Neurocomputing 308 (2018) 1–7

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Context-augmented convolutional neural networks for twitter sarcasm detection

Yafeng Ren[a,*], Donghong Ji[a,b], Han Ren[c,*]

[a] Guangdong Collaborative Innovation Center for Language Research & Services, Guangdong University of Foreign Studies, Guangzhou 510420, China
[b] Computer School, Wuhan University, Wuhan 430072, China
[c] Laboratory of Language Engineering and Computing, Guangdong University of Foreign Studies, Guangzhou 510420, China

ARTICLE INFO

Article history:
Received 7 April 2017
Revised 14 August 2017
Accepted 18 March 2018
Available online 7 May 2018

Communicated by Dr. Y. Chang

Keywords:
Twitter sarcasm detection
Contextual information
Discrete features
Convolutional neural network

ABSTRACT

Sarcasm detection on twitter has received increasing research in recent years. However, existing work has two limitations. First, existing work mainly uses discrete models, requiring a large number of manual features, which can be expensive to obtain. Second, most existing work focuses on feature engineering according to the tweet itself, and does not utilize contextual information regarding the target tweet. However, contextual information (e.g. a conversation or the history tweets of the target tweet author) may be available for the target tweet. To address the above two issues, we explore the neural network models for twitter sarcasm detection. Based on convolutional neural network, we propose two different context-augmented neural models for this task. Results on the dataset show that neural models can achieve better performance compared to state-of-the-art discrete models. Meanwhile, the proposed context-augmented neural models can effectively decode sarcastic clues from contextual information, and give a relative improvement in the detection performance.

1. Introduction

With the development of social media, twitter has become one of the most popular micro-blog services. There are large amounts of valuable information in twitter. So sentiment analysis and data mining based on twitter data has become a heated research topic [1–3]. Recently, sentiment analysis in twitter has received extensively attentions [4–6]. The purpose of twitter sentiment analysis is to automatically analyze the polarity of a tweet. However, sarcastic utterance in twitter can transform the polarity of positive or negative utterance into its opposite. To some extent, this affects the performance of sentiment analysis task. So it is very important to distinguish sarcastic statement from the utterances with positive or negative polarity.

Twitter sarcasm detection has attracted increasing research in the past few years [7–11]. Generally, twitter sarcasm detection task is regarded as a classification problem. Previous work mainly focuses on designing effective features to improve the detection performance according to the tweet content itself. For example, Barbieri et al., uses rich linguistically-motivated features from the target tweet for twitter sarcasm detection [12]. Typically, the used features refer to lexical features (n-gram information), punctuation marks, quotes, emoticons, pronunciations, tweet sentiment information, and word sentiment formation. Moreover, some work tries to design more sophisticated features by using external resources. The used features include POS (Part of Speech) tags, Brown clusters and dependency-based tree structures [13].

The above work has two limitations. First, these work relies on discrete models, requiring large number of manual features, which can be expensive to obtain. Second, these work focuses on designing rich features according to the target tweet itself, which does not utilize contextual information regarding the target tweet or the tweet author. This limits the performance of the task. However, tweets may contain some available contextual information, which includes a conversation or the history tweets of the tweet author. For example, given the following tweet posted by **Erik_in_Raleigh**, which cites **syydsand** and **gretchlol**:

- **Erik_in_Raleigh**: **@syydsand @gretchlol** this seems like a lie, do you no longer associate with yourself?

Does the tweet tend to sarcastic? It is very hard to know if no contextual information is available. However, if contextual information is given, which is shown as follows:

- **syydsand**: I literally will not associate myself with anyone who lies. worst quality ever. (2015-01-26 03:32:57)

* Corresponding authors.
E-mail address: renyafeng@whu.edu.cn (Y. Ren).

https://doi.org/10.1016/j.neucom.2018.03.047
0925-2312/

- **Erik_in_Raleigh**: **@syydsand @gretchlol** this seems like a lie, do you no longer associate with yourself? (2015-01-26 03:53:44)

Here, the target tweet and its contextual tweet form a conversation. Based on this conversation, we can easily infer that this tweet is sarcastic. According to the above example, we know that twitter sarcasm detection can benefit from contextual information.

Recently, some work begins to use contextual features for sarcasm detection [13,14]. In particular, Rajadasingan et al. propose to model sarcasm detection by using a behavioral approach, using a set of statistical indicators extracted from the target tweet and history tweets [14]. In this work, they only use a type of contextual information: the history-based contexts. Different from this work, Wang et al. propose to model the target tweet and its contextual tweets as a sequence, using a sequence labeling algorithm to jointly predict their category labels [13]. In their work, they use two types of contextual information, including the conversation-based contexts and the history-based contexts. Consistent with the above example, these work suggests that contextual information is useful for twitter sarcasm detection. However, these methods mainly focus on discrete models with spare manual features, which can be expensive to obtain. Different from the above context-based models, we will explore the context-augmented neural network models for twitter sarcasm detection.

