기술용어 분산표현을 활용한 특허문헌 분류에 관한 연구

A Study on Patent Literature Classification Using Distributed Representation of Technical Terms

최 윤 수 (Yunsoo Choi)*
최 성 필 (Sung-Pil Choi)**

목 차

초 록

본 연구의 목적은 특허 문헌 분류에 가장 적합한 방법론을 발견하기 위하여 다양한 자질 추출 방법과 기계학습 및 딥러닝 모델을 살펴보고 실험을 통해 최적의 성능을 제공하는 방법론을 분석하는데 있다. 자질 추출 방법으로는 전통적인 BoW 방법과 분산표현 방식인 워드 임베딩 벡터를 비교 실험하고, 문헌 집합 구축 방식으로는 형태소 분석과 멀티그램을 이용하는 방식을 비교 검토하였다. 또한 전통적인 기계학습 모델과 딥러닝 모델을 이용하여 분류 성능을 검증하였다. 실험 결과, 분산표현 방법과 형태소 분석을 이용한 자질추출 방법을 기반으로 딥러닝 모델을 적용하였을 경우에 분류 성능이 가장 우수한 것으로 판명되었으며 섹션, 클래스, 서브클래스 분류 실험에서 전통적인 기계학습 방법에 비해 각각 5.71%, 18.84%, 21.53% 우수한 분류 성능을 보여주었다.

ABSTRACT

In this paper, we propose optimal methodologies for classifying patent literature by examining various feature extraction methods, machine learning and deep learning models, and provide optimal performance through experiments. We compared the traditional BoW method and a distributed representation method (word embedding vector) as a feature extraction, and compared the morphological analysis and multi gram as the method of constructing the document collection. In addition, classification performance was verified using traditional machine learning model and deep learning model. Experimental results show that the best performance is achieved when we apply the deep learning model with distributed representation and morphological analysis based feature extraction. In Section, Class and Subclass classification experiments, We improved the performance by 5.71%, 18.84% and 21.53%, respectively, compared with traditional classification methods.

키워드: 특허문헌 분류, 분산표현, 워드 임베딩 벡터, 딥러닝
Patent Literature Classification, Distributed Representation, Word Embedding Vector, Deep Learning

---

* 경기대학교 일반대학원 문헌정보학과 박사과정(cysoo@kgu.ac.kr / ISNI 0000 0004 6773 2724) (제1저자)
** 경기대학교 휴먼인재융합대학 문헌정보학과 조교수(spchoi@kgu.ac.kr / ISNI 0000 0004 6772 9269) (교신저자)
논문접수일자: 2019년 3월 28일 최초심사일자: 2019년 5월 7일 게재확정일자: 2019년 5월 20일
한국문헌정보학회지, 53(2): 179-199, 2019. [http://dx.doi.org/10.4275/KSLIS.2019.53.2.179]

1. 서 론

전통적 생산요소인 노동과 자본이 배경이 되는 산업사회에서 지식과 정보가 가치를 생산하는 지식기반사회로 접어들면서 지식재산권의 대표적인 형태인 특허에 대한 중요성이 점점 높아지고 있다. 특허는 발명가에게 제한된 기간 동안 발명품에 대한 독점권을 제공하기 위해 사용되며, 발명가는 특허를 통해 발명에 대한 기술적인 세부사항을 공개해야 한다. 특허청(2018) 통계자료에 의하면 국내 특허출원 건수는 2000년대 들어서면서 급증하였으며, 2013년을 기점으로 매년 20만 건을 상회하고 있다.

차세대 기술혁명은 NT(나노기술), BT(생명공학기술), IT(정보기술) 등 신기술간 또는 이들과 타 분야와의 상승적 결합을 통합 "융합기술"이 주도하고 있으며 국가 차원에서 융합기술을 종합적, 체계적으로 육성하기 위한 계획이 수립되고 추진되고 있다.

한국과학기술연구원 융합연구정책센터(2018)에 의하면 융합연구에 대한 예산은 지속적으로 증가하여 2017년 기준으로 정부가 주도하는 전체 R&D예산의 16.88%인 2.5조원이 융합연구 예산으로 할당되었다. 이와 더불어 2개 이상의 범주에 포함되는 특허 문헌 또한 지속적으로 증가하여, 2017년에 다중 범주를 갖는 특허 문헌의 수는 전체 특허의 33.01% 차지하고 있다.

특허 문헌을 분류하기 위해 널리 사용되는 방법은 특허와 관련된 기술의 모든 영역을 포괄하는 복잡한 계층구조로 구성된 국제특허 분류(IPC: International Patent Classification)이다. IPC는 특허 문헌에 대한 국제적으로 균일한 분류를 획득하기 위한 방법으로, 지난 수십 년 동안 갱신되고 정제되었으며, 현재 90여 개 이상의 국가에서 사용되고 있다.

특허 문헌의 양적인 팽창과 함께, 융합연구의 증가로 분류의 난이도 또한 함께 증가하고 있기 때문에, 노동집약적인 수동 특허 분류 작업을 지원하기 위한 자동 분류 도구에 대한 요구가 지속적으로 증가하고 있다. 하지만 특허문헌은 일반적인 신문기사나 기술문헌과는 달리 특허에 사용되는 난해한 법률 용어인 'patentese'의 빈번한 활용이라는 특징을 지니고 있어 자동분류에 어려움이 있다. 특허 출원자의 주된 목적은 발명의 정보를 전달하는 일 외에도 발명에 대한 최대한의 법적 범위와 보호를 확보하는 일이므로, 발명을 기술할 때 일반적으로 잘 사용되지 않는 기술용어를 이용해 작성되거나, 의도적으로 검색이 되지 않도록 관련 기술용어를 직접 사용하지 않고 풀어 작성하기도 한다. 예를 들어, "펌프"라는 용어 대신에 "유체 운송 장치"라는 복합 단어를 선택할 수 있다.

이에, 특허 문헌을 구성하고 있는 단어들로부터 분류를 위해 유용한 자질을 추출하는 방법이 가장 중요하고, 추출된 자질을 학습하고 실제 분류를 수행하는 분류 모델 또한 중요하다. 본 논문은 특허 문헌으로부터 분류를 위한 더욱 적합한 특징을 추출하고, 더 좋은 분류 성능을 제공하는 모델을 알아보기 위해 다음과 같이 연구를 진행하였다.

첫째, 난해한 용어로 구성된 특허 문헌으로부터 분류를 위해 좋은 변별력을 지닌 자질 추출 방법을 살펴본다. 문헌 분류를 위한 첫 단계는 문헌으로부터 분류를 위한 자질을 추출하는 작업이다. 전통적인 기계학습 방법에서 이는 문헌으로부터 단어를 추출하고, 빈도수에 기반하여

추출된 단어에 가중치를 부여하는 형식으로 진행된다. 최근 들어 단어의 주변 문맥으로부터 단어에 대한 분산 표현을 추출하는 워드 임베딩 방식이 주목을 받고 있다. 이 2가지 방식에 대하여 단어 표현력을 살펴보고, 특허 문헌 분류에 대한 성능을 조사한다.

둘째, 특허 문헌 분류를 위해 어떤 분류 모델을 사용하는 것이 더 효과적인가를 조사한다. 문헌 분류를 위해 전통적인 기계학습 방법인 나이브베이즈(Naive Bayes), kNN(k-Nearest Neighbor), SVM(Support Vector Machines)이 주로 사용되어 왔고, 최근 들어 딥러닝 방법으로 FCNN(Fully-Connected Neural Networks), CNN(Convolutional Neural Networks), RNN(Recurrent Neural Networks) 등과 같은 모델들이 자연어 처리 분야에서도 널리 활용되고 있다. 본 논문에서는 기계학습 모델과 딥러닝 모델들을 특허 문헌 분류에 적용하여 분류 성능을 평가하고 적합한 모델을 분석한다.

셋째, 단어에 대한 분산표현인 워드 임베딩 벡터를 구축하기 위하여 널리 사용되는 방법은 CBOW(Mikolov et al. 2013), Skip-Gram(Mikolov et al. 2013), GloVe(Pennington, Socher and Manning 2014), fastText(Bojanowski et al. 2017) 4가지가 있으며, 워드 임베딩 벡터 구축 시 벡터 크기와 윈도우 크기에 따라 다양한 임베딩 벡터를 생성할 수 있다. 특허 문헌으로부터 구축된 다양한 워드 임베딩 벡터 중에서 어떤 방법으로 구축된 벡터가 특허 문헌 분류를 위해 표현력이 가장 좋은지를 실험하고 결과를 살펴본다.

