감성 분석을 위한 어휘 통합 합성곱 신경망에 관한 연구

윤주성*, 김현철*
*고려대학교 컴퓨터학과
e-mail : adullam705@gmail.com

A Study on Lexicon Integrated Convolutional Neural Networks for Sentiment Analysis

Joo-Sung Yoon*, Hyeon-Cheol Kim*
*Dept. of Computer Science and Engineering, Korea University

요 약

최근 딥러닝의 발달로 인해 Sentiment analysis분야에서도 다양한 기법들이 적용되고 있다. 이미지, 음성인식 분야에서 높은 성능을 보여주었던 Convolutional Neural Networks (CNN)은 최근 자연어처리 분야에서도 활발하게 연구가 진행되고 있으며 Sentiment analysis에도 효과적인 것으로 알려져 있다. 기존의 머신러닝에서는 lexicon을 이용한 기법들이 활발하게 연구되었지만 word embedding이 등장하면서 이러한 시도가 점차 줄어들게 되었다. 그러나 lexicon은 여전히 sentiment analysis에서 유용한 정보를 제공한다. 본 연구에서는 SemEval 2017 Task4에서 제공한 Twitter dataset과 다양한 lexicon corpus를 사용하여 lexicon을 CNN과 결합하였을 때 모델의 성능이 얼마큼 향상되는지에 대하여 연구하였다. 또한 word embedding과 lexicon이 미치는 영향에 대하여 분석하였다. 모델을 평가하는 metric은 positive, negative, neutral 3가지 class에 대한 macroaveraged F1 score를 사용하였다.

1. 서론

최근 딥러닝의 발달로 인해 자연어처리 분야에서도 딥러닝을 활용한 연구가 활발하게 진행되고 있다. 자연어 처리의 분야 중 한 분야인 감성 분석은 텍스트 안에 담겨진 내용이 긍정인지 부정인지 판단하는 것을 말한다. 최근 발달된 Twitter와 같은 Social media를 통해 많은 사람들이 의견을 공유하고 있으며 이러한 의견이 담긴 데이터는 금융, 제품에 대한 시장 반응 분석 등 다양한 영역에서 활용되고 있다. 기존에는 Naïve Bayes, SVM등의 방법들이 사용하여 분석하였지만 딥러닝의 발달로 인해 Recurrent Neural Network, Recursive Neural Network등의 방법 또한 사용되었다. 최근 제안된 Convolutional Neural Networks (CNN)은 간단한 구조에도 불구하고 뛰어난 성능을 보여주었으며 컴퓨터 비전, 음성인식뿐 만 아니라 자연어처리 연구에서도 많은 연구가 이루어지고 있다. 본 연구에서는 이러한 CNN모델과 Lexicon을 함께 사용하여 SemEval-2017 Task4에서 제공한 Tweets Dataset에 대해 Positive, Negative, Neutral 3가지 감성 분석을 수행하였다. 우리 본 연구의 Contribution은 감성분석을 위한 새로운 구조의 모델을 제안하고 lexicon이 감성분석에 끼치는 영향 측정하여 lexicon의 CNN 모델에서 중요한 feature로 사용 될 수 있음을 알아낸 것이다.

2. 모델

2.1 Word embedding

본 모델에서는 word2vec을 skip-gram과 negative sampling을 이용하여 구현하였으며 Sentiment140 dataset으로부터 얻은 160M개의 tweets corpus를 통해 각각의 다른 차원 (50, 100, 200, 400)의 word embedding을 얻었다.

2.2 Lexicon embedding

Lexicon은 Word embedding이 syntactic과 semantic feature를 표현해주었기 때문에 상대적으로 덜 사용되었지만 여전히 의미 있는 feature를 표현해주기 때문에 Lexicon embedding을 본 모델에서 사용하게 되었다. Lexicon embedding은 어휘에 따른 Sentiment score의 vector로 구성되었으며, Sentiment score는 -1부터 +1까지의 범위를 가진다. -1은 negative, 0은 neutral, +1은 positive score를 의미한다.

