ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 44, No. 7, pp. 742-747, 2017. 7
https://doi.org/10.5626/JOK.2017.44.7.742

Doc2Vec과 Word2Vec을 활용한 Convolutional Neural Network 기반 한국어 신문 기사 분류

(Categorization of Korean News Articles Based on Convolutional Neural Network Using Doc2Vec and Word2Vec)

김 도 우 †　　구 명 완 ††
(Dowoo Kim)　　(Myoung-Wan Koo)

요 약 본 논문에서는 문장의 분류에 있어 성능이 입증된 word2vec을 활용한 Convolutional Neural Network(CNN) 모델을 기반으로 하여 문서 분류에 적용 시 성능을 향상시키기 위해 doc2vec을 함께 CNN에 적용하고 기반 모델의 구조를 개선한 문서 분류 방안을 제안한다. 먼저 토큰화 방법을 선정하기 위한 초보적인 실험을 통하여, 어절 단위, 형태소 분석, Word Piece Model(WPM) 적용의 3가지 방법 중 WPM이 분류율 79.5%를 산출하여 문서 분류에 유용함을 실증적으로 확인하였다. 다음으로 WPM을 활용하여 생성한 단어 및 문서의 벡터 표현을 기반 모델과 제안 모델에 입력하여 범주 10개의 한국어 신문 기사 분류에 적용한 실험을 수행하였다. 실험 결과, 제안 모델이 분류율 89.88%를 산출하여 기반 모델의 분류율 86.89%보다 2.99% 향상되고 22.80%의 개선 효과를 보였다. 본 연구를 통하여, doc2vec이 동일한 범주에 속한 문서들에 대하여 유사한 문서 벡터 표현을 생성하기 때문에 문서의 분류에 doc2vec을 함께 활용하는 것이 효과적임을 검증하였다.

키워드: doc2vec, word2vec, word piece model(wpm), convolutional neural network(cnn)

Abstract In this paper, we propose a novel approach to improve the performance of the Convolutional Neural Network(CNN) word embedding model on top of word2vec with the result of performing like doc2vec in conducting a document classification task. The Word Piece Model(WPM) is empirically proven to outperform other tokenization methods such as the phrase unit, a part-of-speech tagger with substantial experimental evidence (classification rate: 79.5%). Further, we conducted an experiment to classify ten categories of news articles written in Korean by feeding words and document vectors generated by an application of WPM to the baseline and the proposed model. From the results of the experiment, we report the model we proposed showed a higher classification rate (89.88%) than its counterpart model (86.89%), achieving a 22.80% improvement. Throughout this research, it is demonstrated that applying doc2vec in the document classification task yields more effective results because doc2vec generates similar document vector representation for documents belonging to the same category.

Keywords: doc2vec, word2vec, word piece model(wpm), convolutional neural network(cnn)

· 이 논문은 2015년 정부(미래창조과학부)의 재원으로 국가과학기술연구회 융합연구단 사업(No. CRC-15-04-KIST)의 지원을 받아 수행된 연구임

† 비 회 원 : 서강대학교 정보통신대학원
iigreen@idtc.co.kr
†† 종신회원 : 서강대학교 컴퓨터공학과 교수(Sogang Univ.)
mwkoo@sogang.ac.kr
(Corresponding author임)

논문접수 : 2017년 2월 7일
(Received 7 February 2017)
논문수정 : 2017년 5월 11일
(Revised 11 May 2017)
심사완료 : 2017년 5월 16일
(Accepted 16 May 2017)

1. 서 론

기존에는 문서 분류의 자동화를 위하여, 단순히 문서에 나타나는 단어의 빈도를 이용하여 적합한 범주를 지정하는 통계적인 분류 방법을 이용하거나[1], 분류에 필요한 주요 단어들을 추출하고 추출된 단어들을 기반으로 K-NN, 의사결정 트리, 베이지언 네트워크, 인공 신경망 등의 데이터 마이닝 알고리즘을 이용한 연구가 진행되었다[2]. 최근에는 자연어 처리에 딥러닝 알고리즘인 컨볼루션 신경망(Convolutional Neural Network, CNN, 이하 CNN으로 표기)이 효과적이라는 것이 알려지면서, 2014년에 Kim Yoon은 단어를 벡터(vector)로 표현하는 방법인 word2vec[3]과 CNN을 이용한 문장 분류 모델[4]을 제안하고 논문을 통해 우수한 성능을 보여주었다[4].