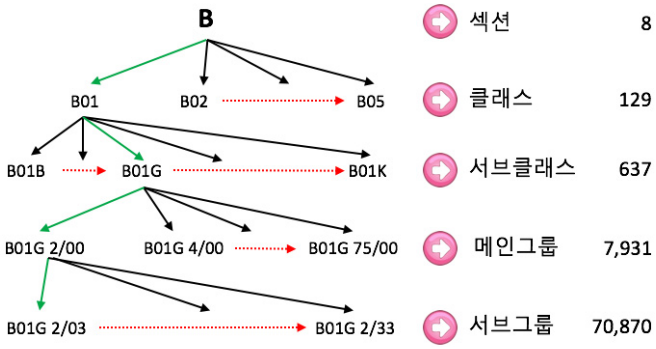
넷째, 특허 문헌으로부터 분류를 위한 자질을 추출하기 위해, 다양한 단어 선택 방법이 존재한다. 이 중에서 형태소분석기를 이용한 품사태깅을 통해 명사, 형용사, 동사 등의 단어를 선택하여 사용하는 방법과, 바이그램이나 트라이그램을 사용하여 단어를 선택하는 방법이 일반적으로 많이 사용된다. 어떤 단어 선택 방법이 특허 문헌 분류에 가장 영향력이 있는지를 실험하고 그 성능을 분석한다.

다섯째, 특허 문헌을 구성하는 가장 기본적인 필드는 영문제목, 국문제목과 초록이다. 초록만을 사용하여 실험한 결과와, 영문제목과 국문제목에 대해 FCNN, CNN, RNN 3가지 딥러닝 모델을 조합하여 특허 분류 성능에 미치는 영향을 조사한다.

2. 관련연구

전 세계적으로 특허의 활용성이 증가하면서, 특허문헌에 대한 효과적인 검색과 활용을 위한 특허분류체계의 필요성이 증가되어, 국제적으로 통일된 특허분류체계로 국제특허분류(IPC: International Patent Classification)가 도입되었다. IPC는 1975년 '특허분류에 관한 스트라스부르그 협정'을 통하여 최초 제정되었고, 이후 주기적으로 개정 과정을 거치면서 IPC 코드 폐지, 합병, 신규 생성 등의 작업이 진행되어 왔으며, 전세계 90개국 이상의 특허 사무소에서 특허 문헌 분류를 위해 사용되고 있다.

현재 IPC 분류체계는 〈그림 1〉과 같은 계층 구조로 구성된다. 최상위 분류체계는 '섹션(Section)'으로 명명되고, '클래스(Class)', '서브클래스(Subclass)', '메인그룹(Main group)', '서브그룹(Subgroup)' 5개의 단계로 문헌을 세



〈그림 1〉 국제특허 분류(IPC) 분류 체계

분화하여 분류하고 있다. 최상위 분류체계인 섹션은 8개의 범주값을 갖고 있으며 하위분류로 내려가면서 증가하여, 최하위 분류체계인 서브그룹은 70,870개의 범주값으로 세분화된다.

1960년대 이후 문헌에 대한 자동 분류는 텍스트마이닝과 자연어처리 분야에서 많은 관심을 받아왔고, 문헌 분류 분야에서 다양하고 좋은 성능을 제공하는 분류 알고리즘들이 연구되어 왔으며, 특허 문헌 분류와 관련하여 전통적인 기계학습 방법론을 활용한 연구들이 다수 수행되었다.

Larkey(1999)는 미국특허문헌을 대상으로 분류에 대한 연구를 수행하였다. 단어와 명사구의 조합을 색인을 위한 용어로 선정하였으며, 빈도수를 기반으로 용어에 대한 가중치를 부여하고 임계치 이상인 용어들만을 분류를 위해 사용하였다. 특허 문헌 중에서 “speech signal processing” 서브클래스 수준에 대한 분류 실험에서 최고 32%의 성능을 얻었다.

Koster and Secutter(2003)은 유럽특허문헌에 대한 분류를 수행하였다. 자질 추출 방법으로 단어와 명사구에 대한 비교 실험을 수행하였으며, 분류 모델은 퍼셉트론의 변형인 Winnow을 사용하였고, 섹션 수준에 대한 분류에서 79.0%의 정확률을 보였다.

Fall et al. (2003)은 WIPO-alpha에 대한 분류 연구를 수행하였고, 나이브베이즈, SVM, kNN 등의 여러 기계학습 알고리즘의 성능을 비교하였다. 제목, 초록, 청구항의 처음 300 단어를 분류를 위한 자질로 선택하여 실험하였고, 초록을 사용한 경우 서브클래스에서 42%의 성능을 보여주었다.

Tikk, Biró and Törcsvári(2008)는 특허문헌 분류를 위해 독자적인 HITEC 모델을 도입하여 분류를 수행하였다. HITEC은 택소노미에 기반한 구조로서, IPC 분류의 계층 구조를 신경망을 이용하여 모델링하고, 입력되는 문헌에 대해 상위 계층부터 분류를 수행하여 하위 계층으로 전달한다. 분류기는 각 분류 수준에서 활성화되어 있는 분류노드들에 대해서만 문헌의 분류 점수를 결정하고, 이 점수에 의해 다음 분류 수준에서 활성화되는 분류노드를 결정한다. WIPO-alpha 집합에 대하여 성능 평가를 수행하였고, 서브클래스 수준에서 53.25%의 정확도를 보여주었다.

Chen and Chang(2012)는 자질추출로 TF-ICF[1)]

를 이용하였고, 세 단계로 구성된 하이브리드 분류 모델을 제시하였다. 첫 번째와 두번째 단계에서 SVM 모델을 학습시켜 특허 문헌을 계속해서 여러 하위 클래스로 분류하고, 마지막 세 번째 단계에서 kNN 모델을 이용하여 선택된 후보에 기반하여 주어진 특허 문헌에 최종 범주값을 할당하는 방법으로, 서브클래스에서 36.89%의 성능을 나타냈다.

김재호, 최기선(2005)은 BoW[2] 기반 문헌 표현이 용어 간 관계 전달의 문제점을 해결하기 위하여 의미론적 구조 정보를 도입한 유사문서 검색 형태의 독자적인 분류 모델을 제안하였으며 일본 특허 문헌의 서브클래스 수준에서 분류 연구를 수행하여 51.26%의 성능을 확보하였다.

박찬정, 김기영, 성동수(2014)는 특징선택 방법으로 정보이득, 카이제곱 통계량, 상호정보량 그리고 우세정보량에 대한 성능 비교를 수행하였다. 한국특허문헌의 섹션 수준에서 분류 실험을 수행하였고, kNN 모델을 적용하여 k=2, 우세정보량 10%를 선택한 경우에 43%의 정확도를 보였다.

임소라, 권용진(2017)은 TF-ICF를 이용하여 단어들에 랭킹을 부여하고, 서브클래스별로 상위 100개의 단어를 추출하여 분류 자질로 선택하였다. 나이브베이즈 모델을 분류모델로 사용하였고, 제목, 초록, 청구항, 기술분야, 배경기술 필드의 조합에 대해 성능 실험을 수행하여 청구항을 제외한 제목, 초록, 기술분야, 배경기술 조합에서 가장 좋은 70.00%의 성능을 보여주었다.

〈표 1〉에서 보는 바와 같이, 대부분의 기존 연구들은 특허 문헌의 분류 체계 중 클래스와 서브클래스를 중심으로 수행되었음을 알 수 있다. 분류를 위해 수집하고 구축한 학습집합은 다양하였고, 사용된 필드를 살펴보면, 제목, 초록은 모든 연구에서 공통으로 사용하였고, 나머지 필드들은 선택적으로 활용되었다.

자질추출 부분에서 용어 빈도에 기반한 가중치를 이용하는 방법을 채택하고 있고, 분류모델은 전통적인 기계학습 방법인 나이브베이즈, kNN, SVM이 주로 사용되었으며, 일부는 독자적인 모델을 구축하였다.

학습문헌이 표준화되어 있지 않고, 동일하게 보이는 학습문헌일지라도 문헌의 수집 범위가 다양하고, 평가 기준 또한 상이하기 때문에 기존 연구들에 대한 직접적인 성능 비교는 어렵지만, 개별 연구들의 최고 성능을 기준으로 살펴보았을 때, 섹션수준에서 43%~79.0%, 클래스 수준에서 55%~65%, 서브클래스 수준에서 32%~70.00%의 분류 성능을 제공하고 있는 것으로 나타났다.

본 논문에서는 최근 자연어 처리 분야에서 좋은 평가를 받고 있는 워드 임베딩 벡터를 자질추출 방법으로 사용하고, 딥러닝 모델을 이용한 특허 문헌 분류 시스템을 제안한다. 전통적인 자질추출 방법과 기계학습 모델과의 성능 비교를 통해 특허 문헌 분류에 적합한 자질추출 방법과 분류 모델을 살펴본다. 분류 모델의

---

1) term-frequency-inverse category frequency: 전체 문헌에서 출현하는 단어의 빈도수와 특정 범주에 출현하는 단어의 역빈도수를 이용하여 단어에 대한 가중치를 계산하는 방법.

