The Korean Journal of Applied Statistics (2020) DOI: https://doi.org/10.5351/KJAS.2020.33.5.615
33(5), 615–625

Document classification using a deep neural network in text mining

Bo-Hui Lee[a] · Su-Jin Lee[b] · Yong-Seok Choi[b,1]

[a]Department of Advertising and Public Relations, Silla University;
[b]Department of Statistics, Pusan National University

(Received June 3, 2020; Revised July 13, 2020; Accepted August 21, 2020)

Abstract

The document-term frequency matrix is a term extracted from documents in which the group information exists in text mining. In this study, we generated the document-term frequency matrix for document classification according to research field. We applied the traditional term weighting function term frequency-inverse document frequency (TF-IDF) to the generated document-term frequency matrix. In addition, we applied term frequency-inverse gravity moment (TF-IGM). We also generated a document-keyword weighted matrix by extracting keywords to improve the document classification accuracy. Based on the keywords matrix extracted, we classify documents using a deep neural network. In order to find the optimal model in the deep neural network, the accuracy of document classification was verified by changing the number of hidden layers and hidden nodes. Consequently, the model with eight hidden layers showed the highest accuracy and all TF-IGM document classification accuracy (according to parameter changes) were higher than TF-IDF. In addition, the deep neural network was confirmed to have better accuracy than the support vector machine. Therefore, we propose a method to apply TF-IGM and a deep neural network in the document classification.

Keywords: document classification, deep neural network, term weighting, text mining, keyword extraction

1. 서론

빅데이터 시대가 도래하면서 SNS, 신문, 이메일과 같은 비정형 데이터의 양이 기하급수적으로 증가하여 텍스트 마이닝(text mining)의 활용 분야 또한 다양해지고 있다. 텍스트 마이닝은 자연어 처리 기술과 문서 처리 기술을 적용하여 유용한 정보를 얻는 기법으로, 그 중 문서 분류는 자연어 처리 분야의 중요한 영역 중 하나이다. 문서 분류를 위한 문서-용어 빈도행렬(document-term frequency matrix)은 그룹 정보가 존재하는 문서들의 용어를 추출한 것으로 문서가 행으로, 용어가 열로 이루어져 있으며 해당 문서에 해당 용어의 빈도 값을 알려준다. 하지만 문서-용어 빈도행렬에서 문서와 용어의 빈도 값만으로 그 중요도를 알기 어렵다. 따라서 용어 가중치(term weighting)를 적용하여 문서 용어들의 중요도를 수치형 자료로 변환해야 한다.

This work was supported by the Phase Four of the Brain Korea 21 Project in 2020.
[1]Corresponding author: Department of Statistics, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-Gu, Busan 46241, Korea. E-mail: yschoi@pusan.ac.kr

이러한 텍스트 마이닝 과정을 거친 문서-용어 빈도행렬에서 문서 분류를 위한 알고리즘으로는 먼저 Jung 등 (2019)의 연구에서와 같이 개체들의 시각적 군집화 기법인 다차원척도법(multidimensional scaling)과 군집분석(cluster analysis), 판별분석(discriminant analysis) 등이 있다. 그리고 머신러닝(machine learning)은 텍스트, 음성, 이미지 등의 여러 자료를 분류하거나 수치적 예측 및 패턴 탐지, 군집화 등 다양한 용도로 사용되는데, 텍스트 분류에 활용될 수 있는 머신러닝 기법으로는 k-최근접 이웃 알고리즘(k-nearest neighbors algorithm), 나이브 베이즈 분류(naïve Bayes classifier), 결정 트리 학습(decision tree learning), 서포트 벡터 머신(support vector machine; SVM) 등이 있다 (Lee 등, 2018). Jeong 등 (2019)의 연구에서는 문서-용어 빈도행렬에 SVM을 적용하여 용어 가중치 함수의 성능을 비교하였다. 또한 최근 대용량 데이터들의 급증으로 딥러닝 알고리즘인 심층 신경망(deep neural network; DNN), 합성곱 신경망(convolution network), 순환 신경망(recurrent neural network) 등 다양한 딥러닝 학습 모델들이 탄생하였고 자연어처리, 컴퓨터비전, 음성인식 등의 분야에 적용되어 놀라운 결과들을 보여주고 있다 (Joo, 2018). 본 연구에서는 문서 분류에서 높은 성능을 보이며 가장 많이 이용되었던 머신러닝 알고리즘 중 하나인 SVM과 최근 각광을 받는 딥러닝 알고리즘 중 하나인 DNN을 사용하여 문서 분류를 실시하고, 이들의 성능을 정확도로 비교하고자 한다.

따라서 본 연구에서는 용어 가중치 함수로 term frequency-inverse document frequency (TF-IDF)와 term frequency-inverse gravity moment (TF-IGM)을, 그리고 문서 분류 알고리즘으로 SVM과 DNN을 적용하여 더 높은 정확도를 보이는 최적의 조합을 찾는 것이 목적이다. 이를 위해 2장에서는 문서-용어 빈도행렬에서 중요도를 수치화하기 용어 가중치 및 문서-핵심어 가중행렬을 생성하는 과정을 소개한다. 3장에서는 실제 텍스트 데이터를 활용하여 문서-용어 빈도행렬을 생성하고 용어 가중치를 적용한다. 또한 SVM과 DNN을 적용하여 이들의 성능과 용어 가중치의 성능을 비교하여 문서 분류에서 최적화된 방법을 찾아보고자 한다. 끝으로 4장의 결론에서는 본 연구 내용을 정리 및 요약한다.

