World Wide Web (2019) 22:59–81
https://doi.org/10.1007/s11280-018-0529-6

Attention based hierarchical LSTM network for context-aware microblog sentiment classification

Shi Feng[1] · Yang Wang[1] · Liran Liu[1] · Daling Wang[1] · Ge Yu[1]

Received: 8 May 2017 / Revised: 29 October 2017 / Accepted: 21 January 2018 /
Published online: 29 January 2018

Abstract Analyzing sentiment polarities of microblog has become a hot research topic for both academic and industrial communities. Most of the existing algorithms regard each microblog as an independent training instance. However, the sentiments embedded in short tweets are usually ambiguous and context-aware. Even a non-sentiment word might convey a clear emotional tendency in certain microblog conversations. In this paper, we regard the microblog conversation as sequence, and develop a Context Attention based Long Short-Term Memory (CA-LSTM) network to incorporate preceding tweets for context-aware sentiment classification. The CA-LSTM network has a hierarchial structure for modeling microblog sequence and allocates the words and tweets with different weights using attention mechanism. Our proposed method can not only alleviate the sparsity problem in feature space, but also capture long distance sentiment context dependency in microblog conversations. Experimental evaluations on a public available dataset show that the proposed CA-LSTM network with context information can outperform other strong baselines by a large margin.

✉ Shi Feng
fengshi@cse.neu.edu.cn

Yang Wang
leotywy@163.com

Liran Liu
lyran_summer@sina.com

Daling Wang
wangdaling@cse.neu.edu.cn

Ge Yu
yuge@cse.neu.edu.cn

[1] School of Computer Science and Engineering, Northeastern University, No.195 Chuangxin Road, Hunnan District, Shenyang, China

Keywords Sentiment classification · Context-aware sentiment · Twitter mining · Long short-term memory

1 Introduction

In recent years, people become more eager to publish their attitudes and feelings in online social media rather than just passively browse and accept information. Because of the rich user-generated information in social media such as blogs and review websites, how to provide an effective way to analyze users' sentiments has received significant attentions from both academic researchers and commercial companies. Inferring sentiment polarity (e.g. positive or negative) of a document in social media is the fundamental task for sentiment analysis and a lot of literature have been published for classifying polarity based on sentiment lexicons or machine learning methods [23].

A crucial problem for polarity classification task is that the same word may express quite opposite polarities in different context. For example, the adjective "unpredictable" indicates an obvious negative polarity in the phrase "unpredictable steering" from a car review; On the other hand, "unpredictable" can also express a positive orientation in the phrase "unpredictable plot" from a movie review [30]. Previous studies have paid close attention to building context-sensitive lexicons and classifiers [12, 17, 18, 21, 37]. However, most of these existing methods need manually designed word or phrase level contextual features. Otherwise, the classifiers are usually trained on topic-specific domains, such as movie reviews or car reviews.

With the advent of microblogging services such as Twitter[1] and Weibo,[2] users can conveniently express and share their personal feelings and opinions about all kinds of hot topics in real time [2, 13]. The informal, colloquial and length limited characteristics have made microblogs evolve into conversation streams with fragmented sentiment expressions. Therefore, there are more implicit sentiment features in microblog. That is to say, the context of a target microblog is usually implicit in the preceding conversation or embedded in the information during interactions. Let us consider the following example, in which a tweet from `Bill` cites `Allen`:

`Bill` : @**Allen** *It seems that I don't need to buy winter clothes in the near future! Yeah!*

Obviously, this tweet is a reply to the preceding one. We can infer that this tweet is subjective. The traditional machine learning based methods tend to classify this tweet into positive category mainly because of the word *Yeah*. Figure 1 show the entire conversation of this tweet.

In Figure 1, the user `Allen` posts a neural microblog that is mainly about Trump's withdrawal from Paris agreement. `Bill` particularly leverages an ironic tone to express negative sentiments with non-negative words or even positive word. `Thomas` follows `Bill`'s ironic tone and satirizes Trump's decision. Both Tweet (b) and (c) express negative sentiments. However, without the help of the preceding tweets and topic semantics in the conversation, we can hardly infer context-aware meanings of the sentences. From this running example, we observe that how to merge context knowledge, capture topic semantics and enrich feature space of microblog are critical problems in sentiment analysis task for microblog conversations.

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

[2] http://www.weibo.com/.

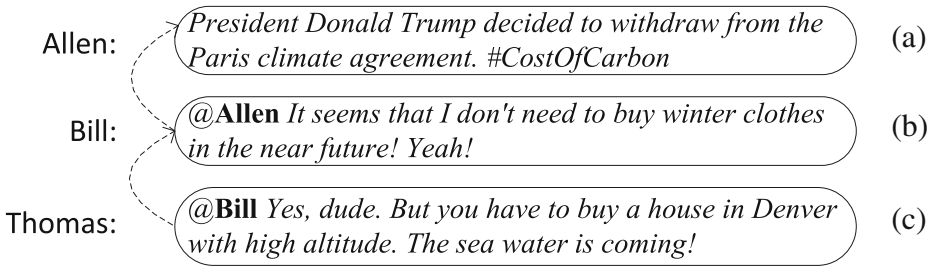


Figure 1 A toy example of microblog conversation between three anonymized users

Different from traditional context-aware sentiment analysis problem in product reviews whose solutions mainly 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 corresponding vectors generated by tweets are extremely sparse, which set obstacles for directly building classifiers from vector space model with manually designed features.

Context-aware sentiment The sentiments embedded in microblog conversations are usually implicit and context-aware. Even a single non-sentiment word can express obvious sentiments given context in certain conversation.

Long distance dependency Microblog conversations may be quite long, but usually share common topics, namely background topics such as a new mobile product or a football game. Authors often omit topic words during the conversation. If a tweet we want to determine its polarity (dubbed as target tweet) is in the tail of conversation chain, the background topics and sentiment indicators could be long distance away from target tweet. Traditional bag-of-words based machine learning methods have difficulties in distinguishing implicit or hidden dependency in these long conversations.

To tackle these challenges, in this paper we regard the microblog conversations as sequences and propose an Context Attention based Long Short-Term Memory (CA-LSTM) network to build context-aware sentiment classifiers for microblogs. We firstly build a hierarchical LSTM network to generate tweet-level and conversation-level representation. Then we incorporate an attention mechanism over microblog conversation sequence and determine the weight of each representation component, which can significantly improve the performance of context-aware microblog sentiment classification task. In summary, the main contributions of this paper are as follows:

– We model microblog conversations as sequences ordered according to time and utilize preceding tweets to enrich the content of target tweets, which can alleviate sparsity problem and incorporate context semantic information.
– We develop a Context Attention based Long Short-Term Memory (CA-LSTM) network, which gives microblogs a continuous representation and considers not only the order of the words in tweets, but also the order of tweets in microblog conversations using hierarchical structure for long distance sentiment dependency.
– We incorporate the attention mechanism into the CA-LSTM network for determining different sentiment weights of words and tweets in the conversation. As far as we know,

this is the first attempt to explore the feasibility and effectiveness of attention based models for context-aware sentiment classification in microblog conversations.
– We conduct extensive experiments on a public available benchmark microblog dataset. The empirical results show that the proposed method outperforms the strong baselines by a large margin.

The rest of the paper is organized as follows. Section 2 summarizes related works in the field of context-aware sentiment analysis tasks in traditional social media and microblog. Section 3 introduces the preliminary. In Section 4, we propose the CA-LSTM network model for context-aware sentiment analysis. Section 5 reports the experimental setup and results on the benchmark dataset. Finally, Section 6 draws the conclusion and outlines the future work.

2 Related works

The context based sentiment analysis has been a hot research topic since the sentiment analysis task emerged. In this section, we first review the traditional context based sentiment analysis methods for blogs and reviews. Then we give a brief summarization of recently published papers for context based sentiment analysis in microblogs.