More recently, neural networks models have been successfully applied for many NLP (Natural Language Processing) tasks, achieving competitive results [15–18]. Excellent performance on these tasks shows potentials of neural network models for sarcasm detection. Compared to traditional discrete models, neural network models mainly have two advantages for sarcasm detection. One is neural layers can automatically induce features, avoiding manual feature engineering [15,16,18,19]. The second is neural models use real-valued word vectors, which can be trained from large scale raw texts, solving the feature sparsity problem of discrete models to some extent.

In this paper, we explore the context-augmented convolutional neural network models for twitter sarcasm detection based on three questions. First, neural representation can give strong performance for many NLP tasks [2,20,21], but it's not clear whether neural models can achieve better performance for twitter sarcasm detection. Second, we want to know whether the context-augmented neural models can capture more sarcastic clues from contextual tweets, comparing with discrete context-based models. Third, we want to explore the effects on different contextual information for twitter sarcasm detection. Apparently, the conversation-based contexts and the history-based contexts have different effects for capturing sarcastic evidence.

Results show that neural models outperform state-of-the-art discrete model, demonstrating the advantage of automatically capturing sarcastic clues. Moreover, the proposed context-augmented neural models further improve the detection performance, showing the usefulness of contextual information for twitter sarcasm detection.

2. Related work

In this section, we will introduce related work from two perspectives, including sarcasm detection and neural network models.

2.1. Sarcasm detection

Sarcasm detection has been extensively investigated in recent years. Generally, sarcasm detection task is treated as a standard text classification problem. Existing models mainly focus on designing effective features for improving the detection performance. For example, [22] studied lexical features for sarcasm detection [22], and found that interjections and punctuation were very effective for the task. Following this work, Carvalho et al. introduced an automatic system for sarcasm detection [23]. Note that this work focuses on sarcastic style detection in newspaper articles. Later, Veale and Hao [24] proposed to separate sarcastic from non-sarcastic similes [24].

With the development of social media, researchers begin to detect sarcasm in twitter. Typically, Gonzalez-Ibanez et al. explored the combinations of unigrams, dictionary-based features and pragmatic factors to achieve better performance by using SVM (Support Vector Machine) [8]. Recently, Reyes et al. proposed to perform twitter sarcasm detection by defining four groups of features: style, unexpectedness, signatures, and emotional scenarios [9]. Moreover, Barbieri and Saggion proposed to detect sarcasm in twitter by designing a linguistically motivated set of features [12].

External information sources have also been utilized to improve the detection performance. For example, Tsur et al. studied the features based on semi-supervised syntactic patterns [25]. Based on the work of Tsur and Davidov extracted similar features from sarcastic tweets [7]. Later, Riloff et al. focused on one type of sarcasm: contrast between a positive sentiment and negative situation [10]. They used a bootstrapping algorithm to acquire the tweets that meet the above conditions.

More recently, contextual features also has been used for sarcasm detection [13,26,27]. First, contextual features from history tweets of the tweet author has been shown the effectiveness for twitter sarcasm detection [14,28]. Second, the conversation-based context, which contains a certain number of tweets in a discussion thread, has also been shown the usefulness for improving the detection performance [13]. However, these methods with contextual information are all based on discrete models, requiring a large amount of hand-crafted features. In this paper, we explore the context-augmented neural network models for twitter sarcasm detection. Note that Wang et al. construct a dataset, which contains the conversation-based contexts and the history-based contexts, to evaluate the contribution of different contextual information for improving the detection performance. In this paper, we adapt this dataset as our experimental dataset to verify the effectiveness of contextual information in the neural network models.

2.2. Neural network models

Recently, neural network models have been used for learning dense feature representation for a variety of NLP tasks [6,19,21,29]. Many methods have been proposed to learn the representations of phrases, sentences and documents from distributed word representations [30]. For example, Le and Mikolov proposes to learn the paragraph vectors for representing the document by extending the word embedding methods of Mikolov and Le [31]. To learn sentence-level semantic compositionality, Socher et al. first introduced recursive neural networks [15]. Later, there has been a series of follow-up research based on recursive neural network of Socher et al.. For example, Irsoy et al. proposed a deep recursive neural network, which is constructed by stacking multiple recursive layers, and evaluated the proposed model on the task of fine-grained sentiment classification. Paulus et al. introduced a global belief recursive neural networks, and evaluated the effectiveness of this model on the task of contextual sentiment analysis.

In addition to recursive neural networks, convolutional neural networks (CNN) have been widely used for automatically capturing n-gram information in semantic composition [19,32], giving competitive results in various NLP tasks. For example, for modeling sentences, CNN gives the best performance for sentiment analysis task [17,32]. Besides, recurrent neural network (RNN) or long short-term memory (LSTM) have also been used for recurrent semantic

