Context-Aware Chinese Microblog Sentiment Classification with Bidirectional LSTM

Yang Wang[1], Shi Feng[1,2(✉)], Daling Wang[1,2], Yifei Zhang[1,2], and Ge Yu[1,2]

[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
wangyang@stumail.neu.edu.cn, {fengshi,wangdaling,zhangyifei,yuge}@cse.neu.edu.cn

[2] Key Laboratory of Medical Image Computing (Northeastern University), Ministry of Education, Shenyang 110819, China

Abstract. Recently, with the fast development of the microblog, analyzing the sentiment orientations of the tweets has become a hot research topic for both academic and industrial communities. Most of the existing methods treat each microblog as an independent training instance. However, the sentiments embedded in tweets are usually ambiguous and context-aware. Even a non-sentiment word might convey a clear emotional tendency in the microblog conversations. In this paper, we regard the microblog conversation as sequence, and leverage bidirectional Long Short-Term Memory (BLSTM) models to incorporate preceding tweets for context-aware sentiment classification. Our proposed method could not only alleviate the sparsity problem in the feature space, but also capture the long distance sentiment dependency in the microblog conversations. Extensive experiments on a benchmark dataset show that the bidirectional LSTM models with context information could outperform other strong baseline algorithms.

Keywords: Context-aware sentiment · Recurrent neural networks · Bidirectional long short-term memory · Sentiment classification

1 Introduction

Sentiment analysis, also known as opinion mining, is a fundamental task in natural language processing and computational linguistics. Analyzing the sentiment orientations of the user generated content on the Web has become a hot research topic for both academic and industrial communities in recent years. A crucial problem for sentiment analysis task is that the same word may express quite opposite orientations in different context. For example, the adjective "unpredictable" indicates a negative polarity in the phrase "unpredictable steering" from a car review; On the other hand, it could also expresses a positive orientation in the phrase "unpredictable plot" from a movie review [1]. A lot of literature have been published for building context-sensitive sentiment lexicons

F. Li et al. (Eds.): APWeb 2016, Part I, LNCS 9931, pp. 594–606, 2016.
DOI: 10.1007/978-3-319-45814-4_48

and classifiers [2–5]. However, most of these existing methods need manually designed word or phrase level contextual features or specific topic domains.

With the popular microblogging services such as Twitter and Weibo, users can conveniently express their personal feelings and opinions about all kinds of issues in real time. The user relationships in microblog may be asymmetrical, and every user is able to follow other users, comment on and retweet the microblogs. Microblog could evolve into widespread real-time conversation streams because of the informal, colloquial, asymmetrical and fragmented characteristics. The microblog data is length limited, sparse, and lack of enough context. Therefore, the microblog has more implicit context of the sentiment related features. The context of a current microblog is usually implicit in the preceding microblogs or the information during the interactions. Let us consider the following tweets from two microblog users:

Allen: @**Thomas** *Exactly!*

Bill: @**Thomas** *You're right! I can't agree with you more!!*

Obviously, these two tweets are replies to the preceding ones. We can infer that these tweets may contain strong sentiment orientations. However, it is really difficult for the traditional machine learning based methods to classify the orientations in these sparse tweets that have no explicit lexical nor syntactic sentiment features. The Fig. 1 show the entire conversations of these tweets.

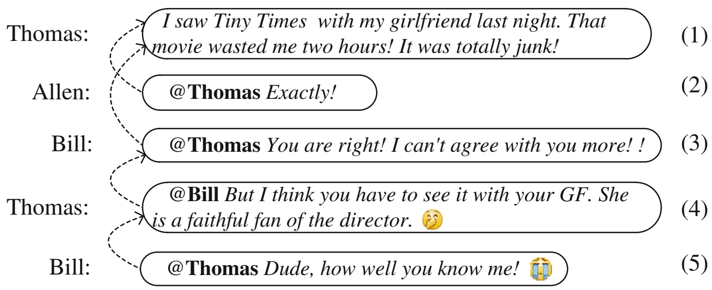

Fig. 1. An illustration of the microblog conversations between three users.

