The 2017 Conference on Computational Linguistics and Speech Processing
ROCLING 2017, pp. 244-253

Multi-Channel Lexicon Integrated CNN-BiLSTM Models for Sentiment Analysis

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

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

Abstract

We improved sentiment classifier for predicting document-level sentiments from Twitter by using multi-channel lexicon embedidngs. The core of the architecture is based on CNN-BiLSTM that can capture high level features and long term dependency in documents. We also applied multi-channel method on lexicon to improve lexicon features. The macro-averaged F1 score of our model outperformed other classifiers in this paper by 1-4%. Our model achieved F1 score of 64% in SemEval Task 4 (2013-2016) datasets when multi-channel lexicon embedding was applied with 100 dimensions of word embedding.

Keywords: Deep Learning, Lexicon, Multi-Channel, CNN-BiLSTM, Sentiment analysis

1. Introduction

Sentiment analysis, known as opinion mining is a task of natural language processing (NLP) aimed to identify sentiment polarities expressed in documents. Numerous amounts of opinioned texts are created on social media every day. For instance, Twitter users generate over 500 million tweets daily. It is important to analyze these opinioned texts because they give useful information such as response for specific product, opinion for candidates and etc.

However, in sentiment analysis, sarcasm is difficult to distinguish. Usually, sentiment classifier can identify polarity better in the case of clear expression than in the case of sarcasm. Contextualization and informal language in social media are additional complicating factors to sentiment classifier (Deriu et al, 2017).

To solve this problem, our approach focuses on high level features of document extracted by

CNN and the context considered by BiLSTM that capture long term dependency which helps to understand the context. Therefore, we propose a Multi-Channel Lexicon Integrated CNN-BiLSTM (MCLICB) model for sentiment analysis.

Our contributions are:

(i) To improve performance of sentiment classifier

(ii) To introduce multi-channel lexicon embeddings and analyze influence for sentiment analysis.

2. Related Works

The first success of sentiment analysis based on convolutional neural networks (CNN) was triggered by text classification (Kim, 2014). This work provided simple and effective architecture for text classification. Convolutional layer can extract local *n*-gram features. After this research, various modified models based on CNN have been proposed.

One of the modified models is lexicon integrated CNN model with attention (Shin and Lee and Choi, 2016). In the traditional setting, where statistical models are based on hand-crafted features, lexicon is a useful feature, consisting of words and their sentiment scores. CNN architecture of Shin showed that lexicon embedding still can be a useful feature for sentiment analysis.

CNN based methods have been successful in many NLP tasks. However, it has limitations in respect of long term dependency. In contrast, Long Short-Term Memory (LSTM) (Hochreiter et al., 1997; Tai et al., 2015) can capture semantic information with long term dependency.

In order to consider local *n*-gram features and long term dependency, various models which combined both CNN and LSTM were proposed (Zhang, 2017). Our model improves this approach.

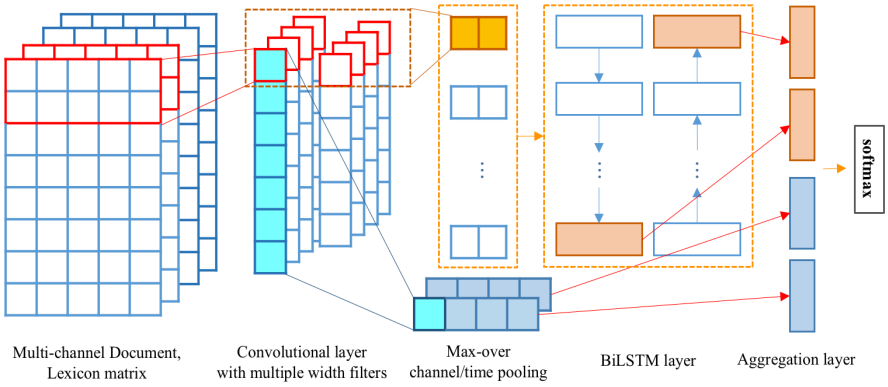


Figure 1: The architecture of our model

3. MCLICB

The architecture of MCLICB consists of a multi-channel embedding layer, a CNN-BiLSTM layer, an aggregation layer, and softmax layer.

3.1 Multi-channel embedding layer

The input of our model (document, lexicon matrix) are based on two multi-channels:

(i) Multi-channel word embedding,

(ii) Multi-channel lexicon embedding.

Multi-channel word embedding is the same as the architecture of Kim (2014) which is both static and non-static. We used word2vec (w2v) trained by skip-gram (Mikolov, 2013). In the similar manner, we applied multi-channel method on lexicon to improve lexicon feature for sentiment analysis. As the coverage of lexicon is low, multi-channel method is more useful because it resolves sparseness in lexicon embedding. The word document matrix is $s \in \mathbb{R}^{n \times d}$, where n is the number of words in a document and d is the dimension of word embedding. The lexicon document corresponding to each word in a document is $s_l \in \mathbb{R}^{n \times e}$, where e is the dimension of lexicon embedding determined by the number of lexicon corpus in section 4.2.