그림 1은 [4]에서 제안한 word2vec을 활용한 CNN 모델을 도식화한 것이다. [4]에서 제안한 모델은 다양한 크기(region size)의 필터(filter)들을 여러 개 적용한 convolution layer와 1-max pooling layer가 하나로만 이루어져 있는 one-layer CNN이며, fully-connected layer도 은닉층(hidden layer)이 없고 softmax output layer만을 가지는 간단한 구조로 구성되어 있다[5]. 그림 1에서 볼 수 있듯이 [4]의 모델은 구조가 단순하기 때문에 훈련 시간이 빠르고 Logistic Regression(이하 LR로 표기) 및 Support Vector Machine(이하 SVM으로 표기)에 비해 높은 성능의 장점을 가진다[6]. 그러나 [4]의 모델은 논문을 통해 영어 문장을 대상으로 한 성능 평가 결과만을 제시하여 한국어 문서 분류에 대한 성능은 확인할 수 없으며, [4]의 모델이 CNN을 기반으로 하기 때문에 문서 분류에 적용 시 효율성이 저하되는 문제점이 발생한다. 다시 말하면 CNN은 입력으로 고정 길이를 요구하기 때문에 데이터셋 내에서 최장 길이 문서가 포함하는 토큰(token)수로 CNN의 입력 길이를 설정해야 하며 이로 인해 CNN의 입력 길이를 맞추기 위해서 대부분의 입력 문서들에 대해 많은 수의 zero-padding을 추가함으로써 자원 사용량 낭비 및 훈련 시간 증가와 같은 비효율이 발생하는 것이다. 본 연구에서 사용한 데이터셋을 예로 들면 최장 길이 문서가 포함하는 토큰

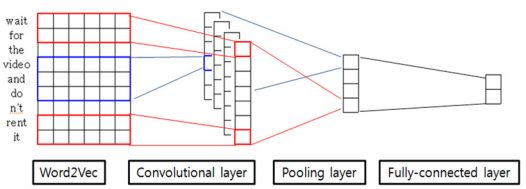

그림 1 word2vec을 활용한 CNN 모델[4]
Fig. 1 CNN model on top of word2vec[4]

수는 3,539개인데 반해 90%의 문서들은 700개 이하의 토큰을 포함하여 데이터셋의 90%에 해당하는 문서들이 CNN의 입력 길이를 맞추기 위해 약 2,839개 이상의 토큰에 해당하는 zero-padding을 추가해야 한다. 이로 인해 자원이 낭비되고 추가된 zero-padding에 해당하는 입력의 수만큼 학습되는 가중치(weight) 수도 증가하여 훈련시간이 증가하게 된다. 따라서 [4] 모델의 장점을 이유로 [4] 모델을 한국어 문서 분류에 적용함에 있어 위와 같은 문제점을 해결하기 위해 CNN의 입력 길이를 성능을 저하시키지 않는 토큰수로 제한하고 문서의 길이를 제한한 것을 보완하기 위해 문서에 포함된 토큰의 개수와 상관없이 고정된 크기로 문서 자체를 하나의 벡터로 표현하는 방법인 doc2vec[7]의 활용을 고려해보았다. doc2vec은 문서의 의미가 반영된 유사도에 기반하여 고정 크기의 문서 벡터 표현을 생성하는 알고리즘으로써, 본 연구로 하여금 문서 분류를 위하여 추가적인 접근 방식을 구상하는 계기가 되었다.

이에 본 논문에서는 doc2vec을 활용하게 된 동기에 대한 정당성과 [4] 모델을 문서 분류에 적용시 발생 가능한 문제점에 대한 해결방안을 제시하고, 문장의 분류에 있어 성능이 입증된 word2vec을 활용한 CNN 모델(이하 기반 모델로 표기)을 기반으로 하여 문서 분류에 적용 시 성능을 향상시키기 위해 CNN의 입력 길이를 제한하고 doc2vec을 함께 적용함에 있어 [5]에서 수행한 기본적인 실험을 바탕으로 최적화된 솔루션을 만들기 위하여 모델의 구조를 개선한 방안을 제안한다.

