워드 임베딩 기반 연구 논문 분류 기법

비스와스 딥또*, 길준민**†
*대구가톨릭대학교 컴퓨터소프트웨어학과
**대구가톨릭대학교 컴퓨터소프트웨어학부
*dipto.biswas94@gmail.com, **jmgil@cu.ac.kr

Research Paper Classification Scheme based on Word Embedding

Biswas Dipto*, Joon-Min Gil**
*Dept. of Computer Software Engineering, Daegu Catholic University
**School of Computer Software Engineering, Daegu Catholic University

요 약

텍스트 분류(text classification)는 원시 텍스트 데이터로부터 정보를 추출할 수 있는 기술에 기반하여 많은 양의 텍스트 데이터를 관심 영역으로 분류하는 것으로 최근에 각광을 받고 있다. 본 논문에서는 워드 임베딩(word embedding) 기법을 이용하여 특정 분야의 연구 논문을 분류하고 추천하는 기법을 제안한다. 워드 임베딩으로 CBOW(Continuous Bag-of-Word)와 Sg(Skip-gram)를 연구 논문의 분류에 적용하고 기존 방식인 TF-IDF(Term Frequency-Inverse Document Frequency)와 성능을 비교 분석한다. 성능 평가 결과는 워드 임베딩에 기반한 연구 논문 분류 기법이 TF-IDF에 기반한 연구 논문 분류 기법보다 좋은 성능을 가진다는 것을 나타낸다.

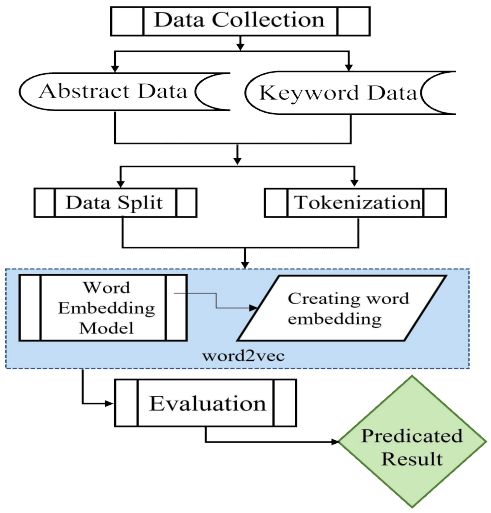
1. 서론

텍스트 분류 기술은 원시 텍스트 데이터로부터 정보를 추출할 수 있는 기술에 기반하여 많은 양의 텍스트를 관심 영역으로 분류하는 것으로 최근에 각광을 받고 있다. 텍스트 분류는 자연어 처리의 한 분야로서 이미지 분류, 감성 분석, 트위터와 페이스북의 댓글 리뷰 분류, 뉴스 기사의 주제 분류 등의 분야에서 활용되고 있다. 그러나, 텍스트로부터 의미 있는 정보를 추출하는 것은 쉽지 않은 작업이다. 기존의 이미지 분류나 음성 인식에서 활용되는 기술은 텍스트 분류를 위해 사용되는 자연어 처리 기술과 다르다. 특히, 문장 내의 단어들은 주변 단어에 의해 영향을 받아 표현될 수 있기 때문에 문장 내에서 단어간 관계로 정의하는 것이 중요하다.

따라서, 본 논문에서는 문장 내에서 단어 간의 관계를 수치적으로 표현할 수 있는 워드 임베딩(word embedding) 기술로 CBOW(Continuous Bag-of-Word)와 Sg(Skip-gram) 알고리즘을 연구 논문의 분류에 적용하고자 한다. 아울러, 이들 워드 임베딩 알고리즘에 기반한 연구 논문 분류 방식을 TF-IDF(Term Frequency-Inverse Document Frequency)에 기반한 연구 논문 분류 방식과 비교 분석하고자 한다.

2. 시스템 모델

본 논문의 워드 임베딩 기반의 연구 논문 분류에 관한 전반적인 흐름은 그림 1과 같다.