3.2 CNN-BiLSTM layer

To combine advantages of CNN and LSTM, the input local n-gram features were extracted by

CNN. We added padding to the output of CNN because different size of filters produced different size of feature map. Then, max pooling over channels was applied to the padded output of CNN.

To consider long term dependency, bidirectional LSTM were applied to the output of max pooling layer. We set the hidden size h as 150 for all BiLSTM layers. In the case of lexicon embedding, when multi-channel lexicon embedding was convolved by filters, separate convolution approach of Shin (2016) was used.

3.3 Aggregation layer

While LSTMs are advantageous for capturing long term dependency, CNNs generally outperformed in capturing high level features in short text.

To consider various document lengths, we concatenated the outputs of CNN which were produced by max pooling over time and the outputs of CNN-BiLSTM which were generated from last hidden states at aggregation layer. We used different filters between CNN and CNN-BiLSTM to capture improved representations.

3.4 Softmax layer

In softmax layer, the outputs of aggregation layer were converted into classification probabilities. In order to compute the classification probabilities, softmax function was used. The output dimension is 3 (positive, negative and neutral classes).

4. Experiments

In this section, we evaluated our model on sentiment analysis task. We first introduced the implementation of our model in section 4.1. Then, we demonstrated data, preprocessing, training and hyperparameters in section 4.2 and 4.3.

4.1 Implementation

To conduct experiments, we used PyTorch which can fully utilize the GPU computing resource to train our model. We trained our model on a single GTX 1080 8GB GPU with CUDA

(Nickolls et al., 2008) and cuDNN (Chetlur and Woolley, 2014).

4.2 Data and Preprocessing

Tweets which were provided by the SemEval-2017 competition were used for training and as test datasets. The training datasets were from Twitter 2013 to 2016 train/dev and the rest were the test datasets in Table 1.

Table 1. Overview of datasets

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

Lexicons used in the proposed model consist of eight types of sentiment lexicons which include sentiment score. Some lexicons were preprocessed to normalize sentiment score to the range from -1 to +1. If words are not in the lexicon vocabulary, neutral sentiment score of 0 were assigned. The following lexicons are used in our model:

- 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).
- Macquarie Semantic Orientation Lexicon (2009).

- NRC Word-Emotion Association Lexicon (2010).

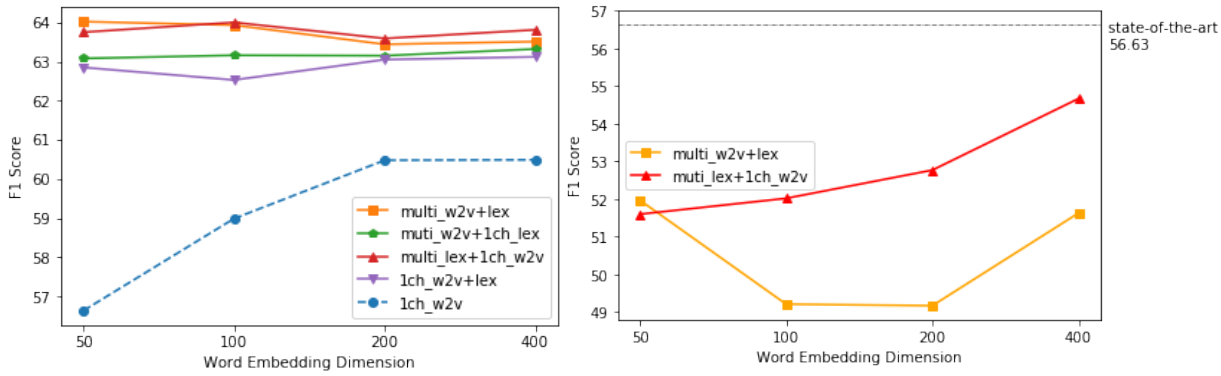
Preprocessing were applied to tweets and lexicon datasets before extracting features using the following procedures:

- Lowercase: characters in tweets and lexicons were converted to lowercase.
- Tokenization: all tweets were tokenized by using NLTK twitter tokenizer.
- Cleaning: URLs and '#' token in hashtags were removed.
- Replacement: for the out-of-vocabulary (OOV) words, they were replaced by <UNK> token.

4.3. Training and Hyperparameters

The parameters were trained by Adam optimizer (Diederik et al. 2014). The following configuration is our hyperparameters:

- Word embedding dimension d = (50, 100, 200, 400) for pre-trained word2vec.
- Lexicon embedding dimension e = (8) for considering lexicon features.
- Hidden size h = (150) for hidden states of BiLSTM.
- Filter size = (2, 3, 4, 5) for capturing n-gram features.
- Number of filters = (200) for convolving the document and lexicon matrix.
- Number of layers = (2) for number of BiLSTM layers.
- Batch size = (100) for calculating losses.



