Adullam at SemEval-2017 Task 4: Sentiment Analyzer based on Lexicon Integrated Convolutional Neural Networks with Attention

Joosung Yoon
Korea University
Seoul, South Korea
xelloss705@gmail.com

Kigon Lyu
Korea University
Seoul, South Korea
gon0121@korea.ac.kr

Hyeoncheol Kim
Korea University
Seoul, South Korea
hkim64@gmail.com

Abstract

We propose a sentiment analyzer for the prediction of document-level sentiments of English micro-blog messages from Twitter. The proposed method is based on lexicon integrated convolutional neural networks with attention (LCA). Its performance was evaluated using the datasets provided by SemEval competition (Task 4). The proposed sentiment analyzer obtained an average F1 of 55.2%, an average recall of 58.9% and an accuracy of 61.4%.

1 Introduction

Sentiment analysis is necessary to interpret the vast number of online opinions on social media platforms such as Twitter. This will allow governments and corporations to manage public relations and policies effectively. Existing sentiment analyzers are based on naïve bayes, SVM, RNN (Irsoy, 2014) and in particular convolutional neural networks (CNNs) (Kim, 2014).

In order to improve on existing CNN based sentiment analyzer, lexicon embedding and attention embedding were integrated into the proposed sentiment analyzer. Lexicon embedding allows extraction of sentimental score for each word and attention embedding enables the global view of the sentence.

The proposed LCA was both trained and evaluated using corpus from Twitter 2013 to 2016 provided by the SemEval-2017. Figure 1 shows the overview of the proposed sentiment analyzer. It consists of embedding, CNNs, concatenation, fully connected and softmax layer.

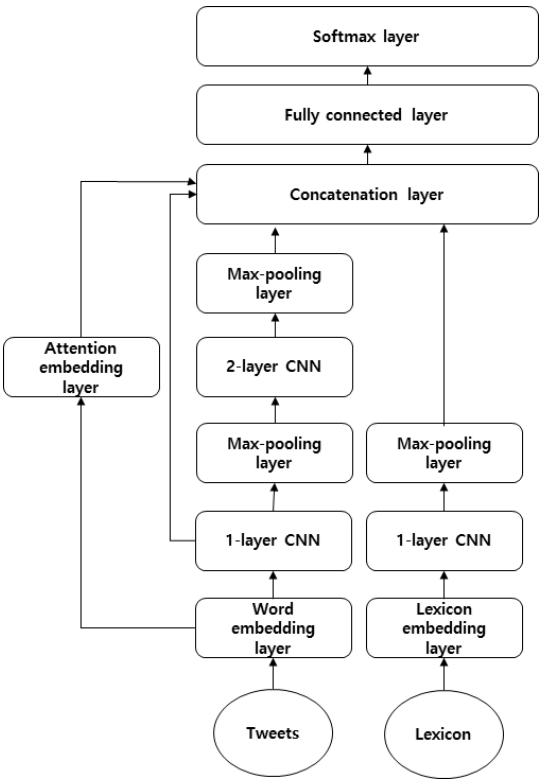

Figure 1: Architecture of proposed LCA.

2 Input Features & Architecture

The proposed LCA consists of three input features

(i) Word embeddings
(ii) Lexicon embeddings
(iii) Attention embeddings.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 732–736,
Vancouver, Canada, August 3 - 4, 2017.

Word embeddings are trained by implementation of word2vec using skip-gram (Mikolov, 2014) and negative sampling. The word embeddings are trained using an unlabeled corpus of 1.6M tweets from Sentiment 140 dataset with different dimensions (50, 100, 200, 400). The dimensions of word embeddings are d and the number of words in a document is n

Lexicon embeddings are considered because they are useful features. Lexicon embeddings consist of set of words each paired with a score ranging from -1 to +1. Where a score of -1 represents a negative sentiment and +1, a positive sentiment. The lexicon document corresponding to each word is $s_l \in \mathbb{R}^{n \times e}$, where e is the dimension of lexicon embeddings and it is set by the number of lexicon corpus.

Attention embeddings are important for Deep Learning in terms of performance and explanation of models (Kelvin, 2015). CNN uses several filters which have length l. It considers l-gram features, but it only takes local views into account not considering the global view of sentence. Sentiment analysis must consider transitional cases such as negation. While attention embeddings can capture keywords to improve sentiment analysis, it also considers the global view of sentence. In order to do so, CNN for attention embedding used 1 as the length of filter. Then, it executes max pooling for each row of attention matrix. The output of max pooing is an attention vector which has probabilities assigned to each word vector that has d-dimension.

The architecture of LCA consists of

(i) a word and lexicon embedding layer,
(ii) CNNs,
(iii) a concatenation layer,
(iv) a fully connected layer
(v) and a softmax layer.

Word and lexicon embedding layer transform input data into vector representation. The input to our model is a document, treated as a sequence of words. Instead of hand-crafted features, we used word2vec (w2v) to represent words to vectors. We also converted lexicons to vectors, containing sentiment score. The Input document matrix is $s \in \mathbb{R}^{n \times d}$ where n is the number of words in a document.

Convolutional neural networks are effective for extracting high level features. We modified the LCA architecture of Shin (2016). The proposed LCA consists of two layer CNNs with a nonlinearity, max pooling layers, a concatenation layer and a softmax classification layer with respect to the word embedding layer. The architecture of the proposed LCA was chosen empirically. The document matrix s is convolved by the filter $c \in \mathbb{R}^{l \times d}$, where l is the length of filters. In convolving lexicon embeddings by the filter, we used the separate convolution approach of Shin (2016).