In Fig. 1, Thomas expresses a clear negative sentiment toward the movie "Tiny Times" in tweet (1). Then Allen and Bill agree with Thomas in tweet (2) and (3). Both of these two microblogs are discussing the same movie, but they omit the topic words and express consistent orientations with the preceding one. Only with the help of the context tweets, we can infer that the tweet (2) and (3) express negative sentiments. From this example, we can find that how to merge the context knowledge, and enrich the features of microblog are critical problems in the context-aware sentiment analysis task for microblog conversations.

Different from the traditional context-aware sentiment analysis problem whose solutions rely on explicit semantic features and domain-sensitive words, the context-aware sentiment analysis task in microblogs has brought in several new challenges: **Sparsity**. Users have very limited space to express their feelings in microblogs. Therefore, the vectors formed by tweets are extremely sparse, which set obstacles for directly building classifiers from vector space model; **Context-aware sentiments**. The sentiments embedded in microblog conversations are usually implicit and context-aware. Even a single non-sentiment word can express obvious sentiments given certain context; **Long distance dependency**. The microblog conversations may be quite long. So the background topics and sentiment indicators could be long distance away from the target tweet. The traditional bag-of-words based machine learning methods could not distinguish the implicit or hidden dependency in the long conversations.

To tackle these challenges, in this paper we regard the microblog conversations as sequences and leverage the Recurrent Neural Network (RNN) models to build context-aware sentiment classifiers for Chinese microblogs. The RNN based model with its variants have achieved promising results for many sentiment analysis tasks. However, to the best of our knowledge, little literature has studied the context-aware sentiment classification problem by using the RNN based models. The main contributions of this work are as follows: (1) We model the microblog conversations as sequences and utilize preceding tweets to enrich the content of the target tweets, which could alleviate the sparsity problem and incorporate context information. (2) We develop improved RNN based model bidirectional Long Short-Term Memory (BLSTM), which could give microblogs a continuous representation and consider not only the order of the words in tweets, but also the order of tweets in microblog conversations. (3) Empirical results on a benchmark dataset show that the proposed method outperforms the strong baselines by a large margin.

2 Related Work

2.1 Traditional Context Based Sentiment Analysis

For context based sentiment analysis problems, most of the existing literature are based on the traditional user generated content, such as reviews and blogs. McDonald et al. [2] treated the sentiment labels on sentences as the sequence tagging problem, and utilized CRFs model to score each sentence in the product reviews. Katz et al. [3] used the information retrieval technology to identify the keywords with the emotional information, and then used these keywords and the context to construct the corresponding features. Yang et al. [4] considered both local and global context information to model the complex sentence. They put the vocabulary and discourse knowledge into CRFs model. To deal with the problem that the same sentiment words might have different polarities in the different context, Lu et al. [5] proposed a method for combining information from different sources to learn context-aware sentiment lexicon.

Most of these existing methods relied on explicit semantic features or topic-focused domains. However, the sentiments embedded in microblogs are usually more abbreviated and obscure without obvious sentiment words. Furthermore, the context sentiment related information in microblog conversations could be long distance dependent with each other and usually have scattered topics, which bring new challenges for the sentiment analysis task.

2.2 Sentiment Analysis Based on Microblog Context

Recently, some researches have already treated the microblog content with the dialogue relationship as the context information. Vanzo et al. [6] regarded the context based sentiment classification problem as the sequence labeling problem of the Twitter stream. They used the bag of words model, word meanings and sentiment information as features and used the SVM^{hmm} as the classifier. Mukherjee et al. [7] combined the conjunctions with dialogue information to improve the accuracy of classification results, and discussed the impact of negative words. Wu et al. [8] proposed a new framework to extract context knowledge from unlabeled data for the microblog classification problem, then they defined the relationship between words and words and the relationship between words and emotions. Li [9] classified the microblog text by using the sentiment dictionary, and considered the influence of the user who made comments. Wu et al. [10] utilized the social context information, such as users' historical tweets and friend relations, to tackle the shortness and noise problem of the microblog data.