2. 문서-핵심어 가중행렬 생성 과정

2.1. 문서-용어 빈도행렬

수집된 문서들의 집합인 말뭉치(corpus)는 비정형 자료이므로 사전처리 작업을 통해 정형화 자료로 변환해야 한다. 사전처리 방법으로 공란처리, 문장부호 및 특수문자 제거, 불용용어 제거 등이 있다. 사전처리가 끝난 문서의 명사들을 대상으로 문서-용어 빈도행렬을 생성할 수 있다.

전체 문서의 수는 $n = \sum_{r=1}^{G} n_r$으로 G는 개체의 수, n_r, $r = 1, \ldots, G$은 개체별 문서의 수이다. 문서-용어 빈도행렬은 n개 문서에 대한 p개 용어로 이루어진 크기가 $n \times p$이고 식 (2.1)과 같이 $\mathbf{X}$라 정의하면, $x_{(r)ij}$은 r번째 개체 내에서 i번째 문서의 j번째 용어의 빈도를 나타내며 $\mathbf{X}_r$은 $\mathbf{X}$의 r번째 부분행렬이다.

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \hline \mathbf{X}_2 \\ \hline \vdots \\ \hline \mathbf{X}_G \end{bmatrix} = (x_{(r)ij}), \quad i = 1, \ldots, n_r;\ j = 1, \ldots, p;\ r = 1, \ldots, G. \tag{2.1}$$

2.2. 문서-용어 가중행렬

2.1절에서 생성한 문서-용어 빈도행렬만으로도 문서 분류로 활용할 수 있지만, 이러한 경우 각각의 개별 문서와 개체를 구별하지 못하고 문서에 대한 중요도를 반영하지 못한다는 한계점이 있다. 따라서 문

서-용어 빈도행렬에 용어 가중치 함수를 적용하여 용어에 중요도를 부여할 필요가 있다.

문서-용어 빈도행렬에서 문서와 용어의 관계를 설명하기 위해 다음 식 (2.2)와 같이 지역적 가중치 함수(local weight function)와 전역적 가중치 함수(global weight function)로 구분하였다.

$$w_{(r)ij} = l_r(i,j) \times g(j), \tag{2.2}$$

여기서 $\mathbf{W} = (w_{(r)ij})$는 문서-용어 가중행렬로 크기가 $n \times p$인 문서-용어 빈도행렬 전체에 용어 가중치 함수를 적용한 행렬이다. $l_r(i,j)$는 i번째 문서에서 추출한 j번째 용어에 대한 지역적 가중치이고 $g(j)$는 개체 또는 문서에 대한 전역적 가중치이다.

2.2.1. Term frequency-inverse document frequency (TF-IDF) TF-IDF는 특정 용어에 대한 중요도로 용어가 출현한 횟수(term frequency)에 비례하고 그 용어가 있는 모든 문서의 수(document frequency)에 반비례한다고 정의한다 (Lee와 Bae, 2002). 2.1절에서 생성한 문서-용어 빈도행렬에 TF-IDF를 적용하기 위한 식을 식 (2.3)에 정리하였다.

$$l_r(i,j) = x_{(r)ij}, \quad g(j) = \log\left(\frac{n}{\mathrm{DF}(n,j)}\right), \tag{2.3}$$

여기서 $x_{(r)ij}$는 i번째 문서에서 추출한 j번째 용어의 발생빈도로 TF라 일컫는다. n은 전체 문서의 수이고, $\mathrm{DF}(n,j)$는 n개의 문서 중 j번째 용어가 포함된 문서의 개수를 말한다. 식 (2.3)에서 로그 값의 분모인 $\mathrm{DF}(n,j)$가 0일 경우 n개의 문서에서 j번째 용어가 한 번도 등장하지 않은 경우이므로 $x_{(r)ij}$값 또한 0이 되기 때문에 $w_{(r)ij}$값은 0으로 계산된다. 특정 용어가 모든 문서에 등장하는 흔한 용어(예를 들어, '방법, 이유, 결과'와 같은 일반 명사 용어)의 가중치를 낮추기 위해 전체 n개의 문서에 대한 j번째 용어가 포함된 문서의 수의 비율 $\mathrm{DF}(n,j)/n$에 역수 변환을 한다. 이때 역수를 취하게 되면 전체 문서의 수가 많아질수록 IDF의 값이 기하급수적으로 커지므로 로그 변환을 고려하면 최종적으로 식 (2.3)과 같이 표현할 수 있다.

