Coooolll: A Deep Learning System for Twitter Sentiment Classification*

Duyu Tang†, Furu Wei‡, Bing Qin†, Ting Liu†, Ming Zhou‡
†Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
‡Microsoft Research, Beijing, China
{dytang, qinb, tliu}@ir.hit.edu.cn
{fuwei, mingzhou}@microsoft.com

Abstract

In this paper, we develop a deep learning system for message-level Twitter sentiment classification. Among the 45 submitted systems including the SemEval 2013 participants, our system (**Coooolll**) is ranked 2nd on the Twitter2014 test set of SemEval 2014 Task 9. Coooolll is built in a supervised learning framework by concatenating the sentiment-specific word embedding (**SSWE**) features with the state-of-the-art hand-crafted features. We develop a neural network with hybrid loss function [1] to learn SSWE, which encodes the sentiment information of tweets in the continuous representation of words. To obtain large-scale training corpora, we train SSWE from 10M tweets collected by positive and negative emoticons, without any manual annotation. Our system can be easily re-implemented with the publicly available sentiment-specific word embedding.

1 Introduction

Twitter sentiment classification aims to classify the sentiment polarity of a tweet as positive, negative or neutral (Jiang et al., 2011; Hu et al., 2013; Dong et al., 2014). The majority of existing approaches follow Pang et al. (2002) and employ machine learning algorithms to build classifiers from tweets with manually annotated sentiment polarity. Under this direction, most studies focus on designing effective features to obtain better classification performance (Pang and Lee, 2008; Liu, 2012; Feldman, 2013). For example, Mohammad et al. (2013) implement diverse sentiment lexicons and a variety of hand-crafted features. To leverage massive tweets containing positive and negative emoticons for automatically feature learning, Tang et al. (2014) propose to learn sentiment-specific word embedding and Kalchbrenner et al. (2014) model sentence representation with Dynamic Convolutional Neural Network.

In this paper, we develop a deep learning system for Twitter sentiment classification. Firstly, we learn sentiment-specific word embedding (**SSWE**) (Tang et al., 2014), which encodes the sentiment information of text into the continuous representation of words (Mikolov et al., 2013; Sun et al., 2014). Afterwards, we concatenate the SSWE features with the state-of-the-art hand-crafted features (Mohammad et al., 2013), and build the sentiment classifier with the benchmark dataset from SemEval 2013 (Nakov et al., 2013). To learn SSWE, we develop a tailored neural network, which incorporates the supervision from sentiment polarity of tweets in the hybrid loss function. We learn SSWE from tweets, leveraging massive tweets with emoticons as distant-supervised corpora without any manual annotations.

We evaluate the deep learning system on the test set of Twitter Sentiment Analysis Track in SemEval 2014 [2]. Our system (**Coooolll**) is ranked 2nd on the Twitter2014 test set, along with the SemEval 2013 participants owning larger training data than us. The performance of only using SSWE as features is comparable to the state-of-the-art hand-crafted features (detailed in Table 3), which verifies the effectiveness of the sentiment-specific word embedding. We release the sentiment-specific word embedding learned

* This work was partly done when the first author was visiting Microsoft Research.

[1] This is one of the three sentiment-specific word embedding learning algorithms proposed in Tang et al. (2014).

[2] http://alt.qcri.org/semeval2014/task9/

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 208–212,
Dublin, Ireland, August 23-24, 2014.

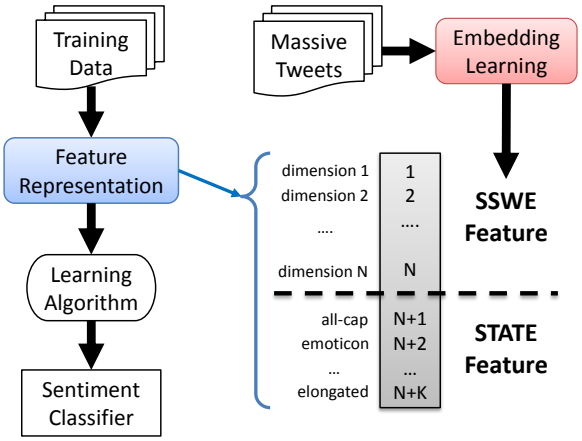

Figure 1: Our deep learning system (Coooolll) for Twitter sentiment classification.

from 10 million tweets, which can be easily used to re-implement our system and adopted off-the-shell in other sentiment analysis tasks.

