Doc2Vec을 활용한 CNN기반 한국어 신문기사 분류에 관한 연구

김도우○, 구명완
서강대학교, 정보통신대학원
iigreen@idtc.co.kr, mwkoo@sogang.ac.kr

A Study on Categorization of Korean News Article based on CNN using Doc2Vec

Do-Woo Kim○ ,Myoung-Wan Koo
Sogang University, Graduate School of Information & Technology

요 약

본 논문에서는 word2vec과 doc2vec을 함께 CNN에 적용한 문서 분류 방안을 제안한다. 먼저 어절, 형태소, WPM(Word Piece Model)을 각각 사용하여 생성한 토큰(token)으로 doc2vec을 활용하여 문서를 vector로 표현한 후, 초보적인 문서 분류에 적용한 결과 WPM이 분류율 79.5%가 되어 3가지 방법 중 최고 성능을 보였다. 다음으로 CNN의 입력자질로써 WPM을 이용하여 생성한 토큰을 활용한 word2vec을 범주 10개의 문서 분류에 사용한 실험과 doc2vec을 함께 사용한 실험을 수행하였다. 실험 결과 word2vec만을 활용하였을 때 86.89%의 분류율을 얻었고, doc2vec을 함께 적용한 결과 89.51%의 분류율을 얻었다. 따라서 제안한 모델을 통해서 분류율이 2.62% 향상됨을 확인하였다.

주제어: Convolutional Neural Network(CNN), word2vec, doc2vec, Word Piece Model(WPM)

1. 서론

정보기술과 통신기술이 발전함에 따라 뉴스 기사 및 멀티미디어 데이터 같은 뉴스보도에 관련된 데이터를 체계적으로 저장 및 관리가 가능하게 되었고 사용자들은 인터넷, 핸드폰, 휴대용단말기 등을 이용하여 언제 어디서나 신속한 뉴스기사 서비스를 받을 수 있게 되었다. 이에 따라 국내외 언론사들은 인터넷 기사 서비스를 위한 별도의 체계를 구성·운영하여 정규 뉴스보도 외에도 사용자들이 시간과 장소에 구애받지 않고 신속하게 뉴스 서비스를 이용할 수 있도록 지원하고 있다. 대부분의 언론사에서는 기사를 인터넷에 게시하기 전에 분류 전문가를 통해 기사를 분류하고 검증하는 단계를 거친다. 그러나 이러한 수작업 처리 방법은 정보시스템의 급속한 발달로 인해 처리해야 할 정보와 문서의 양이 점점 방대해지고 복잡해지는 현대 시대에 빠르게 전달해야 하는 뉴스의 속도를 저하시킬 뿐만 아니라 인력 자원의 투입으로 인한 많은 비용을 소비하고 있다. 따라서 문서 분류의 자동화에 대한 필요성은 더욱 증대되고 있다[1].

문서 분류의 자동화를 위하여 기존에는 단순히 문서에 나타나는 단어의 빈도를 이용하여 분류 범주를 지정하는 통계적인 분류방법을 이용하거나[1], 분류에 필요한 주요 단어들을 추출하고 추출된 단어들을 기반으로 K-NN, 의사결정 트리, 베이지언 네트워크, 인공신경망 등의 데이터 마이닝 알고리즘을 이용한 연구가 진행되었다[2]. 최근에는 딥러닝 알고리즘인 컨볼루션 신경망(Convolutional neural network, CNN)이 자연어 처리에 효과적이라는 것이 알려지면서, 문서에 포함된 단어들을 각각 vector로 표현하는 방법인 word2vec[5]과 CNN을 이용한 문서 분류 방법[3]이 제안되었고, 실제로 놀라운 결과를 보여주었다[3]. 그러나, CNN을 이용한 문서 분류 방법에서 문서 자체를 vector로 표현하는 방법인 doc2vec[6]의 활용은 고려되지 않았다.