Concatenation layer consists of 1-layer CNN, 2-layer CNN, lexicon and attention outputs. We deliberately designed our model so that the output of 1-layer CNN captures low level feature for getting additional information. The dimension of concatenation layer is $D_{concat} \in \mathbb{R}^{2m+d \times n_l}$, where m is the number of filters with the same length and n_l is the number of filters with different lengths.

Fully connected layer (FC) is used to create nonlinear combinations with rectified linear unit (ReLU) (Nair and Hinton, 2010). The input of fully connected layer is the output of concatenation layer. The dimension of weight is $W_{fc} \in \mathbb{R}^{D_{concat} \times n_c}$ and bias is $b_{fc} \in \mathbb{R}^{n_c}$, where n_c is the number of class.

Softmax layer is used to convert the output of FC layer into classification probabilities. In order to compute the probabilities, softmax function was used:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}}$$

The output dimension is 3 because our model classified tweets into 3 classes (positive, neutral and negative).

Regularization is achieved by L_2 regularizer. In order to prevent overfitting from our CNN model, dropout is used at the output of CNN and fully connected layer. To do this, each node is randomly removed. We also apply L_2 regularization to the cost function by adding the term $\lambda \|\theta\|_2^2$, where λ

is the regularization strength and $\theta \in \Theta$ are the fully connected neural network parameters.

3. Data and Preprocessing

Tweets are used as the training and development dataset from Twitter 2013 to 2016 (The training and development dataset were provided by the SemEval-2017 competition.) In addition, sentiment 140 corpus are added for training word embedding.

Lexicons in the proposed LCA have six types of sentiment lexicons (that include sentimental score). Some lexicons only contain positive and negative sentiment polarities. Sentiment scores were normalized to the range from -1 to +1 because some lexicons have different scales. If some words are missing in a lexicon, we assigned neutral sentiment score of 0.

- 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).

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	6	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

Table 1: Overview of datasets

The following preprocessings were applied to every tweets and lexicon in the corpus:

- Lowercase: all the characters in tweets and lexicons are converted in lowercase.
- Tokenization: all tweets were tokenized using tokenizer.
- Cleaning: URLs and '#' token in hashtag were removed to reduce sparse representation.

4. Training and Hyperparameters

The parameters of our model were trained by *Adam* (Diederik et al., 2014) optimizer. To anneal the learning rate over time, the learning rate were calculated by exponential decay. The following configuration is our hyperparameters:

- Embedding dimension = (50, 100, 200, 400) for both word and attention embeddings.
- Filter size = (2,3,4,5,6) for capturing more n-gram features.
- Number of filters = (128) for convolving the document matrix s combined with lexicon and attention embeddings.
- Batch size = (64) for calculating losses to update weight parameters.
- Number of epochs = (80) for training our models.
- Starter learning rate = (0.0001) for updating weight parameters.
- Exponential decay steps and rate = (3000, 0.96) for annealing the learning rate.
- Dropout rate = (0.5) for avoiding overfitting from the last layer of CNN and FC layer
- L_2 Regularization lambda = (0.005) for avoiding overfitting from FC layer

5. Evaluation

The evaluation metric consisted of

(i) macro-averaged F1 measure,
(ii) recall
(iii) and accuracy in the competition across the positive, negative and neutral classes.

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

Table 2: F1 scores corresponding to the dimension of word embedding

Model	**Twt2013**	**Twt2014**	**Twt2015**	**Twt2016**
All features	0.6116	**0.6202**	0.6109	0.6194
w/o lexicon	0.5460	0.5501	0.5414	0.5682
w/o w2v	0.5872	0.5825	0.5811	0.5810
w/o both	0.5256	0.5409	0.5187	0.5327

Table 3: Test score of F1

6. Results

The result of competition showed that our model was overfitting because our experimental results were higher than the actual result. In our experiment, lexicon and word embedding feature showed that it could improve our model. Table 2 presents the various dimensions of word embeddings that could change performance which is high when the dimension of word embedding is 100. Table 3 shows lexicons as the feature more important than word2vec because the overall performances of model with lexicon were higher than the overall performance with word2vec. Since the sentiment score of missing words (such as 0; neutral) has been replaced, the lexicon feature is not perfect. Nonetheless, lexicon is still an important and essential feature for sentiment analysis.

7. Conclusion

This paper proposes the integration of lexicon with attention on CNN as an approach to sentiment analysis. We considered various features to capture improved representations by concatenating the output of 1-layer and 2-layer CNN. Lexicon and word embedding showed that these features improved the model performance significantly.

Additional enhancements are viable by gathering more training dataset or lexicon dataset with distant supervision (Deriu et al, 2016), because it will extend the coverage of our model. Furthermore, in the aspect of models, the combined CNN-CRF model, recursive neural network and ensembles of multi-layer CNN can be applied.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B4003558).

References

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

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

Mickael Rouvier, Benoit Favre. 2016. Polarity embedding fusion for robust sentiment analysis. *Proceedings of SemEval (2016)*: 202-208.

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

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.

Kingma, Diederik, and Ba Jimmy. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980 (2014).*

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified lin- ear units improve restricted boltzmann machines. In *ICML 2010 - Proceedings of the 27th International Conference on Machine Learning*, pages 807–814.

Tomas, Mikolov, et al. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems. 2013.*

Kelvin, Xu, et al. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML. Vol. 14. 2015.*

rsoy, Ozan, and Cardie Claire. 2014. Opinion Mining with Deep Recurrent Neural Networks. In *EMNLP 2014 - Empirical Methods in Natural Language Processing*, pages 720-728.

Socher, Richard, et al. 2013. 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. 2013.*