2.1 Context based sentiment analysis in blogs and reviews

For context based sentiment analysis problems, most of the existing literature focus on traditional user generated content, such as reviews and blogs. At the very beginning, Turney et al. [30] utilized search engine statistical information to infer context-sensitive sentiment polarities of the words and phrases. McDonald et al. [18] treated the sentiment labels on sentences as sequence tagging problem, and leveraged CRFs model to score each sentence in the product reviews. Katz et al. [12] used the information retrieval technology to identify the keywords with the emotional information, and then utilized these keywords and context to construct the corresponding sentiment features. Yang et al. [37] considered both local and global context information to model complex sentence. They incorporated the vocabulary and discourse knowledge into CRFs model. Lu et al. [17] proposed a method for combining information from different sources to learn a context-aware sentiment lexicon.

The previous studies have already realized that the same word may convey quite different sentiment polarities in different context. Most of these existing methods relied on explicit semantic features or topic-focused domains of product reviews. However, the sentiments embedded in microblogs are usually more abbreviated and obscure without clear sentiment words. Furthermore, the context related information in microblog conversations could be long distance dependent with each other, which brings new challenges for context-aware sentiment analysis task.

2.2 Context based sentiment analysis in microblogs

Recently, some researchers treated the microblog content with dialogue relationship as context information. Vanzo et al. [31] regarded context based sentiment classification as sequence labeling problem of Twitter stream. The authors manually designed context-sensitive features and employed SVM^{hmm} as the classifier to assign sentiment polarity to the tweets in sequences. Mukherjee et al. [22] combined the conjunctions with dialogue

information to improve the accuracy of classification results, and further discussed the impact of negative words.

Besides context tweets in corresponding conversation, previous studies also resorted to other external context features, such as user profiles and social relations. Wu et al. [36] proposed a new framework to extract word-word and word-sentiment contextual knowledge from unlabeled data. The contextual knowledge was formulated as regularization terms in supervised learning algorithms for further improving the performance of the classifier. The opinions of friends towards the same topic usually tend to be similar. Based on the above observation, Wu et al. [35] further attempted to utilize the social context information, such as users historical tweets and friend relations, to tackle the shortness and noise problem of microblog data. Similarly, Tan et al. [28] utilized the contextual information extracted from the follower/followee network in Twitter and the "@" mention relationship to build a sentiment classifier that could capture user-level characteristics. Li [15] incorporated topic factor and social influence in a probabilistic framework for predicting users' future opinions on some specific topic.

The previous literature for context based microblog sentiment analysis usually need to manually design context based features. Otherwise, existing papers resort to user profiles and social relations to enrich feature set, but they ignore the word orders and other inner textual features of sentences, which are very important for the polarity classification task.

2.3 Deep learning models for sentiment classification

The deep learning models have achieved promising results for many sentiment analysis tasks [29, 32]. Similar to our motivation, Huang et al. [11] proposed a hierarchical LSTM model to capture the long distance sentiment dependency in the microblog conversations. The performance of their proposed model was further improved by incorporating additional social and text-based context features. Zhao et al. [40] leveraged the following relationship in Twitter to build a heterogeneous network and incorporated random walks into the LSTM network for personalized sentiment classification. Ren et al. [24] extracted the words with high TF-IDF values in training corpus as context features which were further integrated with textual features using Convolutional Neural Network model for context-sensitive tweet sentiment classification. These LSTM and CNN based deep learning methods outperform traditional shallow learning algorithms. However, the previous models do not explicitly consider the weights of hidden units in context sequence.

Different from hierarchical LSTM [11], more syntactic information is considered for recursive tree-structured LSTM models [16, 27, 42]. Built on dependency tree, the Tree-LSTM can better model semantic relationship between words in the sentence [11]. However, instead of dependency tree-structure, the sentences in a tweet or the tweets in a conversation branch share a sequential chain structure, as shown in Figure 1.

In this paper, we integrate attention mechanism into hierarchical LSTM network models to classify context-aware sentiments in microblogs. Different from previous literature that incorporating history tweets, social relations or user profiles, the proposed CA-LSTM model does not resort to external features and only focus on microblog conversations themselves. 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.

The attention mechanism for neural network models has achieved outstanding performance for many sentiment classification tasks [1, 34, 41]. Zhou et al. proposed a framework with two hierarchical LSTM components for simultaneously modeling the parallel

cross-lingual corpus. The attention vectors are incorporated at the high level of the network to determine the weights of the sentences and words [41]. Chen et al. utilized global attention vectors to capture characteristics of products and users for review classification [1]. Wang et al. appended the aspect embedding into input each word vector and the final output of the LSTM network was weighted sum of hidden vectors with the help of global attention mechanism [34]. The basic idea of attention based sentiment classification methods is designing a task-specific neural network structure and leveraging attention vector to allocate composition weights of output vectors. Based on this intuition, we model microblog conversation as sequence, and incorporate the preceding tweets as context information. To the best of our knowledge, the attention based model has never before been reported for context-aware sentiment classification task.

3 Preliminary

3.1 Problem formulation

The traditional microblog sentiment classification methods treat each post equally and independently in the dataset and do not care about the order of posts. However, the examples in Figure 1 show that without context information we can hardly detect the sentiments embedded in microblog (b) and (c), whose polarities highly rely on the preceding microblogs. Hence, the context-aware sentiment analysis problem is that sentiment is not only determined by target microblog, but also should be inferred with the help of the context in conversation. In this paper, we model the incoming microblog stream with retweeting or referring relationship as a message 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 in Figure 2) with an original tweet as root and a number of branches as conversations.

Given training dataset $D = \{(m_1; y_1) \cdots (m_n; y_n)\}$, the sentiment classification task can be defined as learning a mapping function $f : M \rightarrow Y$ where microblog $m \in M$ and M is microblog set; the sentiment polarity label $y \in Y$ and $Y = \{Positive, Neutral, Negative\}$. As aforementioned, conversation branches in D are chain microblogs regarded as message sequences ordered according to time. To capture the context information for target microblog m and tackle the sparsity problem, we incorporate the preceding tweets of m to enrich the target microblog's representation. Therefore, each instance in D is transformed to $(pre(m), m; y)$ where $pre(m)$ denotes the preceding tweets

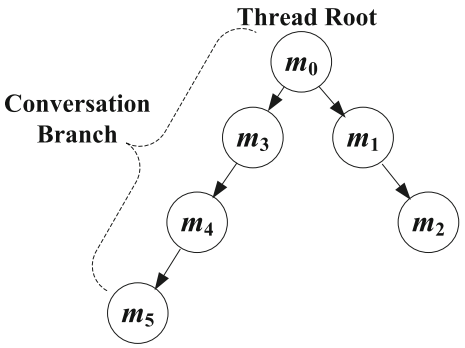


Figure 2 An example of conversation tree in training dataset D. Each node represents a microblog post and the link in the tree denotes retweet or '@' actions in microblog stream. There is one original tweet acting as thread root and retweetting and referring actions have formed two conversation branches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

of m in the conversation branch. If m is the root, $pre(m)$ is $Null$. 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 microblogs in testing set T. Note that during training and testing we ignore sentiment labels of preceding tweets in the conversation. Moreover, different from previous literature [11, 24, 35, 36], we do not consider author profiles or social relations either. We only focus on text in the conversations for sentiment classification problem.

3.2 Recurrent neural network and LSTM

Based on our problem formulation, the key issue is how to model the sequence of microblog conversation. The traditional language models such as N-Gram only take into account limited number of words around current word, so they cannot model the long distance dependency relationships well in microblog conversations. On the other hand, bag-of-words model does not consider the order of words in the conversation sequence. In this subsection, we provide a brief overview of the basic Recurrent Neural Network and LSTM models for context-aware sentiment analysis task and introduce the notations used in the paper.

Recurrent neural network RNN [8] is a kind of neural network model that is suitable for modeling sequential data. The RNN model maintains a inner memory based on history information, which enables the model to predict 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 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 is able to reflect not only the order of the words in sentences, but also the order of the tweets in conversation. The word x is the input of Recurrent Neural Network and calculation formula of hidden state $\mathbf{h}_t \in \mathbb{R}^d$ at time step t is as follows:

$$\mathbf{h}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \tag{1}$$

where $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{b}$ are parameters of the model, $\mathbf{x}_t$ is the distributed representation of word x at step t. The historical information is stored in the variable $\mathbf{h}_{t-1}$. The RNN model has demonstrated promising results in some sequence labeling tasks [3, 7, 26]. However, due to the existence of gradient vanishing problem, directly learning long-range dependencies with a vanilla RNN is still very difficult.