본 논문에서는, word2vec과 CNN을 이용한 기존의 분류 방법[3]을 수정하여, 한국어 신문기사로부터 doc2vec을 활용하여 문서의 vector 표현을 생성하고 수정된 CNN에 word2vec을 활용한 단어의 vector 표현과 함께 적용함으로써 이를 바탕으로 기사를 적합한 범주로 자동 분류하는 방안을 제안하며, word2vec만을 활용한 기존 CNN 기법에 비해 문서의 vector 표현을 함께 사용했을 때 분류율이 향상됨을 검증하는 것이 목적이다. 분류율은 검증에 사용한 전체 문서 중에서 정확하게 분류된 문서가 차지하는 비율을 말하며 다음 식으로 나타낸다[1].

$$\text{분류율} = \frac{\text{정확하게 분류된 문서수}}{\text{검증에 사용한 전체 문서수}} \times 100\,[1]$$

본 논문의 구성은 다음과 같다. 2장과 3장에서는 WPM과 word2vec을 활용한 기존 CNN 모델 및 제안하는 모델에 관하여 소개하고, 4장에서는 2장과 3장에서 소개한 모델 별로 실험하고 결과를 분석한다. 마지막으로 5장에서 본 연구의 결론을 제시한다.

2. 관련연구

2.1 Word Piece Model(WPM)

WPM[8]은 2012년 구글에서 제안한 구글 일본과 구글 한국에 적용된 성공적인 음성 검색 시스템 구축을 위한 방법이다. 기존의 자연어 처리에서는 형태소 분석, 통사 분석, 의미 분석, 화용 분석의 4단계로 진행되나, WPM은 음절을 기반으로 유닛을 코드화 시킨 후 사용빈도수에 따라 조합을 하여 새로운 유닛을 생성한다. 즉, WPM은 음절을 기반으로 통계적 기법을 활용하여 사용빈도수가 높은 음절을 합쳐서 사전을 만드는 방법이다. WPM의 장점은 언어에 독립적이며, 통계적인 방식을 사용하므로 특정 도메인 또는 아직 의미를 파악하지 못한 상태에서

도 적용할 수 있다[9].

2.2 Word2Vec을 활용한 CNN 모델

원래 컴퓨터 비전을 위해 고안된 CNN 모델이 자연어 처리에 효과적이라는 것이 알려지고, semantic parsing, search query retrieval, sentence modeling, 그리고 다른 전통적인 자연어 처리에 있어서 놀라운 결과를 이루었다[3]. 최근에, Yoon Kim은 그의 논문[3]에서 문장 단위(sentence-level) 분류작업을 위해 미리 훈련된(pre-trained) 단어의 vector 표현을 활용한 CNN을 제안하였고, 7개 데이터셋 중 sentence analysis와 question classification을 포함한 4개 데이터셋에서 놀라운 향상을 이루었다[3]. 그림 1은 Yoon Kim이 제안한 word2vec을 활용한 CNN 모델을 나타낸다.

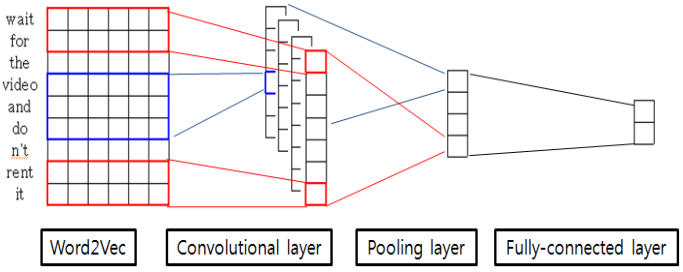

그림 1. word2vec을 활용한 CNN 모델[3]

그림 1의 CNN 모델은 다양한 크기(filter region size)의 feature map들을 여러 개 적용한 하나의 convolutional layer와 1-max pooling layer로 이루어져 있으며, fully-connected layer도 hidden layer 없이 softmax output layer만을 가지는 간단한 구조로 이루어져 있다[3]. 본 논문에서는 doc2vec을 활용함으로써 문서의 범주 분류율이 향상됨을 비교 검증하는데 그 목적이 있으므로, 입력자질로써 non-static channel의 사용은 실험대상에서 제외하였다.

3. 제안 모델

2장에서 기술한 모델은 단순하고, 빠른 훈련·예측 시간의 장점이 있기 때문에, SVM 및 Logistic Regression과 같이 잘 구축된 baseline model의 대체가 가능하다[4]. 그러나, 문서 자체를 vector로 표현하는 방법인 doc2vec의 활용은 고려되지 않았다. 이에 본 논문에서는 그림 2와 같이 doc2vec을 함께 활용한 CNN 모델을 제안한다.