(그림 1) 연구 논문 분류의 전반적인 흐름

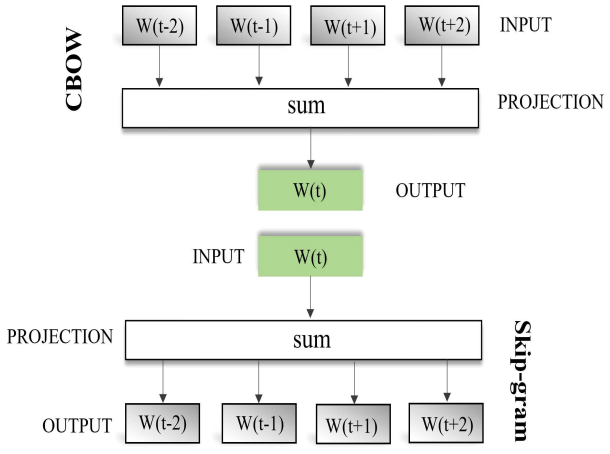
그림 1에서 볼 수 있듯이, 논문 정보에 해당하는 데이터셋은 “Science Direct” 웹사이트에서 제공하고 있는 FGCS(Future Generation Computer System)

† 교신저자

저널의 데이터를 웹크롤링을 통해 수집하여 활용한다. 약 1,000개의 연구 논문에서 수집된 논문 데이터 셋은 크게 두 가지 유형으로부터 구성된다. 하나는 논문의 초록(abstract) 데이터이며 다른 하나를 논문의 키워드(keyword) 데이터이다. 다음으로, 수집된 데이터를 Gensim 라이브러리[1]를 활용하여 토큰화를 수행하며 이를 통해 단어 단위로 분할한다. 그런 다음, 문장 내의 단어들 중에 URL, 숫자, 구두점, 불용어 등 불필요한 단어들을 제거한다. 또한, 워드 임베딩 작업 준비를 위해 각 단어들을 명사형으로 변환한다. 본 논문에서는 word2vec에 기반한 워드 임베딩 방식[2]을 사용한다. 마지막으로 논문의 초록 데이터와 키워드 데이트를 기반으로 성능 평가를 수행하고 결과를 도출한다.

3. 워드 임베딩

워드 임베딩(word embedding)은 단어들의 관계를 추출하기 위해 단어들을 벡터로 표현한 자연어 처리 기술 중의 하나이다[3, 4]. 일반적으로 워드 임베딩은 유사한 의미를 가진 단어들을 군집으로 구성할 수 있도록 각 단어를 실수값 형태의 벡터로 표현하며, 이렇게 바뀌어진 벡터는 계산이 용이하다는 장점이 있다[5]. 한편, 워드 임베딩은 텍스트 분류를 위해 word2vec 형태로 개발되어 왔으며, 대표적 방식으로는 Mikolov[3]에 의해서 제안된 CBOW(Continuous bag of words)와 Sg(Skip-gram) 알고리즘이 있다. 그림 2는 이들 두 알고리즘 수행 절차를 도식적으로 보여준다.



(그림 2) CBOW와 Skip-gram의 수행 절차

CBOW는 주변 단어를 기반으로 중간 단어를 예측하여 모델을 구성하며, 반면 Sg는 중간 단어로 주변 단어를 예측하고 이를 벡터로 나타낸다. CBOW와 Sg의 기본 아이디어는 단어들이 서로 유사할수록 좀더 유사한 벡터 값을 가지도록 하는 것이다.

3.1 word2vec 모델

본 논문에서는 연구 논문 초록의 모든 단어를 사용하여 word2vec 모델을 구성하였다. 이를 위해 윈도우 크기를 5로 설정하였으며, 50회 이상 나오는 단어만을 대상으로 100번의 반복을 수행하였다. 이런 설정을 기반으로 CBOW과 Sg 모델을 각각 적용하였으며, 가중치 행렬과 TF-IDF와 관련된 문서-단어 행렬(DTM; Document-Term Matrix)을 계산한 후 문서와 관계성이 높은 단어를 추출한다.

3.2 문서 단어 행렬

DTM은 여러 문서에 나타난 각 단어의 빈도를 나타낸다. DTM에서 각 행은 특정 문서를 나타내며 각 열은 특정 단어를 나타낸다. 표 1은 DTM의 구성 예를 보여준다.