Long short term memory LSTM [8, 10] model solves the gradient vanishing problem by introducing memory cells c_t at each time step t and three gate structures: input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$ and output gate $\mathbf{o}_t$:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^i\mathbf{x}_t + \mathbf{U}^i\mathbf{h}_{t-1} + \mathbf{b}^i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^f\mathbf{x}_t + \mathbf{U}^f\mathbf{h}_{t-1} + \mathbf{b}^f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o\mathbf{x}_t + \mathbf{U}^o\mathbf{h}_{t-1} + \mathbf{b}^o) \\
\mathbf{g}_t &= \tanh(\mathbf{W}^g\mathbf{x}_t + \mathbf{U}^g\mathbf{h}_{t-1} + \mathbf{b}^g) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{2}$$

where $\sigma(\cdot)$ is sigmoid function, $\tanh(\cdot)$ is hyperbolic tangent function and $\odot$ is element-wise multiplication operator. The memory cell $\mathbf{c}_t$ is computed additively, thus the error derivatives can flow in another path, and avoid gradient vanishing problem. The parameters of LSTM model are $\mathbf{W}^j, \mathbf{U}^j, \mathbf{b}^j$ for $j \in \{i, f, o, g\}$. Similar with RNN model, the input of

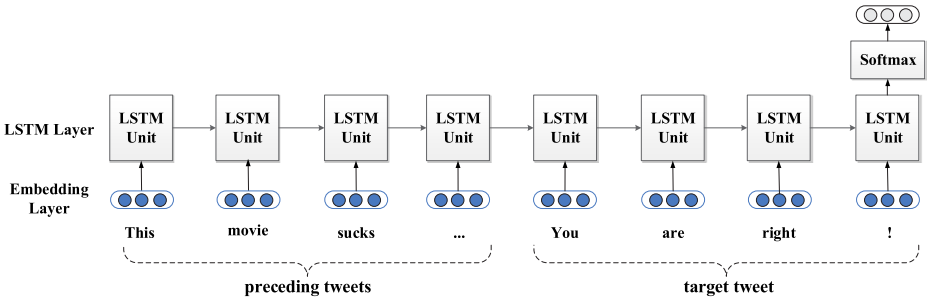


Figure 3 The LSTM model for sentence classification

LSTM is the embedding representation of word x in conversation sequence. LSTM is capable of mapping word sequences with variable length to a fixed-length vector by recursively transforming current word vector $\mathbf{x}_t$ with the output hidden vector of the previous step $\mathbf{h}_{t-1}$.

The key idea behind the LSTM architecture is a memory cell $\mathbf{c}$ which can maintain its state over time, and the non-linear gating units that regulate the information flow into and out of the cell. With the help of the powerful cell and the gating units, the LSTM model is able to model the long-range dependency in the word sequence. By adding a softmax layer at the end of the network, the LSTM model can deal with the sequence labeling and sentiment classification problems as shown in Figure 3.

4 Context attention based LSTM

We now describe our method for modeling microblog conversations with improved LSTM variants. We first introduce the hierarchical LSTM model for sentiment classification. Then, we incorporate attention mechanism into the hierarchical LSTM model for further capturing the context information in the microblog conversations.

4.1 Hierarchical LSTM

The vanilla LSTM model represents the microblog conversations using chain structure. However, according to the principle of compositionality [5], the semantic of the microblog conversations can be better modeled by a hierarchical structure composed of word-level, tweet-level and conversation-level information. Capturing this hierarchical sequential structure for text [30] can potentially give the RNN models higher predictive accuracies, as we have seen in previous works [4, 9, 20, 42]. The similar hierarchical LSTM network has been adopted by Huang et al. [11] for modeling the microblog conversations. Following Huang's work, the hierarchical LSTM we used in this paper is depicted in Figure 4.

In Figure 4, the input of the model is the word embedding corresponding to each word. The first word-level LSTM layer can generate the representation of a single tweet by the hidden state of the last word. And the second tweet-level LSTM layer is able to model the context information of the target tweet. Specifically, the inputs for the tweet-level LSTM are the representations of tweets in the conversation that is composed of the preceding tweets $pre(m)$ and the target tweet m in a sequence. The first one is the root tweet, while the others are the retweets or references of the root tweet ordered according to time with the target tweet m being the last. Finally, the output hidden vector of the last LSTM unit in tweet-level

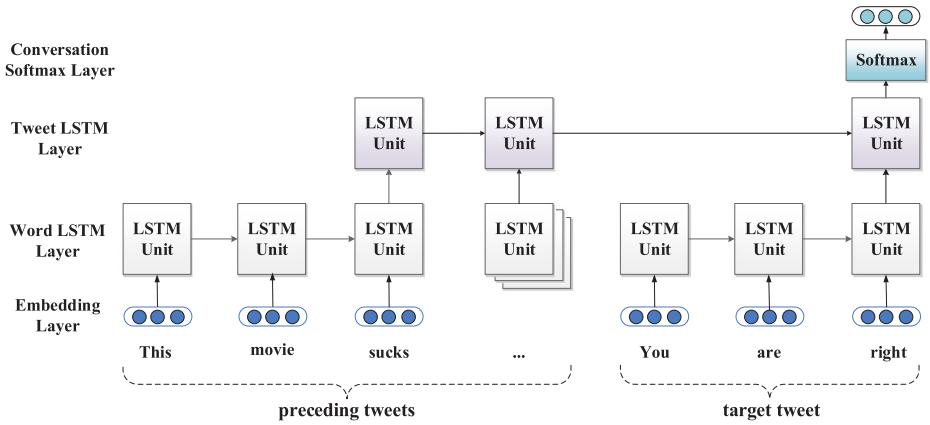


Figure 4 The hierarchical LSTM model for context-aware sentiment classification

is the distributed representation of the conversation sequence, which is further fed into the softmax function for the polarity classification.

The memory cells in the LSTM units can capture the long distance dependency in the sequence. Therefore, by considering the retweet structure, the hierarchical LSTM model is able to combine both the local features of a tweet and the long-range semantic and dependency information in conversation to perform context-aware sentiment classification.

4.2 Incorporating context attention

The hierarchical LSTM (abbreviated as HLSTM) model has achieved the state-of-the-art result for the context-aware sentiment classification task [11]. However, in HLSTM the context information is mainly implicitly embedded in the memory cells and hidden vectors of the LSTM units. It is obvious that not all the tweets in the conversation sequence contribute equally to the target microblog's polarity. For example, the root and the adjacent tweet usually play more critical role than the other tweets in the sequence. Inspired by previous work [38], we incorporate the context attention into the hierarchical LSTM network to capture the crucial components over different semantic levels for the context-aware sentiment classification problem. The overall architecture of our proposed Context Attention based hierarchical LSTM (abbreviated as CA-LSTM) model is shown in Figure 5.

In Figure 5, instead of using the last LSTM unit hidden state as the representation of the corresponding tweet/conversation, we introduce a context attention mechanism to aggregate the representation of those LSTM unit outputs to form a tweet/conversation vector. Note that we have a word-level and a tweet-level context attention vector respectively in CA-LSTM model.

