텍스트 문서에 대해서 이를 자동으로 분류할 방법이 필요해졌으며 이를 자동화하고자 기계학습기법을 도입하게 되었다 [4]. SVM [5]은 이진 분류와 같은 선형 및 비선형 다항식에서의 벡터들의 최대 마진을 구하는 평면을 찾는 형식으로 훈련의 속도가 느림에도 불구하고 다른 기계학습에 비해 과적합되는 경우가 낮아 많은 분류에서 사용이 되고 있다. 또한,

다중 분류에 대해서도 충분히 좋은 결과 (약 90% 내외)를 보인다고 알려져 있다 [6].

2.2 Deep Neural Network Algorithms

최근 들어서 하드웨어의 발전과 딥러닝 (Deep Learning)의 연구가 활발해지며 대량의 데이터를 한 번에 처리 가능한 환경이 생기며 뉴럴넷 모델들이 주목을 받기 시작했다. 대표적으로 Convolutional Neural Network (CNN), Recurrent Neural Network (RNN)이 있다.

CNN은 이미지 처리를 위해 처음 사용되었는데, 텍스트 분야에서는 [2]의 연구에서 처음 소개되었으며 이미지 처리와는 다르게 1차원의 Convolution을 이용하여 계산한다. 최근, CNN 하나만 사용하는 것이 아닌 CNN과 RNN 혹은 CNN과 SVM을 결합하는 하이브리드 형태의 결합기법이 연구되기도 하였다.

RNN모델[3]은 문장의 순서에 따른 텍스트 데이터의 시간적 상관성을 얻어올 수 있으며. 텍스트 분류뿐만이 아닌 다양한 텍스트 학습에서 사용되고 있다.

3. 제안 기법

3.1 국가과학기술표준분류체계 (K-NSCC)

국가과학기술표준분류체계는 과학기술 관련 정보. 연구 개발사업 등을 효율적으로 관리하기 위해 만들어졌으며 2015년 개정 기준 33개의 대분류와 369개의 중분류 2,899개의 소분류로 이루어져 있다[7]. 대분류는 연구와 적용 분야로 2차원 분류체계를 도입하면서 OECD 연구개발 활동조사지침 및 대다수 국가의 R&D 통계 범위와 인문, 사회과학 분야가 포함되어 있다. 중, 소분류는 분야별 자체 분류체계와의 호환성을 제공하며 소분류 복수선택 및 가중치 도입을 통한 융합기술 등 신기술의 발전 추세를 보다 정확하게 표현하고자 하였다.

이러한 표준분류체계 외에도 연구관리 전문기관 등에서 각 수요 주체의 실무를 위해 세분화 되어있는 자체 분류체계를 활용하였으며 국가연구개발사업의 연구기획, 평과 및 관리, 과학기술예측 및 기술 수준 평가, 과학기술 지식, 정보의 관리 유통 중에 활용이 될 수 있다.

하지만 이 분류체계의 활용도는 아직 미흡한 수준이며, 관련 오픈소스나 공개된 자동분류 시스템은 존재하지 않는 실정이다.

Table 1. National Science and Technology Standard Classification System, as of 2015 by Ministry of Science and ICT

Code System			Section	Division	Group
Field of Research	Science and Technology	Nature	4	47	347
		Life	3	49	447
		Artificial Object	9	111	854
	Humanities and Social Sciences	Human	5	61	547
		Society	9	88	634
		Human Science and Technology	3	13	70

3.2 데이터의 선정 및 전처리

본 연구에서는 NTIS 과제보고서 중 2013년부터 2017년까지 최근 5개년도의 데이터를 대상 데이터로 선정하였다. 본 연구에서 사용한 데이터는 Table 1에서 보는 바와 같이 국가과학기술표준분류체계의 중분류를 기준으로 최소한이 학습이 가능한 100개 이상의 문서를 가지고 있는 과학기술 분야 210개의 중분류코드를 선정하였다. 단, 중분류가 '00'이거나 '99'인 경우 대분류 내에서 명확한 분류 기준이 없는 분류이기 때문에 해당 분류는 제외하였다.