성능 평가 척도로 분류율을 사용하고, 분류율은 검증에 사용한 전체 문서 중에서 정확하게 분류된 문서가 차지하는 비율을 말하며 다음 식으로 나타낸다[1].

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

그리고 본 논문에서 사용한 개선 효과의 계산식은 다음과 같다.

$$\text{개선 효과} = \frac{\text{개선 후 분류율} - \text{개선 전 분류율}}{1 - \text{개선 전 분류율}} \times 100$$

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 모델에 관하여 기술한다. 3장에서는 기반 모델과 제안 모델을 대상으로 비교 실험하고 결과를 분석한다. 마지막으로 4장에서 본 연구의 결론 및 향후 연구 계획을 제시한다.

2. 제안 모델

본 논문에서는 word2vec을 활용한 CNN 모델의 장점을 이유로 문서 분류에 적용 시 앞서 기술한 비효율 문제점을 해결하고 성능을 향상시키기 위하여 다음과 같은 방안을 제안한다.

1. 성능을 저하시키지 않는 토큰수로 CNN의 입력 길이 제한
2. 입력 자질(input feature)의 추가 : doc2vec의 활용
3. 기반 모델의 구조 개선 : fully-connected layer에 은닉층 추가
4. 문서 분류에 유용한 토큰화 방법의 비교 선정 및 적용

CNN의 입력 길이 제한, 즉 CNN의 입력으로 문서 전체가 아닌 일부만 전달하는 것을 보완하고 성능을 향상시키기 위해 doc2vec을 활용하는 것은 doc2vec이 동일 범주에 속한 문서들에 대해 유사한 문서 벡터 표현을 생성해야 하는 가정을 전제한다. 이에, 실험 데이터셋에 대해 doc2vec 알고리즘을 이용하여 생성한 문서 벡터 표현의 군집화 경향을 확인해보기 위해 그림 2와 같이 산포도(scatter plot)를 그려보았다.

doc2vec을 이용하여 생성한 문서의 벡터 표현이 300차원이기 때문에, 산포도는 시각적으로 확인이 가능하도록 PCA(Principal Component Analysis)[8]를 이용해 2차원으로 차원을 감소하여 표현하였다. 또한 데이터의 군집화 경향을 확인하기 좋도록 하기 위해 범주는 5개, 범주별 데이터는 100개로 한정하였다. 점들의 색깔은 범주를 나타낸다. 그림 2에서 볼 수 있듯이, 비록 모든 범주의 문서 벡터들이 명확하게 구분되어 군집화 되지 않거나 또는 다른 범주의 문서 벡터들과 섞여있는 경우도 있으나, 그럼에도 불구하고 doc2vec을 활용한 문서의 벡터 표현은 범주별로 문서 벡터들이 대부분 군집화 되어 벡터 공간상에 위치하기 때문에 doc2vec의 활용이 문서 분류에 유용성이 있다고 직관적으로 판단할 수 있다.

이에 본 논문에서는, word2vec을 활용한 CNN 모델을 문서의 분류에 적용하기 위해서 doc2vec 알고리즘을 이용하여 생성한 문서의 벡터 표현을 입력 자질로 추가하는 것이 문서의 분류에 긍정적인 영향을 줄 수 있을 것이라는 관점으로, 그림 3과 같이 doc2vec과 word2vec을 함께 활용한 CNN 모델을 제안한다.