Word-level context attention Specifically, at word level, the context attention vector associates each hidden word representation with different weight according to its overall contribution to the sentiment polarity of the tweet. Formally, given a tweet m_i with l_i words, the enhanced tweet representation is a weighted sum of hidden word states as:

$$\mathbf{m}_i = \sum_{j=1}^{l_i} \alpha_j^i \mathbf{h}_j^i \tag{3}$$

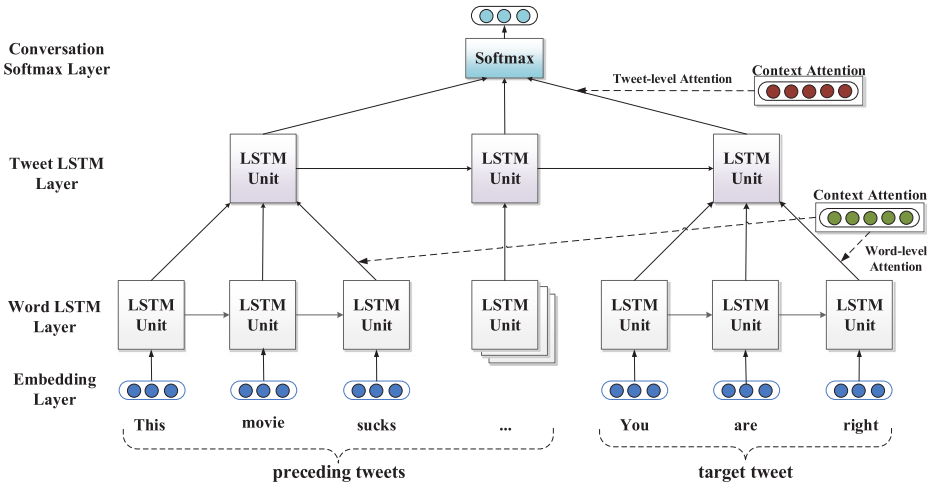


Figure 5 The overall architecture of the Context Attention LSTM model

where $\mathbf{m}_i$ is the distributed representation of tweet m_i, and the context weight α_j^i measures the importance of the j-th word for determining sentiment polarity. We embedded the word-level context attention as continuous real valued vector $\mathbf{ca}_w \in \mathbb{R}^d$, where d is the dimensions of embedding vector. Then the context weight α_j^i for the corresponding hidden state $\mathbf{h}_j^i$ of word w_j^i can be calculated as:

$$\alpha_j^i = \frac{\exp(f(\mathbf{ca}_w, \mathbf{h}_j^i))}{\sum_{k=1}^{l_i} \exp(f(\mathbf{ca}_w, \mathbf{h}_k^i))} \tag{4}$$

where $\mathbf{ca}_w$ is the word-level context vector; the function $f(\cdot)$ is utilized to calculated the relation scores between the context vector and the word-level hidden vectors for composing sentence representation. The function $f(\cdot)$ is defined as:

$$f(\mathbf{ca}_w, \mathbf{h}_j^i) = \mathbf{ca}_w^{\mathrm{T}} \tanh(\mathbf{W}_w \mathbf{h}_j^i + \mathbf{b}_w) \tag{5}$$

where $\mathbf{W}_w$ is weight matrix and $\mathbf{b}_w$ is bias. In Formula (5), we feed word LSTM unit state $\mathbf{h}_j^i$ and bias $\mathbf{b}_w$ into nonlinear hyperbolic tangent activation function and then the output is multiplied with the context vector $\mathbf{ca}_w$ to score the importance of word w_j^i in representation of tweet m_i. Based on $f(\cdot)$, in Formula (4) we leverage the softmax function to calculate the normalized importance weight α_j^i of w_j^i in m_i using the function score $f(\mathbf{ca}_w, \mathbf{h}_j^i)$. Finally, the context-aware embedding of the tweet m_i is calculated by weighted sum of the word representation using Formula (3).

The word-level context vector $\mathbf{ca}_w$ is randomly initialized. After training with large enough dataset, $\mathbf{ca}_w$ is able to distinguish informative words in tweets for sentiment classification task and gives the tweet a better distributed representation. To further compose the representation of tweet conversation, we need the help of tweet-level context attention.

Tweet-level context attention Given a microblog conversation sequence that composed of n tweets $\{m_1, m_2, \cdots, m_n\}$ and the target tweet for sentiment analysis is m_a where $1 \leq a \leq n$. We feed the preceding tweets $pre(m_a)$ and m_a into CA-LSTM sequentially. Similarly, the tweet-level context attention vector $\mathbf{ca}_m$ can also help us to reward

the context tweets that are crucial for inferring sentiment of the target tweet. Hence, given the tweet-level context attention vector $\mathbf{ca}_m$ and target tweet m_a, we have the conversation sequence s:

$$\mathbf{s} = \sum_{i=1}^{a} \beta_i \mathbf{m}_i \tag{6}$$

$$\beta_i = \frac{\exp(f(\mathbf{ca}_m, \mathbf{m}_i))}{\sum_{k=1}^{a} \exp(f(\mathbf{ca}_m, \mathbf{m}_k))} \tag{7}$$

$$f(\mathbf{ca}_m, \mathbf{m}_i) = \mathbf{ca}_m^{\mathrm{T}} \tanh(\mathbf{W}_m \mathbf{m}_i + \mathbf{b}_m) \tag{8}$$

where $\mathbf{W}_m$ is weight matrix, $\mathbf{b}_m$ is bias vector, and $\mathbf{m}_i$ is the context-aware hidden representation of tweet m_i in sequence $\{m_1, m_2, \cdots, m_a\}$ with m_a as the tail tweet. Therefore, we leverage Formula (6) to embed the conversation sequence (till m_a) as $\mathbf{s}$ by the weighted sum of the tweet hidden representations. The normalized importance weight β is calculated by the context attention vector $\mathbf{ca}_m$ and tweet embedding $\mathbf{m}_i$ through a softmax function. Similarly, $\mathbf{ca}_m$ is randomly initialized and learned using training dataset.

4.3 Context-aware sentiment classification

Based on CA-LSTM model, the enhanced vector $\mathbf{s}$ considers not only the features in target tweet m_a, but also the sequential context information in the preceding tweets, which gives target tweet a better context-aware distributed representation. With the help of the context attention, CA-LSTM model can automatically learn informative words and tweets with long-range dependency for context-aware sentiment classification task. We feed the distributed representation vector $\mathbf{s}$ into a softmax layer to obtain sentiment distribution of target tweet.

$$p_c = \frac{\exp(\boldsymbol{\theta}_c^{\mathrm{T}} \mathbf{s} + \mathbf{b}_c)}{\sum_{k=1}^{C} \exp(\boldsymbol{\theta}_k^{\mathrm{T}} \mathbf{s} + \mathbf{b}_k)} \tag{9}$$

where C is the number of sentiment classes, $\mathbf{b}_c$, $\mathbf{b}_k$ are bias items, and p_c is the predicted probability of sentiment class c. We utilize cross-entropy as the loss function for training CA-LSTM model. Specially, the probability of gold standard sentiment distribution p_c^g is multiplied by the logarithm of CA-LSTM predicted probability p_c. Given training tweet dataset D with gold standard labels, we intend to optimize cross-entropy loss function as follows.

$$L = -\sum_{m \in D} \sum_{c=1}^{C} p_c^g(m) \cdot \log(p_c(m)) \tag{10}$$

where m is training tweet in D. The CA-LSTM models are trained by back propagation through time (BPTT) [8]. It is worth noting that different from the traditional tweet classification task, the training instances in D of context-aware sentiment classification are not independent with each other and D are composed by a number of conversation threads. In each training step, the target tweet with its preceding ones are fed into the model.

5 Experiment

In this section, we first give a brief introduction of a benchmark dataset for context-aware microblog sentiment classification task. Then we describe our experiment settings and several baseline algorithms including the state-of-the-art method for comparisons. Finally, we

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%

introduce the empirical results with corresponding discussions. Our experiment is conducted on a commodity PC Server with Ubuntu 14.10, Intel Xeon CPU E5-2680 v3 and 8GB RAM.

5.1 Dataset

We evaluate the effectiveness of our proposed CA-LSTM model on COAE-2015 Context-Sensitive Microblog Sentiment Classification Task dataset.[3] The official training set that crawled from the largest Chinese microblogging platform Weibo.com contains 2,800 microblogs with *Positive*, *Negative* and *Neutral* labels. The organizers of the COAE evaluation task have provided a benchmark testing set. However, they do not disclose the gold standard labels of the microblogs in the testing set. Therefore, we ask two graduate students who majoring in opinion mining to manually label part of the testing data. There are about 65% of the results having consistent sentiment labels, and we remove the remaining microblogs with inconsistent annotations. Finally we have 4,248 labeled Chinese microblogs, which belong to 555 topic threads formed by retweet and reference '@' relationships.