각 문서에서 사용 가능한 데이터는 Table 2에서 보는 바와 같이 여러 개의 범주로 국문 과제명, 영문과제명, 연구목표 요약, 연구내용 요약, 기대효과 요약, 과제 한글 키워드, 과제 영문키워드가 있다. 문서의 카테고리를 분류하기 위해 모든 범주를 사용하여 학습을 시도하였으며 이때 모든 텍스트가 문장의 형식으로 이루어져 있는 것이 아니기 때문에 문서를 분류하기 위해 본 연구에서는 모든 데이터를 일괄적으로 처리하는 방식과 각 데이터를 별도로 처리하는 방법을 고려하였다. 모든 메타 데이터를 하나의 데이터로 합쳐 사용할 시 텍스트의 길이가 길어지며, 이는 CNN과 같은 알고리즘에서 고정 길이로 입력데이터를 제한할 때 핵심단어나 다른 데이터에 비해 많은 양의 데이터를 가지고 있는 데이터의 손실이 발생할 수 있다.

데이터를 일괄적으로 처리할 시에는 해당 문서의 자주 출연하는 단어 및 키워드에 대해서 임베딩 시 좋은 결과를 보여줄 수 있지만 각 카테고리를 연결하는 과정이나 데이터를 정규화해주는 과정에서 데이터의 손실이 쉽게 일어난다. 반대로 각 데이터를 따로 처리할 시에는 데이터의 손실을 하나의 데이터로 합쳐 사용하는 방법보다는 줄어들지만, 각 데이터별 처리방식이 다르며 새로 들어오

는 문서에 대해서 데이터가 없을 시 학습 과정에서 모델에 큰 영향을 줄 수가 있다.

연구보고서의 메타 데이터들에서는 과제의 키워드와 과제명에 자주 나타나는 단어는 대부분 범주를 대표하는 경향을 보였다. 이러한 근거에 기반하여, 각 범주에 해당하는 대표적 단어들을 선별·분리 등의 과정을 거쳐 학습을 진행하였으며, 이때 단어의 토큰화 및 워드 임베딩은 모든 데이터에 대해서 진행하였다.

3.3 적용 알고리즘들

본 장에서는 제안한 TK_CNN기법과 더불어 텍스트 분류에서 좋은 성능을 보여주고 있는, SVM, CNN, LSTM/GRU 등의 알고리즘에 대해 소개한다. 특히, 성능 향상을 위해 선택적으로 사용되는 워드임베딩 기법과 연구보고서 각 필드 별 특성을 감안하기 위해 Concatenation Model에 대해서도 소개한다.

1) Linear Support Vector Classification (Linear SVC): 텍스트 분류 태스크에서 매우 높은 성능 [8]을 보이고 있는 알고리즘이다. SVM은 과적합을 사용하여 기능의 수에 의존하지 않은 SVM은 서로 관련이 없는 고차원 공간에서 큰 효율을 보이지 못하지만[8] 텍스트 분류에서 각각의 단어는 관련이 없는 경우가 희소하기 때문에 좋은 성능을 보인다. 또한, 대부분의 텍스트 분류 문제는 선형으로 분리할 수 있으며 SVM은 이러한 선형 분리에 대해서 효율적인 성능을 보여주어 텍스트 분류 연구에 있어서 많이 사용되고 있다. SVM에 기반을 두어 만들어진 Support Vector Classification (SVC)[9]는 학습 데이터의 부분집합에만 의존하여 학습을 진행하면서 손실함수를 벗어난 training point를 고려하지 않으며 대규모 데이터 셋에 적합하다고 알려져 있다. 본 연구에서는 SVM계열의 많은 변형알고리즘들 중 LinearSVC가 가장 높은 성능을 보여 비교 알고리즘으로 채택하였다.