2) 자연어 처리 분야에서 문서를 단어들의 다중집합으로 표현하는 방법으로, 문법이나 단어 순서 등의 정보는 무시된다.

〈표 1〉 특허문헌 분류에 대한 관련연구

저자	학습집합	사용된 필드	분류모델	분류수준	성능[3] (precision)
Larkey (1999)	USPTO	제목, 초록, 배경지식, 청구항 (처음20줄)	kNN	서브클래스	25%* ~32%*
Koster and Seutter (2003)	EPO1A	초록	winnow	섹션수준	63%~79.0%
Fall et al. (2003)	WIPo-alpha	제목, 초록, 청구항 (처음300단어)	NB, kNN, SVM	클래스 서브클래스	45%~55% 34%~42%
Tikk, Biró and Törcsvári(2008)	WIPO-alpha	발명자, 특허권자, 제목, 초록, 청구항	HITEC (NB, kNN)	클래스 서브클래스 메인그룹	62.81%~65.75% 49.41%~53.25% 32.28%~36.89%
Chen and Chang (2012)	WIPO-alpha	제목, 초록	kNN, SVM	서브클래스	30.02%~36.07%*
김재호, 최기선 (2005)	일본특허문헌	기술분야, 목적, 해결방법, 청구항, 설명	독자적인 모델 (유사문서검색)	서브클래스	37.44%~51.26%
박찬정, 김기영, 성동수 (2014)	한국특허문헌	제목, 초록, 청구항	kNN	섹션	36% ~ 43%
임소라, 권용진 (2017)	한국특허문헌	제목, 초록, 청구항, 기술분야, 배경기술	NB	서브클래스	58.65%~70.00%

※ 성능 중에서 '*' 첨자가 붙은 수치는 정확도(accuracy)임

성능에 대한 평가는 조화평균(f1-score)을 사용하며, 정확률과 재현율을 부가적인 성능으로 제공한다.

3. 분산표현 기반 특허문헌 분류

3.1 대상 데이터 수집 및 가공

특허문헌 수집은 특허정보넷 키프리스 홈페이지[4]를 통해 수행되었다. 1996년부터 2017년도까지 22년 동안 전체 2,038,553건의 특허문헌을 수집하였다. 수집된 특허문헌에서 문헌 분류를 위해 필요한 '제목'[5]과 '초록' 필드를 추출하고, '국문제목'과 '초록'은 〈표 2〉와 같이 4가지 방법[6]으로 단어 추출을 진행하여 각각 연도별로 특허 문헌집합을 구축한다.

NOUN과 NAV(Noun, Adjective, Verb)는 형태소 분석기의 품사태깅[7] 결과를 이용하여 추출된 단어들로 특허 문헌 집합을 생성한다. 〈표 3〉은 형태소 분석기를 이용하여 초록에 대한 품사 태깅이 수행된 문헌의 예를 보여준다.

3) 최저 성능과 최고 성능을 함께 표기함(최고 성능~최고 성능)
4) http://www.kipris.or.kr/khome/main.jsp
5) 제목은 국문제목과 영문제목으로 구성되어 있어 이를 분리하는 과정이 필요하다.
6) 학습집합 구성 시 단어 선택 방법에 따른 성능 비교를 위해 4가지 방법으로 문헌집합을 구성한다.
7) 문헌에 대한 품사태깅을 위해 KoNLPy 패키지 중에서 한나눔 형태소 분석기를 이용하였다.

〈표 2〉 추출형식에 따른 어휘 통계 자료

일련 번호	추출형식	단어수	단어 출현 빈도		
			최대값	최소값[8]	평균값
1	명사 (NOUN)	85,417	6,202,410	5	2,268.68
2	명사, 형용사, 동사 (NAV)	97,532	6,202,410	5	2,530.86
3	바이그램 (2GRAM)	171,300	6,216,671	5	2,756.04
4	트라이그램 (3GRAM)	1,145,193	5,999,687	5	304.49

〈표 3〉 특허 문헌에 형태소 분석기 적용한 예

등록번호	20-2014-0007965
분류코드	A61N 1/18(2006.01), A45D 44/22(2006.01), A61K 8/02(2006.01)
제목	무선전기 공급을 받아 작동되는 침패치를 부착한 마스크팩(Mask pack attached needle patch operated with wireless electric)
초록	본 고안은 무선전기 공급을 받아 작동되는 침패치를 부착한 마스크팩에 있어서, 특히 화장용 마스크팩의 표면에 침패치를 부착하되...
초록 (형태소 분석 수행 후)	본/Noun 고안/Noun 은/Josa 무선/Noun 전기/Noun 공급/Noun 을/Josa 받아/Verb 작동/Noun 되는/Verb 침/Noun 패치/Noun 를/Josa 부착/Noun 한/Josa 마스크/Noun 팩/Noun 에/Josa 있어/Adjective 서/Eomi ,/Punctuation 특히/Adverb 화장/Noun 용/Noun 마스크/Noun 팩/Noun 의/Josa 표면/Noun 에/Josa 침/Noun 패치/Noun...

NOUN은 형태소 분석 결과에서 품사가 명사인 단어만을 추출하고, NAV는 명사, 형용사, 동사를 추출하여 문헌 집합을 구성한다. 예를 들어, 〈표 3〉의 문헌의 경우 전자는 초록에서 명사로 품사태깅되어 있는 "본", "고안", "무선", "전기", "공급" 등의 단어들이 명사로 추출하여 하나의 문헌으로 구성한다.

2GRAM과 3GRAM은 형태소 분석을 사용하지 않고, 원본 문헌에 대하여 멀티그램을 적용하여 단어를 추출하는 방법이다. 초록 문장의 단어들을 공백으로 분리한 뒤, 각 분리된 단어에 대하여 바이그램과 트라이그램을 적용하여 어휘집을 구축한다. 예를 들어, "전기자동차"의 경우 바이그램을 적용하면 "전기", "기자", "자동", "동차" 4개의 단어를 추출하고, 트라이그램을 적용하면 "전기자", "기자동", "자동차" 3개의 단어를 추출한다. 트라이그램의 경우 다른 추출 방법과 비교하여 고유한 단어 수가 급격히 증가하지만, 단어의 평균 출현 빈도는 상대적으로 매우 낮음을 알 수 있다.

수집된 특허 문헌 중에서 1996년부터 2016년까지의 문헌들이 단어에 대한 분산표현인 워드 임베딩 벡터를 생성하기 위해 사용되었다.[9] 4가지 워드 임베딩 벡터 모델(CBOW, Skip-Gram, GloVe, fastText), 4가지 벡터공간(50, 100, 200, 300), 그리고 3가지 단어 문맥 윈도우(10, 15,

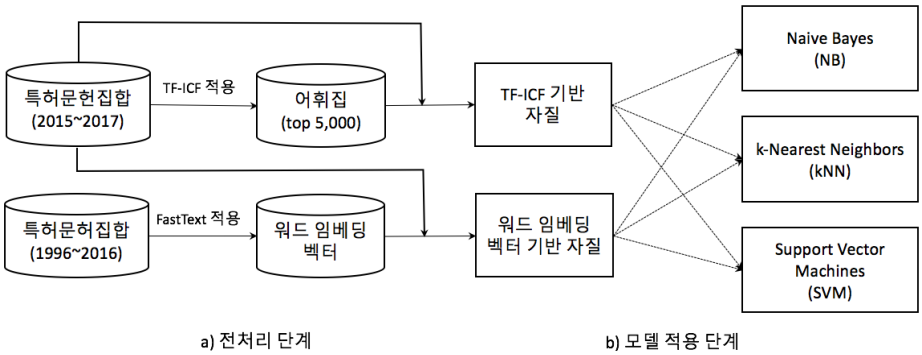
8) 단어 벡터를 구축 시 전체 문헌 집합에서 최소 출현 빈도를 5로 지정하였기 때문에 최소값은 항상 일정하다.
9) 2017년도 특허 문헌은 테스트집합으로 사용되므로 워드 임베딩 벡터 구축에서 122,280건을 제외한 191,916,278건(1996~2016)이 사용되었다.

20)의 조합을 사용하여 4가지 문헌 집합 당 각각 48가지의 워드 임베딩 벡터를 생성하였다.

3.2 기계학습 기반 특허문헌 분류 모델

기존 특허문헌 분류 연구에서 사용하였던 문헌 집합들은 개별 연구마다 자신만의 문헌집합을 구축하고 실험을 수행하였기 때문에, 동일한 환경에서 특허문헌에 대한 분류 성능을 평가하기 어렵다. 이에 본 논문에서는 기존 연구에서 많이 사용되었던 전통적인 기계학습 모델을 이용하여 특허 문헌에 대한 분류 성능을 측정하고, 이를 기반으로 딥러닝 모델의 분류 성능과 비교하기 위하여, 〈그림 2〉와 같이 기계학습 기반의 특허문헌 분류 모델[10]을 구현하였다.