제안하는 모델에서는, doc2vec을 활용한 문서의 vector 표현이 문서 분류 수행시마다 한번만 사용되고, 분류시 doc2vec vector의 값들이 최대한 활용될 수 있도록, 그림 2와 같이, word2vec vector들에 대한 convolutional layer와 pooling layer의 출력 vector와 doc2vec vector를 연결(concatenation)한 vector를 fully-connected layer의 입력자질로써 활용하도록 구성하였다.

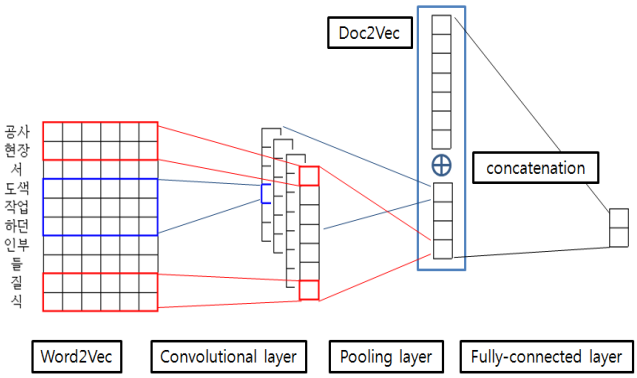

그림 2. word2vec과 doc2vec을 함께 활용한 CNN 모델

4. 실험 및 결과

2013년 5월에서 7월 사이에 77개 신문사에서 작성된 한국어 신문기사 528,735개중 범주가 분류되어 있는 146,691개를 대상으로, 범주별 기사수가 1,000개 이상인 10개 범주를 선정한 후, 범주별로 1,000개의 기사를 추출하여 데이터셋을 구성하고 실험을 수행하였다. 선정된 범주와 label은 표 1과 같다.

표 1. 실험 데이터셋의 범주 및 label

범주	label
사회일반	1
경제일반	2
정치일반	3
행정 · 자치	4
방송 · 연예	5
교육	6
사람	7
스포츠	8
문화	9
사설 · 오피니언 · 칼럼	10

NewsML[7]형식으로 작성된 신문기사를 XML 파싱(parsing)한 후 제목, 본문, 범주 정보를 추출하여 범주별로 파일에 저장하였다. 파일에 저장시 HTML tag를 제거하고, 각 기사를 하나의 라인으로 바꾸어 저장하였다.

단어와 문서의 vector 표현 생성을 위해 word2vec과 doc2vec 알고리즘을 사용하기 위해서는, 먼저 문서를 token으로 나누어야 한다. 문서 분류에 더 나은 성능을 보이는 tokenizing 방법을 찾기 위해, 기사를 어절 단위, 형태소 분석, WPM 적용의 3가지 방법으로 tokenizing한 후, 해당 token들로 doc2vec 알고리즘을 이용하여 생성한 문서의 vector 표현을 적용하여 Logistic Regression(LR) 분류기로 분류율을 실험해보았다. tokenizing 방법을 찾는 실험시, 데이터셋은 9:1로 나누어 90%를 훈련에 사용하고, 10%로 테스트를 수행하였다. 수행 결과는 표 2과 같이 WPM을 적용한 결과가 분

류율이 79.5%로 가장 높았기 때문에, 이어지는 실험에서는 WPM을 적용하여 생성한 token들을 doc2vec과 word2vec의 입력으로 사용하였다. 리소스 사용 및 성능을 고려하여, 각각 token별로 vector는 300차원으로 생성하였다.

표 2. tokenizing 방법별 분류율 비교

구분	어절 단위	형태소 분석기 적용 (komoran - 명사만 추출)	WPM 적용
전체 token 수	2,418,996	1,726,752	4,160,186
unique token 수	407,301	51,340	73,299
하나의 기사내 최대 token 수	2,055	1,665	3,539
LR 분류기를 적용한 분류율	74.8%	77.6%	**79.5%**

WPM을 적용하여 문서들을 tokenizing한 결과 token 개수별 문서수의 비율은 그림 3과 같다. 그림 3을 통해서, token 개수 600개 이하 문서들이 전체 데이터셋의 80%, token 개수 700개 이하인 문서들이 전체 데이터셋의 약 90%를 차지함을 알 수 있다.