2) Convolutional Neural Network (CNN): CNN을 이용한 텍스트 분류의 경우는 [2]의 논문에서 소개되었으며 문장의 단어 벡터에 대해서 임베딩된 데이터를 이용하여 학습을 진행한다. [2]의 경우 단어의 개수에 따라 사전이 너무 커지는 문제가 발생하거나 사전에 없는 단어가 나타날 경우 문제가 발생할 수 있는데, 이러한 방법을 해결하기 위해 [10]에서는 데이터의 단위를 단어에서 문자(character) 단위로 고려하는 연구를 진행하였다. 이 방법은 사전의 크기를 줄일 뿐만 아니라 새로운 데이터 셋에 대해서도 별다른 처리 없이 사용할 수 있기에 텍스트 분류 분야에서 많이 사용되는 기법이다. 본 연구에서는 두 기법에 대해서 동일한 데이터를 이용하여 실험을 진행 하였으며 1차원 Convolution Layer을 이용하여 얻은 값 중 Pooling이전에 Dropout을 이용하여 일정 수치 이하의 값을 제거해 주었다. 차원을 줄이기 위해 Pooling은 각 채널의 평균값을 추출해내는 AveragePooling을 이용하여 값을 얻어 냈다.

2-1) Title - Keyword Convolutional Neural Network (TK_CNN): 연구보고서의 메타정보 중, 문서를 대표하고 분류의 기준을 보여주는 단어는 제목과 키워드에 대부분 포함되어 있었으며 연구목표, 연구내용에도 키워드가 포함되어 있었으나 대분류를 구별할 수 있는 단어가 많이 포함되어 있기 때문에 중분류까지의 분류에 있어서는 오히려 혼동을 야기할 수 있다. 따라서, 핵심단어가 존재하는 제목과 키워드 필드만을 입력으로 하여 CNN 모델[2]을 구성하였다.

Table 2. NTIS Research Data Sample

Category	Content
Korean Title	유니버셜 스테이지 및 동작 인식 센서 모듈을 이용한 고령자의 균형감각 증진용 운동 시스템 개발
English Title	The development of balance sence enhancement training system using universal stage and motion capture sensor module
Research Area Classification Code	LC0506 (대분류:LC, 중분류: LC05, 세분류: LC0506)
Research Area Classification	재활훈련기기
Summary of Research Goals	○ 신체 기능 중 균형감각 증진을 위해 multi axis가 적용되어 실시간으로 고령자의 Ground reaction force를 측정할 수 있는 유니버셜 스테이지 개발 ○ 상지모션 및 하지 기울어짐을 독립적으로 인식하는 동작인식 센서 (depth sensor) 를 융합한 센서 모듈의 개발 ...
Summary of Research	-Ground reaction force(G.R.F.) 측정 센서의 multi axis 적용을 위한 메커니즘 설계 및 제작 -유니버셜 스테이지의 축변환에 대응하여 변화되는 G.R.F.를 실시간 획득하는 기술개발
Expected Effect Summary	□ 기술 분야 ○ 센싱 및 소프트웨어 처리, 장치 제어 및 상태 추정, 영상 콘텐츠와 같은 타 분야 시장에 활용 가능 ○ High Tech기반의 고령자 신체 기능 증진 시스템 개발 기술로 국제 경쟁력 향상과 고부가가치 실현 ...
Korean Keywords	유니버셜 스테이지,지면반력,깊이 감지기,센서융
English Keywords	Universal stage, Ground reaction force, Depth sensor, Multi-sensor

3) Long-Short Term Memory (LSTM) / Gated Recurrent Unit (GRU): LSTM [11]은 순차적인 정보를 저장하고 출력할 수 있다. 이 알고리즘은 RNN의 학습 시 역전파 과정에서 gradient가 점차 줄어들어 학습 능력을 크게 저하시키는 gradient vanishing문제를 보완하여 기존 RNN기법의 성능을 크게 향상시켰다. GRU [12] 는 LSTM의 장점을 유지하면서도 계산 복잡성을 낮춘 구조이다. LSTM과 유사하지만, 게이트 일부를 생략한 형태로써, 파라미터의 개수가 적어서 이른 시간에 그리고 적은 데이터로도 학습이 가능한 장점이 있다. 이러한 순차적인 정보를 가진 데이터를 LSTM과 같은 기법을 통해 사용할 때는 이전의 정보를 이용하여 학습을 진행한다. 가장 최신 기법인 Bidirectional- LSTM/GRU [13]는 데이터의 양방향 정보를 활용하고자 하였고 이 방법을 통해 이전의 정보와 이후의 정보를 모두 저장하고 사용할 수 있게 하였다.