<표 1> DTM의 구성 예

	model	**system**	**…**	**sensor**
doc 1	0	1	...	0
…	…	…	…	…
doc 1000	1	1	...	0

3.3 가중치 행렬

가중치 행렬은 단어 간의 관계를 나타내는 것으로 다음의 수식으로 계산된다.

$$W_{ij} = \exp\left(-\frac{d(x_i,x_j)^2}{2\sigma^2}\right) \qquad (2)$$

여기서, $W_{i,j}$는 단어 x_i와 x_j사이의 가중치 값을 의미하며, 정규분포 모양을 따르도록 $2\delta^2$ 값으로 나눈다. 그리고 $d(x_i,x_j)$는 단어 x_i와 x_j 사이의 거리를 나타내며 본 논문에서는 유클리디안 거리(Euclidean distance)를 사용한다.

본 논문에서는 가장 많이 나오는 N개의 단어(즉, Top-N 단어)를 기반으로 수식 (2)의 가중치 행렬을 유도하였다. N값으로 10, 20, 30, 40, 50을 사용하여 가중치 행렬을 계산하였다. 표 2는 가중치 행렬의 구성 예를 보여준다.

<표 2> 가중치 행렬의 구성 예

	model	system	...	sensor
model	1.000000	-0.167018	...	-0.183092
...	...	...	...	...
security	-0.245440	0.126722	...	0.083788

3.4 Score 값 계산

최종적으로 1,000개 문서에 대해서 상위 빈도의 10개, 20개, 30개, 40개, 50개 단어 각각의 Score 값을 계산한다. 앞서 제시한 표 1의 DTM과 표 2의 가중치 행렬을 바탕으로 두 행렬의 내적에 의해서 계산된다. 표 3은 문서별 단어에 대한 Score 값의 구성 예를 보여준다.

<표 3> Score 값의 구성 예

	model	system	network	...
doc 1	**3.89E+00**	-2.68E+00	5.05E-01	...
doc 641	2.01E-01	**2.73E+00**	-3.13E-01	...
doc 28	2.93E+00	-2.80E-01	**3.08E+00**	...
...	...	...	...	...

표 3과 같이 얻어진 Score 값은 특정 단어에 대해서 주요 문서를 검색하는데 사용된다. 즉, 연구 논문의 분류에 적용하자면, 특정 단어 별로 주요 연구 논문을 검색하고 특정 단어에 대해서 유사한 연구 논문끼리 분류하는데 사용될 수 있다.

4. 단어 빈도-역 문서 빈도

이 절에서는 본 논문의 워드 임베딩과 비교 대상이 되는 TF-IDF(Term Frequency-Inverse Document Frequency)에 대해서 살펴본다. 텍스트 처리에 적용되는 TF-IDF는 단어별로 문서의 상대적 빈도 수에 기반으로 TF 값와 IDF 값의 곱으로 계산된다. 먼저, 문서 내의 단어 빈도를 나타내는 TF를 다음과 같이 계산한다.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (3)$$

여기서, $tf_{i,j}$는 문서 j에서 단어 i의 TF 값을 나타내며, $n_{i,j}$는 문서 j에서 단어 i가 나오는 빈도 수를 나타낸다.

IDF는 단어 t를 포함하고 있는 문서 수(df_t)에 대한 역수의 의미로서 다음과 같이 계산된다.

$$idf(d,t) = \log\left(\frac{N}{df_t}\right) \quad (4)$$

여기서, $idf(d,t)$는 문서 d에 대한 단어 t의 IDF 값을 나타내며, N은 문서의 전체 수를 나타낸다.

본 논문에서는 CBOW와 Sg에서 사용한 상위 10개, 20개, 30개, 40개, 50개 단어를 기반으로 TF-IDF 값을 계산한다.

