형태소 임베딩과 SVM을 이용한 뉴스 기사 정치적 편향성의 자동 분류

조단비*, 이현영*, 박지훈**, 강승식*
*국민대학교 컴퓨터공학과
**다하미 커뮤니케이션즈
daanv319@kookmin.ac.kr, hyunyoung2@kookmin.ac.kr,
hoonzinope@dahami.com, sskang@kookmin.ac.kr

Automatic Bias Classification of Political News Articles by using Morpheme Embedding and SVM

Dan-Bi Cho*, Hyun-Young Lee*, Ji-Hoon Park**, Seung-Shik Kang*
*Dept. of Computer Science, Kookmin University
**Dahami Communications Co.

요 약

딥러닝 기술을 이용한 정치적 성향의 편향성 분류를 위하여 신문 뉴스 기사를 수집하고, 머신러닝을 위한 학습 데이터를 구축하였다. 학습 데이터의 구축은 보수 성향과 진보 성향을 대표하는 6개 언론사의 뉴스에서 정치적 성향을 이진 분류 데이터로 구축하였다. 뉴스 기사의 수집 방법으로 최근 이슈들 중에서 정치적 성향과 밀접하게 관련이 있는 키워드 15개를 선정하고 이에 관한 뉴스 기사들을 수집하였다. 그 결과로 11,584개의 학습 및 실험용 데이터를 구축하였으며, 정치적 편향성 분류를 위한 머신러닝 모델을 설계하였다. 머신러닝 기법으로 학습 및 실험을 위해 형태소 단위의 임베딩을 이용하여 문장 및 문서 임베딩으로 확장하였으며, SVM(Support Vector Machine)을 이용하여 정치적 편향성 분류 실험을 수행한 결과로 75%의 정확도를 달성하였다.

1. 서론

입력 문장을 분석 또는 생성하기 위해 문장들을 토큰 단위로 표현하여 벡터로 구성하는 기법을 사용한다. 영어에서는 문장의 의미를 표현하는 최소 단위로 단어를 하나의 토큰으로 임베딩한다.[1,2] 특히, 머신러닝 모델에서는 토큰을 연속적인 벡터 공간에 표현함으로써 모델의 입력값으로 사용한다.[3,4]

단어 임베딩 방법은 TF-IDF와 같이 단어 쌍이 함께 출현하는 빈도수를 기반으로 임베딩하는 방법과 주변 단어들로부터 단어를 예측하여 벡터를 구성하도록 하는 예측 기반의 임베딩 방법으로 나누어진다.[6] 예측 기반의 임베딩 기법으로는 Mikolov(2013)가 제안한 CBOW(Continuous Bag Of Words)와 skip-gram의 word2vec과 GloVe, FastText 등이 있다.[3,4,7] 대부분의 연구에서 이와 같은 예측 기반 임베딩이 보다 높은 성능을 나타낸 것으로 알려져 있으며[8], Kim(2014)과 Santos(2014)는 어절 단위 토큰의 단어 임베딩으로 skip-gram을 사용하여 실험을 진행하였다.[1,2]

굴절어인 영어와 달리, 교착어인 한국어의 어절은 형태소들의 조합으로 이루어진다. 이러한 점에서 한국어는 어절보다 형태소 단위 토큰이 다양한 언어적 의미를 표현할 수 있다.[5] 본 논문에서는 정치적 성향의 편향성을 분류 실험에서 형태소 단위 토큰화를 이용한 한국어 임베딩 방법론을 제안한다.