4) Concatenation Model (CM): CM 모델은 연구보고서 메타 데이터의 각 필드별 특성을 고려하기 위해 제안한 모델로 각 항목별 나타나는 단어의 차이와 문장형식의 데이터와 단어 나열의 데이터를 고려하여, 각 필드별 특성을 고려하는데 주안점을 두고 설계했다. 이를 위해 필드별로 학습모델을 다르게 설정하고 이를 통합하는 실험을 수행하였다. 이 경우, 같은 대분류에서 중분류로 나뉠 때 생길 수 있는 혼돈과 오차를 위해 학습된 결과를 다시 1차원으로 나열하여 Hidden Layer를 거쳐 결과를 추출하고 하였다. 이렇게 제안된 모델에서 단어의 나열 형태로 나오는 필드의 경우에는 CNN 기법을 적용하였으며 연구의 목표 요약, 연구의 내용 요약과 같은 순서가 있는 데이터의 경우는 각각의 데이터 특성을 고려하여 LSTM, GRU 등의 기법을 적용하였다. 이와 같은 방법을 적용하여 제안 모델은 한 분류에 대해서 여러 가지의 항목이 있는 경우 각 데이터의 구조를 무시한 일괄적인 처리 및 중간점을 찾은 후의 처리방식보다는 Fig. 1과 같이 각 데이터를 별도로 처리하는 목적에서 제안되었다.

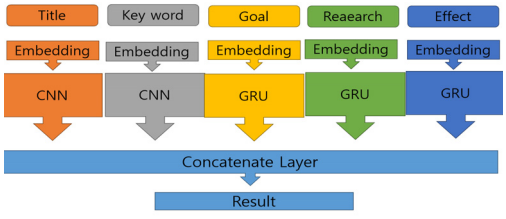

Fig. 1. Our Concatenation Model

5) Word Embedding: 단어의 임베딩을 위해 Word2Vec[14], Glove[15]를 이용하거나, 경우에 따라서 임베딩 자체를 적용하지 않는 실험조합도 고려하였다. 데이터를 처리하는 과정에서 Word2Vec과 같은 워드 임베딩이 영어권에서는 기계학습에 있어 긍정적인 효과를 보여주고 있지만, 한글 뉴스나 연구보고서 메타정보에서는 좋은 성능을 나타내지 못하고 있다고 한다. [16]의 연구에서는 데이터가 방대했을 때 Word2Vec과 같은 표현으로 부족한 코퍼스를 보완해주는 것이 성능향상에 도움이 되지 않는다고 한다. 이를 고려하여 임베딩을 문서 전체, 각 문서의 각 필드별로 실험을 진행하였다.

3.4 실험 과정

본 연구에서 과학기술표준분류를 자동화된 방식으로 분류하기 위해 2013년부터 2017년까지의 된 연구보고서 데이터를 기준으로 선정된 210개의 중분류에 해당하는 총 212,385개의 데이터를 이용하였으며, 10-fold cross evaluation을 위해 학습 셋 (train set)과 테스트 셋 (test set)의 비율을 9:1로 분할하였다. 그리고 검증 셋 (validation set)이 요구되는 실험에서는 학습 셋을 다시 7:3의 비율로 랜덤하게 분할하여 사용하였다.

본 연구에서 다양한 범주를 고려해 학습을 진행하였다. 각 범주에 대해서 단어의 개수는 Table 3에서 보는 바와 같이 이루어져 있으며 이때 포함된 데이터는 한글, 영어 그리고 숫자로 이루어져 있으며 모든 특수문자는 제외하였다. 한글은 Konlpy[17]의 Mecab을 이용하여 형태소를 분리 하였으며 이때 총 단어 토큰 개수는 96382개이다. 이때 대부분의 핵심단어는 Title와 Keyword에 포함되어 있어 해당 범주를 기반으로 최대 30000개의 단어 사전을 구축하였다.