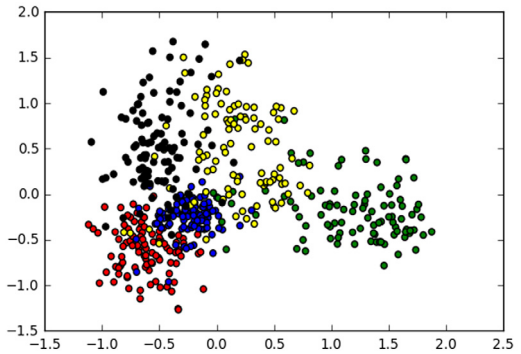

그림 2 범주별 문서 벡터의 산포도

Fig. 2 Scatter plot of document vectors per category

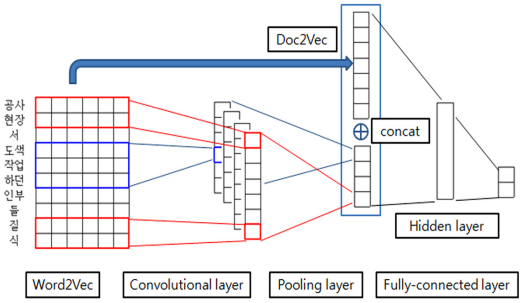

그림 3 doc2vec과 word2vec을 함께 활용한 CNN 모델

Fig. 3 CNN model on top of doc2vec and word2vec

제안 모델은 그림 3과 같이 doc2vec을 활용하여 생성한 문서의 벡터 표현이 문서 분류 수행시마다 한번만 사용되고, fully-connected softmax layer(이하 FC layer로 표기)에서 최종 분류 작업 수행 시 문서의 벡터 값들이 최대한 활용될 수 있도록 하기 위해서 word2vec을 이용해 생성한 토큰의 벡터 표현들로 이루어진 매트릭스에 대한 convolution layer 및 pooling layer의 출력 벡터와 doc2vec을 통해 생성된 문서의 벡터를 연결(concatenation)하여 top-level feature 벡터를 생성하고, 이 벡터를 FC layer의 입력 자질로 전달되도록 구성하였다. FC layer에 정규화(regularization)의 수단으로 dropout을 적용하였다. doc2vec의 활용으로 인해 FC layer 입력 자질의 크기가 늘어난 것을 감안하여, 학습 능력 및 성능 향상을 위해 은닉층이 없는 기반 모델의 FC layer를 개선하여 은닉층을 추가하였다. 그림 3에는 표현되지 않았으나 CNN의 입력 길이를 제한함에 따라 토큰의 벡터 표현으로 구성된 입력 매트릭스의 높이(height)는 고정된다.

3. 실험 및 결과

본 연구에 사용된 데이터셋은, 2013년 5월에서 7월 사이에 77개 신문사에서 NewsML[9]형식으로 작성된 한국어 신문기사 528,735개중 범주가 분류되어 있는, 즉 기자가 기사 작성 시에 기사의 범주를 지정한 기사 146,691개를 대상으로 범주별 기사수가 1,000개 이상인 10개의 범주를 선정한 후, 범주별로 1,000개의 기사를 추출하여 전체 10,000개의 기사로 데이터셋을 구성하고 실험을 수행하였다[5]. 선정된 범주는 사회일반, 경제일반, 정치일반, 행정·자치, 방송·연예, 교육, 사람, 스포츠, 문화, 사설·오피니언·칼럼이며, 실험은 아래와 같이 4단계로 진행하였다.

1. 문서 분류에 유용한 토큰화 방법 선정
2. 은닉층을 제외한 제안 모델에서 최고 성능을 산출하는 토큰의 개수 선정(CNN의 입력 길이 제한)

3. 제안 모델에 적합한 은닉층의 계층수 및 뉴런(neuron)의 개수 선정
4. 기반 모델과 제안 모델의 성능 비교

3.1 토큰화 방법 선정

단어와 문서의 벡터 표현을 생성하는 word2vec과 doc2vec을 사용하기 위해서는 문서의 토큰화가 선행되어야 한다. 따라서 문서 분류에 더 나은 성능을 보이는 토큰화 방법을 선정하기 위해 doc2vec을 활용하여 생성한 문서의 벡터 표현을 입력 자질로 전달하여 초보적인 문서 분류율 측정 실험을 수행하였다. 기사를 어절 단위, 형태소 분석[10], WPM[11] 적용의 3가지 방법으로 토큰화한 후, 해당 토큰들로 doc2vec 알고리즘을 이용하여 생성한 문서의 벡터 표현을 LR 분류기에 전달하여 분류율을 측정하였다. 리소스 사용 및 성능을 고려하여, 벡터의 크기는 300차원으로 생성하였다. 수행 결과는 표 1과 같이 WPM을 적용한 결과가 분류율 79.5%로 가장 높은 성능을 산출하였다[5].