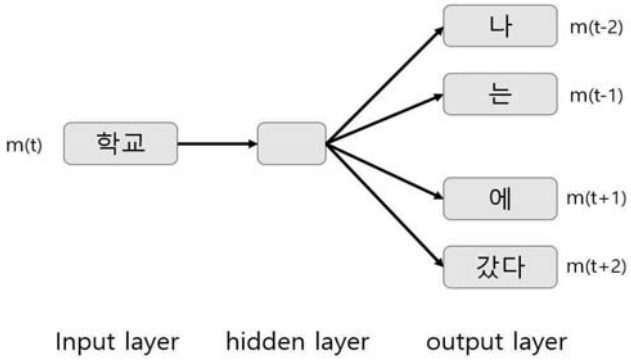
2. 정치적 편향성의 분류 모델

skip-gram[1]은 어절 단위 토큰을 사용하여 중심 단어 벡터로부터 주변 단어를 예측하는 방식으로, 연속적인 벡터 공간에 각각의 독립적인 단어들을 벡터로 표현한다.[3] 이처럼 단어들마다 독립적인 벡터를 할당하기 때문에 어절 내부의 형태학적 정보를 포함하지 못하므로 어절을 구성하는 형태소 단위 토큰을 입력 벡터로 구성하는 형태소 임베딩을 제안한다.

형태소 임베딩은 skip-gram을 확장한 모델로 각각의 단어가 어절이 아닌 형태소 단위의 토큰으로 입력된다. “나는 학교에 갔다”를 형태소 분석기[2]

1) gensim을 이용하여 window size 5, min-count 1, negative sampling 5, ns_exponent 0.75의 skip-gram으로 파라미터를 조정하여 벡터를 구성하였다. (https://radimrehurek.com/gensim/)

Okt로 분석하면 ['나', '는', '학교', '에', '갔다']의 형태소 토큰이 생성되며, 이러한 형태소 임베딩의 예시는 그림 1과 같다.



(그림 1) skip-gram을 이용한 형태소 임베딩

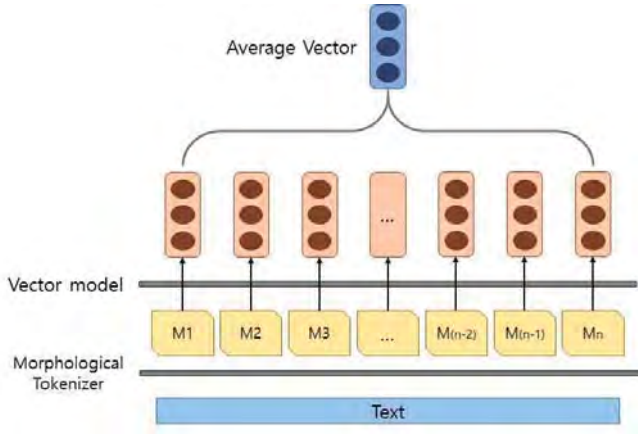
학습 데이터 M를 구성하는 토큰 {m1, m2, m3,…mT}에 대하여 중심 단어 c의 벡터를 Vc, 윈도우 크기에 속하는 주변 단어 s의 벡터를 Vs라고 할 때, 중심 단어 토큰 mc와 주변 단어 토큰 ms의 등장 확률 값은 softmax 함수로 계산되며 (1)과 같이 정의된다. 이는 목적 함수 (2)를 계산할 때 학습 데이터의 크기만큼 연산 비용이 소요된다는 단점이 존재하며, 효율적인 연산을 위해 확률을 근사적으로 계산하는 negative sampling 기법의 목적함수를 사용하였다.[9]

$$P(m_s|m_c) = \frac{\exp(v_s^T v_c)}{\sum_{m=1}^{M} \exp(v_m^T v_c)} \quad (1)$$

$$J(\Theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{-s \le j \le s, j \ne 0} \log P(m_{t+j}|m_t) \quad (2)$$

정치적 편향성을 분류하기 위해서는 문서 벡터를 구성하여야 한다. 형태소 임베딩을 통해 생성된 문서 내 형태소 토큰들의 벡터를 (v1, v2, ..., vn)라고 할 때, 문서 벡터 V는 average(v1, v2, ..., vn)와 같이 토큰 벡터들의 평균으로 표현된다. 이처럼 형태소 임베딩을 통해 문서 벡터를 구성하는 모델은 그림 2와 같이 설계하였다.