이하 임베딩의 계산에서도 효율성을 위해 구축된 사전을 기준으로 최대 단어의 개수를 3만 개로 제한하였는데, 9만여 개 단어를 모두 사용한 것과 비교하여 성능 차이가 거의 없었다. 단어 임베딩을 위해 Gensim [18]의 Word2Vec과 Glove 두 가지 방법을 이용하여 임베딩을 시도하였으며 이때 300차원에 대해서 단어 사전에 등록된 30000개의 단어에 대해서 임베딩을 적용하였으며 이때 [16]와같이 데이터의 수가 방대해질 경우 워드임베딩이 오히려 성능을 저하시킬 수 있다.

Table 3. Word Count Distribution of Data

	Title	Key word	Goal	Contents	Effect	All
Max Sentence Length	116	203	1,687	2,146	1799	4698
Min Sentence Length	37	29	81	106	39	296
Average Sentence Length	28	32	179	378	237	858
Word Count	32,844	43,607	63,706	80,266	62,279	96,382

4. 실험 결과

실험 및 평가를 위해 Linear SVM, CNN[2], Char-CNN[10], 및 Bidirectional-GRU[13]기법과 Concatenation 모델의 실험을 진행하였으며, 이 결과는 Table 4와 같다.

Table 4. Evaluation Results

Algorithms	Precision	Recall	F1	Top-3
LinearSVC[19]	0.74	0.74	0.74	0.85
ALL_CNN[1]	0.71	0.71	0.71	0.81
ALL_GLOVE_CNN[1]	0.7	0.69	0.69	0.80
ALL_W2V_CNN[1]	0.69	0.67	0.67	0.81
***TK_CNN**	0.75	0.75	0.75	0.86
ALL_char_CNN[10]	0.69	0.68	0.68	0.81
Bidirectional-GRU[13]	0.7	0.7	0.7	0.81
Concatenation (CM)	0.69	0.68	0.68	0.79

- 모든 데이터를 사용할 시에는 SVM 계열의 알고리즘 중 scikit-learn 패키지에서 제공하는 Linear SVC[19]가 가장 좋은 정확도를 보여주었는데, 이는 210개 중분류 코드들에 대응함에 있어 단어의 수에 의존하지 않아 큰 텍스트의 특징 공간을 처리하는 데 있어 매우 효율적 작동된 것으로 판단된다. 그 성능 또한 다른 딥러닝 기법들과 비교하여 매우 우수한 수준이다.

- 핵심 데이터인 과제명과 키워드를 사용하였을 때에는 워드 임베딩이 적용되지 않은 CNN 모델이 가장 높은 정답률을 보여주었는데, 이는 영어권 문서에서 워드임베딩이 문서분류에 긍정적인 효과를 미친것과는 다소 다르다. 이를 보다 면밀히 살피기 위해 5가지 조합으로 실험을 진행하였다. (조합1: 모든 필드사용+워드임베딩 미적용, 조합2: 모든 필드사용 + Glove 임베딩 적용, 조합3: 모든 필드사용 + Word2Vec 임베딩적용, 조합4: 과제명 및 키워드 필드만 사용 (임베딩 미적용), 조합5: 모든 필드사용 (임베딩 미적용)) 과제명과 키워드에 대해 나오는 단어의 경우는 모든 데이터를 사용하는 것에 비해 단어의 정보가 명확하여 모든 데이터를 사용하여 사전의 크기가 커지는 것에 비해 생성된 단어 사전을 통한 데이터 정규화 과정과 불필요한 단어의 소실로 인한 데이터의 크기가 줄어드는 것과 같은 좋은 효율을 보여준다. 여러 조합의 CNN 실험들 중에서 과제명과 키워드를 워드임베딩 없이 사용한 경우가 가장 높은 성능(F1=75%, Top-3 정확도 86%)을 보였다.

- 흥미로운 것은 워드임베딩을 채택한 CNN보다 Bidirectional-GRU가 좀 더 나은 결과를 보이고 있는 부분이다. 이때 데이터의 길이와 과제명과 키워드의 경우는 연속적인 문장의 형식보다는 단어를 나열해둔 형식의 데이터인 반면, 그 외의 연구 목적, 연구의 요약과 같은 데이터는 연구의 내용을 기반으로 연속적으로 내용을 서술하고 있다. 이는 시간적 순서의 측면을 고려할 수 있는 RNN계열의 기법에서 좀 더 높은 효율을 보여준 것으로 분석된다.