The previous literature have already proposed the idea of modeling the microblogs as sequences. However, most of these methods usually need to manually design context based features and resorted to the user profiles and social relations to enrich the feature set. In this paper, we leverage the RNN based models to classify the context-aware sentiments in microblogs. Although the RNN based methods have achieved promising results for many sentiment analysis tasks [11,12], to the best of our knowledge, they have never before been reported for context-aware sentiment classification. Our proposed algorithm considers not only the order of words in tweets but also the order of the tweets in conversations, which is capable of learning continuous representations without feature engineering and meanwhile capturing context dependency.

3 Preliminary

The traditional microblog sentiment classification methods treat each post equally in the dataset and do not care about the orders of the posts. However, we can see from the examples in Fig. 1 that without context information we can hardly detect the sentiments embedded in microblog (2) and (3), whose polarities are highly rely on the preceding microblog (1). In this paper, we model the incoming microblog stream as a sequence. Suppose that we have a labeled training set D with microblogs from *Positive, Neutral, Negative* categories. There are several threads in D, and each thread contains sequential microblogs that forming a conversation tree (shown below in Fig. 2) with an original tweet as root and a number of branches as conversations.

Given the training dataset $D = \{(m_1; y_1) \cdots (m_n; y_n)\}$, the sentiment classification problem can be defined as learning a mapping function $f : M \rightarrow Y$

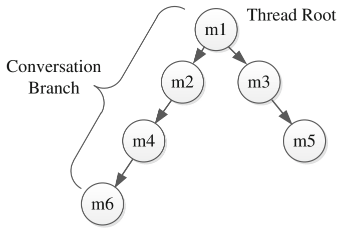

Fig. 2. An example of conversation tree in the training dataset D. Each node is representing a microblog post and the link in the tree denotes the retweet or the '@' actions in microblog stream. There is one original tweet acting as the thread root and the retweet and reference actions have formed the conversation branches in the tree.

where the microblog $m \in M$, the sentiment polarity label $y \in Y$ and $Y = \{Positive, Neutral, Negative\}$. As aforementioned, the conversation branches in D are chain microblogs regarded as sequences. To capture the context information for each m and tackle the sparsity problem, we incorporate the preceding tweets of m to enrich the microblog representations. Therefore, each instance in D is transformed to $(pre(m), m; y)$ where $pre(m)$ denotes the preceding tweets of m in the conversation branch. We aim to learn a context-aware sentiment classification model with the help of the expanded representation as context to detect the polarity of target microblog in the testing set T. Note that we do not consider the sentiment labels of preceding tweets in the conversation.

4 Context-Aware Sentiment Analysis for Microblogs

4.1 Modeling Context-Aware Sentiments in Chinese Microblog

Based on our problem formulation, one key problem is how to model the sequence of the microblog conversation. The traditional language models such as N-Gram can only take into account the limited number of words around the current word, so they cannot model the long term dependency relationships well in the long microblog conversations. On the other hand, the bag-of-words model does not consider the order of the words in the conversation sequence. In this section, we first provide a brief overview of the basic recurrent neural network model for the context-aware sentiment analysis task and introduce the notations used in the paper. Afterwards, we leverage the LSTM and bidirectional LSTM to model the conversation branches in the sequential microblog stream.