In final dataset, 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 sequential conversations have the length of two or three. But there are still quite a number of conversations contain more than three sequential microblogs.

There are about 12 words on average for each microblog. That means most of microblogs in the dataset are very short. After concatenating preceding tweets, each input conversation has about 58 words on average. In the dataset, 86.9% data are 'non-root' tweets whose embedded sentiments are usually context-aware.

The statistics information of sentiment polarity drift between the root microblog and the retweet/reference microblog are shown in Table 2. The horizontal header represents root microblog polarity information and the vertical header represents retweet/reference microblog polarity information. The contents of table are the numbers of microblogs in corresponding categories. We can see that a lot of microblogs retain root microblogs' polarities, but there are still many microblogs have completely reversed orientations.

We also make statistics on sentiment drift between microblog and their preceding adjacent neighbor in the conversation chain, as shown in Table 3. It can been seen that about 45% microblogs have changed polarities compared with their neighbors in the conversations. These polarity drifts are indicated by features such as contrastive connectives or context dependencies, which we think can be captured by our CA-LSTM model.

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

Table 2 The sentiment drift information of retweet/reference and the root microblog

Retweet/Reference \ Root	*Positive*	*Neutral*	*Negative*
Positive	725	629	52
Neutral	287	908	141
Negative	126	554	217

Finally we select 4/5 of the whole dataset as training set and the remaining as testing set. The dataset used in this paper has been made public.[4]

5.2 Experiment setup

The proposed CA-LSTM and other baseline RNN models are implemented by Theano.[5]s The input of CA-LSTM models are 100 or 200 dimensions word embedding vectors constructed by selecting the 10,000 most frequent features in the dataset. Note that we do not need to conduct complex feature engineering work or add manually designed syntactic features into the model. For regularization, early stopping and Dropout [10] are used during the training procedure. In CA-LSTM model, we compared the impact of Dropout and weight regularization mechanism on the neural network structures. Finally, based on preliminary empirical results we set the Dropout rate to 0.5 on the input-to-hidden layer and hidden-to-output layer with the L2 regularization method. The classification results are measured by metrics Accuracy, Precision, Recall and MacroF1.

5.3 Comparison methods

We compare our CA-LSTM model with several baselines including the state-of-the-art method for context based microblog sentiment classification. The comparison methods not only include the traditional classification algorithms such as SVM and CRF, but also include recently published variants of RNN models, such as vanilla RNN, GRU, Bidirectional LSTM and so on.

Bag-of-words and SVM We build a Support Vector Machine (SVM) based classifier on training dataset. 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 machine learning tool set Scikit-learn[6] to get the document-term matrix and calculate the TFIDF weight respectively. The SVM is implemented by LinearSVC function in Scikit-learn.

[4] http://github.com/leotywy/coae2015.

[5] http://deeplearning.net/software/theano.

[6] http://scikit-learn.org/stable/.

Table 3 The sentiment drift information of the target and preceding adjacent microblog

Target Tweet \ Preceding Tweet	*Positive*	*Neutral*	*Negative*
Positive	764	286	120
Neutral	509	855	466
Negative	60	157	338

Conditional random fields CRFs [14] are a class of statistical modeling methods often applied in pattern recognition and machine learning. The CRFs are usually used for structured prediction, whose the most common use-case is to predict the chain-structured outputs. These occur naturally in sequence labeling tasks, such as Part-of-Speech tagging, shallow semantic parsing or word segmentation in natural language processing.

We regard the microblog conversation branch with the retweet/reference relationship as input sequence, and the sentiment polarity label of each microblog in the conversation branch as output sequence. We use the linear chain CRF model to solve context-aware sentiment classification problem. Each microblog is a node in the chain, and the microblogs with retweet/reference relationship are connected with an edge. Therefore, the conversation branch described in Section 3 has formed the chain in CRF model. The length of the chain varies according to the number of microblogs with retweet/reference relationship, just the same as conversation branches in Figure 2. The CRF model with linear chain structure is implemented by Pystruct[7] package.

Note that during training process, each node in CRF chain is associated with a sentiment label and in the testing set we can predict the labels of all the nodes in a conversation branch at the same time. This is a different learning framework from the other models in this paper, where we only add the contents of the preceding microblogs into the target one and ignore preceding microblogs' labels. Furthermore, in the other models we only predict the label of the target microblog in the conversation during the testing procedure.

Gated recurrent units (GRU) In order to make a comparison with other deep learning methods, we utilize an alternative RNN model called GRU [8] to train a sentiment classifier. GRU is designed to have more persistent memory thus theoretically making it easier to capture long-term dependencies. The GRU model has two types of gates: the update gate z_t and the reset gate r_t as follows.

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}^z \mathbf{x}_t + \mathbf{U}^z \mathbf{h}_{t-1} + \mathbf{b}^z) \\ \mathbf{r}_t &= \sigma(\mathbf{W}^r \mathbf{x}_t + \mathbf{U}^r \mathbf{h}_{t-1} + \mathbf{b}^r) \\ \mathbf{g}_t &= \tanh(\mathbf{W}^g \mathbf{x}_t + \mathbf{r}_t \odot \mathbf{U}^g \mathbf{h}_{t-1} + \mathbf{b}^g) \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{g}_t + \mathbf{z}_t \odot \mathbf{g}_{t-1} \end{aligned} \tag{11}$$

where $\sigma(\cdot)$ is sigmoid function and $\tanh(\cdot)$ is hyperbolic tangent function. $\odot$ is element-wise multiplication operator. Since the matrix $\mathbf{W}$ and $\mathbf{U}$ are smaller, GRUs have fewer parameters to train and the training time can be less than the vanilla LSTM model.

Bidirectional long short term memory (BLSTM) We utilize the BLSTM model to make full use of the past and future context information. The basic idea of BLSTM is to

[7] http://pystruct.github.io/.

connect two LSTM hidden layers of opposite directions (forward hidden sequence $\overrightarrow{\mathbf{h}}$ and backward hidden sequence $\overleftarrow{\mathbf{h}}$) to the same output. Because the BLSTM 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 [33].

The general bidirectional RNN model can be represented as the following equations:

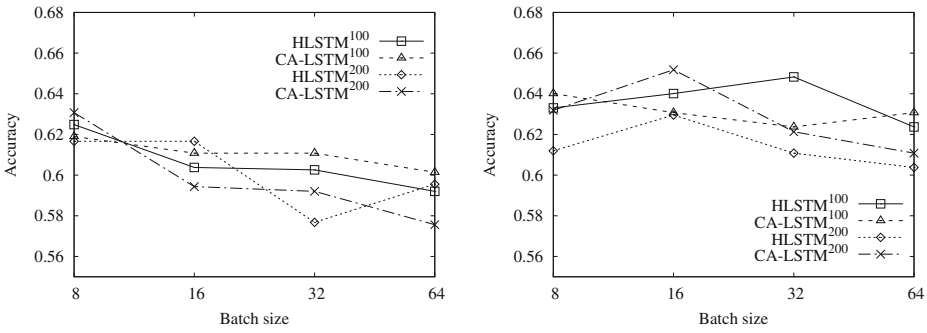
$$\begin{aligned} \overrightarrow{\mathbf{h}_t} &= f(\mathbf{W}_{x\overrightarrow{h}}\mathbf{x}_t + \mathbf{U}_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overrightarrow{h}}) \\ \overleftarrow{\mathbf{h}_t} &= f(\mathbf{W}_{x\overleftarrow{h}}\mathbf{x}_t + \mathbf{U}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{\mathbf{h}}_{t-1} + \mathbf{b}_{\overleftarrow{h}}) \\ P(y=j|s) &= \frac{\exp(\overrightarrow{\mathbf{h}_t}\cdot\overrightarrow{\mathbf{p}}^j + \overleftarrow{\mathbf{h}_t}\cdot\overleftarrow{\mathbf{p}}^j + \mathbf{q}^j)}{\Sigma_{j'\in\{-1,0,1\}}\exp(\overrightarrow{\mathbf{h}_t}\cdot\overrightarrow{\mathbf{p}}^{j'} + \overleftarrow{\mathbf{h}_t}\cdot\overleftarrow{\mathbf{p}}^{j'} + \mathbf{q}^{j'})} \end{aligned} \tag{12}$$

where $\mathbf{p}^j$ is the j-th column of $\mathbf{P} \in \mathbb{R}^{d\times 3}$ and q^j is the j-th element of $\mathbf{q} \in \mathbb{R}^3$. $\mathbf{P}$, $\mathbf{q}$, $\mathbf{U}$, $\mathbf{W}$ are parameters of the model to be learned during training strategy. $\{-1, 0, +1\}$ represents categories $\{Negative, Neutral, Positive\}$ respectively. By replacing the hidden states in the general bidirectional RNN with LSTM memory blocks, we can get the aforementioned BLSTM models for context-aware sentiment classification task.