5. 실험 결과

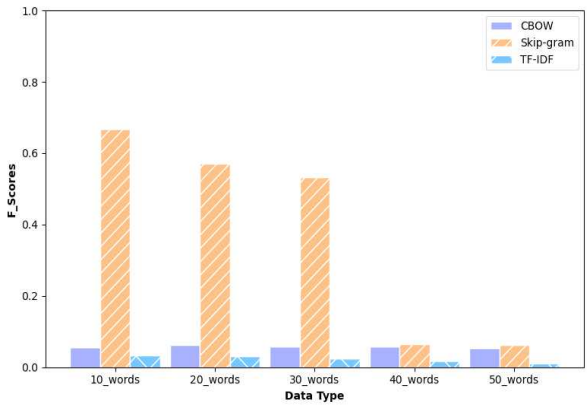
이 절에서는 본 논문에서 제안하는 워드 임베딩 기반 연구 논문 분류 기법에 대한 성능 평가를 수행한다. 본 논문의 제안 기법에 대한 성능을 측정하기 위해 수식 (5)와 수식 (6)으로 각각 정의되는 정밀도(precision) 및 재현율(recall)에 기반한 수식 (7)의 F-Score[6]에 이용하여 제안 기법과 TF-IDF 기법의 성능을 평가한다.

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \quad (5)$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \quad (6)$$

$$\text{F-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})} \quad (7)$$

여기서, {relevant documents}는 검색하고자 하는 단어와 관련된 문서의 수를 전문가 판단에 의해 얻은 것을 의미하며, 본 논문에서는 연구 논문의 키워드와 매칭되는 문서의 수로 계산하였다. 그리고 {retrieved documents}는 검색하고자 하는 단어와 관련된 문서의 수를 의미하며, 본 논문의 제안 기법인 CBOW, Sg, 그리고 기존 기법인 TF-IDF로 구한다.



(그림 3) CBOW, Sg와 TF-IDF의 성능 평가 결과

그림 3은 CBOW, Sg, TF-IDF의 성능 평가 결과를 보여준다. 그림 3의 결과를 분석하면, Top-N으로 10개, 20개, 30개 단어에 대해서 Sg 기법이 다른 기법에 비해서 높은 F-Score 값을 보여주었다.

즉, 전체 문서에서 30개까지 많이 나오는 단어들에 대해서 좋은 성능을 보여주었다. 이는 30개 이하의 단어들에 대해서는 비교적 단어 간의 관계성을 잘 표현하여 비교적 높은 성능을 보이지만, 40개 이상의 단어들에 대해서는 단어 간의 관계성을 비교적 잘 표현하지 못하여 낮은 성능을 나타내는 것으로 보인다. 한편, 워드 임베딩에 기반한 CBOW와 Sg 기법은 Top-N의 모든 경우에 대해서 TF-IDF 기법보다 높은 성능을 가짐을 알 수 있다.

6. 결론

본 연구에서는 워드 임베딩에 기반한 연구 논문 분류를 바탕으로 연구 논문의 추천 방법을 제안하였다. word2vec에 기반한 CBOW와 Sg을 연구 논문의 분류에 적용하였으며, TF-IDF에 기반한 연구 논문의 분류 기법과 F-Score를 활용하여 비교·분석하였다. 성능 평가 결과는 word2vec에 기반한 Sg 기법이 CBOW 기법보다 높은 성능을 보여주었으며, 이는 Sg기법이 중간 단어(핵심어)로부터 주변 단어들의 관계성을 비교적 잘 표현하여 주어진 단어에 대해서 유사 단어를 가진 연구 논문을 비교적 잘 분류할 수 있기 때문인 것으로 보인다. 한편, 워드 임베딩을 사용한 CBOW와 Sg 기법은 TF-IDF 기법에 비해 모든 경우에 대해서 높은 성능을 가짐을 보여주었다.

Acknowledgment

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1F1A1062039).

참고문헌

[1] Gensim, https://pypi.org/project/gensim/.

[2] Word2Vec, https://radimrehurek.com/gensim/index.html.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” Proceeding of the International Conference on Learning Representations (ICLR), 2013.

[4] H. Liu. “Sentiment analysis of citations using word2vec,” University of Nottingham, Malaysia Campus, 2017.

[5] E. Nalisnick, B. Mitra, N. Craswell, and R. Caruana, “Improving document ranking with dual word embeddings,” Proceedings of the 25th International Conference Companion on World Wide Web, pp. 83-84, 2016.

[6] Morgan and Claypool, “Information Retrieval evaluation,” nlp.stanford.edu/IR-book/pdf/08eval.pdf, pp. 151-175.