2.2.2. Term frequency-inverse gravity moment (TF-IGM) Chen 등 (2016)은 개체 간의 관점으로 계산하여 개체 구분을 정확하게 하는 용어 가중치 TF-IGM을 소개하였다. 2.1절에서 생성한 문서-용어 빈도행렬에 TF-IGM을 적용하기 위한 식을 식 (2.4)에 정리하였다.

$$l_r(i,j) = x_{(r)ij}, \quad g(j) = 1 + \lambda \times \frac{d_{(1)j}}{\sum_{r=1}^{G}(d_{rj} \times R_{rj})}, \tag{2.4}$$

여기서 $x_{(r)ij}$은 i번째 문서에서 추출한 j번째 용어의 발생빈도로 2.2.1에서 설명한 TF와 동일하다. $g(j)$에서 λ는 조정 계수(adjustable coefficient)로 5.0에서 9.0의 사이의 값을 가진다. 본 연구에서는 Jeong 등 (2019)의 문서 분류 정확도에서 가장 좋은 결과를 보인 7.0을 사용하였다. d_{rj}는 j번째 용어가 적어도 한 번이라도 출현한 r번째 개체에 포함된 문서의 수이며 총 문서 수는 $d_{1j}, d_{2j}, \ldots, d_{Gj}$이다. R_{rj}는 총 문서수의 순위를 의미하며 같은 값을 가지는 경우는 평균 순위를 적용한다. 총 문서의 수를 내림차순하면 $d_{(1)j} \geq d_{(2)j} \geq \cdots \geq d_{(G)j}$로 표현할 수 있고 $d_{(1)j}$는 j번째 용어가 포함된 개체 중에 가장 많이 출현한 문서의 수를 말한다.

2.2.1절과 2.2.2절에서 언급한 TF-IDF와 TF-IGM을 Table 2.1과 같이 정리할 수 있고, 2.1절의 문서-용어 빈도행렬에 Table 2.1의 [Scheme M1]과 [Scheme M2]를 적용하여 문서-용어 가중행렬 $\mathbf{W}_1$과 $\mathbf{W}_2$을 생성하였다.

Table 2.1. Document-term weighted matrix generation scheme M1 and M2

Local weight function	Global weight function	Term weighting scheme	Weighted matrix
$x_{(r)ij}$	$\log\left(\frac{n}{\mathrm{DF}(n,j)}\right)$	[Scheme M1] TF-IDF	$\mathbf{W}_1$
	$1+\lambda\times\frac{d_{(1)j}}{\sum_{r=1}^{G}(d_{rj}\times R_{rj})}$	[Scheme M2] TF-IGM	$\mathbf{W}_2$

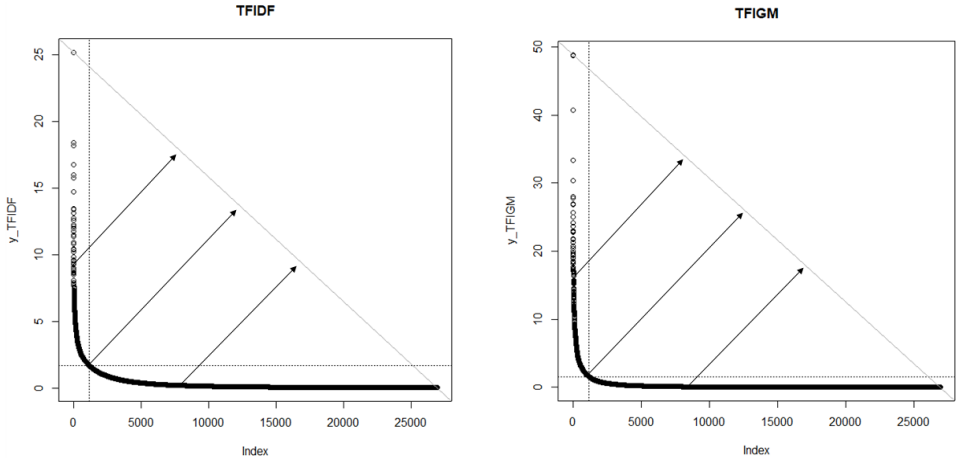

Figure 2.1. The process of finding an elbow point.

2.3. 문서-핵심어 가중행렬

2.2절에서 생성된 문서-용어 가중행렬 $\mathbf{W}$는 모든 개체와 용어를 포함하고 있기 때문에 차원의 수가 매우 크고 0의 값이 많은 행렬이다. 따라서 TF-IDF와 TF-IGM만으로 핵심어가 추출된다고 보기 어렵다. 또한 차원의 수가 클 경우 분석 결과의 질이 낮아지고 소요 시간이 길어질 수 있다.

이러한 단점을 극복하기 위해 Cho 등 (2015)이 소개한 팔꿈치 지점(elbow point)을 기준으로 핵심어를 필터링(filtering)하여 분석의 질을 높이고 소요시간을 단축시키는 방법을 사용하고자 한다. 이 방법은 팔꿈치 지점을 기준으로 하여 필터링하면 $\mathbf{W}$에서 각 용어들의 평균 가중점수를 산출한 후 점수가 급격하게 감소하는 지점을 기준으로 점수가 높은 상위 용어를 핵심어로 선정하는 것이다. 하지만 팔꿈치 지점은 해당 연구자들의 주관적인 판단에 따라 핵심어가 추출된다. 따라서 Jung 등 (2019)에서는 Satopaa 등 (2011)의 Kneedle algorithm을 활용하여 그래프의 변곡점을 찾고 이를 팔꿈치 지점으로 정의하며 핵심어 선정에 대한 객관성을 보였다.

용어 필터링 방법을 통해 핵심어를 선정하려면 먼저 문서-용어 가중행렬 $\mathbf{W}$에서 각 용어 평균 가중점수로 이루어진 평균 벡터 $\bar{\mathbf{w}}$를 계산해야 한다.

$$\bar{\mathbf{w}} = (\bar{w}_1, \bar{w}_2, \ldots, \bar{w}_p)^t = n^{-1}\mathbf{W}^t\mathbf{1}_n. \tag{2.5}$$

$\bar{\mathbf{w}}$의 원소를 큰 값에서 작은 값 순으로 내림차순하면 $\bar{w}_{(1)} \geq \bar{w}_{(2)} \geq \cdots \geq \bar{w}_{(p)}$와 같고 이를 바탕으로 그래프를 그린 후 그래프의 처음과 끝 지점인 $\bar{w}_{(1)}$과 $\bar{w}_{(p)}$를 관통하는 직선을 이어주면 Figure 2.1과 같다.