2 A Deep Learning System

In this section, we present the details of our deep learning system for Twitter sentiment classification. As illustrated in Figure 1, Coooolll is a supervised learning method that builds the sentiment classifier from tweets with manually annotated sentiment polarity. In our system, the feature representation of tweet is composed of two parts, the sentiment-specific word embedding features (SSWE features) and the state-of-the-art hand-crafted features (STATE features). In the following parts, we introduce the SSWE features and STATE features, respectively.

2.1 SSWE Features

In this part, we first describe the neural network for learning sentiment-specific word embedding. Then, we generate the SSWE features of a tweet from the embedding of words it contains.

Our neural network is an extension of the traditional C&W model (Collobert et al., 2011), as illustrated in Figure 2. Unlike C&W model that learns word embedding by only modeling syntactic contexts of words, we develop **SSWE$_u$**, which captures the sentiment information of sentences as well as the syntactic contexts of words. Given an original (or corrupted) ngram and the sentiment polarity of a sentence as the input, SSWE$_u$ predicts a two-dimensional vector for each input ngram. The two scalars ($\boldsymbol{f}_0^u$, $\boldsymbol{f}_1^u$) stand for language model score and sentiment score of the input ngram, respectively. The training objectives of SSWE$_u$ are that (1) the original ngram should obtain a higher language model score $\boldsymbol{f}_0^u(t)$ than the corrupted ngram $\boldsymbol{f}_0^u(t^r)$, and (2) the sentiment score of original ngram $\boldsymbol{f}_1^u(t)$ should be more consistent with the gold polarity annotation of sentence than corrupted ngram $\boldsymbol{f}_1^u(t^r)$. The loss function of SSWE$_u$ is the linear combination of two hinge losses,

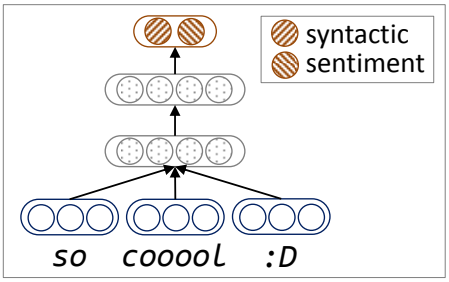

Figure 2: Our neural network (SSWE$_u$) for learning sentiment-specific word embedding.

$$loss_u(t, t^r) = \alpha \cdot loss_{cw}(t, t^r) + (1-\alpha) \cdot loss_{us}(t, t^r) \quad (1)$$

where where t is the original ngram, t^r is the corrupted ngram which is generated from t with middle word replaced by a randomly selected one, $loss_{cw}(t, t^r)$ is the syntactic loss as given in Equation 2, $loss_{us}(t, t^r)$ is the sentiment loss as described in Equation 3. The hyper-parameter α weighs the two parts.

$$loss_{cw}(t, t^r) = max(0, 1 - f^{cw}(t) + f^{cw}(t^r)) \quad (2)$$

$$loss_{us}(t, t^r) = max(0, 1 - \delta_s(t)\boldsymbol{f}_1^u(t) + \delta_s(t)\boldsymbol{f}_1^u(t^r)\ ) \quad (3)$$

where $\delta_s(t)$ is an indicator function reflecting the sentiment polarity of a sentence, whose value is 1 if the sentiment polarity of tweet t is positive and -1 if t's polarity is negative. We train sentiment-specific word embedding from 10M tweets collected with positive and negative emoticons (Hu et al., 2013). The details of training phase are described in Tang et al. (2014).

After finish learning SSWE, we explore min, $average$ and max convolutional layers (Collobert et al., 2011; Socher et al., 2011; Mitchell and Lapata, 2010), to obtain the tweet representation. The result is the concatenation of vectors derived from different convolutional layers.