기계학습 기반의 모델을 이용하여 본 논문에서 제안하는 워드 임베딩 벡터의 성능을 검증하기 위하여 나이브베이즈, kNN, SVM 3가지 모델을 사용하였고, 학습집합은 "초록"만을 이용하여 구성하였다. 기계학습 모델을 학습시키고 테스트하기 위해, 2015년도와 2016년도 문헌은 학습집합과 검증집합으로 사용하고, 2017년도 테스트집합으로 사용하였다. 학습집합과 검증집합은 2015~16년도를 통합한 문헌집합을 8:2 비율로 분할하여 생성하였고, 전체 학습집합의 구축 정보는 〈표 4〉와 같다.[11]



〈그림 2〉 기계학습 기반 특허 문헌 분류 실험 구성도

〈표 4〉 특허 문헌 분류를 위한 학습집합

분류체계	학습집합	검증집합	테스트집합	전체[12]	범주수
섹션	173,276	43,319	122,280	338,875	8
클래스	158,296	39,574	111,000	308,870	84
서브클래스	148,883	37,221	102,945	289,049	350

10) 기계학습 기반 모델은 파이썬으로 구현된 sklearn 라이브러리(https://scikit-learn.org)를 이용하였다.
11) 딥러닝 모델에서는 학습집합, 검증집합, 테스트집합이 그대로 이용되고, 기계학습모델에서는 학습집합과 검증집합이 통합된 형태로 제공된다.
12) 클래스와 서브클래스의 경우, 학습집합, 검증집합, 테스트집합 일부에서만 나타나는 범주값은 제외 시켰다. 이에 클래스는 전체 문헌의 94.98%, 서브클래스는 92.98%로 집합 크기가 축소되었다.

본 논문에서 기계학습 모델은 기존 기계학습 방법에서 사용하던 방법 중의 하나인 TF-ICF와 워드 임베딩 벡터를 이용하는 2가지 방법을 사용하여 자질을 추출하고 성능을 비교한다.

3.2.1 TF-ICF를 이용한 학습 자질 구성

정보검색과 텍스트마이닝 분야에서 단어에 대한 가중치 할당을 위해 TF-IDF(Term Frequency-Inverse Frequency)를 주로 사용한다. TF-IDF는 한 단어가 특정 문헌내에서 중요한 정도를 나타내는 통계적 수치로써, 다음과 같이 계산된다.

$TF(t, d) = freq(t, d)$:

한 문서 d에서 단어 t의 총 출현빈도

$IDF(t, D) = \log \frac{|D|}{|\{d \in D: t \in d\}|}$:

한 단어 t를 포함하는 문헌빈도의 역

$TF-IDF = TF(t, d) \times IDF(t, D)$

TF-IDF는 한 단어가 특정 문헌에서의 출현빈도는 높으면서, 전체 문헌 집합에서는 소수의 문헌에만 출현하는 경우에 높은 가중치를 갖게 된다. TF-ICF는 TF-IDF를 범주(Category)에 관한 수식으로 변형한 형태로써, 여러 범주를 갖는 분류 문제에서 유용한 단어를 선별하기 위해 사용되며, 그 식은 아래와 같다.

$TF_{ij} = avg(freq_{ij})$: 각 범주(c_j)내에서 해당 단어(f_i)의 빈도수 평균

$ICF_i = \log_2(N/n_i)$: N은 전체 범주의 수이고, n_i는 해당 단어가 출현한 범주의 수

$w(f_i c_j) = TF_{i,j} \times ICF_i$: 각 범주($c_j$)내에서 해당 단어($f_i$)의 가중치($w_{i,j}$)

전통적인 기계학습 방법에서 학습집합에서 출현하는 모든 단어들을 이용하여 학습하는 것이 불가능하기 때문에, 문헌 분류를 위해 유용한 단어들을 선별하는 작업을 먼저 수행하고, 이후 이 단어들에 가중치를 부여하는 작업을 수행하게 된다. 본 논문에서는 유용한 단어들을 선별하기 위하여 Chen and Yang(2012)와 임소라, 권용진(2017)의 연구에서 사용되었던 TF-ICF을 이용하였다.

〈표 5〉 H02 클래스의 서브클래스들에서의 높은 TF-ICF값을 갖는 상위 10단어

	H02B	H02G	H02H	H02J	H02K	H02M	H02N	H02P	H02S
1	분전	중배	선로	배터리	권선	컨버터	압전	지령	태양광
2	차단기	전선	전류	전력	스테이	전압	열전	전동기	태양전지
3	수배	케이블	배전	전압	로터	스위칭	척	인버터	발전
4	전반	배전	차단기	무선	요크	인버터	전소	계자	인버터
5	부스	용부	계전기	충전기	계자	직류	전극	전압	집광
6	배전	해저케이블	전압	충전	정류자	커패시터	압전체	권선	태양
7	내진	애자	계전	직류	영구자석	역률	하베스터	전류	집광판
8	모선	송전	직류	방전	초전도	전류	발전	모터	솔라셀
9	외함	선로	서지	계통	코일	정류	하베	역기	지붕
10	지진	포설	한류	컨버터	발전기	트랜스포머	정전	스위칭	수상

최상위 분류체계인 섹션에서 살펴보면, A부터 H까지 8개의 범주에 대하여 각 단어들에 대한 TF-ICF값이 계산되고, 각 범주별로 최상위 TF-ICF값을 갖는 단어들부터 추출하여 전체 5,000[13]개의 단어를 추출하는 과정을 수행한다. 이때 중복되는 단어는 제거되고, 전체 개수가 5,000개가 될 때까지 반복적으로 작업이 진행된다. 〈표 5〉는 H02의 서브클래스에서 추출된 단어 들 중 각 범주별로 최상위 TF-ICF 값을 갖는 10개의 단어를 보여준다. 이 작업이 완료된 후, 추출된 5,000개의 단어를 이용하여 자질을 추출할 때, 각 단어에 대한 가중치는 TF-IDF 값을 사용하여 할당한다.

3.2.2 워드 임베딩 벡터를 이용한 학습 자질 구성

워드 임베딩 벡터를 이용하면 문헌내의 단어를 표현하기 위해 필요한 공간은 워드 임베딩 벡터에 할당된 벡터의 크기로 축소된다. 예를 들어, 워드 임베딩 벡터 구축 시 사용한 벡터 크기가 200이라면 한 단어를 표현하기 위한 공간은 전체 단어의 개수 85,417차원이 아닌 200차원이 된다. 학습 자질을 추출하기 위해 현재까지 가장 성능이 좋다고 알려진 fastText(벡터 크기 200, 윈도우 크기 15)를 사용하였다. TF-ICF를 사용한 경우와 달리, 문헌 내의 모든 단어들에 워드 임베딩 벡터를 적용한 값들을 합산한 후 이에 대한 평균값을 사용하여 문헌을 표현하는 벡터로 구성하였다.

〈표 6〉은 구축된 워드 임베딩 벡터를 이용하여 "유전자"라는 단어와의 유사도를 계산하여 가장 유사도가 높은 상위 20개의 단어를 보여주고 있다. "발현", "단백질", "서열" 등의 단어가 가장 높은 유사도를 갖는 단어로 나타나고 있으며, 관련성이 있는 단어들에 유사한 워드 임베딩 벡터가 할당되었음을 볼 수 있다.

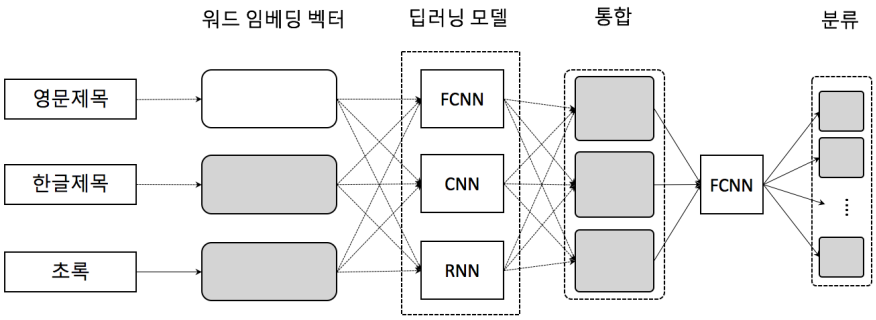
3.3 딥러닝 기반 특허 문헌 분류 모델

〈그림 3〉은 본 논문에서 사용한 딥러닝 기반

〈표 6〉 "유전자"와 유사도가 높은 상위 20 단어 목록

순번	단어	유사도	순번	단어	유사도
1	발현	0.9485	11	벡터	0.6289
2	단백질	0.8837	12	신규	0.5455
3	서열	0.8616	13	코딩	0.5377
4	세포	0.8111	14	동물	0.5150
5	항체	0.7791	15	분자	0.5104
6	바이러스	0.7626	16	배양	0.4814
7	효소	0.7337	17	진단	0.4687
8	면역	0.7066	18	질환	0.4485
9	조합	0.6583	19	전이	0.4432
10	인자	0.6322	20	수준	0.4385

13) 상위 5,000개 단어의 출현빈도는 전체 73,615개 단어의 출현빈도의 83.09%를 차지하며, 이는 전체 문서에 대한 충분한 표현력이 있다고 예상된다.