- 마지막으로 각 데이터의 특성을 고려한 Concatenation 모델은 Table 4의 마지막 줄에서와 같이 F1=68%의 결과를 보여주었다. 해당 결과를 분석하기 위해 각 분류별 적용한 기법을 확인하였을 때, 과제명과 키워드 부분에 대해서 가장 높은 정답률을 보였으나 다른 데이터들과 합쳐지는 과정에서 오히려 정답률이 떨어지는 결과를 보여주고 있어 과제명과 키워드에 대한 학습결과의 가중치를 올려 학습을 진행하였을 때는 큰 변화는 보여주지 못 하였다. 이렇게 많은 항목의 데이터를 사용할 시 오히려 정답률을 감소시키는 문제를 고려하여 가장 비중이 높은 단어를 포함하고 있는 과제명과 키워드만을 이용하여 실험을 진행하였으며, 이때 워드임베딩을 적용하였을 때의 결과는[16]과 같이 단어의 수가 너무 방대하기 때문에 오히려 성능이 저하되었다.

5. 토론

Table 4에서의 실험 결과들에서 볼 때, Word2Vec나 Glove와 같은 워드 임베딩은 한글 연구 보고서 메타데이터를 이용한 분류 학습에서는 좋은 성능을 보이기보다는 오히려 성능을 저하시키는 결과를 보였다. 또한, 연구 보고서 메타 데이터에 있어서 CNN과 같은 딥러닝 기법보다 Linear SVC와 같은 기존의 기계학습 알고리즘이 좋은 효율을 보여 해당 데이터의 분류 학습에 있어서 딥

러닝 기법이 항시 좋은 성능을 보여주지는 않았다.

실험을 위해 사용된 NTIS 연구보고서 메타 데이터에서 100개 이상의 데이터를 가진 210개의 중분류코드를 이용하여 실험을 진행하였지만, 데이터를 분할하는 과정에서 데이터가 적은 코드들은 다른 데이터에 비해 낮은 정확도를 나타내었는데, 이러한 데이터의 불균형(Imbalanced Data) 문제를 해결하여 추후에 실험을 다시 진행할 필요가 있다.

메타정보를 구성하는 각 필드별 특성을 고려하여 학습을 진행하는 Concatenation Model은 그리 좋은 성능을 보여주지 못하였다. 하지만 이러한 형태의 모델은 여러 메타 데이터를 가지고 있거나 이미지와 텍스트와 같은 서로 다른 유형의 데이터를 함께 사용하여 학습하는 것과 같은 다양한 시도를 할 수 있다고 생각한다.

이번 연구에서 여러 메타 데이터를 이용하여 학습을 진행하였을 때 과제명과 키워드만을 이용하여 학습을 진행하였을 때 좋은 성능을 보였다. 이러한 결과를 보았을 때, 메타데이터는 모든 데이터를 사용하는 것은 반드시 효율적인 결과를 보여주는 것이 아니며, Feature Engineering이 불필요하다고 여겨지는 딥러닝의 경우에서도 영향력 있는 데이터를 선별하여 학습에 사용하는 것이 학습에 좀 더 좋은 효율을 보여준다.

과학기술표준분류체계와 같이 대규모의 분류체계를 대상으로 진행하는 텍스트 분류는 [20]의 연구에 대해서도 비슷하게 진행되었다. 하지만, 해당 연구에서는 GRU가 가장 높은 성능을 보였으며 오히려 SVM이 더욱 낮은 성능을 보여주었다. 따라서 데이터의 특성에 맞는 전처리 및 학습기법을 선택하는 것이 매우 중요하다고 하겠다.

6. 결론 및 향후 연구

최근 딥러닝에 관한 지속적인 연구를 통해 기계적인 방법을 통한 텍스트 분류에 대해서 좋은 성능을 보여주고 있지만 대부분 적은 개수의 카테고리에 대해서 문장에 대해 좋은 결과를 보여주고 있으며, 특히 국내문서기반으로 큰 규모의 분류체계를 대상으로는 연구가 미미하다.