표 1 토큰화 방법별 분류율 비교[5]
Table 1 Comparison of tokenizing methods[5]

	phrase unit	part-of-speech tagger	WPM
Classification rate	74.8%	77.6%	79.5%

이와 같은 초보적인 실험 결과로 WPM을 적용한 토큰화 방법이 문서 분류에 유용하다고 결론을 내리기엔 무리일 수도 있다. 하지만, [12]에서도 WPM이 분류 성능에 있어 효과적임을 이미 실증적으로 입증하였고, 본 실험 결과에서도 가장 높은 성능을 산출한 점에 기반하여 이어지는 실험에서는 WPM을 적용하여 생성한 토큰들을 doc2vec과 word2vec의 입력으로 사용하였다.

3.2 토큰의 개수 선정

기반 모델과 제안 모델(은닉층 제외)에 대해 성능을 저하시키지 않는 CNN의 입력 토큰수를 선정하는 실험은 토큰의 개수를 100개 단위로 증가시키며, zero-padding을 적용하여 수행하였다[5]. CNN의 입력 토큰수에 관하여 예를 들면, 토큰 개수 100은 입력 문서가 포함한 토큰의 개수가 100개 이상인 경우에는 첫 번째 토큰부터 100번째 토큰까지 CNN의 입력으로 전달하고, 문서가 포함한 토큰의 개수가 100개 이하인 경우에는 zero-padding을 적용하여 CNN의 입력 길이를 100으로 만들었음을 의미한다. 토큰의 개수가 700개 이상이 되면 더 이상의 성능 개선이 발견되지 않음으로 토큰 개수 1,000개까지 실험을 수행한 후 종료하였다. 두 모델의 매개변수(hyper-parameter) 설정은 [4]에서 제안한 모델의 기본 설정을 준용하였고 표 2와 같다[5].

표 2 기본 설정[6]
Table 2 Baseline configuration[6]

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

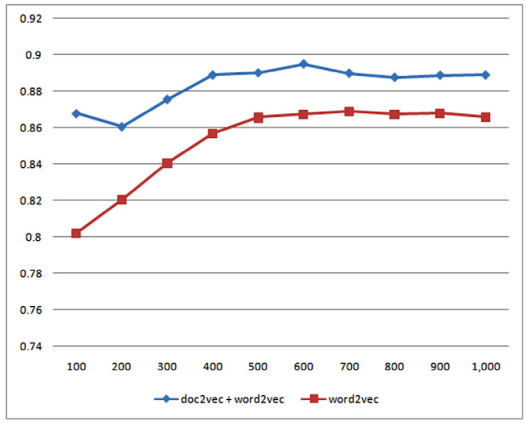

그림 4 두 모델의 토큰 개수별 성능 비교[5]
Fig. 4 Comparison of performance results per number of tokens of the two models[5]

성능 평가 척도는 앞서 기술하였듯이 분류율이며, 학습 데이터와 테스트 데이터의 비율을 90%, 10%로 설정하고 10-fold cross validation 방법으로 10번 수행 후 평균값을 계산하여 분류율을 측정하였다. 모델별 실험결과는 그림 4와 같다.

word2vec만 사용한 CNN 모델(기반 모델)은 토큰 개수 700개에서 분류율 86.89%로 최고 성능을 보이고 이후 유지되는 경향을 보이며, doc2vec을 함께 사용한 CNN 모델(제안 모델)은 토큰 개수 600개에서 분류율 89.51%로 최고 성능을 보이며, 이후 유지되는 경향을 보였다[5]. 따라서, 제안 모델에 적합한 은닉층의 계층수 및 뉴런의 개수를 선정하는 실험에서는 토큰의 개수를 600개로 고정하여 진행하였다.

3.3 제안 모델의 은닉층 계층수와 뉴런수 선정

FC layer의 은닉층의 계층수는 1과 3으로, 은닉층의 뉴런수는 300개, 600개, 900개로 늘려가며 총 6개의 경우에 대하여 성능을 측정하였다. 측정 결과는 그림 5와 같이 뉴런의 개수 600개로 구성된 은닉층을 한 계층 추가할 때 89.88%로 최고 성능을 보이며, 은닉층 추가 전 성능(89.51%)보다 성능이 0.37%가 향상되었다.