Recurrent Neural Network. RNN [14] is a kind of neural network model which is suitable for modeling the sequence information, and it can maintain a memory based on history information, which enables the model to predict the current output conditioned on long distance features. At each time step t, the state of the model's hidden layer is not only related to the current input, but also related to the state of hidden layer at the previous time step $t-1$. Given the expanded representation $(pre(m), m; y)$ of a training instance in D, we concatenate the words in $pre(m)$ and m as a sequence $x_1 x_2 \cdots x_n$, which could reflect not only the order of the words in sentences, but also the order of the tweets in conversations. The word x is the input of the recurrent neural network and the calculation formula of hidden state $h_t \in \mathbb{R}^m$ at time step t is as follows:

$$h_t = f(Wx_t + Uh_{t-1} + b) \tag{1}$$

where W,U and b are the parameters of the model, x_t is the embedding distributed representation of word x at step t. The historical information is stored in the variable h_{t-1}. RNN has demonstrated promising results in some sequence labeling task. However, due to the existence of gradient vanishing problem, learning long-range dependencies with a vanilla RNN is very difficult.

Long Short Term Memory. LSTM [13] model solves the gradient vanishing problem by introducing the memory cells c_t at each time step t and three gate structures: input gate i_t, forget gate f_t and output gate o_t:

$$\begin{aligned}
i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\
f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o) \\
g_t &= \tanh(W^g x_t + U^g h_{t-1} + b^g) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned} \tag{2}$$

Here $\sigma(\cdot)$ is the sigmod function and $\tanh(\cdot)$ is the hyperbolic tangent function. $\odot$ is the element-wise multiplication operator. The memory cell c_t is computed additively, so the error derivatives can flow in another path, thus avoid the gradient vanishing problem. Parameters of the LSTM are W^j,U^j,b^j for $j \in \{i, f, o, g\}$. Similar with RNN model, the input of LSTM is the embedding representation of the word x in the conversation sequence. LSTM is capable of mapping word sequences with variable length to a fixed-length vector by recursively transforming current word vector x_t with the output vector of the previous step h_{t-1}. With the help of the powerful cell and the three gates, the LSTM is able to model the long dependency in the word sequence. During training, the dropout technique that randomly drop units from the neural network is used to avoid overfitting problem and epochs measure how many times every example has been seen during training.

Bidirectional Long Short Term Memory. Considering we need to use the history information of microblogs, we apply the bidirectional LSTM(**BLSTM**) model for making use of the past and future context information. The basic idea of BLSTM is to connect two LSTM hidden layers of opposite directions to the same output. Bidirectional LSTM model is a 2 layers LSTM network, which is composed of forward hidden sequence $\overrightarrow{h}$ and backward hidden sequence $\overleftarrow{h}$ of

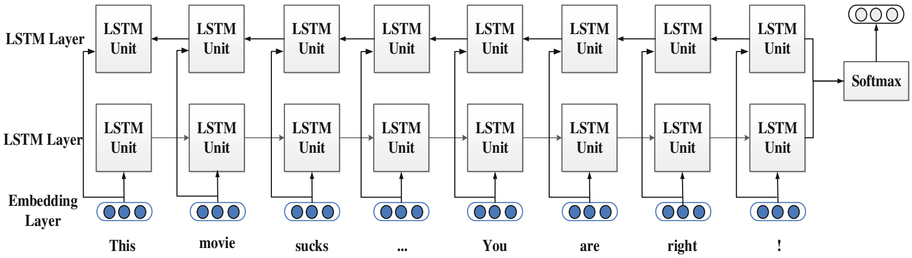

Fig. 3. The bidirectional LSTM model. The input of the model is the word embedding corresponding to each word. Softmax function is used to predict the sentiment polarity.

LSTM models, both of which are connected to the output layer as shown in Fig. 3. The details of the equation of BLSTM model are discussed in Sect. 4.2. Because the bidirectional LSTM model looks at the sequence twice, from left to right and right to left, the output layer can get information from past and future states and the context information can be handled well. The bidirectional LSTM models are trained by back propagation through time(BPTT) [14].