본 연구에서는 여러 개의 필드를 포함하는 메타 데이터를 가지고 있는 텍스트에서 210여 개에 이르는 중분류 분류체계를 대상으로 적합한 분류코드 할당하는 연구를 진행하였으며, ALL_CNN(모든 메타 데이터를 사용한 CNN기법)에서 F1=0.71, Top-3 정확도 81%를 보여주었으나, 좀 더 의미가 있는 선별된 데이터를 적용한 TK_CNN (과제명과 키워드에 해당하는 핵심 필드만을 사용한 CNN 기법)에서 F1=0.75, Top-3 정확도 86%가 더 높은 성능을 보였다. 따라서, 연구문서에 대한 카테고리 분류에 있어서 분류에 사용되는 메타 데이터를 선별이 여전히 중요함을 확인하였다.

하지만, 성능 부분에서는 아직 개선의 여지가 많으며, 최근 제안된 기법들 중에 추후 성능 향상을 위해 결합이 가능한 모델은 Doc2Vec을 이용한 문서의 유사도 비교[21], 2계층 Bi-LSTM을 사용한 한글 문서분류[22], 개수가 부족한 카테고리에 대한 특징 선택을 통한 정확도 향상[23]이 등이 있다. 또한, 학습기법에 대해서 Hierarchical Text Classification [24], BERT[25], 지식기반의 임베딩[26] 그리고 RCNN[27] 등과 같은 방법을 이용하여 성능 향상을 시도해볼 필요가 있다.

활용 측면에서 볼 때, 높은 정확도의 과학기술표준분류기반 문서분류기 개발은 향후 연구보고서를 자동 분석하여, 연구 활동의 목적 및 산업과의 연계성 파악 및 과학기술 동향 분석 등에 다각적으로 이용될 수 있을 것이다.

References

[1] C. H. Song, and S. S. Sung. 2006. “A Study on the Problems of Current National Standard Classification of Science and Technology for National Science and Technology Information System.” : pp.496-513.

[2] Y. Kim. 2014. “Convolutional Neural Networks for Sentence Classification.” EMNLP 2014: 1746-51. DOI: https://doi.org/10.3115/v1/D14-1181

[3] P. Liu, X. Qiu, and X. Huang. 2016. “Recurrent Neural Network for Text Classification with Multi-Task Learning.” AAAI Publications, Twenty-Ninth AAAI Conference on Artificial Intelligence: 2267-2273.

[4] S. Fabrizio. 2002. “Machine Learning in Automated Text Categorization.” ACM Computing Surveys 34: 1-47. DOI:https://doi.org/10.1145/505282.505283

[5] L. Saitta. 1995. Nov “Support-Vector Networks.” Machine Learning 20(3): 273-97. DOI: https://doi.org/10.1007/BF00994018

[6] C. Nello, J. Shawe-Taylor, and B. Williamson. 2001. “On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines.” Machine Learning Research 2: 265-92. DOI: https://doi.org/10.1007/BF00994018

[7] Y. H. Kim, S. Y. Kang , and M. J. Choi. 2015. “Improvement of National Science and Technology Standard Classification System in 2015” Research and Development, Korea Institute of Science and Technology Evaluation and Planning, Korea, pp.1-221.

[8] J. Weston, et al. 2000. "Feature Selection for SVMs." Advances in Neural Information Processing Systems 13: 668-674.

[9] Scikit learn's SVC, Available at https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[10] X. Zhang, J. Zhao, and Y. LeCun. 2015. Character-level convolutional networks for text classification. arXiv preprint arXiv:1509.01626.

[11] S. Hochreiter, and J. Schmidhuber. 1997. "Long Short-Term Memory." Neural Computation 9(8): p.1735-1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[12] J. Y. Chung , G. Caglar, K. H. Cho , and Y. Bengio. 2014. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." NIPS 2014 Workshop on Deep Learning: p.1-9.