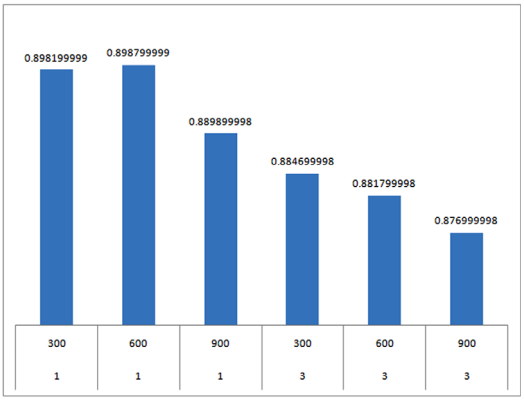

그림 5 은닉층의 계층수와 뉴런수에 따른 성능
Fig. 5 Performance results of model per number of hidden layer(s) and neurons

3.4 모델별 성능 비교 분석

그림 4를 통해 word2vec만 적용한 모델보다 doc2vec을 함께 적용한 모델이 항상 높은 성능을 산출하는 것을 확인할 수 있다. 이 실험 결과에 기반하여, word2vec만 활용하는 것보다 word2vec과 doc2vec을 함께 활용하는 것이 문서의 분류에 있어 성능 향상에 효과적임을 알 수 있다. 이것은 doc2vec이 동일한 범주에 속한 문서들에 대해 유사한 문서 벡터 표현을 생성하고 이러한 역할을 수행하는 doc2vec을 이용하여 생성한 문서의 벡터 표현과 word2vec을 이용하여 생성한 단어 벡터 표현들의 CNN 출력값이 연결되어 시너지 효과를 일으킴으로써 문서의 범주 분류 성능을 향상시킨 것으로 판단된다. 그리고 FC layer에 은닉층을 추가함으로써 성능이 더 향상되어 최종적으로 제안한 모델의 분류율(89.88%)이 기반 모델의 분류율(86.89%)보다 2.99% 높게 산출되고 22.80%의 개선 효과를 보였으며, t 검증 결과 또한 p값이 0.000186255를 보임으로써 유의수준 5%에서 개선후 분류율은 유의미한 차이가 있다는 결론을 도출할 수 있다.

4. 결 론

본 논문에서는 문장 분류에 있어 성능이 입증된 word2vec-CNN 모델을 문서 분류에 적용 시 발생하는 비효율 문제를 해결하고 성능을 향상시키기 위해 CNN의 입력 길이를 제한하고 doc2vec을 함께 활용하며 은닉층을 추가한 개선된 모델을 제안한다.

먼저 토큰화 방법을 선정하기 위해 수행한 실험을 통하여 음성인식기에 효과적이라고 알려진 WPM을 활용한 경우가 분류율 79.5%를 산출하여 어절 단위와 형태소 분석을 이용한 경우보다 문서의 분류에 더 유용함을 실증적으로 확인하였다. 그리고 앞서 수행한 일련의 실험을 통해 제안 모델의 성능(89.88%)이 기반 모델의 성능(86.89%)보다 2.99% 향상되고 22.80%의 개선 효과를 보임을 검증하였다. 이는 doc2vec이 동일한 범주에 속한 문서들에 대해 유사한 문서 벡터 표현을 생성하고 이러한 성질을 가진 doc2vec이 word2vec과 시너지 효과를 냄으로써 최종 성능에 긍정적인 영향을 끼친 것으로 판단된다. 또한 한국어 신문 기사를 대상으로 한 성능 측정 결과로 두 모델 모두 우수한 성능을 보임으로써 한국어 문서 데이터셋의 분류에 유효함을 알 수 있다.

본 연구를 통해 doc2vec을 함께 활용하는 것이 문서 분류의 성능 향상에 효과적임을 검증하였다. 향후 문서의 분류 작업에 doc2vec이 활용되길 기대한다.