SVM[3]은 대표적인 분류 모델이다. 이는 분류 모델의 입력 데이터를 기저 벡터라고 할 때, 각 정치적 성향의 기저 벡터들과 분류 경계면 간의 거리,



(그림 2) 정치적 성향의 편향성을 위한 분류 모델

즉 마진을 최대화하고자 한다.[10] 분류 경계면의 직선 판별함수는 $f(x) = w^T x - w_0$과 같다. 이 판별함수를 통해 계산되는 값을 score라고 할 때, 보수 성향에 속하는 기저 벡터의 score는 0보다 큰 값이고, 진보 성향에 속하는 기저 벡터의 score는 0보다 작은 값이 된다. SVM의 분류 모델을 최적화하기 위한 손실함수는 hinge loss를 사용하였으며, (3)과 같이 계산된다. SVM은 이러한 손실 값을 최소화하도록 모델을 학습한다.

$$L_i = \sum_{j \ne y_i} \begin{cases} 0 & \text{if } s_{y_i} \ge s_j + 1 \\ s_j - s_{y_i} + 1 & otherwise \end{cases} = \sum_{j \ne y_i} \max(0,\ s_j - s_{y_i} + 1) \quad (3)$$

3. 실험 및 평가

3.1 데이터 구축

정치적 성향의 편향성을 분류하기 위해 정치적 성향을 나타내는 키워드를 기반으로 뉴스 기사를 크롤링하여 데이터를 수집하였다. 보수 성향과 진보 성향을 대표하는 6개 언론사의 뉴스에서 정치적 성향과 밀접한 관련이 있는 키워드를 선정하고, 각 키워드를 통해 검색되는 기사를 추출하여 구축하였다. 키워드는 인물, 사건, 주요 단어를 기준으로 추출하였으며, 총 15개의 키워드를 기준으로 하여 11,584개의 기사를 수집하였다. 정치적 편향성 관련된 키워드는 나무 위키[4]에서 선정하였으며, 선정한 키워드 15개는 다음과 같다.

2) https://konlpy-ko.readthedocs.io/ko/v0.4.3/
3) SVM은 Cornell 대학에서 구현한 SVM-Light 모델을 사용하였다. (http://svmlight.joachims.org/)
4) https://namu.wiki/w/분류:2019년%20사건

<표 1> 정치적 편향성 기사의 문장과 어절 수

	보수	진보
기사 수	5,792	5,792
문장 수	122,544	163,264
어절 수	2,082,330	2,748,160

<표 2> 토크나이저 별 중복 제거한 토큰 수

	Train(9,267)	Test(2,317)
Hannanum	201,139	79,450
Komoran	62,181	35,325
Okt	114,594	57,998

<표 3> SVM을 이용한 정치적 편향성 분류 정확도(%)

	Hannanum			Komoran			Okt		
	100	200	300	100	200	300	100	200	300
100	69.23	69.57	71.21	72.59	72.42	72.64	71.69	72.42	72.81
200	70.18	69.36	69.83	73.93	74.28	74.71	73.72	72.77	73.41
300	65.69	71.86	68.80	73.24	74.06	74.75	74.54	75.27	74.58
400	70.69	71.56	67.33	75.18	75.31	75.44	73.50	**75.70**	72.90
500	73.72	66.98	74.02	75.49	75.49	74.41	74.94	75.57	75.31

- 인물: 홍익표, 이명박, 손혜원, 트럼프, 조국
- 사건: 지소미아, 패스트트랙, 파업, 하명수사, 검찰 개혁
- 주요어: 교과서, 판문점, 탈북인, 여경, 부동산

수집한 데이터에서 특수 기호, 기자 이름, 날짜 등의 텍스트를 제거하고 구두점으로 끝나는 문장들로만 뉴스 기사의 본문을 구성하도록 정제하여 데이터를 구축하였다. 구축한 데이터의 보수 및 진보 성향 기사의 문장 수와 어절 수는 표 1과 같다. 보수 성향과 진보 성향의 데이터 크기는 이진 분류를 위해 동일한 크기로 사용하여 데이터의 균형을 맞추었다. 형태소 분석기를 이용한 토크나이저 별 중복을 제거한 토큰의 수는 표 2와 같으며, 학습 데이터와 훈련 데이터는 8:2의 비율로 분할하여 모델을 학습하고 자동 분류 정확도를 평가하였다.