[13] P. Zhou et al. 2016. "Text Classification Improved by Integrating Bidirectional LSTM with Two-Dimensional Max Pooling." Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics 2(1): 3485-95.

[14] T. Mikolov, et al. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." Advances in Neural Information Processing Systems 26 (NIPS 2013): 1-9.

[15] J. Pennington, R. Socher, and C. D. Manning. 2014. "GloVe : Global Vectors for Word Representation." EMNLP: 1532-1543. DOI: https://doi.org/10.3115/v1/D14-1162

[16] H. Jo, et al. 2015. "Large-Scale Text Classification Methodology with Convolutional Neural Network." Korean Information Science Society: 792-94. DOI: http://dx.doi.org/10.5626/KTCP.2017.23.5.322

[17] E. J. Park, and S. Z. Cho . 2014. "KoNLPy : Korean Natural Language Processing in Python." Annual Conference on Human and Language Technology: pp.133-136.

[18] Gensim Word2Vec, Available at https://radimrehurek.com/gensim/models/word2vec.html

[19] Scikit learn's Linear SVC, Available at https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[20] H. Y. Jo, et al. 2017. "Large-Scale Text Classification with Deep Neural Networks." KIISE Transactions on Computing Practices 23: 322-27. DOI: https://doi.org/10.5626/KTCP.2017.23.5.322

[21] J. S. Jeong et al. 2019. "Related Documents Classification System by Similarity between Documents." The Korean Society Of Broad Engineers 24(1): 77-86. DOI: https://doi.org/10.5909/JBE.2019.24.1.77

[22] K. Y. Kim and C. J. Park. 2019. "Automatic IPC Classification of Patent Documents Using Word2Vec and Two Layers Bidirectional Long Short Term Memory Network." THE JOURNAL OF KOREAN INSTITUTE OF NEXT GENERATION COMPUTING 15(2): 50-60.

[23] M. J. Seo, G. S. Ahn, and S. Hur. 2019. "Feature Selection Method from Multiclass Text with Class Imbalance Problem." Journal of the Korean Institute of Industrial Engineers (April): 1-8.

[24] K. Kowsari et al. 2017. "HDLTex : Hierarchical Deep Learning for Text Classification." 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA): 364-71. DOI: https://doi.org/10.1109/ICMLA.2017.0-134

[25] Jacob, Devlin, Ming-wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." NAACL-HLT: 4171-4186.

[26] R. A. Sinoara et al. 2019. "Knowledge-Based Systems Knowledge-Enhanced Document Embeddings for Text Classification." Knowledge-Based Systems 163: 955-71. DOI: https://doi.org/10.1016/j.knosys.2018.10.026

[27] S. Lai, L. Xu, K. Liu, and J. Zhao. 2015. "Recurrent Convolutional Neural Networks for Text Classification." Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Recurrent: 2267-73.

최 종 윤(Jong-Yun Choi) [정회원]

• 2015년 2월 ~ 현재 : 금오공과대학교 컴퓨터공학과 (학사과정)

〈관심분야〉
딥러닝/AI, 자연어처리, 크롤러

한 혁(Hyuk Hahn) [정회원]

• 1994년 2월 : 고려대학교 농경제학과 (경제학사)
• 2001년 2월 : 한국과학기술원 Techno-MBA (경영학석사)
• 2004년 7월 ~ 현재 : 한국과학기술정보연구원

〈관심분야〉
R&D투자분석, 계량정보분석, 빅데이터, 적정기술

정 유 철(Yuchul Jung) [정회원]

- 2003년 2월 : 아주대학교 정보 및 컴퓨터공학부 (공학사)
- 2005년 2월 : 한국과학기술원 정보통신공학 (공학석사)
- 2011년 2월 : 한국과학기술원 전산학 (공학박사)
- 2009년 1월 ~ 2013년 7월 : 한국전자통신연구원 선임연구원
- 2013년 8월 ~ 2017년 8월 : 한국과학기술정보연구원 선임연구원
- 2017년 8월 ~ 현재 : 금오공과대학교 컴퓨터공학과 조교수

〈관심분야〉
인공지능, 기계학습, 정보검색, 자연어처리