본 연구에서 사용한 문서 토큰화 방법의 선정은 [12]의 실험 결과와 본 연구에서 수행한 초보적인 실험 결과에 기반하였기 때문에 WPM이 문서 분류에 끼치는 영향은 향후 분석이 필요할 것으로 보인다. 또한 본 논문은 doc2vec의 활용이 문서 분류 성능에 끼치는 영향을 검증하는 것이 목적으로 doc2vec의 튜닝(tuning)을 비롯하여 모델의 매개변수 튜닝 및 FC layer 외 구조 변경을 통한 성능 향상 연구는 연구 대상에서 제외하였으나, 향후 관련 연구를 지속할 계획이다. 그리고 다른 문서 데이터셋을 제안 모델에 적용해 봄으로써 데이터셋의 종류에 상관없이 제안 모델이 문서 분류에 있어 성능 향상에 유용한지 여부를 향후에 검증해보고자 한다.

References

[1] K.H.Joo, E.Y.Shin, J.I.Lee, W.S.Lee, "Hierarchical Automatic Classification of News Articles based on Association Rules," *Journal of Korea Multimedia Society*, Vol. 14, No. 6, pp. 730-741, Jun. 2011. (in Korean)

[2] Y.G.Beak, "A Study on Automatic Classification System of Hangul Internet News Articles," *Korea University, Graduate School of Department of Business Administration*, a master's thesis, 2003. (in Korean)

[3] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, "Efficient Estimation of word Representations in Vector Space," arXiv:1301.3781v3, Sep. 2013.

[4] Yoon Kim, "Convolutional Neural Network for Sentence Classification," *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pp. 1746-1751, Oct. 2014.

[5] D.W.Kim, M.W.Koo, "A Study on Categorization of Korean News Article based on CNN using Doc 2Vec," *the 28th Annual Conference on Human and Cognitive Language Technology*, pp. 67-71, 2016. (in Korean)

[6] Ye Zhang, Byron C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional

Neural Networks for Sentence Classification," arXiv:1510.03820v4, Apr. 2015.
[7] Quoc Le, Tomas Milokov, "Distributed Representations of Sentences and Documents," *Proc. of the 31st International Conference on Machine Learning*, 2014.
[8] scikit-learn developers, "2.5. Decomposing signals in components (matrix factorization problems) scikit-learn 0.18 documentation," [Online]. Available:http://scikit-learn.org/stable/modules/decomposition.html (downloaded 2016, Oct. 28)
[9] M.J.Choi, H.S.Park, T.S.Jeong, S.H.Jeong, Y.H, S.H.Lee, Y.J.Hwang, "Understanding of NewsML (policy data report 2007-01)," *Korea Press Foundation*, pp. 10-18, 2007. (in Korean)
[10] E.J.Park, S.Z.Cho, "KoNLPy: Korean natural language processing in Python," *the 26th Annual Conference on Human and Cognitive Language Technology*, 2014. (in Korean)
[11] Mike Schuster and Kaisuke Nakajima, "JAPANESE AND KOREAN VOICE SEARCH," Google Inc, USA, 2012.
[12] J.H.Park, M.W.Koo, "A Study on the Sentiment analysis Google Play Store App Comment Based on WPM(Word Piece Model)," *the 28th Annual Conference on Human and Cognitive Language Technology*, pp. 291-295, 2016. (in Korean)

김 도 우

2000년 인하대학교 전자계산공학과(학사) 2000년~2001년 하늘사랑 개발팀. 2001년~2002년 이룸소프트 개발팀. 2002년~2003년 대우증권 e-biz 시스템부 CRM팀. 2003년~2009년 SK커뮤니케이션즈 개발팀. 2009년~현재 인천도시공사 정보화사업팀. 2017년 서강대학교 정보통신대학원(석사). 관심분야는 지능형 음성대화 인터페이스

구 명 완

1982년 연세대학교 전자공학과(학사). 1985년 KAIST 전기및전자공학(석사). 1991년 KAIST 전기및전자공학(박사). 1996년~1997년 미국 벨연구소 포스트닥. 1985년~2012년 KT 중앙연구소 상무보. 2012년~현재 서강대학교 컴퓨터공학과 부교수. 관심분야는 지능형 음성대화 인터페이스