4.2 Context-Aware Sentiment Classification

In our model, each word corresponds to a specific index, then we turn indexes into dense word vectors of fixed size, and each microblog with its preceding tweets is encoded as a sequence of word embeddings. The input of the model at time t is the input word embedding x_t, which is one column of the embedding matrix X. The bidirectional RNN model can be represented as the following equations:

$$
\begin{aligned}
\overrightarrow{h_t} &= f(W_{x\overrightarrow{h}} x_t + U_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \\
\overleftarrow{h_t} &= f(W_{x\overleftarrow{h}} x_t + U_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \\
P(y=j|w_{1:t}) &= \frac{exp(\overrightarrow{h_t} \cdot \overrightarrow{p}^{j} + \overleftarrow{h_t} \cdot \overleftarrow{p}^{j} + q^j)}{\Sigma_{j\in\{-1,0,1\}} exp(\overrightarrow{h_t} \cdot \overrightarrow{p}^{j'} + \overleftarrow{h_t} \cdot \overleftarrow{p}^{j'} + q^{j'})}
\end{aligned}
\tag{3}
$$

where p^j is the j-th column of $P \in \mathbb{R}^{m\times 3}$ and q^j is the j-th element of $q \in \mathbb{R}^3$. P,q,X are parameters of the model to be learned during training strategy. $\{-1, 0, +1\}$ represents categories $\{Negative, Neutral, Positive\}$ respectively. The final orientation of the microblog is predicted by the softmax function at the top layer (as shown in Fig. 3). By replacing the hidden states in the bidirectional RNN with LSTM memory blocks, we can get the aforementioned BLSTM models as the context-aware sentiment representation learning layer in our proposed framework.

5 Experiment

5.1 Datasets

Our experiments were performed on the COAE-2015 Task-1 dataset[1]. The training set that crawled from Weibo contains 2,800 Chinese microblogs with *Positive*, *Negative* and *Neutral* labels. The organizers of the evaluation task have provided a benchmark testing set. However, they do not disclose the gold standard labels of the microblog in the set. We ask two graduate students who majoring in opinion mining to manually label part of the test data. There are about 65 % of the results having the consistent sentiment labels, and we remove the remaining microblogs with the inconsistent annotations. Finally we have 4,248 labeled Chinese microblogs, which belong to 555 threads formed by retweet and reference '@' relationship. Among the data set, there are 1,571 positive, 1,030 negative, and 1,647 neutral microblog respectively. The statistics information of the conversation length are shown in Table 1. We can see that most of the sequential conversations have the length of two or three. But there are still quite a number of conversations contain more than 3 sequential microblogs.

Table 1. The statistics of the conversation branch length in the dataset

	Length 2	Length 3	Length 3+
Percentage	57.8 %	25.1 %	17.1 %

The statistics information of the sentiment polarity drift between the root microblog and the retweet microblogs are shown in Table 2. The horizontal header represents the source microblog polarity information and the vertical header represents the retweet microblog polarity information.

Table 2. The sentiment drift information of the source and retweet microblog

Retweet	Source		
	Positive	*Neutral*	*Negative*
Positive	725	629	52
Neutral	287	908	141
Negative	126	554	217

We can see that a lot of microblogs retain the source microblogs polarities, and there are still quite a number of microblogs have the reverse orientations. These polarity drifts can be indicated by features such as contrastive connectives

[1] http://www.ccir2015.com/.

or context dependencies, which could be captured by our LSTM models. Finally we select 4/5 of the whole dataset as the training set and the remaining as the validation set. The dataset used in this paper has been made public[2].

5.2 Experiment Setup

The input of LSTM models are 50 or 100 dimensions word embedding vectors that constructed by selecting the 10,000 most frequent features in the dataset. We choose the Adam algorithm as the optimizer and choose the softmax as the top layer to predict the final sentiment labels. For regularization, early stopping and dropout [13] are used during the training procedure. In the LSTM model, we compared the impact of the dropout and weight regularize mechanism on the neural network structures. We set the dropout rate to 0.5 on the input-to-hidden layers and hidden-to-output layer and use the L2 regularization.