〈그림 3〉 딥러닝 기반 특허문헌 분류 모델 구성도

특허문헌 분류 모델에 대한 구성도이다.[14] 본 논문에서는 특허 문헌의 필드 중 초록을 특허 분류의 기본 자질로 사용하고, 영문제목, 국문제목을 추가 자질로 사용한다. 이에 딥러닝 모델에 대한 입력은 영문제목, 국문제목, 초록으로 구성되며, 이 중 초록은 모든 실험에서 항상 사용되고, 영문제목과 국문제목은 4.6 실험에서만 추가적으로 사용한다.

국문제목과 초록은 기 구축된 워드 임베딩 벡터를, 영문제목은 무작위로 초기화된 벡터를 이용하고, 학습을 진행 하는 동안 해당 워드 임베딩 벡터를 동적으로 학습한다. 입력된 문헌은 임베딩 벡터를 거쳐 축소된 차원으로 변경되고, 실험에서 선택된 3가지 딥러닝 모델 중 한 가지를 적용하여 계산을 수행하고 그 결과가 통합되어 최종 분류를 위한 FCNN 모델로 입력되고 분류가 수행된다.

〈그림 3〉의 특허 문헌 분류 모델의 세부 구조를 입력 자질 정보의 구성, 각 입력 자질 분석을 위한 인코딩 층, 각 자질 벡터를 병합하여 최종 처리하는 통합 분석층으로 나누어 설명한다.

3.3.1 입력 자질 정보 구성

특허 문헌 분류 성능에 미치는 여러 조건들을 실험하기 위해서는 실험대상이 되는 조건이외에 나머지 조건들은 단순하게 설정되어야 한다. 이에 4.6을 제외한 나머지 실험은 초록만을 입력으로 설정하여 해당 실험 조건에 대한 성능을 파악하도록 하였다.

특허 문헌내의 단어들은 해당되는 임베딩 벡터를 이용하여 수치 벡터로 치환하는 과정을 거친다. 임베딩 벡터는 기 학습한 임베딩 벡터를 활용하는 방법과 자체적으로 무작위로 구성한 임베딩 벡터를 활용하는 방법을 제공한다. 두 방법 모두 임베딩 벡터를 동적으로 학습하도록 하여, 학습 과정에서 학습집합에 최적화된 임베딩 벡터를 구성하게 된다.

3.3.2 딥러닝 모델 층

본 논문에서는 입력된 임베딩 벡터에 대해 다양한 딥러닝 모델을 적용하여 실험할 수 있도록 구성하였다. 활용할 수 있는 딥러닝 구조는 가장 기본적인 모델인 완전연결 네트워크(FCNN: Fully-Connected Neural Networks),

14) 딥러닝 모델은 파이썬으로 구현된 텐서플로우 라이브러리(https://tensorflow.org)를 이용하였다.

CNN(Convolutional Neural Networks), RNN(Recurrent Neural Networks) 3가지이다.

FCNN은 딥러닝의 가장 기반이 되는 네트워크 구조이다. 입력 벡터의 노드와 출력 벡터의 노드가 모두 완전 연결을 이루는 형태로, 일반적인 자질 정보를 추출하는데 사용되는 모델이다. 입력된 단어와 벡터의 곱으로 이루어진 2차원 자질 벡터를 1차원으로 평탄화하여 입력하고, FCNN을 거친 후 출력 벡터로 변환된다. FCNN의 활성화 함수는 사전실험을 통하여 ReLU(Rectified Linear Unit)로 선택하였다.

CNN은 최초 이미지 분석을 위해 개발된 딥러닝 구조로, 일정한 크기의 가중치 필터를 활용하여 개별적으로 계산한 결과를 모두 합한 값을 구하는 연산이다. 이미지 분석에서 주로 활용되던 CNN은 Collobert and Weston(2008)의 연구를 기점으로 점점 자연어 처리에도 활용되면서, 문장의 구조 분석을 활용한 연구에서 우수한 성능을 보였다. 본 논문에서 입력된 2차원 자질 벡터는 이미지와 유사한 2차원 벡터로 구성되고, 실험에서 설정된 필터 단위로 1차원적인 컨볼루션 연산을 수행하고 그 결과에 최대값 풀링을 적용하여 결과 벡터를 생성한다.

RNN은 연속적인 입력에 대한 분석에 우수한 성능을 보이는 딥러닝 구조로, 자연어 처리 연구 분야에서 전반적으로 가장 우수한 성능을 보이는 네트워크 구조이다. RNN은 일정한 연산을 수행하는 셀(Cell)을 두고, 입력 값을 순차적으로 연산하면서 가중치를 갱신하는 형태의 네트워크로, 문맥이나 단어 목록 내의 연결성을 분석한다. 본 논문에서는 가장 일반적으로 활용되는 RNN 셀 구조인 LSTM(Long Short-Term Memory)을 통해 정방향 RNN 연산을 수행하는 방법을 선택하였다. 입력된 벡터는 단어 단위로 LSTM 셀 내부 구조를 통해 연산되며, 이전 입력 벡터의 결과와 결합되어 현재 상태와 출력 값을 생성하게 된다. 마지막 입력 벡터에 대한 연산이 종료되면 해당 셀의 결과가 최종 출력 벡터값으로 다음 층으로 전달된다.

3.3.3 출력 통합 및 분류

3.3.2 과정을 통하여 3가지의 딥러닝 구조에 대해 각 입력 자질은 실험에서 선택한 딥러닝 네트워크 연산을 수행한다. 초록만을 입력으로 한 실험에서는 결과 벡터를 그대로 활용하게 되고, 국문제목, 영문제목을 함께 실험하는 4.6에서는 각 연산에 대한 결과 벡터가 결합되어 통합 벡터로 변환되어 분류를 위한 입력으로 사용된다.

최종 분류를 위한 딥러닝 모델은 3.3.2에서 설명된 FCNN을 사용하였으며, 다중 분류를 위해 시그모이드(Sigmoid)를 활성함수로 적용하였다. 최종 출력은 해당되는 IPC 계층의 범주 갯수 만큼의 차원을 갖는 배열이 되고, 각 배열의 값이 임계치 0.5 이상인 경우 이 문헌의 범주값으로 사용하게 된다.

4. 특허문헌 분류 성능 평가

4.1 성능평가 대상 및 방법

본 논문에서는 특허문헌 분류를 위해 사용되는 다양한 파라미터들 중에서 가장 좋은 파라미터를 찾기 위하여 〈표 7〉과 같이 성능평가 대상을 구분하여 실험하였다.

〈표 7〉 특허 문헌 분류를 위한 실험 종류

순번	성능평가 대상	대상	사용 모델	목적
실험1 (4.2)	TF-ICF와 워드임베딩 벡터	초록	기계학습 모델 (NB, KNN, SVM)	전통적인 기계학습 방법에서 주로 사용하였던 TF-ICF와 워드 임베딩 벡터에 대한 성능 비교
실험2 (4.3)	기계학습 모델과 딥러닝 모델	초록	기계학습 모델 (NB, KNN, SVM) 딥러닝 모델 (FCNN, CNN, RNN)	동일한 입력(초록)에 대하여 기계학습 모델과 딥러닝 모델의 성능 비교
실험3 (4.4)	워드 임베딩 벡터	초록	딥러닝 모델 (CNN)	다양한 워드 임베딩 벡터에 대한 성능 비교(실험2에서 최고 성능을 보여준 모델과 하이퍼 파라미터 사용)
실험4 (4.5)	단어추출 방법	초록	딥러닝 모델 (CNN)	학습집합 구성에 따른 성능 비교 (실험2에서 최고 성능을 보여준 모델과 하이퍼 파라미터 사용)
실험5 (4.6)	영문제목, 국문제목 추가	초록 제목	딥러닝 모델 (CNN)	영문제목과 국문제목 추가에 따른 성능 비교 (실험2에서 최고 성능을 보여준 모델과 하이퍼 파라미터 사용)

4.2 TF-ICF 와 워드 임베딩 벡터 성능 비교

기존 기계학습 방법에서 사용한 자질 추출방법 중의 하나인 TF-ICF와 본 논문에서 제안한 워드 임베딩 벡터를 자질로 사용하는 방법을 전통적인 기계학습 모델을 이용하여 성능을 비교하였다. 학습집합은 NOUN을 사용하였고, 기계학습 모델로는 나이브베이즈, kNN, SVM을 이용하였다. 워드 임베딩 벡터는 CBOW, Skip-Gram, GloVe, fastText 4가지[15]를 각각 비교하였다.