2.3 Word and lexicon embedding layer

모델의 Input data 형태는 단어로 구성된 document이다. Document는 각 단어는 vector로 나타내며 이때 document matrix는 $s \in \mathbb{R}^{n \times d}$로 정의한다. 이때 d는 Word embedding의 차원을 n은 문서내의 단어 개수를 의미한

다. 각 단어는 hand-crafted feature가 아닌 word2vec으로부터 distributed representation을 가진 vector로 표현되었다. Lexicon은 sentiment score에 대한 정보를 담고 있으며 lexicon으로 구성된 document는 $s_l \in \mathbb{R}^{n \times e}$로 정의한다. 이때 e는 Lexicon embedding의 차원을 나타내며 lexicon corpus의 개수로 정의한다.

2.4 Convolutional neural networks

CNN의 기본적인 구조는 Kim (2014)이 제안한 모델을 변형해서 사용했다. 본 모델은 2-layer CNN으로 구성되어있으며 max-pooling layer와 concatenation layer를 통해서 최종 feature를 추출하였다. 본 모델의 구조는 실험을 통해 결과 값을 고려하여 결정하였다. CNN의 각 filter는 document matrix s의 feature들을 추출하게 되며 이때 filter는 $c \in \mathbb{R}^{l \times d}$로 나타낸다. l은 filter의 길이를 나타낸다. feature를 추출할 때는 multichannel 방법을 사용하여 첫 번째 document matrix channel은 학습된 word2vec을 고정하여 사용하였고 나머지 document matrix는 학습을 통해 embedding이 변할 수 있게 하였다. Lexicon embedding에 대해서 feature를 추출할 때는 Shin(2016)이 제안한 separate convolution 방법을 사용하였다. 각 convolutional layer를 통과한 값에 대한 Activation function은 ReLU를 사용하였다.

2.5 Concatenation layer

1-layer CNN에서는 low level feature를 얻을 수 있으며 여기서 나온 결과 값 또한 추가적인 feature로 사용될 수 있으므로 1-layer CNN의 결과 값을 사용하였다. 또한 2-layer CNN의 결과 값은 상대적으로 high level feature를 얻을 수 있으므로 사용하였고 lexicon으로부터 얻은 결과 값까지 합쳐서 하나의 vector로 재구성하였다. Concatenation layer은 $D_{concat} \in \mathbb{R}^{3n}$로 나타낸다.

$$D_{concat} = X_{1layer\,CNN} \oplus X_{2layer\,CNN} \oplus X_{lexicon} \quad (1)$$

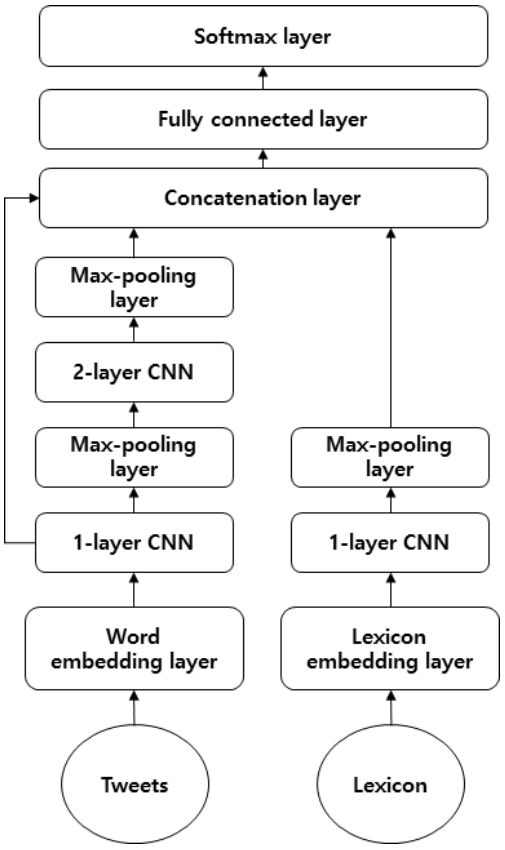
여기서 $\oplus$은 concatenation operator를 의미하며 $X_{1layer\,CNN}$과 $X_{2layer\,CNN}$은 tweet데이터에 대해서 CNN의 각 1-layer, 2-layer를 지난 후 max-pooling한 결과를 의미하고 $X_{lexicon}$은 lexicon데이터에 대해서 CNN의 1-layer를 통과한 후 max-pooling된 결과값을 의미한다.