Figure 2.1은 TF-IDF와 TF-IGM 행렬에서 팔꿈치 지점을 찾는 과정을 그래프로 표현한 것이다. 여기서 화살표로 표시한 것과 같이 그래프에서 직선까지 이르는 거리가 최대가 되는 그래프 상의 지점인 변

Table 3.1. Classification of the nineteen government-funded research institutes

Research fields	Research institutes
Economic policy (C1)	대외경제정책연구원, 산업연구원, 한국개발연구원
Resources·Infrastructure (C2)	정보통신정책연구원, 한국교통연구원, 한국해양수산개발원, 한국환경정책평가연구원
Public policy (C3)	과학기술정책연구원, 통일연구원, 한국행정연구원, 한국형사정책연구원
Human resources (C4)	육아정책연구소, 한국교육개발원, 한국교육과정평가원, 한국노동연구원, 한국보건사회연구원, 한국여성정책연구원, 한국직업능력개발원, 한국청소년정책연구원

곡점을 팔꿈치 지점으로 간주할 수 있다. 팔꿈치 지점을 기준으로 평균 가중점수가 높은 상위 용어 q개를 핵심어로 선정한다. 이를 통해 행렬 $\mathbf{W}$로부터 문서-핵심어 가중행렬 $\mathbf{Y} = (y_{(r)it})$, $t = 1, \ldots, q$를 얻게 된다.

2.2절에서 언급한 Table 2.1의 문서-용어 가중행렬 $\mathbf{W}_1$와 $\mathbf{W}_2$에 용어 필터링 방법을 적용한 문서-핵심어 가중행렬을 $\mathbf{Y}_1$과 $\mathbf{Y}_2$로 정의한다.

3. 활용 사례

3.1. 연구 자료

본 연구의 자료는 Jung 등 (2019)에서의 연구 자료를 인용하였다. 경제·인문사회연구회 소속 정부출연연구기관에서 2016년 동안 발간된 정기간행물 중 텍스트 추출이 불가능한 간행물 자료를 제공하는 건축도시공간연구소, 국토연구원, 에너지경제연구원, 한국농촌경제연구원, 한국법제연구원, 조세재정연구원, KDI 국제정책연구원은 제외하였다. 총 19개 연구기관을 분석대상으로 선정하였으며, 한국법제연구원 홈페이지에 게재된 공고문(2008)을 바탕으로 연구 분야의 성격에 따라 경제정책, 자원·인프라, 공공정책, 인적자원으로 분류하여 Table 3.1에 정리하였다. 네 가지로 분류된 기관들을 C1−C4로 정의하여 각 연구기관의 성격에 알맞게 분류가 되었는지 2.3절에서 생성된 문서-핵심어 가중행렬에 심층 신경망을 활용하여 확인하고자 한다.

본 연구에서는 파이썬(Python) 프로그램을 이용하여 각 기관들의 정기간행물들의 PDF 파일들을 크롤링(crawling)하고 TXT 파일로 저장하여 활용하였고, 파이썬에 가장 많이 사용되는 한국어 자연어 처리 패키지 KoNLPy(Korean NLP in Python)를 사용하였다. KoNLPy는 한나눔(Hannanum), 꼬꼬마(Kkma), Komoran, Mecab, Twitter와 같은 5개 형태소 분석기를 지원하고 있다. 이중 2010년에 만들어진 꼬꼬마 형태소 분석기는 띄어쓰기 오류에 덜 민감한 한국어 형태소 분석기로 알려져 있으며 (Jung, 2017), 1998–2007년 정부 지원 과제(21세기 세종 계획)에서 구축된 세종 말뭉치를 기반으로 하는 검증된 형태소 분석기이다 (Lee 등, 2010). 따라서 본 연구에서는 꼬꼬마 형태소 분석기를 사용하여 말뭉치에서 한국어 형태소 품사 중 일반 명사와 고유 명사만을 추출하였으며, 이 과정에서 미등록어, 오·탈자, 띄어쓰기, 영문 및 기호 등을 제거하였다. 최종적으로 Table 3.2의 기관별 대표 정기간행물 문서 343개와 문서에서 추출한 용어 26,915개로 구성된 문서-용어 빈도행렬을 분석에 사용하였다.