5.4 Hyper-parameter tuning

Hyper-parameter tuning is a crucial step for deep learning models. Firstly, we explore the impacts of hyper-parameters batch size and word vector dimension in CA-LSTM model as well as different word vector initialization settings. The dimensions of hidden states and memory cell states in our LSTM units are set to 200. AdaDelta algorithm [39] is utilized to update parameters during training. The training epochs of HLSTM and CA-LSTM models are set to 2 because the preliminary experiment results show that using the early stopping method will be a good choice on this small training dataset. In the following experiments, we fix the Dropout rate to 0.5 and the impacts of batch size and word vector dimension are illustrated in Figure 6.

The LSTM models for text classification employ the word vectors as inputs and the word embedding methods have great effects on final classification results of LSTM models. In Figure 6, we study on the random and pre-trained word embedding methods. First of all,



(a) Glorot initialization word embeddings (b) Pre-trained word embeddings

Figure 6 The performances of HLSTM and CA-LSTM models for hyper-parameter tuning. The superscript number represents the dimension of the input word vectors. For example, the input word vectors in $\mathrm{CA-LSTM}^{100}$ have 100 dimensions

we randomly initialize input word vectors by Glorot normalized initialization method [6] where the scale of the data is determined by the size of the word embedding matrix.

$$scale = \sqrt{\frac{6}{fan_{in} + fan_{out}}} \tag{13}$$

In Formula (13), fan_{in} and fan_{out} represent the number of words in the glossary and the dimension of the word vectors respectively. Based on Formula (13), the word vectors are initialized by random values with uniform distribution in the interval $[-scale, scale]$. The experiment results with different hyper-parameter settings using Glorot random initialization word embeddings are shown in Figure 6a.

In deep learning models, batch size defines the number of training instances that going to be propagated through the network. Figure 6a depicts that when the batch size grows bigger, the accuracies of both models decrease slightly. The best performance is achieved by CA-LSTM model using 200 dimensional word vectors with batch size 8.

We also pre-train the 100/200 dimensional word embeddings on our COAE-2015 training dataset using the CBOW model [19], and the experiment results are shown in Figure 6b. Generally speaking, the models with pre-trained word vectors have achieved better accuracies than the models with randomly initialized word vectors. In Figure 6b when the batch size grows, the accuracies fail to show clear upward or downward trending. The best performance is achieved by CA-LSTM model using 200 dimensional pre-trained word vectors with batch size 16.

Secondly, we study the effects of Dropout and L2 regularization on LSTM models. Dropout is a technique that can randomly drop hidden units from neural network during training, which together with regularization can effectively prevent the neural network from overfitting problem [25]. Figure 7 depicts the best performances achieved by different LSTM models with/without Dropout and L2 regularization techniques.

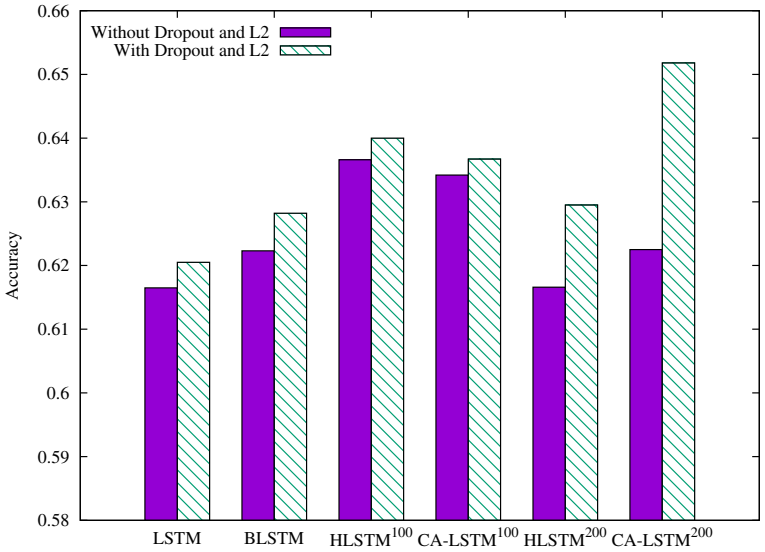


Figure 7 The performances of LSTM models trained with different Dropout and L2 regularization settings. The superscript number represents the dimension of the input word vectors. For example, the input word vectors in $CA-LSTM^{100}$ have 100 dimensions

It can been seen from Figure 7 that Dropout and L2 regularization techniques can significantly improve the performances of all the LSTM models, especially when the models have larger word vectors. Compared with HLSTM^{100} and $\text{CA}-\text{LSTM}^{100}$, the accuracies of HLSTM^{200} and $\text{CA}-\text{LSTM}^{200}$ are much worse without Dropout and L2 regularization. However, when using Dropout and L2 regularization techniques, the performances of the models with 200 dimensional word embeddings have been dramatically improved. This is because larger word embeddings have brought in more parameters and consequent overfitting. The Dropout technique along with L2 regularization can prevent LSTM units from co-adapting too much by making the presence of any particular hidden unit unreliable and thus alleviate the overfitting problem.

5.5 Experiment results

We compare the results of our proposed CA-LSTM model with the traditional methods and the state-of-the-art LSTM models for context based microblog sentiment classification. The hyper-parameters are tuned on all these models using training dataset to achieve each model's the best performance. For the bag-of-word features in SVM, we consider the target microblog alone as well as combining the target microblog with all of its preceding tweets. For the RNN models, we compare the performances of vanilla RNN, GRU, LSTM, BLSTM, HLSTM. The experimental results are shown in Table 4.

Table 4 Experiment results of the traditional methods and RNN models

Model	*Context*	Accuracy	Precision	Recall	MacroF1
SVM-a	No	0.6235	0.6134	0.6079	0.6082
	Yes	0.5847	0.5699	0.5677	0.5683
SVM-b	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	No	0.5800	0.5722	0.5718	0.5702
	Yes	–	–	–	–
GRU	No	0.5894	0.5959	0.5839	0.5794
	Yes	0.6188	0.6059	0.6012	0.6019
LSTM	No	0.6153	0.6100	0.5900	0.5815
	Yse	0.6205	0.6145	0.6058	0.6057
BLSTM	No	0.6118	0.5921	0.5855	0.5802
	Yes	0.6282	0.6196	0.6189	0.6183
$\text{HLSTM}^{\S}$	Yes	0.6249	0.6119	0.6119	0.6117
HLSTM	Yes	0.6401	0.6123	0.6320	0.6101
$\text{CA-LSTM}^{\S}$	Yes	0.6307	0.6336	0.6344	0.6271
CA-LSTM	Yes	**0.6518**	**0.6368**	**0.6369**	**0.6362**

The symbol § in HLSTM and CA-LSTM models denotes that the input word embeddings are randomly initialized with Glorot's method. The HLSTM and CA-LSTM models without symbol § mean that their input word embeddings are pre-trained on the COAE-2015 training dataset

The bold emphasis means that the value achieves the best performance in the corresponding category

In the Table 4, the *context* column represents whether to incorporate preceding tweets or not. SVM-a and SVM-b denote the SVM algorithms that using 1 to 3-gram as features and 1 to 5-gram as features respectively. Table 4 shows that as strong baseline SVM algorithm can achieve a relatively good performance when using 1 to 3-gram as features without context information. However, when incorporating preceding tweets, the performances decrease dramatically. This may be because that bag-of-word based SVM algorithm does not consider the order of words and sentences in microblogs. Hence, including preceding tweets has brought in more noise instead of useful features for the classifier. The performance of CRF based method is dramatically dragged down by the sparse feature space and limited training instance sequences.