2.2 STATE Features

We re-implement the state-of-the-art hand-crafted features (Mohammad et al., 2013) for Twitter sentiment classification. The STATE features are described below.

- *All-Caps*. The number of words with all characters in upper case.
- *Emoticons*. We use the presence of positive (or negative) emoticons and whether the last unit of a segmentation is emoticon [3].
- *Elongated Units*. The number of elongated words (with one character repeated more than two times), such as *gooood*.
- *Sentiment Lexicon*. We utilize several sentiment lexicons [4] to generate features. We explore the number of sentiment words, the score of last sentiment words, the total sentiment score and the maximal sentiment score for each lexicon.
- *Negation*. The number of individual negations [5] within a tweet.
- *Punctuation*. The number of contiguous sequences of dot, question mark and exclamation mark.
- *Cluster*. The presence of words from each of the 1,000 clusters from the Twitter NLP tool (Gimpel et al., 2011).
- *Ngrams*. The presence of word ngrams (1-4) and character ngrams (3-5).

3 Experiments

We evaluate our deep learning system by applying it for Twitter sentiment classification within a supervised learning framework. We conduct experiments on both positive/negative/neutral and positive/negative classification of tweets.

[3]We use the positive and negative emoticons from SentiStrength, available at http://sentistrength.wlv.ac.uk/.

[4]*HL* (Hu and Liu, 2004), *MPQA* (Wilson et al., 2005), *NRC_Emotion* (Mohammad and Turney, 2013), *NRC_Hashtag* and *Sentiment140Lexicon* (Mohammad et al., 2013).

[5]http://sentiment.christopherpotts.net/lingstruc.html

3.1 Dataset and Setting

We train the Twitter sentiment classifier on the benchmark dataset in SemEval 2013 (Nakov et al., 2013). The training and development sets were completely in full to task participants of SemEval 2013. However, we were unable to download all the training and development sets because some tweets were deleted or not available due to modified authorization status. The distribution of our dataset is given in Table 1. We train sentiment classifiers with LibLinear (Fan et al., 2008) on the training set and dev set, and tune parameter $-c$, $-wi$ of SVM on the test set of SemEval 2013. In both experiment settings, the evaluation metric is the macro-F1 of positive and negative classes (Nakov et al., 2013).

	Positive	Negative	Neutral	Total
Train	2,642	994	3,436	7,072
Dev	408	219	493	1,120
Test	1,570	601	1,639	3,810

Table 1: Statistics of our SemEval 2013 Twitter sentiment classification dataset.

The test sets of SemEval 2014 is directly provided to the participants, which is composed of five parts. The statistic of test sets in SemEval 2014 is given in Table 2.

	Positive	Negative	Neutral	Total
T1	427	304	411	1,142
T2	492	394	1,207	2,093
T3	1,572	601	1,640	3,813
T4	982	202	669	1,939
T5	33	40	13	86

Table 2: Statistics of SemEval 2014 Twitter sentiment classification test set. T1 is LiveJournal2014, T2 is SMS2013, T3 is Twitter2013, T4 is Twitter2014, T5 is Twitter2014Sarcasm.

3.2 Results and Analysis

The experiment results of different methods on positive/negative/neutral and positive/negative Twitter sentiment classification are listed in Table 3. The meanings of T1$\sim$T5 in each column are described in Table 2. *SSWE* means the approach that only utilizes the sentiment-specific word embedding as features for Twitter sentiment classification. In *STATE*, we only utilize the existing features (Mohammad et al., 2013) for building the

Method	Positive/Negative/Neutral					Positive/Negative				
	T1	T2	**T3**	**T4**	T5	T1	T2	**T3**	**T4**	T5
SSWE	70.49	64.29	68.69	66.86	50.00	84.51	85.19	85.06	86.14	62.02
Coooolll	72.90	67.68	**70.40**	**70.14**	46.66	86.46	85.32	**86.01**	**87.61**	56.55
STATE	71.48	65.43	66.18	67.07	44.89	83.96	82.82	84.39	86.16	58.27
W2V	55.19	52.98	52.33	50.58	49.63	68.87	71.89	74.50	71.52	61.60
Top	74.84	70.28	72.12	70.96	58.16	- -	- -	- -	- -	- -
Average	63.52	55.63	59.78	60.41	45.44	- -	- -	- -	- -	- -