2.6 Fully connected layer

Concatenation layer에서 합쳐진 feature들 분류하기 위해 fully connected(FC) layer를 사용하였다. 본 layer에 사용된 weight는 $W_{fc} \in \mathbb{R}^{D_{concat} \times c}$로 표현되며 bias는 $b_{fc} \in \mathbb{R}$로 표현된다. 이 때, c는 class 개수를 의미한다.

2.7 Softmax layer

모델은 tweet에 대해 3가지 class (positive, negative, neutral)로 분류하며 FC layer의 결과 값을 classification 확률로 변환하기 위해 softmax function을 사용하였다.



(그림 1) 모델의 전체 구조

2.8 Regularization

본 모델에 대한 Overffiting을 막기 위해 CNN의 결과 값과 FC layer에 대해 dropout을 사용하였으며 각 노드를 50%의 확률로 제거했다. 또한 L_2 normalization을 적용하였으며 cost function에 대해 $\lambda \| \theta \|_2^2$의 term을 추가했다. 이 때, λ는 regularization의 영향을 결정하는 parameter이며 $\theta \in \Theta$는 FC layer의 parameter를 의미하며 CNN layer의 parameter를 사용할 경우 학습 과정이 불안정하였기 때문에 제외하였다.

3. 실험 및 분석

본 모델은 SemEval 2017 Task4에서 제공한 데이터 셋을 통해 실험 및 분석을 진행하였으며 평가를 위해 Positive, negative, neutral 3가지 class에 대한 macroaveraged F1 measure를 사용하였다.

3.1 Dataset

3.1.1 Tweets

본 모델에서는 SemEval 2017에서 2013년부터 2016년도의 Twitter로부터 얻은 영어로 된 training set과 development set을 사용하였다. 추가로 word embedding을 training시키기 위해 sentiment140 corpus를 사용하였다.

Corpus	Total	Positive	Negative	Neutral
Train 2013	9,684	3,640	1,458	4,586
Dev 2013	1,654	575	340	739
Train 2015	489	170	66	253
Train 2016	6,000	3,094	863	2,043
Dev 2016	1,999	843	391	765
DevTest 2016	2,000	994	325	681
Test 2013	3,547	1,475	559	1,513
Test 2014	1,853	982	202	669
Test 2015	2,390	1,038	365	987
Test 2016	20,632	7,059	3,231	10,342
TwtSarc 2014	86	33	40	13
SMS 2013	2,094	492	394	1,208
LiveJournal 2014	1,142	427	304	411

<표 1> Overview of datasets

3.1.2 Lexicons

Sentiment score를 포함하고 있는 총 7가지 Lexicon을 사용하였다. 단순히 positive, negative로만 되어있는 lexicon의 경우 +1, -1로 변환하였으며 -1부터 +1까지의 구간을 넘어가는 스케일을 가진 score의 경우 -1과 +1사이로 정규화 하였다. 추가로 lexicon에 포함되지 않은 단어의 경우 neutral score로 지정하여 0을 부여하였다.

- SemEval-2015 English Twitter Sentiment Lexicon (2015).
- National Research Council Canada (NRC) Hashtag Affirmative and Negated Context Sentiment Lexicon (2014).
- NRC Sentiment140 Lexicon (2014).
- Yelp Restaurant Sentiment Lexicons (2014).
- NRC Hashtag Sentiment Lexicon (2013).
- Bing Liu Opinion Lexicon (2004).

3.2 Preprocessing

Dataset에 속한 모든 tweets과 lexicon에 대해 다음과 같이 전처리를 진행하였다.

- Lowercase: tweet 및 lexicon내의 모든 문자에 대해서 소문자로 변환
- Tokenization: NLTK의 Tweet tokenizer를 이용하여 모든 tweet에 대해 토큰화 진행
- Cleaning: URLs 또는 hashtag내의 '#' 토큰을 제거하여 representation의 sparseness 제거

3.3 Training and hyperparameters

Adam optimizer를 통해 본 모델을 훈련시켰으며 exponential decay 방법을 사용해서 learning rate를 조절했다. Training에 사용된 configuration은 다음과 같다.