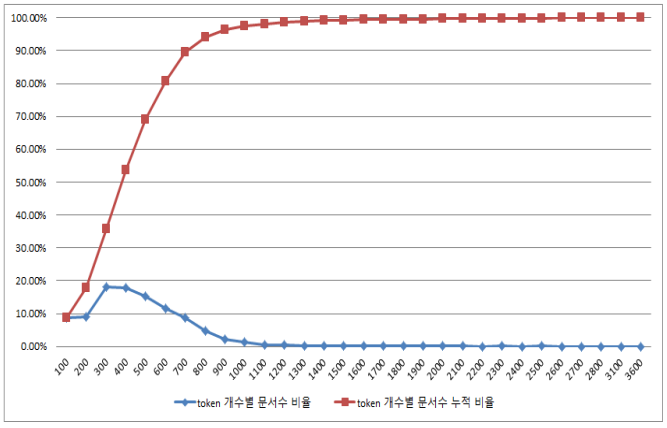

그림 3. token 개수별 문서수 비율 및 문서수 누적 비율

신문기사는 가변길이로 token의 개수가 고정되어 있지 않으나, CNN은 입력 자질로 고정길이를 요구한다. 따라서, word2vec을 활용한 CNN 모델과 doc2vec을 함께 활용한 CNN 모델의 성능 비교 실험시, CNN의 입력자질로 입력되는 기사의 token 개수를 고정하여 수행하였다. token의 개수는 100개부터 1,000개까지 100개 단위로 증가시키며, zero-padding을 적용하였다. token 개수 100은 문서내 token의 개수가 1개부터 100개 사이인 경우를 의미한다.

두 모델의 설정은 Yoon Kim[3]이 제안한 모델의 기본 설정(baseline configuration)을 기반으로 하여 적용하였고, 표 3에 기술하였다.

표 3. baseline configuration[4]

설정명	설정값
filter region size	(3,4,5)
feature maps	100
activation function	ReLU
pooling	1-max pooling
dropout rate	0.5

실험 데이터의 개수가 적은 것을 고려하여 10-fold cross validation을 이용하여 분류율을 측정하였다.

우선, doc2vec만을 활용하였을 때보다 doc2vec과 word2vec을 함께 활용하는 모델의 성능이 더 향상됨을 확인하기 위하여, 그림 4와 같이 제안 모델에서 word2vec 적용 부분을 제외하고, fully-connected layer의 입력자질로 doc2vec만을 이용하여 실험한 결과 81.81%의 분류율을 얻었다.

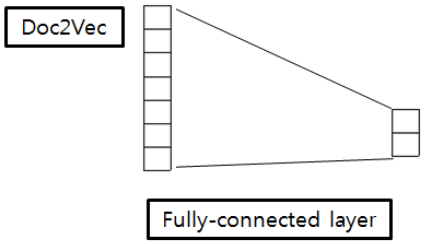

그림 4. Fully-connected layer의 입력자질로 doc2vec만을 활용한 모델

모델별 실험결과는 표 4, 그림 5와 같다.

표 4. word2vec과 doc2vec을 함께 사용한 모델과 word2vec만 사용한 모델의 token 개수별 분류율 비교

token 개수	Doc2Vec을 함께 적용(1)	Word2Vec만 적용(2)	차이 (1)-(2)
100	0.8678	0.8022	0.0555
200	0.8606	0.8206	0.0399
300	0.8756	0.8406	0.0349
400	0.8890	0.8570	0.0319
500	0.8902	0.8656	0.0246
600	**0.8951**	0.8675	0.0276
700	0.8900	**0.8689**	0.0211
800	0.8877	0.8675	0.0202
900	0.8889	0.8680	0.0209
1,000	0.8890	0.8660	0.0230