Different from bag-of-word based SVM algorithm, all of the Recurrent Neutral Network based methods can achieve better results when incorporating context information. This interesting observation indicates that RNN based methods are good at modeling context-aware sentiments in microblog conversations. Note that in Table 4 the vanilla RNN does not have a context-aware results because when considering preceding tweets, the training procedure does not converge in two epoches for vanilla RNN. The HLSTM and CA-LSTM models only have the context-aware results because these two models naturally embed context information using the hierarchical structure.

As we discussed in the hyper-parameter tuning section, we can randomly initialize input word embeddings or pre-train word embeddings using training corpus. In Table 4, the symbol § in HLSTM and CA-LSTM models denotes that input word embeddings are randomly initialized with Glorot's method [6]. The HLSTM and CA-LSTM models without the symbol § mean that their input word embeddings are pre-trained on the COAE-2015 training dataset with CBOW [19].

In Table 4, the GRU and LSTM models are able to map the word sequences into fixed-length word vectors and capture the context-aware sentiments in microblogs with long distance dependencies. The performance of GRU and LSTM is quite close, but still worse than that of the BLSTM [33] which considers both the past and the future context information for every timestep in the input sequence. The reported state-of-the-art method HLSTM [11] outperforms our previous work BLSTM [33] by considering the hierarchical conversation structure. The best performance is achieved when using our proposed CA-LSTM model with pre-trained word vectors. To be specific, CA-LSTM achieves more than 4% MacroF1 relative improvements over state-of-the-art model HLSTM and the other well-designed Recurrent Neural Network models. These results demonstrate that with the help of attention mechanism, CA-LSTM model is more effective to capture the word orders and the hierarchical context information in microblog conversations, which is crucial for the context-aware sentiment classification task.

It is observed that generally the models with pre-trained word embeddings (CA-LSTM, HLSTM) outperform their counterparts (CA-LSTM§, HLSTM§) with random Glorot initialization. We argue that the pre-trained word embeddings can alleviate the sparsity problem

Table 5 The statistics of HLSTM and CA-LSTM classification results

# of test data	850
# of both HLSTM and CA-LSTM are correct	493
# of HLSTM succeeds while CA-LSTM fails	42
# of HLSTM fails while CA-LSTM succeeds	61
# of both HLSTM and CA-LSTM are wrong	264

of feature space and give neural model a better training initialization with reasonable topic semantics. Note that in Table 4 even without the pre-trained vectors, our proposed CA-LSTM still outperforms the strong baseline methods BLSTM and LSTM.

We give in-depth analysis of the classification results of HLSTM and CA-LSTM models, as shown in Table 5. CA-LSTM has more microblogs that are correctly classified. But there are still a number of microblogs that HLSTM succeeds while CA-LSTM fails.

5.6 Case study

To demonstrate the effectiveness of our context attention model, we provide a microblog conversation sequence in COAE-2015 testing dataset for example. Table 6 shows the content text of the micrblogs and their retweet relationships in the example conversation.

In Table 6, there are six users who are discussing new released iPhone 6. The tweet ID is started at 0; The Parent column represents ID that current microblog retreet where '-1' denotes that the current microblog is the root; The Child column represents IDs that retweet the current microblog where '–' denotes that no more microblog retweet the current tweet. Therefore, the retweet relationship in the conversation of Table 6 can be illustrated as Figure 8.

In Table 6, Tweet 0 is an informative microblog that is mainly about the iPhone 6 release information and the price. Tweet 0 is labeled as Neutral because it only presents the facts. Tweet 1 and 3 retweet Tweet 0 with their own negative comments, whose polarities are obscure and context-aware. For example, Tweet 3 particularly employs an ironic tone to express negative sentiments with extremely limited words (You have to sell two kidneys for buying a new expensive iPhone). Similarly, Tweet 2 conveys positive sentiment while Tweet 4 and 5 express negative sentiments. The short length of the these tweets and the resulting semantic ambiguity are critical limitations and make the classification task very complex.

Table 6 The example microblog conversation with retweet relationship and polarity labels

User	ID	Parent	Child	Text	Label
A	0	−1	1,3	*Today, iPhone 6 has been released at ZOL.com with price as low as 14,000 Yuan. The Deluxe Edition's price is up to 20,000 Yuan.*	Neutral
B	1	0	2	*So what? It only has a good operation system. Why is it so expensive?*	Negative
C	2	1	–	*Finally I can use my iPhone 6 that bought two month ago in USA. Excellent!*	Positive
D	3	0	4	*Please prepare two kidneys...*	Negative
E	4	3	5	*From iPhone 1 to 5S, I have used seven or eight kidneys. So two more this time?*	Negative
F	5	4	–	*If I win an iPhone 6 as a lottery prize, I will definitely sell it. iPhone 6 is too extravagant for me.*	Negative

The text that talking about the new released iPhone are translated from Chinese. It was reported that a Chinese teen sold his kidney to buy a new expensive iPhone. User D and E employed ironic tone and satirized the expensive price

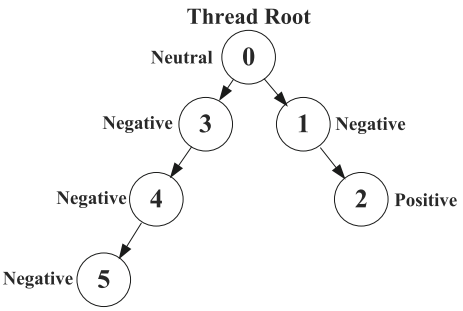


Figure 8 The corresponding conversation tree of the example tweets in Table 6

We utilize the trained BLSTM, HLSTM, CA-LSTM models to classify sentiment polarities of these tweets and the final results are shown in Table 7.

In Table 7, the text in bold means that the classification result is inconsistent with the gold standard label. For Tweet 1 all the models give the wrong results, which are mainly because of the strong but misleading sentiment word *good* in Tweet 1. All the models have correctly classified the Tweet 2 in which the user expresses the sentiments completely and directly. However, without context information, it is really difficult to identify the sentiment orientations of Tweet 3 and 4 since there are no obvious sentiment words in these short microblogs. The HLSTM and CA-LSTM models can capture long-range dependencies between the words in target tweet and the preceding ones, and successfully infer topic semantic and emotional tendency of traditional non-sentiment word with the help of pre-trained word embeddings. To further illustrate the context attention mechanism, we visualize the word-level and tweet-level attention weights of CA-LSTM model for the example microblog conversation in Figure 9.

In Figure 9, the rounded rectangular boxes represent microblogs, whose retweet relationships are represented by the arrows. The black rectangles with different gray scale values denote the tweet-level attention weights and the blue color on words represents the word-level attention weight. The darker color means the higher weight and these attention weights are calculated when analyzing the last tweet in the conversation branch. It is clear that CA-LSTM model locates the important and discriminative words, such as *excellent*, *extravagant*, as well as the corresponding tweets for sentiment classification task. In particular, CA-LSTM model can discover the words that contain context-sensitive meanings. For example, the original non-sentiment word *kidney* conveys obscure negative sentiments in the example conversation, and this phenomenon is accurately localized by our CA-LSTM model.