3.2 평가 및 결과

뉴스 기사의 본문 내용을 형태소 분석기를 통해 형태소 단위로 토큰화하고, 벡터 크기를 각각 100, 200, 300, 400, 500으로 생성한 후에 100, 200, 300의 반복 횟수로 학습하여 토큰 벡터를 구성하였다. 문서를 구성하고 있는 토큰들의 벡터를 평균값으로 문서 벡터를 구성하고 이를 SVM 분류 모델을 통해 실험하였으며 정확도는 표 3과 같다.

SVM을 사용한 자동 분류 결과, 표 4와 같이 벡터 크기 400, 반복 횟수 200으로 Okt 형태소 분석기를 사용하였을 때 정확도 75.7%로 가장 높은 성능을 보였다. 또한, 정치적 편향성의 자동 분류 실험 결과로 전반적으로 Okt와 Komoran은 비슷한 성능을 보였으며, Hannanum의 성능이 가장 낮게 나타났다.

4. 결론

한국어의 교착어 특성을 고려하여 어절을 형태소 단위의 토큰열로 분할하여 각 형태소 분석기 별 성능 비교 실험을 진행하였다. 정치적 성향의 편향성을 분류하기 위해 정치 키워드를 기반으로 검색된 뉴스 기사를 수집하여 데이터를 구축하였으며, 구축한 데이터를 활용하여 형태소 단위 토큰화를 진행하였다. 형태소 토큰을 사용한 머신러닝 기법으로 SVM 모델을 사용하였으며, 형태소 분석기 별 정확도에 따른 성능을 비교하였다. SVM 모델을 사용한 정치적 편향성의 자동 분류 실험에서 Okt의 형태소 분석기를 사용했을 때 가장 높은 성능을 나타냈다.

Acknowledgements

이 논문은 2019년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임(NRF-2019S1A5A2A03046571).

참고문헌

[1] Santos. C. D. & Gatti. M. “Deep convolutional neural networks for sentiment analysis of short texts,” Proceedings of COLING 2014, the 25th International Conferernce on Computational Linguistics: Technical Papers, pp.69-78, 2014.

[2] Kim. Y. "Convolutional neural networks for sentence classification." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP), pp.1746-1751,

2014.
[3] Mikolov. T., Chen. K., Corrado. G., & Dean. J. “Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781, 2013.
[4] Mikolov. T., Sutskever. I., Chen. K., Corrado. G. S., & Dean. J. “Distributed representations of words and phrases and their compositionality.” Advances in Neural Information Processing Systems, pp.3111-3119, 2013.
[5] 이홍식, “형태소와 문법 기술,” 어문학 109호, pp.1-35, 2010.
[6] 이동준, 임유빈, 권태경, “형태소 기반 효율적인 한국어 단어 임베딩,” 정보과학회논문지, 45권 5호, pp.444-450, 2018.
[7] Pennington. J., Socher. R., & Manning. C. D, “Glove: Global vectors for word representation,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing(EMNLP), pp.1532-1543, 2014.
[8] Baroni. M., Dinu. G. & Kruszewski. G., “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors,” Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Vol.1: Long Papers, pp.238-247, 2014.
[9] Y. Goldberg and O. Levy, “word2vec Explained: deriving Mikolov et al.‘s negative-sampling word-embedding method,” arXiv preprint arXiv:1402.3722, 2014.
[10] Joachims. T. *Learning to classify text using support vector machines*, Springer Science & Business Media, 2002.