Table 3: Macro-F1 of positive and negative classes in positive/negative/neutral and positive/negative Twitter sentiment classification on the test sets (T1-T5, detailed in Table 2) of SemEval 2014. The performances of Coooolll on the Twitter-relevant test sets are **bold**.

sentiment classifier. In *Coooolll*, we use the concatenation of SSWE features and STATE features. In *W2V*, we only use the word embedding learned from word2vec[6] as features. *Top* and *Average* are the top and average performance of the 45 teams of SemEval 2014, including the SemEval 2013 participants who owns larger training data.

On positive/negative/neutral classification of tweets as listed in Table 3 (**left** table), we find that the learned sentiment-specific word embedding features (*SSWE*) performs comparable with the state-of-the-art hand-crafted features (*STATE*), especially on the Twitter-relevant test sets (**T3** and **T4**) [7]. After feature combination, *Coooolll* yields 4.22% and 3.07% improvement by macro-F1 on T3 and T4,which verifies the effectiveness of SSWE by learning discriminate features from massive data for Twitter sentiment classification. From the 45 teams, *Coooolll* gets the Rank **5/2/3/2** on T1-T4 respectively, along with the SemEval 2013 participants owning larger training data. We also comparing *SSWE* with the context-based word embedding (*W2V*), which don't capture the sentiment supervision of tweets. We find that *W2V* is not effective enough for Twitter sentiment classification as there is a big gap between *W2V* and *SSWE* on T1-T4. The reason is that *W2V* does not capture the sentiment information of text, which is crucial for sentiment analysis tasks and effectively leveraged for learning the sentiment-specific word embedding.

We also conduct experiments on the positive/negative classification of tweets. The reason is that the sentiment-specific word embedding is learned from the positive/negative supervision of tweets through emoticons, which is tailored for positive/negative classification of tweets. From Table 3 (**right** table), we find that the performance of positive/negative Twitter classification is consistent with the performance of 3-class classification. *SSWE* performs comparable to *STATE* on T3 and T4, and yields better performance (1.62% and 1.45% improvements on T3 and T4, respectively) through feature combination. *SSWE* outperforms *W2V* by large margins (more than 10%) on T3 and T4, which further verifies the effectiveness of sentiment-specific word embedding.

4 Conclusion

We develop a deep learning system (**Coooolll**) for message-level Twitter sentiment classification in this paper. The feature representation of Coooolll is composed of two parts, a state-of-the-art hand-crafted features and the sentiment-specific word embedding (**SSWE**) features. The SSWE is learned from 10M tweets collected by positive and negative emoticons, without any manual annotation. The effectiveness of *Coooolll* has been verified in both positive/negative/neutral and positive/negative classification of tweets. Among 45 systems of SemEval 2014 Task 9 subtask(b), *Coooolll* yields Rank 2 on the Twitter2014 test set, along with the SemEval 2013 participants owning larger training data.

Acknowledgments

We thank Li Dong for helpful discussions. This work was partly supported by National Natural Science Foundation of China (No.61133012, No.61273321, No.61300113).

[6] We utilize the Skip-gram model. The embedding is trained from the 10M tweets collected by positive and negative emoticons, as same as the training data of SSWE.

[7] The result of *STATE* on T3 is different from the results reported in Mohammad et al. (2013) and Tang et al. (2014) because we have different training data with the former and different *-wi* of SVM with the latter.

References

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 49–54.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Ronen Feldman. 2013. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 42–47.

Ming Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. 2013. Unsupervised sentiment analysis with emotional signals. In *Proceedings of the International World Wide Web Conference*, pages 607–618.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. *The Proceeding of Annual Meeting of the Association for Computational Linguistics*, 1:151–160.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *Proceedings of the International Workshop on Semantic Evaluation*, pages 321–327.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, volume 13, pages 312–320.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *The Conference on Neural Information Processing Systems*, 24:801–809.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. *arXiv preprint arXiv:1404.4714*.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 347–354.