기계학습 모델 중 나이브베이즈 모델은 자질 추출의 방법과 상관없이 분류 성능의 차이가 크지 않았으며, 다른 모델들에 비해 매우 낮은 성능을 보여주었다.

SVM의 경우 섹션에 대한 분류에서는 워드 임베딩 벡터로 fastText를 사용하였을 경우 근소하게 좋은 성능을 보여주었으나, 클래스와 서브클래스 분류에서는 전통적인 TF-ICF 방법이 더 좋은 결과를 보여주었다. SVM은 전체 자질 중에서 분류를 위해 필요로 되는 지지벡터들만을 선별하여 사용하는 특징을 지닌 모델이다. 본 실험에서 SVM이 워드 임베딩 벡터의 전체 차원을 제대로 활용하지 못하는 반면, TF-ICF에서는 필요한 소수의 지지벡터들을 잘 선별하였다고 볼 수 있다.

kNN 모델은 모든 실험에서 워드 임베딩 벡터를 사용하였을 경우에 압도적으로 좋은 성능을 보여주었으며, 다른 모델과의 비교에서도 섹션을 제외하고 가장 우수한 성능을 보여주었다. kNN은 분류대상 문헌을 표현하는 모든 벡터들과 학습집합에 있는 모든 문헌들에 대한 자질을 전부 비교하여 분류값을 결정한다. 즉, 자질로 추출된 모든 벡터들이 분류를 위해 사용된다는 점이 SVM과의 차이점이 있고, 워드 임베딩 벡터의 모든 차원(자질)이 분류를 위해 사용되기 때문에, 워드 임베딩 벡터의 성능을 가장 잘 보여주는 모델이라 볼 수 있다.

15) 사전실험을 통하여 벡터크기는 200, 윈도우 크기는 15인 워드 임베딩 벡터를 사용하였다.

〈표 8〉 TF-ICF와 워드벡터를 이용한 자질추출에 대한 성능 비교

학습집합	모델	TF-ICF (f1-score)	워드 임베딩 벡터(f1-score)			
			CBOW	SKIP	GloVe	fastText
섹션	NB	51.13	56.19	59.34	58.61	59.20
	KNN	31.79	53.54	66.57	66.01	68.17
	SVM	69.02	57.92	70.10	59.09	**71.77**
클래스	NB	28.03	29.33	33.77	32.62	33.26
	KNN	15.23	34.79	51.18	50.69	**52.81**
	SVM	49.45	25.86	46.38	46.46	47.39
서브클래스	NB	17.14	12.12	17.18	15.20	15.92
	KNN	9.80	19.47	39.60	38.44	**40.53**
	SVM	34.08	7.71	26.67	25.53	26.23

한편, TF-ICF는 학습집합으로 구성된 데이터만 활용하는데 비해, 워드 임베딩 벡터는 수집된 전체 데이터를 이용하여 선학습 하였기 때문에 공정하지 않은 비교로 보일 수 있다. 학습집합만을 이용하여 구축된 워드 임베딩 벡터를 사용하는 방법과 유사한 실험이 4.4의 무작위로 초기화된 워드 임베딩 벡터를 이용한 실험이다. 무작위로 초기화된 워드 임베딩 벡터를 이용한 방법 또한 충분히 높은 성능[16]을 보여주고 있기 때문에, 특허 문헌에서 워드 임베딩 벡터가 TF-ICF보다 좋은 자질추출 방법임을 알 수 있다.

모든 분류 실험에서 워드 임베딩 벡터 중 fastText를 사용하였을 때 가장 좋은 성능을 보여주었기 때문에, 이후 진행되는 실험에서 4.4를 제외하고 워드 임베딩 벡터를 fastText로 고정하고 진행하였다.

4.3 기계학습 모델과 딥러닝 모델의 성능 비교

4.2 실험에서 여러 워드 임베딩 벡터 중에서 가장 높은 성능을 보여준 fastText를 적용하여 딥러닝의 분류 성능을 진행하였고, 4.2의 fastText를 사용한 기계학습 모델과 성능을 비교하였다. 딥러닝 모델 실험을 위해 사용된 하이퍼 파라미터는 〈표 9〉와 같다.

〈표 10〉은 4.2의 기계학습 모델의 실험결과와 딥러닝 모델의 실험결과를 보여준다. 섹션, 클래스, 서브클래스 모든 분류 실험에서 딥러닝 모델이 기계학습 모델보다 훨씬 좋은 성능을 보여주었고, 딥러닝 모델 중 가장 좋은 성능을 보여준 모델은 CNN이었다. 딥러닝 모델과 기계학습 모델의 성능 차이를 구체적으로 살펴보면, 섹션 분류에서 딥러닝의 CNN 모델과 기계학습의 SVM 모델의 성능차이는 4.36으로 크지 않지만, 클래스 분류에서는 CNN과 kNN이 16.93, 서브클래스 분류에서는 19.51로 그 차이가 더욱 크게 나타났다. 분류할 범주갯수의 증가로 분류 복잡도가 높아질수록 딥러닝 모델이 더 적합한 분류 모델임을 알 수 있다.

16) 분류 성능은 섹션(76.25%), 클래스(67.14%), 서브클래스(54.83%)이며, 4.4에서 상세히 설명한다.

〈표 9〉 딥러닝 모델 실험에 사용된 하이퍼 파라미터 종류와 범위

하이퍼 파라미터	범위	설명	최적 파라미터[17]
워드 벡터	fastText (v200, w15)	fastText의 v옵션은 워드벡터의 크기 w옵션은 워드벡터 구축 시 사용된 문맥의 길이	-
은닉층의 크기	100, 300, 600	워드 임베딩 벡터의 출력 크기, 딥러닝 모델로의 입력 크기	600
딥러닝 모델	FCNN, CNN, RNN	RNN은 LSTM을 사용	CNN
필터 크기	2, 3, 5	CNN에서만 사용된다	5, 5, 3
드롭아웃	0.5, 0.7, 0.9	은닉층의 유지 비율	0.9, 0.7, 0.5
배치 크기	20, 50, 100	학습에서 사용되는 미니배치의 크기	20, 20, 100
최적화 알고리즘	adam, rmsprop, sgd	네트워크 가중치를 최적화하기 위한 알고리즘	adam
학습률	0.01, 0.001, 0.005, 0.0001	학습과정에서 손실값을 최소화하기 위해 가중치를 변경하는 비율	0.0001, 0.0001, 0.001
학습회수	10 (5)	사전 조사를 통해 FCNN, CNN의 경우는 10, RNN의 경우에는 5가 적절한 학습횟수로 선택됨	10, 7, 10

〈표 10〉 기계학습과 딥러닝 모델의 분류 성능 비교표

학습집합	모델		정확률(precision)	재현율(recall)	조화평균(f1-score)
섹션	기계학습	NB	52.55	72.74	59.20
		kNN	78.70	60.32	68.17
		SVM	82.92	63.85	71.77
	딥러닝	FCNN	65.92	84.6	73.84
		CNN	67.25	88.28	**76.13**
		RNN	65.31	86.75	74.16
클래스	기계학습	NB	23.39	82.12	33.26
		kNN	69.38	43.81	52.81
		SVM	78.70	36.80	47.39
	딥러닝	FCNN	67.89	63.26	64.48
		CNN	68.61	71.81	**69.74**
		RNN	65.97	72	68.24
서브클래스	기계학습	NB	9.58	85.01	15.92
		kNN	60.30	32.45	40.53
		SVM	58.20	19.26	26.23
	딥러닝	FCNN	60.02	48.63	52.59
		CNN	64.77	57.87	**60.04**
		RNN	61.17	59.18	58.82

17) 실험 후 가장 좋은 분류 성능에 사용된 하이퍼 파라미터. 섹션, 클래스, 서브클래스 모두 동일한 경우에는 한 가지 값만 표기하였다.

4.4 워드 임베딩 벡터별 성능 평가

워드 임베딩 벡터를 이용한 실험은 2가지 경우로 분리하여 살펴본다. 첫 번째, 무작위로 초기화된 워드 임베딩 벡터와 선학습된 워드 임베딩 벡터를 이용한 분류 성능을 알아본다. 무작위로 초기화된 워드 임베딩 벡터는 학습집합에 대한 데이터에서만 학습이 진행되고, 선학습된 워드 임베딩 벡터는 수집된 전체 문헌집합으로 구축된 상태에서 추가적으로 학습이 진행된다. 딥러닝 모델은 4.3의 실험에서 가장 우수한 성능을 보였던 CNN 모델을 사용하고, 최고 성능을 보여주었을 때 사용하였던 하이퍼 파라미터로 고정한 후, 무작위 초기화된 벡터와 선학습된 워드 임베딩 벡터[18]에 대해 실험을 진행하였다.