5.3 Comparison Methods

Bag-of-words and SVM. We build a Support Vector Machine (SVM) based classifier on the training data provided. There are two kinds of feature space: (1) unigram, bigram and trigram (1 to 3-gram); (2) unigram, bigram, trigram, 4-gram and 5-gram (1 to 5-gram). For implementation, we utilize the TFIDFVectorizer function provided by Scikit-learn[3] to get the document-term matrix and calculate the TFIDF weight respectively. The SVM is implemented by LinearSVC function in Scikit-learn.

Conditional Random Fields. CRFs [15] are a class of statistical modelling method often applied in pattern recognition and machine learning, where they are used for structured prediction. One of the most common use-cases for structured prediction is chain-structured outputs. These occur naturally in sequence labeling tasks, such as Part-of-Speech tagging or segmentation in natural language processing.

We regard the conversation microblog with the retweet relationship as the input sequence, and the sentiment of each microblog as the output sequence. We use the linear chain CRF model to solve the context-aware sentiment classification problem. Each microblog is a node in our chain, and the microblogs with the retweet relationship are connected with an edge. Therefore, the conversation branch described in Sect. 3 has formed the chain in CRF model. The length of the chain varies with the number of microblogs with the retweet relationship, just the same as the conversation branches in Fig. 1. The CRF model with linear chain structure is implemented by Pystruct[4] package.

---

[2] http://github.com/leotywy/coae2015.
[3] http://scikit-learn.org/stable/.
[4] http://pystruct.github.io/.

Gated Recurrent Units (GRU). In order to make a comparison, we utilize an alternative RNN model called GRU [14] to train a sentiment classifier. GRU is designed to have more persistent memory thus making it easier to capture long-term dependencies. GRUs have fewer parameters (W and U are smaller) to train and the training time could be less than the basic LSTM model.

5.4 Experiment Results

We first study the effect of some hyperparameters on the one layer LSTM model with dropout and L2 regularize. We fix the number of hidden unit as 64, batch size as 10, and compare the impact of the dropout rate and epoch of the model.

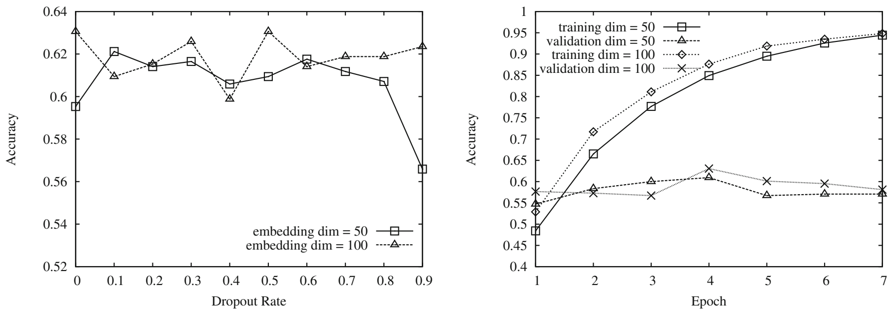

Fig. 4. The performance of 1 layer LSTM model with different dropout rate and epoch.

Figure 4 shows the higher dimensional word embedding (100) generally performs better. From the results, we find that setting the dropout rate to 0.5 and training the model with 4 or 5 epoch can get the good performance with the larger word embedding dimensions. When the epoch is larger than 5, model may be over fitting. Therefore using the early stopping method may be a good choice on this small dataset. Then we fix the dropout rate to 0.5 and explore the impact of the number of hidden units and the batch size as shown in Fig. 5.

In Fig. 5, with different number of hidden units, the impact of the dimension of the word embedding on the results is not very obvious. 100 dimensional word embedding is slightly better than 50 dimensional word embedding. For both of the smaller and larger word embedding, the batch size of training data for 8 or 16 is good enough. In the following experiments, we choose batch size as 8. For the smaller word embedding, the number of neural units in the hidden layer for 16 performs well. For the larger word embedding, the number of neural units in the hidden layer for 32 or 128 performance better than the smaller model by using the smaller batch size, but the model is not as stable as the smaller word embedding model when the batch size changes.