3.2. 심층 신경망을 이용한 문서 분류

심층 신경망은 입력층(input layer)과 출력층(output layer) 사이에 여러 개의 은닉층(hidden layer)들로 이루어진 인공 신경망(artificial neural network)이다 (Schmidhuber, 2015). 하나의 은닉층은 다수

Table 3.2. PDF files and terms of periodical publication by institute

	Korea institute	Periodical publication	Number of files	Number of terms
1	과학기술정책연구원	과학기술정책	12	8,840
2	대외경제정책연구원	정책연구브리핑	23	4,252
3	산업연구원	동향브리프	12	3,354
4	육아정책연구소	육아정책포럼	4	3,485
5	정보통신정책연구원	프리미엄	11	3,397
6	통일연구원	KINU통일	4	5,208
7	한국개발연구원	경제동향	12	1,073
8	한국교육개발원	교육정책포럼	12	6,645
9	한국교육과정평가원	교육광장	4	6,661
10	한국교통연구원	월간교통	12	10,805
11	한국노동연구원	노동리뷰	12	8,712
12	한국보건사회연구원	보건복지포럼	132	9,899
13	한국여성정책연구원	젠더리뷰	4	4,664
14	한국직업능력개발원	HRD	6	7,039
15	한국청소년정책연구원	한국청소년	39	6,262
16	한국해양수산개발원	해양정책연구	5	4,879
17	한국행정연구원	행정포커스	6	8,425
18	한국형사정책연구원	형사정책연구	3	7,430
19	한국환경정책평가연구원	환경정책	30	6,790
	Total		343	117,820
	Number of terms after delete duplication and stopwords			26,915

의 노드들로 구성되어 있으며, 각각의 노드들은 활성화 함수(activation function)를 통해 출력을 내보낸다 (Cho 등, 2018).

일반적으로 심층 신경망에서 활성화 함수로 사용하는 sigmoid 함수는 네트워크가 깊어지면 깊어질수록 기울기 소실(gradient vanishing) 즉, 기울기가 0으로 수렴하여 신경망이 학습되지 않는 문제가 발생할 수 있다. 이러한 문제점을 해결하기 위한 다양한 활성화 함수들 중 ReLU 함수는 0보다 작을 때는 모든 값을 0으로 처리하고, 0보다 큰 값은 x를 그대로 이용하여 여러 은닉층을 거치며 곱해지더라도 0이 되지 않아 끝까지 존재할 수 있는 함수로 식 (3.1)과 같이 표현할 수 있다.

$$f(x) = \begin{cases} x, & (x > 0), \\ 0, & (x \leq 0). \end{cases} \tag{3.1}$$

딥 러닝 모델에서 3 클래스 이상의 분류를 목적으로 하는 활성화 함수로 softmax 함수를 사용한다. 신경망 모델을 통해 i번째 출력값 벡터를 $\hat{\mathbf{y}}_i$라 하고 벡터의 원소 번호를 j라고 했을 때, Jeon (2018)에 의해 식 (3.2)와 같이 나타낼 수 있다.

$$\hat{\mathbf{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{ic})^T, \quad i = 1, \ldots, n; \ j = 1, \ldots, c. \tag{3.2}$$

따라서 softmax 함수는 다음의 식 (3.3)과 같이 정의된다.

$$f(x_{ij}) = \frac{e^{\hat{y}_{ij}}}{\sum_{j=1}^{c} e^{\hat{y}_{ij}}}. \tag{3.3}$$

softmax 함수를 사용하면 $\hat{\mathbf{y}}_i$의 각 원소값 $\hat{y}_{ij}$이 해당 범주가 될 확률 값으로 표현된다. 이 결과를 출력값이 0과 1로만 이루어진 형태로 바꾸어 주는 기법인 원-핫 인코딩(one-hot encoding)과 연결하면 해당하는 출력값은 1로 나머지는 모두 0인 형태로 전환시킬 수 있다.

식 (3.4)는 softmax 함수에서 예측값과 실제값의 차이를 최소화하기 위한 비용함수(cost function)로 다중분류문제에서 사용되는 교차 엔트로피(cross entropy)이다. i는 관측치 번호, j는 관측치 벡터의 원소 번호, $\mathbf{y}_i$는 종속변수 i번째 관측치 벡터, $\hat{\mathbf{y}}_i$는 i번째 출력값 벡터라고 했을 때, 식 (3.4)와 식 (3.5)와 같이 정의된다. 이때 교차 엔트로피 함수는 식 (3.6)과 같이 정의된다.

$$\mathbf{y}_i = (y_{i1}, \ldots, y_{ic})^T, \quad i = 1, \ldots, n;\ j = 1, \ldots, c. \tag{3.4}$$

$$\hat{\mathbf{y}}_i = (\hat{y}_{i1}, \ldots, \hat{y}_{ic})^T, \quad i = 1, \ldots, n;\ j = 1, \ldots, c. \tag{3.5}$$

$$\text{Cross entropy}(y_{ij}, \hat{y}_{ij}) = \sum_{x=1}^{n} \sum_{j=1}^{c} [y_{ij} \log(\hat{y}_{ij})]. \tag{3.6}$$

심층 신경망에서 입력층에서 은닉층, 은닉층에서 출력층까지의 과정을 순전파(forward propagation)라 한다. 순전파를 통해 예측값을 얻어냈으면 예측값과 실제값 사이의 오차를 구하고 경사하강법(gradient descent)을 이용해 가중치를 업데이트하면서 모델을 최적화 시켜야하는데, 이러한 과정을 오차 역전파(back propagation)라 한다. 즉, 심층 신경망은 오차 역전파를 통해 경사를 구하고, 식 (3.7)에 따라 가중치를 업데이트하는 과정을 통해 오차값이 0에 가까워지게 한다.