〈표 11〉에서 보는바와 같이, 모든 분류 체계에서 선학습 벡터를 이용하고 학습을 진행하는 방법이 더 우수한 결과를 보여주었으며, 하위 분류 체계로 갈수록 분류 성능이 더 차이가 나고 있음을 알 수 있다. 분류할 범주의 수가 많아져서 분류 복잡도가 증가할수록 선학습된 임베딩 벡터를 사용하는 방법이 분류 성능에 도움을 준다고 볼 수 있다.

다음으로, 두 번째는 선학습된 워드 임베딩 벡터 48가지에 대한 성능 분포를 살펴본다. 딥러닝 모델은 앞의 실험과 마찬가지로 동일한 CNN 모델과 하이퍼 파라미터를 사용하였고, 48가지의 워드 임베딩 벡터에 대해 실험을 진행하였다.

〈그림 4〉는 전체 48회의 실험에서 성능이 좋은 상위 15개(왼쪽 그래프)와 성능이 나쁜 하위 15개(오른쪽 그래프)에 대한 워드 임베딩 벡터 종류에 대한 분포이다. x축은 워드 임베딩 벡터의 종류를 나타내고, y축은 해당 워드 임베딩 벡터가 포함된 개수이다. 왼쪽 그래프에 많이 나타나고, 오른쪽 그래프에 적게 나타날수록 좋은 워드 임베딩 벡터라고 볼 수 있다.

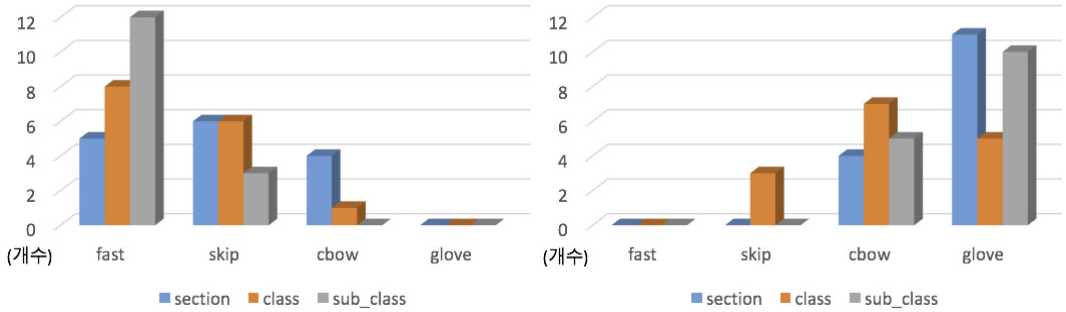
섹션에 대한 분류에서는 fastText, Skip-Gram, CBOW가 좋은 성능을 보여주고 있는 반면, GloVe는 하위 분포에만 나타남을 알 수 있다. 클래스와 서브클래스 분류에서 fastText는 상위 분포에서 가장 많이 나타난 워드 임베딩 벡터 종류였으며, 하위 분포에는 1건도 나타나지 않았다. 이와 같이, 특허 문헌에 대해서 워드 임베딩 벡터는 fastText, Skip-Gram, CBOW, GloVe 순으로 좋은 표현력을 가지고 있는 것으로 나타났다.

〈그림 5〉는 벡터 차원의 크기에 따른 성능 분포이다. 〈그림 4〉와 마찬가지로 왼쪽 그래프

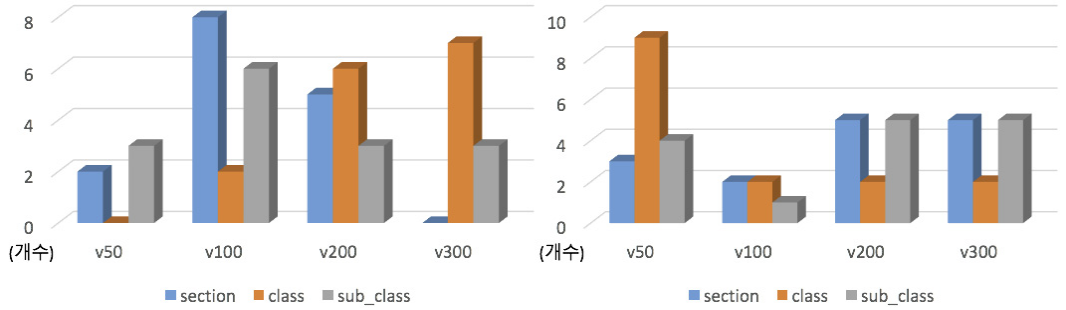
〈표 11〉 워드 임베딩 벡터 적용 방법 방법에 따른 분류 성능

학습집합	적용방법	정확률	재현율	조화평균
섹션	무작위 초기화	71.44	82.34	76.24
	선학습 벡터	67.31	88.22	**76.25**
클래스	무작위 초기화	67.88	67.96	67.14
	선학습 벡터	68.65	72.37	**69.70**
서브클래스	무작위 초기화	26.36	62.50	54.83
	선학습 벡터	64.22	58.75	**60.10**

18) 4.2에서 사용하였던 fastText로 선학습된 벡터 크기 200, 윈도우 크기 15를 사용하였다.



〈그림 4〉 워드 임베딩 벡터 종류에 따른 분포



〈그림 5〉 임베딩 벡터 크기에 따른 성능 분포

는 성능이 좋은 상위 15개에 대한 분포이고, 오른쪽은 성능이 낮은 하위 15개에 대한 분포이다. 섹션, 클래스, 서브클래스에 따라 좋은 성능을 나타내는 벡터 차원의 크기는 상이하지만, 벡터 차원 50을 사용하였을 경우에는 명백하게 좋지 않은 성능을 보여 주고 있다. 또한 벡터 차원 300은 섹션 분류에서 상위 20에 속한 경우가 한번도 나타나지 않았다. 벡터차원 100이나 200을 사용한 경우에 전체적으로 좋은 성능을 보여주었다.

4.5 학습 자질 추출에 따른 성능 비교

딥러닝 모델은 4.3과 마찬가지로 CNN을 사용하였고, 하이퍼 파라미터도 4.3과 동일하게 설정하였으며, 워드 임베딩 벡터는 4.4에서 가장 우수한 성능을 보였던 fastText(v200, w15)를 사용하여 실험을 진행하였다.

모든 분류에서 명사, 형용사, 동사를 추출하여 학습집합을 구축한 NAV 방식이 가장 좋은 성능을 보여주었고, 명사만 대상 단어로 이용한 NOUN 방식이 근소한 차이로 낮은 성능을 보여줌을 알 수 있다. 2GRAM과 3GRAM 방식을 이용한 단어 추출 방식은 NAV와 NOUN 방식에 비해 좋지 않은 성능을 보여주었다.

4.6 제목 추가에 따른 성능 비교

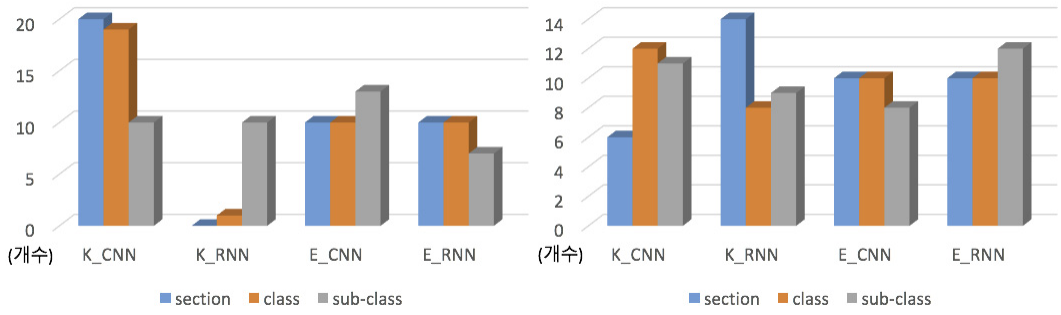
4.5까지는 특허 문헌의 초록만을 대상으로 실험이 진행되었고, 본 실험에서는 영문제목과

〈표 12〉 단어 추출 방법에 따른 분류 성능 비교

학습집합	단어추출방법	정확률	재현율	조화평균
섹션	NOUN	67.25	88.28	76.13
	NAV	67.69	88.38	**76.51**
	2GRAM	66.94	87.13	75.57
	3GRAM	65.92	85.05	74.10
클래스	NOUN	68.61	71.81	69.74
	NAV	68.65	72.80	**70.32**
	2GRAM	67.99	70.42	68.79
	3GRAM	66.59	67.87	66.61
서브클래스	NOUN	64.77	57.87	60.04
	NAV	63.42	59.52	**60.35**
	2GRAM	61.81	56.60	58.05
	3GRAM	63.23	52.63	55.88

국문제목을 추가하여 그 성능을 비교한다. 국문제목과 영문제목을 제외한 나머지 하이퍼 파라미터는 4.5와 동일하고, 학습집합은 4.5에서 좋은 실험결과를 보여준 NOUN과 NAV를 사용하였다. 국문제목과 영문제목에 대한 딥러닝 모델로 CNN과 RNN의 조합을 선택하도록 조건을 구성한 후 실험을 수행하였다.