Then we compare the results of the traditional methods and recurrent neural network models. We perform hyperparameters tuning on all these models. For

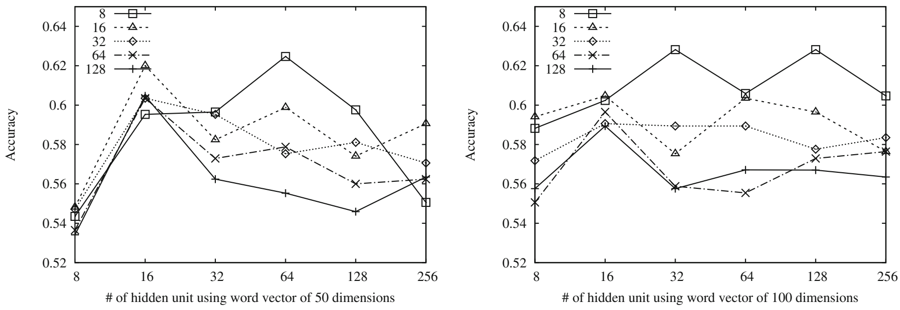

Fig. 5. The performance of BLSTM model trained with different dimension of word embedding, number of hidden units and batch size. Different lines represent different number of batch sizes with the different number of hidden layer units.

the bag of words model in SVM, we combine the target microblog with all of its preceding tweets. For the recurrent models, we compare the performances of RNN, GRU, LSTM and BLSTM. We use Accuracy, Precision, Recall and Macro F1 score as the evaluation metric. The results are shown in Table 3.

In the Table 3, the *context* column represents whether to incorporate preceding tweets or not. From this table we can see that as a strong baseline the SVM based methods could achieve a relatively good performance when using 1 to 3-gram as features without context information. However, when incorporating the preceding tweets, the performances decrease dramatically. This may because that bag of words based SVM does not consider word and sentence orders. Including preceding tweets has brought in more noise instead of useful features. The performance of CRF based method is dragged down by the sparse feature space and limited training instances.

All of the recurrent neutral network based methods could achieve better results when incorporating context information, which indicates that RNN based methods are good at modeling the context-aware sentiments in the microblog conversations. Note that in Table 3 the vanilla RNN does not have a context-aware results because when considering preceding tweets, the training procedure does not converge for vanilla RNN. We do not report some of the context-free classification results due to space limitations.

A good classification performance is achieved when using one layer LSTM with dropout and regularization. Dropout is a technique that can randomly drop units from the neural network during training, which can prevent the neural network overfitting effectively. We argue that the input word embedding could alleviate the sparsity problem of the feature space. The GRU and LSTM model could map the long word sequences into fixed-length word vectors and capture the context-aware sentiments in microblogs with long distance dependencies. The performance of GRU model and LSTM is similar, but slightly worse than that of the BLSTM. We find that the bidirectional LSTM achieve a better performance than the unidirectional LSTM at the context aware sentiment analysis task.