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \alpha \frac{\partial \boldsymbol{\delta}}{\partial \mathbf{w}_{(t)}}, \tag{3.7}$$

여기서 $\mathbf{w}(t)$는 t단계에서의 신경망 가중치이고, $\mathbf{w}(t+1)$은 업데이트 후의 가중치이다. 오차값 $\boldsymbol{\delta}$는 $1/2(\hat{\mathbf{y}} - \mathbf{y})^2$로 정의되고 $\hat{\mathbf{y}} = (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_c)^T$는 출력값, $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_c)^T$는 실제값이며, 오차값이 최소화되도록 출력층에서 입력층 방향으로 역전파하면서 가중치를 수정하게 된다 (Choi, 2017). α는 학습률(learning rate)로 매개변수를 얼마나 업데이트할지 정한다. 위와 같이 심층 신경망은 순전파에서 역전파, 역전파에서 가중치 업데이트 과정을 계속 반복해 나가면서 오차값이 0에 가까워지게 하여 모델을 최적화시켜 실제값과 가장 가까운 예측값을 얻어낸다.

3.3. 분류 분석 결과

본 연구는 R 프로그램에서 Tensoflow기반 keras 라이브러리를 활용하여 심층 신경망을 구현한다. 다음으로 딥 러닝에서 데이터를 정규화하면 학습속도가 빨라지는 장점이 있으므로 변수의 범위를 정규화하는 과정을 거쳤다. 심층 신경망은 데이터 훈련을 통해 매개변수를 업데이트하는 훈련 과정과 업데이트된 매개변수를 통해 성능을 확인하는 실험 과정으로 나눌 수 있다. 본 연구에서는 총 문서수 343개를 7:3의 비율로 훈련표본과 실험표본으로 데이터를 나누었으며, 각 문서에 나타난 용어 26,915개의 특징을 바탕으로 하여 문서 343개가 경제정책, 자원·인프라, 공공정책, 인적자원의 네 가지 연구 분야로 잘 분류되는지를 확인하고자 하였다. 심층 신경망 학습에 대한 매개변수 옵션으로는 먼저 3.2절에서 설명한 활성화 함수를 이용하여 은닉층에서 ReLU 함수를, 출력층에서 softmax 함수를 사용하였다. 비용함수는 다중분류에 적절한 오차 함수인 categorical crossentropy를 사용하고, 최적화 함수로 adam을 사용하였다. 그리고 전체 샘플이 200회 반복될 때까지 실험을 진행하되 한번에 입력되는 값은 5로 하였으며, 과적합을 피하기 위하여 validation split을 0.2로 설정하였다.

심층 신경망 모형을 적합할 때 연구자가 은닉층과 은닉노드의 수 등을 결정해야 하는데, 무조건 은닉노드의 수가 많을수록 성능이 좋아지는 것이 아니라 과적합의 우려가 있다. 또한 Bengio 등 (2013)에 따르면 심층 신경망 구조는 하위 계층에서 상위 계층으로 갈수록 차원이 축소되어야 하며, LeCun 등 (1989)에 따르면 은닉층이 깊으면 깊을수록 정교한 모델 생성이 가능하다고 하였다. 따라서 본 연구에서는 최적의 모수를 찾기 위해 총 문서수 343개를 기준으로 1/4, 1/2, 1, 2, 4배씩 첫 번째 은닉층에 입력하고 절반씩 감소시켜가며 분석을 수행하였다.

Table 3.3. Model accuracy according to the number of hidden layers and hidden nodes

Model	The number of hidden layers and hidden nodes	Accuracy	
		M1 (TF-IDF)	M2 (TF-IGM)
1	86-43-21-10	0.793	0.810
2	172-86-43-21-10	0.801	0.801
3	343-172-86-43-21-10	0.750	0.818
4	686-343-172-86-43-21-10	0.818	0.827
5	1372-686-343-172-86-43-21-10	0.827	0.853

TF-IDF = term frequency-inverse document frequency; TF-IGM = term frequency-inverse gravity moment.

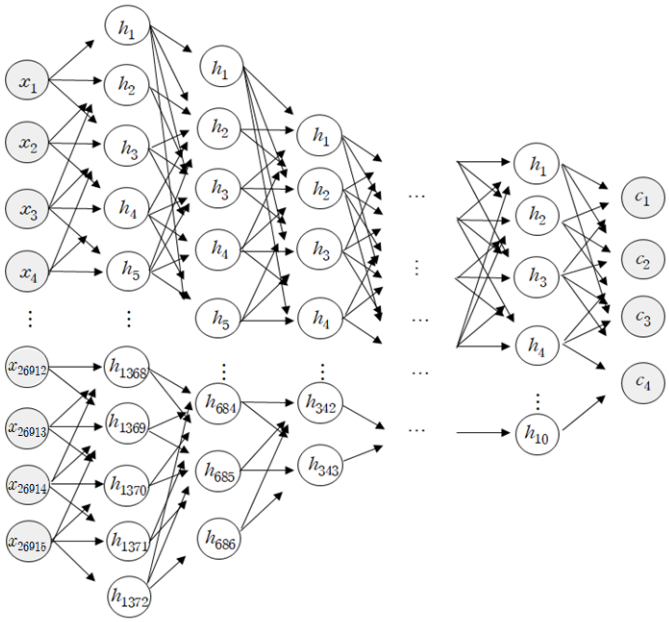

Figure 3.1. Deep neural network structure of Model 5.