(a) Average F1 score of SemEval Task 2013-2016

(b) Twitter Sarcasm Task 2014

Figure 2: The performances of models change across various dimensions of word embedding. In general, as the dimensions of word embedding increase, the performances of multi-channel lexicon models are better than that of multi-channel word embedding (w2v) and lexicon embedding (lex).

- Learning rate = (0.0005) for updating the parameters.
- Number of epochs = (15) for training models.
- Dropout rate = (0.5, 0.65) for avoiding overfitting (Hinton et al., 2012).
- Regularization lambda = (0.0001) for avoiding overfitting.

5. Evaluation

To evaluate the performances of our models in comparison to other classification models, we used the evaluation metric as macro-averaged F1 score across the positive, negative and neutral classes. In our experiment, baseline is 1 layer CNN which is the architecture of Kim (2014) in Table 2.

Table 2. Overall macro-averaged F1 scores of models.
Best (second-best) results of models are highlighted in bold (underlined) face.

	Method	**Test 2013**	**Test 2014**	**Test 2015**	**Test 2016**	**Twt Sarc 2014**	**SMS 2013**	**LiveJournal 2014**
This Paper	1 layer CNN (baseline)	63.22	60.43	61.04	60.41	43.46	65.05	65.18
	1 layer CNN + lex	62.70	61.37	61.76	62.19	46.39	67.07	68.04
	2 layer CNN	61.71	61.84	61.17	60.16	51.20	64.35	66.96
	2 layer CNN + lex	62.63	63.75	61.65	61.91	49.82	67.17	67.99
	Our model	<u>66.59</u>	<u>64.92</u>	<u>62.50</u>	<u>62.53</u>	<u>51.97</u>	**69.55**	**70.08**
Deriu, et al., 2016	FS (state-of-the-art)	**70.01**	**71.55**	**67.05**	**63.30**	**56.63**	-	<u>69.51</u>

5. Results

Our model outperformed other classification models all as shown in Table 2. In the case of sarcasm, modifying embedding dimension and using multi-channel lexicon embedding alone improved our model about 3% which are shown in Figure 2 (b).

F1 score of our model based on multi-channel lexicon embedding was higher than that of our model based on 1 channel word embedding by about 4-7% as shown in Figure 2 (a). In our

experiments, our model achieved the highest F1 score when multi-channel lexicon embedding was applied with 100 dimensions of word embedding in Figure 2 (a).

5. Conclusion

In this paper, we improved our model based on CNN-BiLSTM architecture for predicting document-level sentiments with multi-channel embeddings. Our model outperformed other classifiers in this paper by 1-4%, confirming multi-channel lexicon embedding's effectiveness in improving the performance.

For future work, the application of attention mechanism (Xu, et al., 2015; Yang, et al., 2016), other word embedding method such as fastText (Joulin et al., 2016) and ensemble methods (Deriu, et al., 2016) can be applied to improve our model.

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

[1] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[2] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[3] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word–emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

[4] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catan- zaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," *arXiv preprint arXiv:1410.0759*, 2014.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their composi- tionality," in *Advances in neural*

information processing systems, 2013, pp. 3111–3119.

[6] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Je´gou, and T. Mikolov, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[7] S. Mohammad, C. Dunne, and B. Dorr, "Generating high-coverage semantic orientation lexicons from overtly marked words and a the- saurus," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 599–608.

[8] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification." in *HLT- NAACL*, 2016, pp. 1480–1489.

[9] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic rep- resentations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co- adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[11] J. Deriu, A. Lucchi, V. De Luca, A. Severyn, S. Mu¨ller, M. Cieliebak, T. Hofmann, and M. Jaggi, "Leveraging large amounts of weakly supervised data for multi-language sentiment classification," in *Pro- ceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1045–1052.

[12] B. Shin, T. Lee, and J. D. Choi, "Lexicon integrated cnn models with attention for sentiment analysis," *arXiv preprint arXiv:1610.06272*, 2016.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[15] S.M.Mohammad,S.Kiritchenko,andX.Zhu,"Nrc-canada:Building the state-of-the-art in sentiment analysis of tweets," *arXiv preprint arXiv:1308.6242*, 2013.

[16] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "Nrc-canada- 2014: Detecting aspects and sentiment in customer reviews." in *SemEval@ COLING*, 2014, pp. 437–442.

[17] J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with cuda," *Queue*, vol. 6, no. 2, pp. 40–53, 2008.

[18] S. Rosenthal, P. Nakov, S. Kiritchenko, S. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter." in *SemEval@ NAACL-HLT*, 2015, pp. 451–463.

[19] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.

[20] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[21] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, and M. Jaggi, "Swisscheese at semeval-2016 task 4: Sentiment classifica- tion using an ensemble of convolutional neural networks with distant supervision." in *SemEval@ NAACL-HLT*, 2016, pp. 1124–1128.

[22] H. Zhang, J. Wang, J. Zhang, and X. Zhang, "Ynu-hpcc at semeval 2017 task 4: Using a multi-channel cnn-lstm model for sentiment classification," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 796–801.