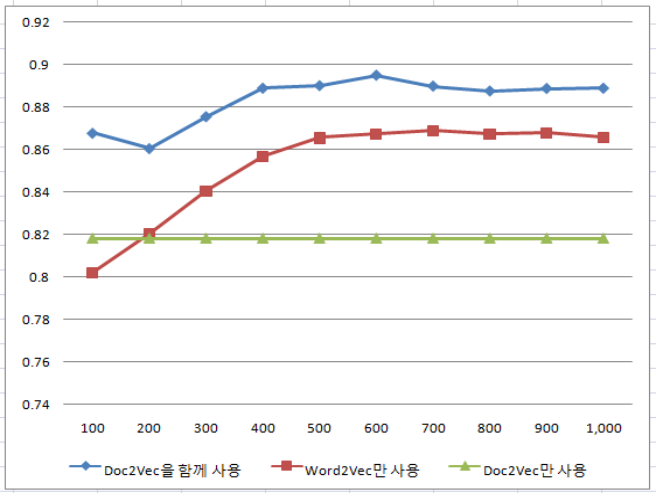

그림 5. word2vec과 doc2vec을 함께 사용한 모델과 word2vec만 사용한 모델, 그리고 doc2vec만 사용한 모델의 token 개수별 분류율 비교 그래프

doc2vec은 문서 전체를 하나의 vector로 표현하기 때문에, doc2vec만 사용한 모델의 분류율은 token 개수와 상관없이 81.81%로 일정하다.

word2vec만 사용한 CNN은 token 개수 700에서 최고 성능을 보이고 이후 유지되는 경향을 보이며, doc2vec을 함께 사용한 CNN은 token 개수 600에서 최고 성능을 보이며, 이후 유지되는 경향을 보였다. 표 4, 그림 5에서 볼 수 있듯이 doc2vec을 word2vec과 함께 적용한 경우, word2vec만을 또는 doc2vec만을 적용한 모델보다 성능이 항상 높게 측정되었다. 두 모델의 최고 분류율의 차이는 doc2vec을 함께 적용한 모델이 word2vec만을 적용한 모델보다 분류율이 2.62% 높았고, 19.98%의 개선율을 보였다.

모델간 비교를 돕기 위하여, 두 모델의 범주별 분류율과 precision, recall, F-measure, accuracy의 비교 결과를 그림 6과 그림 7에 나타내었다. 아래 두 그림은 doc2vec을 함께 사용한 모델의 경우 token 개수가 600개일 때, word2vec만을 사용한 모델의 경우는 token 개수가 700개 일 때 테스트셋의 실험결과를 이용하여 작성하였다.

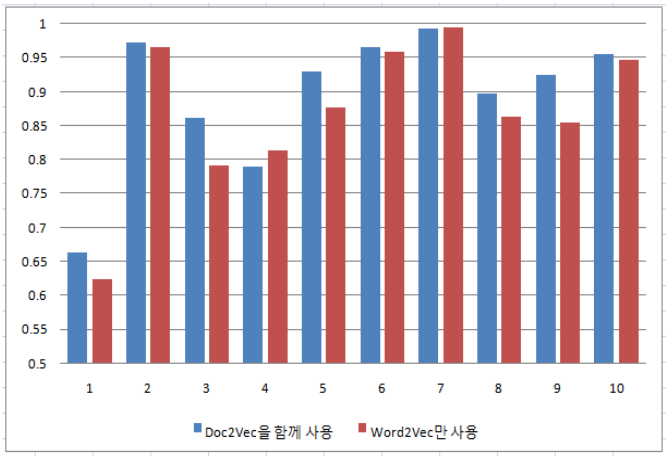

그림 6. word2vec과 doc2vec을 함께 사용한 모델과 word2vec만을 사용한 모델의 범주별 분류율 비교 (X-축은 10개 범주의 label을 나타낸다.)

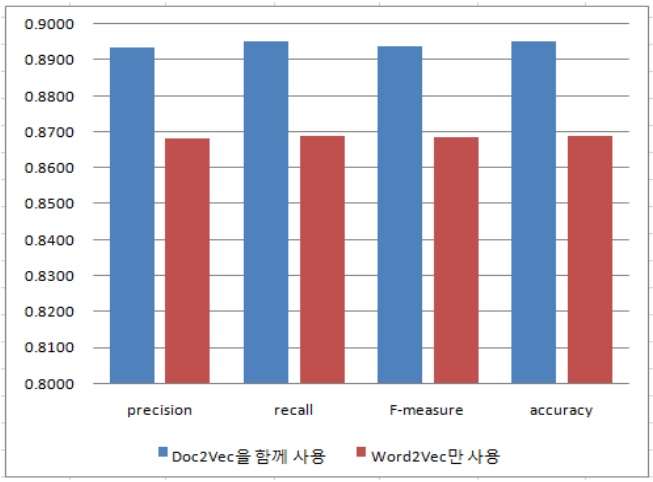

그림 7. word2vec과 doc2vec을 함께 사용한 모델과 word2vec만을 사용한 모델의 precision, recall, F-measure, accuracy 비교