본 연구에서는 분류 모델의 성능을 평가하기 위한 지표로 가장 많이 이용되는 정확도(accuracy)를 이용하였으며, 이는 다음의 식 (3.8)로 계산할 수 있다.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{3.8}$$

여기서 true positive (TP)는 분류 모델에 의해 예측된 범주와 실제 범주를 정확하게 분류한 빈도이고, false positive (FP)는 실제 범주가 거짓인데 참으로 예측한 경우의 빈도, false negative (FN)은 실제 범주가 참인데 거짓으로 예측한 경우의 빈도, 마지막으로 true negative (TN)은 실제 범주가 아닌 범주로 정확하게 분류된 빈도이다. Table 3.3은 은닉층과 은닉노드 변화에 따라 방법 M1과 M2의 모델 정확도를 나타낸 표이다. 그 결과 은닉층 당 노드의 수가 가장 많은 모델 5에서 M1과 M2의 정확도가 약 83%와 85%로 가장 높게 나타나 문서를 가장 잘 분류 한 것으로 볼 수 있다.

Figure 3.1은 Table 3.3에서 정확도가 가장 높았던 모델 5에 대한 심층 신경망 구조이다. 입력층에는 총 26,915개의 용어가 입력되었다. 은닉층의 수는 총 8개로, 첫 번째 은닉노드 수 1,372개에서 절반

Table 3.4. SVM and DNN classification results for M1 (TF-IDF)

Research fields	SVM	DNN
Economic policy (C1)	0.940	1.000
Resources · Infrastructure (C2)	0.822	0.853
Public policy (C3)	0.862	0.991
Human resources (C4)	0.947	0.896

SVM = support vector machine; DNN = deep neural network; TF-IDF = term frequency-inverse document frequency.

Table 3.5. SVM and DNN classification results for M2 (TF-IGM)

Research fields	SVM	DNN
Economic policy (C1)	0.950	0.974
Resources · Infrastructure (C2)	0.892	0.862
Public policy (C3)	0.836	0.974
Human resources (C4)	0.945	0.913

SVM = support vector machine; DNN = deep neural network; TF-IGM = term frequency-inverse gravity moment.

씩 감소시킨 구조이다. 출력층은 Table 3.1에 따라 네 가지 세부 분야로 나눈 경제정책(C1), 자원·인프라(C2), 공공정책(C3), 인적자원(C4)을 뜻한다.

동일한 연구 자료를 이용한 Jung 등 (2019)의 연구는 근접성 데이터(proximity data)를 통해 문서들의 특징을 추출한 것을 바탕으로 K-평균 군집분석(k-means cluster analysis)을 실시하여 문서들을 군집화시킨 것이다. Jung 등 (2019)의 K-평균 군집분석과 본 연구의 심층 신경망 모델 5의 정확도를 비교한 결과, K-평균 군집분석에서는 약 68%, 심층 신경망에서는 약 85%의 정확도를 보임을 확인하였다.

Table 3.4와 Table 3.5는 M1과 M2에 대해 서포트 벡터 머신과 본 연구의 모델 5에 대한 심층 신경망의 개별 범주에 대한 문서 분류 분석 결과이다. 그 결과 Table 3.4의 인적자원과 Table 3.5의 자원·인프라를 제외한 모든 결과에서 심층 신경망의 문서 분류 정확도가 더 높음을 확인하였다. 따라서 가중치를 부여한 문서-용어 빈도행렬의 문서 분류를 하는 데 있어서 서포트 벡터 머신보다 심층 신경망이 더 적합한 방법이라 판단된다.

4. 결론

기존의 연구에서는 문서에서 추출한 특징을 통해 문서 분류를 판단하였다. 그러나 특징들을 추출하는 경우 많은 시간을 투자해야하고 제대로 추출되지 않을 경우에는 사용자가 직접 처리해야하므로 효율이 떨어진다 (Jeon, 2018). 따라서 본 연구에서는 이러한 단점을 보완하기 위해 심층 신경망으로 컴퓨터가 직접 판단하고 찾아 문서를 분류 하는 것을 목표로 하여 설계하였다.

비정형 데이터인 문서에서 추출된 명사들의 빈도를 센 것을 바탕으로 정형 데이터인 문서-용어 빈도행렬을 생성하고, 개체들의 정보가 존재하는 문서-용어 빈도행렬에서 용어 가중치 함수 TF-IDF와 TF-IGM을 적용하여 문서에 대한 용어의 중요도를 반영하였다. 또한 가중치가 적용된 문서-용어 빈도행렬의 문서 분류 정확도 향상을 위해 핵심어를 추출하여 최종 문서 분류 행렬을 생성하였다. 이러한 과정을 거친 문서-핵심어 가중행렬에 최근 각광받고 있는 딥 러닝 기법 중 하나인 심층 신경망을 활용하여 개체들이 각각의 연구 분야 성격에 알맞게 분류되는지 확인하였다.

심층 신경망의 은닉층과 은닉노드에 변화를 주며 M1과 M2 방법 각각의 정확도를 산출하였다. 그 결과 제일 깊은 은닉층을 가진 심층 신경망 모델이 가장 높은 정확도를 보였고, 모든 결과에서 M2의 정확도

가 M1에 비해 높거나 같은 값을 보였다. 또한 제안된 심층 신경망 모델을 문서 분류에서 가장 많이 이용되었던 서포트 벡터 머신과 비교했을 때 대부분의 심층 신경망 결과에서 더 높은 정확도를 보였다. 따라서 개체 정보가 존재하는 문서를 분류하는데 있어서 TF-IGM 용어가중치를 이용하며 심층 신경망을 적용하는 방법을 제안한다.