〈그림 6〉은 국문제목과 영문제목에 CNN과 RNN을 적용[19]한 경우에 대해 상위 표본(왼쪽 그래프)과 하위 표본(오른쪽 그래프)에 대한 분포를 보여준다. 국문제목(K_CNN, K_RNN)의 경우 CNN을 사용한 경우 왼쪽 그래프에 많은 분포를 보이지만, RNN의 경우에는 오른쪽 그래프에서 더 많은 분포를 보이고 있다. 즉, 국문제목에서는 CNN이 RNN에 비해 더 좋은 성능을 보여줌을 알 수 있다. 영문제목(E_CNN, E_RNN)의 경우에는 왼쪽과 오른쪽의 그래프에서 분포 차이가 크지 않다. 즉, CNN과 RNN에 따라 큰 성능의 차이는 없고, 국문제목보다 분류 성능에 미치는 영향이 적다고 볼 수 있다.



〈그림 6〉 딥러닝 모델의 성능 분포

19) 국문제목에 CNN을 적용한 경우는 K_CNN, RNN을 적용한 경우는 K_RNN으로 표시하였다.

〈표 13〉 제목 추가에 따른 분류 성능 비교

학습집합	단어추출방법	정확률	재현율	조화평균
섹션	ABS(NOUN)	67.25	88.28	76.13
	ABS+TI(NOUN)	68.66	89.06	77.41
	ABS+TI(NAV)	68.79	88.98	**77.48**
클래스	ABS(NOUN)	68.61	71.81	69.74
	ABS+TI(NOUN)	69.59	73.45	71.07
	ABS+TI(NAV)	69.97	74.08	**71.65**
서브클래스	ABS(NOUN)	64.77	57.87	60.04
	ABS+TI(NOUN)	64.79	59.76	61.49
	ABS+TI(NAV)	64.45	61.54	**62.06**

〈표 13〉은 국문제목과 영문제목을 추가한 분류 성능에 대한 결과이다. 섹션, 클래스, 서브클래스 모든 실험에서 국문제목과 영문제목을 추가한 경우에 초록만을 사용하여 분류를 수행한 경우보다 좋은 성능을 나타내고 있으며, 4.5 실험과 마찬가지로 NAV를 사용한 경우에 가장 좋은 성능을 보여주었다.

5. 결 론

본 논문에서는 특허 문헌 분류에 적합한 자질과 분류 모델을 알아보기 위하여, 다양한 자질 추출 방법과 기계학습 및 딥러닝 모델을 사용하여 분류 성능을 살펴보았으며, 그 결과를 정리하면 다음과 같다.

첫째, 특허 문헌 분류를 위한 자질 추출 방법에서, 단어에 대한 분산표현인 워드 임베딩 벡터를 사용하였을 경우에 전통적인 BoW 방식보다 더 좋은 성능을 보여주는 것으로 나타났다.

둘째, 딥러닝 모델을 적용하였을 경우 기존 기계학습 모델보다 섹션, 클래스, 서브클래스의 모든 분류 실험에서 좋은 성능을 보였으며, 범주갯수가 많아 분류에 대한 난이도가 더 높아질수록 딥러닝이 더 우수한 결과를 보여주는 것으로 평가되었다.

셋째, 워드 임베딩 벡터에 대한 실험에서는 fastText, Skip-Gram, CBOW, GloVe 순으로 좋은 성능을 보여주는 것으로 나타났다.

넷째, 특허 문헌으로부터 단어 추출 시 2GRAM, 3GRAM 같은 멀티그램 방식보다는 형태소 분석기를 적용한 품사 태깅 결과를 이용한 방법이 유용한 것으로 나타났다.

다섯째, 분류를 위한 자질로 초록만을 사용하는 것보다 국문제목과 영문제목 등의 자질을 추가할 때, 성능이 향상되는 것으로 입증되었다.

분산표현 기법인 워드 임베딩 벡터를 사용하고, 형태소 분석을 이용한 문헌집합 구축을 기반으로 CNN 모델을 적용하였을 경우에 분류성능이 가장 우수한 것으로 나타났으며, 섹션, 클래스, 서브클래스 분류 실험에서 전통적인 기계학습 방법에 비해 각각 5.71%, 18.8%, 21.53% 우수한 분류 성능을 보여주었다.

분산표현과 딥러닝 모델을 신문기사 분류나 감성분류 등에 적용한 연구는 있었지만, 지금까지 특허 문헌 분류에 적용한 사례는 없었다.

본 논문은 딥러닝 모델 중 FCNN, CNN, RNN의 기본적인 기능만을 사용하여 분류 시스템을 구현하였고, 분류 성능을 전통적인 기계학습 모델과 비교하여 그 우수성을 입증하였으며, 이를 통해 향후 특허 문헌 분류 모델 연구에 대한 방향성을 제시하였다.

향후 연구로는 특허 문헌에서 제공하는 배경기술 및 기술분야 등의 필드를 분류를 위한 자질로 추가하고, 현재 서브클래스 보다 하위 수준인 메인그룹이나 서브그룹까지 분류 범위를 확대하고, 기본적인 딥러닝 구조를 확장 및 혼합한 분류 모델에 대한 연구가 수행될 필요가 있다.

참 고 문 헌

[1] 김재호, 최기선. 2005. 문서의 의미적 구조정보를 이용한 특허 문서 분류.『한국정보과학회 언어공학연구회 학술발표 논문집』, 28-34.

[2] 박찬정, 김기용, 성동수. 2014. KNN 을 이용한 융합기술 특허문서의 자동 IPC 분류.『한국정보기술학회논문지』, 12(3): 175-185.

[3] 임소라, 권용진. 2017. 특허문서 필드의 기능적 특성을 활용한 IPC 다중 레이블 분류.『인터넷정보학회지』, 18(1): 77-88.

[4] 특허청. 2018.『2017 지식재산통계연보』. 대전: 특허청.

[5] 한국과학기술원 융합연구정책센터. 2018.『2017년도 국가융합기술 R&D 조사・분석』. 서울: 한국과학기술원 융합연구정책센터.

[6] Bahdanau D., Cho, K. and Bengio, Y. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate." In *Proceeding of ICLR 2015*. [arXiv:1409.0473]

[7] Bojanowski, P. et al. 2017. "Enriching word vectors with subword information." *Transactions of the Association for Computational Linguistics*, 5: 135-146.

[8] Chen, Y. and Chang, Y. 2012. "A three-phase method for patent classification." *Information Processing & Management*, 48(6): 1017-1030.

[9] Collobert, R. and Weston, J. 2008. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning." In *Proceeding of the 25th International Conference on Maching Learning*.

[10] Fall, C. et al. 2003. "Automated categorization in the international patent classification." In *Acm Sigir Forum*, 37(1): 10-25.

[11] Koster, C. and Seutter, M. 2003. "Taming wild phrases." In *Proceedings of the 25th European*

conference on IR research (ECIR'03), 161-176.

[12] Larkey, L. 1999. "A patent search and classification system." In *Proceedings of the fourth ACM conference on Digital libraries*, 179-187.

[13] Mikolov, T., Chen, K., Corrado, G. and Dean, J. 2013. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781.

[14] Pennington, J., Socher, R. and Manning, C. 2014. "Glove: Global vectors for word representation." In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543.

[15] Tikk, D., Biró, G. and Törcsvári, A. 2008. "A hierarchical online classifier for patent categorization." *Emerging technologies of text mining: Techniques and applications. IGI Global*, 244-267.

• 국문 참고자료의 영어 표기
(English translation / romanization of references originally written in Korean)

[1] Kim, Jao-Ho and Choi, Key-Sun. 2005. "Patent Document Categorization based on Semantic Structural Information." *Proc. of the 17th Annual Conference on Human and Cognitive Language Technology*, 28-34.

[2] Park, Chanjeong, Kim, Kiyong and Seong, Dongsu. 2014. "Automatic IPC Classification for Patent Documents of Convergence Technology Using KNN." *Journal of Korean Institute of Information Technology*, 12(3): 175-185.

[3] Lim, Sora and Kwon, Yongjin. 2017. "IPC Multi-label Classification based on Functional Characteristics of Fields in Patent Documents." *Review of Korean Society for Internet Information*, 18(1): 77-88.

[4] Korean Intellectual Property Office. 2018. *Intellectual Property Statistics for 2017*. Daejeon: Korean Intellectual Property Office.

[5] KIST, Convergence Research Policy Center. 2018. *Research and Analysis of National Convergence Technology R & D in 2017*. Seoul: KIST, Convergence Research Policy Center.