- Embedding dimension = (50, 100, 200, 400)
- Filter size = (2,3,4,5,6)
- Number of filters = (128)
- Batch size = (64)
- Number of epochs = (80)
- Starter learning rate = (0.0001)
- Exponential decay steps and rate = (3000, 0.96)
- Dropout rate = (0.5)
- Regularization lambda = (0.005)

	d(50)	*d*(100)	*d*(200)	*d*(400)
F1 score	0.6065	**0.6097**	0.594	0.5841

<표 2> word embedding의 차원과 F1 score

Model	Twt2013	Twt2014	Twt2015	Twt2016
All features	0.6116	**0.6202**	0.6109	0.6194
w/o lexicon	0.5453	0.5741	0.5458	0.5883
w/o WE	0.5872	0.5825	0.5811	0.5810
w/o both	0.5317	0.5392	0.5349	0.5584

<표 3> Test set에 대한 F1 score

4. 결과

실험결과 word embedding과 lexicon feature가 모델의 성능을 향상시킬 수 있음을 확인했다. 표2에서 보여주는 것과 같이 word embedding의 차원은 성능에 영향을 끼침을 알 수 있으며 실험에서는 100차원일 때 가장 좋은 성능을 얻을 수 있었다. 표3에서는 lexicon에 대한 feature가 적용될 때 word embedding보다 전체적으로 더 높은 성능을 보여주었기에 더욱 중요한 feature로 작용할 수

있음을 보여주었다. lexicon feature의 경우 없는 word에 대해 neutral score를 부여한다는 한계에도 불구하고 여전히 sentiment analysis에서 유용하고 필수적인 feature로 사용될 수 있음을 확인 할 수 있었다.

5. 결론

본 논문에서는 lexicon과 CNN을 결합하여 sentiment analysis에 적용한 모델에 대하여 기술하였다. 본 모델에서는 document에 대한 더 나은 representation을 얻기 위해 1-layer CNN과 2-layer CNN의 결과 값을 모두 고려하였고 multichannel을 적용함으로써 data의 manifold를 더 잘 예측하려는 시도를 하였다. Lexicon과 word embedding의 feature는 sentiment analysis에 매우 중요함을 확인하였으며 특히 lexicon이 여전히 중요한 역할을 할 수 있음을 표3을 통해 확인하였다. 본 모델의 성능은 lexicon의 coverage를 높이거나, Attention mechanism 또는 CNN모델에 대한 ensemble 기법 등을 적용할 때 더욱 개선될 수 있을 것으로 보인다.

참고문헌

[1] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP 2014 – Empirical Methods in Natural Language Processing*, pages 1746 – 1751.

[2] Bonggun Shin, Timothy Lee, and Jinho D. Choi. 2016. Lexicon Integrated CNN Models with Attention for Sentiment Analysis. arXiv preprint arXiv:1610.06272.

[3] Jan, Deriu, et al. 2016. Sentiment classification using an ensemble of convolutional neural networks with distant supervision. *Proceedings of SemEval (2016)*: 1124-1128.

[4] XingYi, Xu, Liang HuiZhi , and Baldwin Timothy. An Ensemble of Neural Networks and a Word2Vec Based Model for Sentiment Classification. *Proceedings of SemEval (2016)*: 183-189.

[5] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (Vol. 1631, p. 1642).

[6] Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075.*

[7] Nakov, P., Rosenthal, S., Kiritchenko, S., Mohammad, S. M., Kozareva, Z., Ritter, A., ... & Zhu, X. (2016). Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. *Language Resources and Evaluation, 50*(1), 35-65.

[8] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820.*

[9] Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, 72, 221-230.

[10] Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242.*

[11] Kokkinos, F., & Potamianos, A. (2017). Structural Attention Neural Networks for improved sentiment analysis. *arXiv preprint arXiv:1701.01811.*

[12] Keshavarz, H., & Abadeh, M. S. (2017). ALGA: Adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs. *Knowledge-Based Systems, 122*, 1-16.

감사의 글

" 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017R1A2B4003558)."