본 연구에서 제안한 방법을 텍스트 마이닝 단계와 심층 신경망 모델 구현에 활용한다면 용어 중요도를 확인하고 특징 추출없이 문서를 분류할 수 있을 것이다. 그러나 하나의 연구자료를 분석에 활용하였기 때문에 결과를 일반화하기엔 무리가 있다. 따라서 다양한 문서자료를 이용해 다양한 가중치와 심층 신경망 매개변수 변화 등을 통해 더 나은 문서 분류 기법을 찾는 연구가 필요하다고 생각된다. 또한 비정형 데이터를 정형 데이터로 변환하기 위해 최근 구글에서 개발한 Word2Vec (2013)나 스탠포드에서 개발한 GolVe (2014) 등을 활용하면 다양한 연구 결과를 얻을 수 있을 것이라 생각한다.

References

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, 1798–1828.

Chen, K., Zhang, Z., Long, J., and Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification, *Expert System with Applications*, **66**, 245–260.

Cho, H. Y., Kim, Y. H., and Im, H. H. (2018). Forecast of wind-shear alert using deep neural networks, *Asia-Pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, **8**, 749–757.

Cho, S. G., Cho, J. H., and Kim, S. B. (2015). Discovering meaningful trends in the inaugural addresses of United States Presidents Via text mining, *Journal of Korean Institute of Industrial Engineers*, **41**, 453–460.

Choi, M. J. (2017). *Forecasting the number of tourists in Jeju Island using deep learning algorithm* (Master thesis), Hanyang University.

Jeon, E. K. (2018). *Implementation of arrhythmia classification system using deep neural network* (Master thesis), Soonchunhyang University.

Jeong, H. Y., Shin, S. M., and Choi, Y. S. (2019). Comparison of term weighting schemes for document classification, *The Korean Journal of Applied Statistics*, **32**, 265–276.

Joo, W. K. (2018). *Automatic classification method for atypical texts that include structure information using deep learning* (Doctoral thesis), Chungnam National University.

Jung, M. J. (2017). *A study on clustering methods for proximity data in text mining* (Master thesis), Pusan National University.

Jung, M. J., Shin, S. M., and Choi, Y. S. (2019). Creation and clustering of proximity data for text data analysis, *The Korean Journal of Applied Statistics*, **32**, 451–462.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition, *Neural Computation*, **1**, 541–551.

Lee, D. J., Yeon, J. H., Hwang, I. B., and Lee, S. G. (2010). KKMA: a tool for utilizing Sejong corpus based on relational database, *Communications of the Korean Institute of Information Scientists and Engineers*, **16**, 1046–1050.

Lee, G. G., Ha, H. S., Hong, H. G., and Kim, H. B. (2018). Exploratory research on automating the analysis of scientific argumentation using machine learning, *Journal of the Korean Association for Science Education*, **38**, 219–234.

Lee, M. R. and Bae, H. K. (2002). Design of keyword extraction system using TFIDF, *The Korean Society for Cognitive Science*, **13**, 1–11.

Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. (2011). Finding a "kneedle" in a haystack: detecting knee points in system behavior, *Distributed Computing Systems Workshops (ICDCSW) 2011 31st International Conference on, IEEE*, 166–171.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview, *Neural Networks*, **61**, 85–117.

텍스트 마이닝에서 심층 신경망을 이용한 문서 분류

이보희[a] · 이수진[b] · 최용석[b,1]

[a]신라대학교 광고홍보학과, [b]부산대학교 통계학과

(2020년 6월 3일 접수, 2020년 7월 13일 수정, 2020년 8월 21일 채택)

요 약

문서-용어 빈도행렬은 그룹정보가 존재하는 문서들의 용어를 추출한 것으로 일반적인 텍스트 마이닝에서의 자료이다. 본 연구에서는 연구 분야 성격에 따른 문서 분류를 위해 문서-용어 빈도행렬을 생성하고, 전통적인 용어 가중치 함수인 TF-IDF와 최근 잘 알려진 용어 가중치 함수인 TF-IGM을 적용하였다. 또 용어 가중치가 적용된 문서-용어 가중행렬에 문서분류 정확도 향상을 위해 핵심어를 추출하여 문서-핵심어 가중행렬을 생성하였다. 핵심어가 추출된 행렬을 바탕으로, 심층 신경망을 이용해 문서를 분류하였다. 심층 신경망에서 최적의 모델을 찾기 위해 매개변수인 은닉층과 은닉노드수를 변화해가며 문서 분류 정확도를 확인하였다. 그 결과 8개의 은닉층을 가진 심층 신경망 모델이 가장 높은 정확도를 보였으며 매개변수 변화에 따른 모든 TF-IGM 문서 분류 정확도가 TF-IDF 문서 분류 정확도보다 높은 것을 확인하였다. 또한 개별 범주에 대한 문서 분류 분석 결과를 서포트 벡터 머신과 비교했을 때 심층 신경망이 대부분의 결과에서 더 좋은 정확도를 보임을 확인하였다.

주요용어: 문서 분류, 심층 신경망, 용어 가중치, 텍스트 마이닝, 핵심어 추출

이 논문은 2020년도 4단계 BK21 사업에 의하여 지원되었음.

[1]교신저자: (46241) 부산광역시 금정구 부산대학로 63번길 2, 부산대학교 통계학과.
E-mail: yschoi@pusan.ac.kr