그림 6을 통해서 볼 수 있듯이, 범주 4와 7을 제외한 나머지 범주에서 모두 doc2vec을 함께 활용한 모델의 분류율이 높았으며, precision, recall, F-measure, accuracy 모두 doc2vec을 함께 활용한 모델이 높음을 그림 7을 통해 확인할 수 있다.

5. 결론

본 논문에서는 문서 분류를 위하여 doc2vec과 word2vec을 함께 활용한 CNN 모델을 제안한다.

CNN의 입력자질로 사용할 word2vec과 doc2vec을 생성하기에 앞서, word2vec과 doc2vec에 입력될 token을 생성하기 위한 tokenizing 방법을 선정하기 위해 수행한 실험을 통하여, 음성인식기에 효과적이라고 알려진 WPM을 활용하여 생성한 doc2vec을 이용한 경우가 분류율 79.5%로 어절과 형태소 분석을 이용하는 경우보다 문서 분류에 더 도움이 됨을 확인하였다. 그러나, WPM이 문서 분류시에 끼치는 영향은 향후 분석이 필요할 것으로 보인다.

실험을 통해, doc2vec과 word2vec을 함께 활용하여 CNN에 적용하는 것이, word2vec만을 활용한 CNN보다 분류율 2.62% 향상, 개선율 19.98%로 좀 더 높은 성능을 보임을 검증하였다. 또한 전체 문서의 80%가 포함하는 token 개수 이상이 되면 분류율이 더 높아지지 않고 유지되는 것을 확인 할 수 있었다.

본 연구에서는 word2vec만을 사용한 기존 CNN 모델에 doc2vec을 함께 적용하는 것이 문서 분류율에 끼치는 영향을 검증하는 것이 목적이기 때문에, doc2vec 및 CNN의 튜닝은 연구대상에 포함하지 않았으나, 모델의 튜닝 및 모델의 topology를 개선하는 등의 분류율 향상 시도가 향후 필요할 것으로 보인다. 또한 다른 데이터셋으로 제안 모델의 성능을 실험하여 데이터셋에 독립적으로 성능 향상을 보이는지 여부와 범주 4와 7에서 word2vec만 사용하였을 때보다 분류율이 낮은 이유의 분석 또한 필요할 것으로 생각된다.

참고문헌

[1] 주길홍, 신은영, 이주일, 이원석, “연관규칙을 이용한 뉴스기사의 계층적 자동분류기법”, Journal of Korea Multimedia Society, 2011.

[2] 백용규, “한글 인터넷 뉴스 기사 자동 분류시스템에 관한 연구”, 고려대학교 대학원 경영학과 석사논문, 2003.

[3] Yoon Kim, "Convolutional Neural Network for Sentence Classification", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP), 2014.

[4] Ye Zhang, Byron C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification", arXiv:1510.03820, 2015.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of word Representations in Vector Space", arXiv:1301.3781, 2013.

[6] Quoc Le, Tomas Milokov, "Distributed Representations of Sentences and Documents", Proceedings of the 31st International Conference on Machine Learning, 2014.

[7] 최민재, 박현수, 정태성, 정순한, 허영, 이상헌, 황유지, “NewsML의 이해(정책자료 2007-01)”, 한국언론재단, 2007.

[8] Mike Schuster and Kaisuke Nakajima, “JAPANESE AND KOREAN VOICE SEARCH”, Google Inc, USA, 2012.

[9] 박재훈, 구명완, “WPM(Word Piece Model)을 활용한 구글 플레이스토어 앱의 댓글 감정 분석 연구”, 제28회 한글 및 한국어 정보처리 학술대회 논문집 (2016년), 2016.