Based on the experiment results and the case study, we argue that the proposed CA-LSTM model with pre-trained word embeddings is able to alleviate the sparsity problem

Table 7 Classification results of the example tweets using different models

ID	Label	BLSTM	HLSTM	CA-LSTM
0	Neutral	Neutral	Neutral	Neutral
1	Negative	**Neutral**	**Positive**	**Neutral**
2	Positive	Positive	Positive	Positive
3	Negative	**Neutral**	Negative	Negative
4	Negative	Negative	Negative	Negative
5	Negative	Negative	**Neutral**	Negative

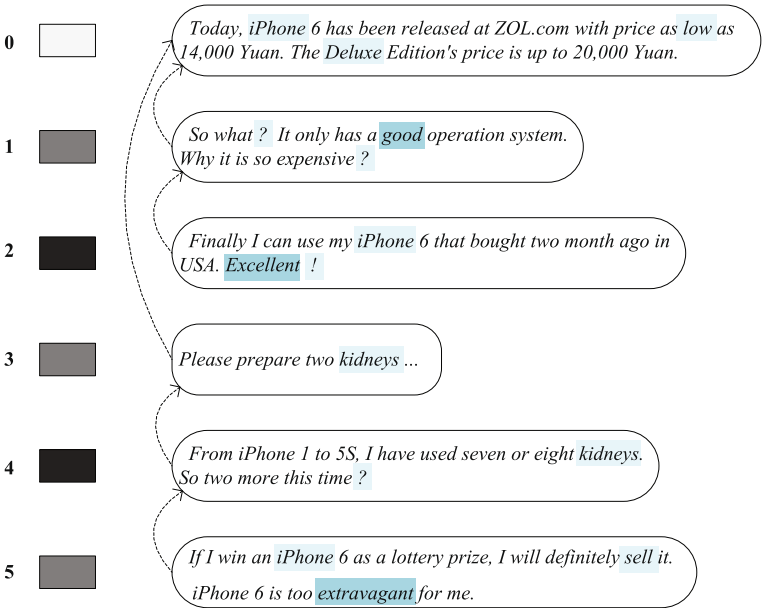


Figure 9 The word and tweet-level attention weights in CA-LSTM for the example microblog conversation. It was reported that a Chinese teen sold his kidney to buy a new expensive iPhone. User D and E employed ironic tone and satirized the expensive price. The darker color means the higher weight. This figure is best viewed in color

of the short text. The hierarchical long short-term memory structure with attention mechanism can discover the long-range topic and sentiment dependencies, and furthermore infer emotional tendencies of the context-sensitive words.

6 Conclusion

In this paper, we propose a CA-LSTM model which incorporates preceding tweets with attention weights for context-aware microblog sentiment classification. With the help of the word-level and tweet-level attentions, our CA-LSTM can capture the long distance dependency in microblog conversation and infer the sentiment polarities of context-sensitive words and tweets. The extensive experiments on a public available benchmark dataset demonstrate that our model achieves consistent improvements compared to other baselines and the state-of-the-art models.

The same word may have different sentiment meanings in different topic context. In the future, we intend to explicitly incorporate the topic information into the context attention vector for context-aware sentiment classification. We will also explore the effectiveness of our model on the emerging new task stance detection, which is also context-sensitive and topic-relevant.

Acknowledgments The work was 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. Chen, H., Sun, M., Tu, C., Lin, Y., Liu, Z.: Neural sentiment classification with user and product attention. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1650–1659 (2016)
2. Cheng, J., Zhang, X., Li, P., Zhang, S., Ding, Z., Wang, H.: Exploring sentiment parsing of microblogging texts for opinion polling on Chinese public figures. Appl. Intell. **45**(2), 429–442 (2016)
3. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. In: Proceedings of Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems (NIPS), pp. 2980–2988 (2015)
4. Fernández, S., Graves, A., Schmidhuber, J.: Sequence labelling in structured domains with hierarchical recurrent neural networks. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), pp. 774–779 (2007)
5. Frege, G.: On sense and reference. Philos. Rev. **57**(3), 1296–1323 (2010)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. J. Mach. Learn. Res. **9**, 249–256 (2010)
7. Goldberg, Y.: A primer on neural network models for natural language processing. arXiv:1510.00726 (2015)
8. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: A search space odyssey. arXiv:1503.04069 (2015)
9. Hihi, S.E., Bengio, Y.: Hierarchical recurrent neural networks for long-term dependencies. In: Proceedings of Advances in Neural Information Processing Systems 8 (NIPS), pp. 493–499 (1995)
10. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580 (2012)
11. Huang, M., Cao, Y., Dong, C.: Modeling rich contexts for sentiment classification with LSTM,. arXiv:1605.01478 (2016)
12. Katz, G., Ofek, N., Shapira, B.: Consent: Context-based sentiment analysis. Knowl.-Based Syst. **84**, 162–178 (2015)
13. Kwak, H., Lee, C., Park, H., Moon, S.B.: What is twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web (WWW), pp. 591–600 (2010)
14. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML), pp. 282–289 (2001)
15. Li, D., Shuai, X., Sun, G., Tang, J., Ding, Y., Luo, Z.: Mining topic-level opinion influence in microblog. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM), pp. 1562–1566 (2012)
16. Li, J., Luong, T., Jurafsky, D., Hovy, E.H.: When are tree structures necessary for deep learning of representations? In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2304–2314 (2015)
17. Lu, Y., Castellanos, M., Dayal, U., Zhai, C.: Automatic construction of a context-aware sentiment lexicon: an optimization approach. In: Proceedings of the 20th International Conference on World Wide Web (WWW), pp. 347–356 (2011)
18. McDonald, R.T., Hannan, K., Neylon, T., Wells, M., Reynar, J.C.: Structured models for fine-to-coarse sentiment analysis. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 432—-439 (2007)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv:1301.3781 (2013)
20. Mnih, A., Hinton, G.E.: A scalable hierarchical distributed language model. In: Proceedings of Advances in Neural Information Processing Systems 21: Annual Conference on Neural Information Processing Systems (NIPS), pp. 1081–1088 (2008)
21. Muhammad, A., Wiratunga, N., Lothian, R.: Contextual sentiment analysis for social media genres. Knowl.-Based Syst. **108**, 92–101 (2016)
22. Mukherjee, S., Bhattacharyya, P.: Sentiment analysis in twitter with lightweight discourse analysis. In: Proceedings of the Conference 24th International Conference on Computational Linguistics (COLING), pp. 1847–1864 (2012)
23. Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowl.-Based Syst. **89**, 14–46 (2015)
24. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Context-sensitive twitter sentiment classification using neural network. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI), pp. 215–221 (2016)

25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems (NIPS), pp. 3104–3112 (2014)
27. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP), pp. 1556–1566 (2015)
28. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 1397–1405 (2011)
29. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1422–1432 (2015)
30. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 417–424 (2002)
31. Vanzo, A., Croce, D., Basili, R.: A context-based model for sentiment analysis in twitter. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING), pp. 2345–2354 (2014)
32. Wang, X., Liu, Y., Sun, C., Wang, B., Wang, X.: Predicting polarities of tweets by composing word embeddings with long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP), pp. 1343–1353 (2015)
33. Wang, Y., Feng, S., Wang, D., Zhang, Y., Yu, G.: Context-aware chinese microblog sentiment classification with bidirectional LSTM. In: Proceedings of 18th Asia-Pacific Web Conference (APWeb), pp. 594–606 (2016)
34. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 606–615 (2016)
35. Wu, F., Huang, Y., Song, Y.: Structured microblog sentiment classification via social context regularization. Neurocomputing **175**, 599–609 (2016)
36. Wu, F., Song, Y., Huang, Y.: Microblog sentiment classification with contextual knowledge regularization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI), pp. 2332–2338 (2015)
37. Yang, B., Cardie, C.: Context-aware learning for sentence-level sentiment analysis with posterior regularization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 325–335 (2014)
38. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pp. 1480–1489 (2016)
39. Zeiler, M.D.: Adadelta: An adaptive learning rate method. arXiv:1212.5701 (2012)
40. Zhao, Z., Lu, H., Cai, D., He, X., Zhuang, Y.: Microblog sentiment classification via recurrent random walk network learning. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI), pp. 3532–3538 (2017)
41. Zhou, X., Wan, X., Xiao, J.: Attention-based LSTM network for cross-lingual sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 247–256 (2016)
42. Zhu, X., Sobhani, P., Guo, H.: Long short-term memory over recursive structures. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 1604–1612 (2015)