Table 3. Experiment results of the shallow learning methods and RNN models

Model	Features/structures	*Context*	Accuracy	Precision	Recall	MacroF1
SVM	1 to 3 gram	No	0.6235	0.6134	0.6079	0.6082
		Yes	0.5847	0.5699	0.5677	0.5683
	1 to 5 gram	No	0.6176	0.6089	0.6027	0.6042
		Yes	0.5965	0.5798	0.5777	0.5779
CRF		Yes	0.4264	0.4327	0.4187	0.4113
RNN	1 layer+dropout+L2	No	0.5800	0.5722	0.5718	0.5702
		Yes	-	-	-	-
GRU	1 layer+dropout+L2	No	0.5894	0.5959	0.5839	0.5794
		Yes	0.6188	0.6059	0.6012	0.6019
LSTM	1 layer	No	0.5965	0.5940	0.5760	0.5804
		Yes	0.6165	0.6145	0.6113	0.6093
	1 layer+dropout+L2	No	0.6153	0.6100	0.5900	0.5815
		Yes	0.6205	0.6145	0.6058	0.6057
	BLSTM	No	0.6024	0.5839	0.5776	0.5757
		Yes	0.6223	0.6114	0.5996	0.6008
	BLSTM+dropout+L2	No	0.6118	0.5921	0.5855	0.5802
		Yes	**0.6282**	**0.6196**	**0.6189**	**0.6183**

The BLSTM provides the output with both the past and the future context information for every timestep in the input sequence.

6 Conclusion

In this paper, we utilize the different LSTM models for the context-aware Chinese microblog sentiment analysis task. The extensive experiments on a benchmark dataset demonstrate that the LSTM based models can be more effective at representing the sequential microblog conversations and modeling the context-aware sentiments. The best performance is achieved when using bidirectional LSTM networks with dropout and regularization.

In the future, we intend to use the pretrained word embedding with larger corpus as the initial input to the recurrent network. We will also consider attention-based RNN models for further improving the sequential representation of the microblog conversations. Our proposed method is language-independent, and we will conduct more experiments to evaluate the models on Twitter corpus.

Acknowledgments. The work is supported by National Natural Science Foundation of China (61370074, 61402091), the Fundamental Research Funds for the Central Universities of China under Grant N140404012.

References

1. Turney, P.D.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL, pp. 417–424 (2002)
2. McDonald, R., Hannan, K., Neylon, T., Wells, M., Reynar, J.: Structured models for fine-to-coarse sentiment analysis. In: Proceedings of ACL, pp. 432–439 (2007)
3. Katz, G., Ofek, N., Shapira, B.: ConSent: context-based sentiment analysis. Knowl.-Based Syst. **84**, 162–178 (2015)
4. Yang, B., Cardie, C.: Context-aware learning for sentence-level sentiment analysis with posterior regularization. In: Proceedings of ACL, pp. 325–335 (2014)
5. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.: Automatic construction of a context-aware sentiment Lexicon: an optimization approach. In: Proceedings of WWW, pp. 347–356 (2011)
6. Vanzo, A., Croce, D., Basili, R.: A context-based model for sentiment analysis in twitter. In: Proceedings of COLING, pp. 2345–2354 (2014)
7. Mukherjee, S., Bhattacharyya, P.: Sentiment analysis in twitter with lightweight discourse analysis. In: Proceedings of COLING, pp. 1847–1864 (2013)
8. Wu, F., Song, Y., Huang, Y.: Microblog sentiment classification with with contextual knowledge regularization. In: Proceedings of AAAI, pp. 2332–2338 (2015)
9. Li, D., Shuai, X., Sun, G., Tang, J., Ding, Y., Luo, Z.: Mining topic-level opinion influence in microblog. In: Proceedings of CIKM, pp. 1562–1566 (2012)
10. Wu, F., Huang, Y., Song, Y.: Structured microblog sentiment classification via social context regularization. Neurocomputing **175**, 599–609 (2016)
11. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of EMNLP, pp. 1422–1432 (2015)
12. Wang, X., Liu, Y., Sun, C.: Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of ACL, pp. 1343–1353 (2015)
13. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving Neural Networks by Preventing Co-adaptation of Feature Detectors (2012). arXiv preprint. arXiv:1207.0580
14. Greff, K., Srivastava, R.K., Koutnk J., Steunebrink B.R., Schmidhuber J.: LSTM: A Search Space Odyssey (2015). arXiv preprint. arXiv:1503.04069
15. Lafferty, J., McCallum, A.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceeding of ICML, pp. 282–289 (2001)