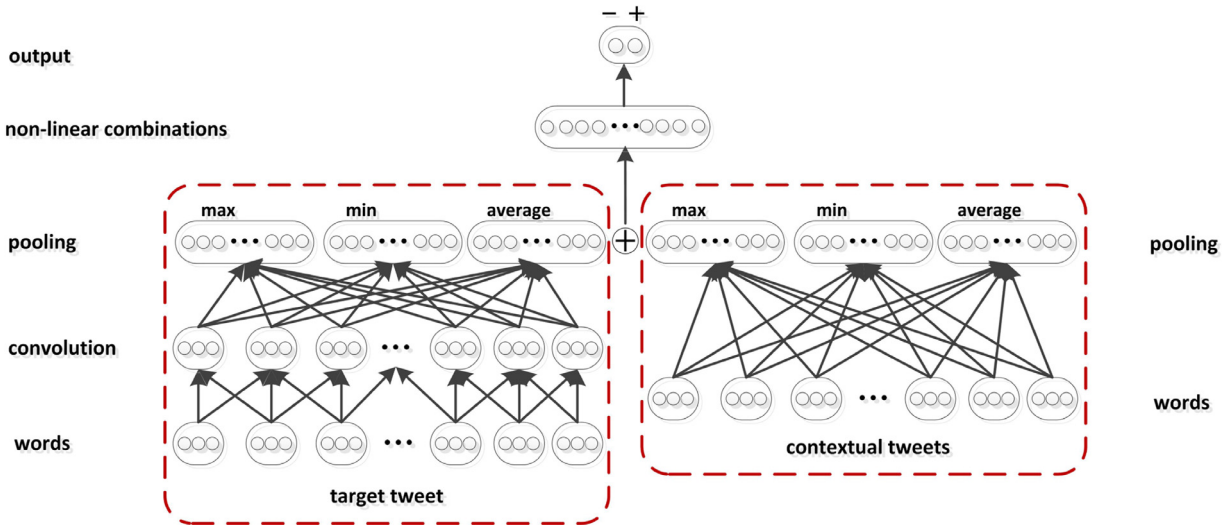

Fig. 1. The context-augmented neural network model CANN-KEY for twitter sarcasm detection.

composition [18,33]. For example, LSTM gives the best performance for question answering task [34]. CNN, RNN and LSTM are typically used for representing semantic composition of phrases, sentences and documents. However, RNN and LSTM have been used more frequently for modeling discourse structures [33,35]. In this paper, we use CNN models because this paper aims to detect the message-level sarcasm in twitter. Based on CNN models, we propose two different context-augmented neural models to fully utilize contextual information.

3. Approach

This paper aims to explore the neural network models for twitter sarcasm detection. In this paper, we explore the usefulness of contextual information from two aspects. Based on two types of contextual information, we propose two context-augmented neural network models to fully capture sarcastic cues from contextual information. First, we think that sarcastic evidence can be easily captured by using some key information of the history-based contexts. Intuitively, the number of the history-based context is relatively large for the target tweet. There are some redundant information in these tweets. So we only consider some keywords when we model the context-augmented neural network model for sarcasm detection. The first model is called the context-augmented neural model by integrating key contextual information (named CANN-KEY in this paper). Second, we think that sarcastic clues can be well captured by using all information of the conversation-based contexts. This is because only a small percentage of the target tweets carry the conversation-based contexts, which usually contain only 1–2 contextual tweets. So we consider all information when we model the context-augmented neural network model for sarcasm detection. The second model is called the context-augmented neural model by integrating all contextual information (named CANN-ALL in this paper).

3.1. The Model CANN-KEY

The model CANN-KEY is illustrated in Fig. 1, including two main parts. The left part is a local sub network, using the information from the target tweet, while the right part is a contextual sub network, using the information from contextual tweets. For contextual sub network, we will automatically induce the features from the conversation-based contexts and the history-based contexts, respectively.

In our proposed model CANN-KEY, the local sub network consists of five layers, which are named as input layer, convolution layer, pooling layer, non-linear combination hidden layer and an output layer, respectively. Next, we will introduce the detail of each layer.

3.1.1. Input layer

For input layer, each node denotes a word in a tweet, and the order of nodes follows in their original order. In a typical neural model, words are represented by low-dimensional real-valued vectors (generally called word embeddings). For each word w_i, a look-up matrix $\mathbf{L}$ is used to obtain its embedding $e(w_i) \in R^{\mathrm{D}}$, where $\mathbf{L} \in R^{\mathrm{D}\times V}$ is a model parameter, D is the dimension of the word embedding and V is the vocabulary size. Typically, $\mathbf{L}$ can be initialized by pre-training with a large scale raw corpus. We will discuss how to generate the word embeddings in experimental setting section.

3.1.2. Convolution layer

N-grams information have been shown very useful for many NLP tasks [2,20,36]. So the convolution layer has been commonly used to capture lexical n-grams information [16,29]. In this paper, we apply the convolution action in our neural network. Given the input layer $e(w_1)\dots e(w_n)$, the convolution action is used to obtain a hidden sequence $h_1^1 \dots h_n^1$, which is computed as:

$$h_i^1 = tanh(\mathbf{W}^1 \cdot [e'(w_{i-1}), e'(w_i), e'(w_{i+1}), 1]'),$$

where e' represents the transpose of the vector e, $\mathbf{W}^1 \in R^{\mathrm{C}\times(3\mathrm{D}+1)}$ is a model parameter, and C is the output dimension. In this paper, the window size of the convolution operation is set 3, and *tanh* is used as the activation function.

3.1.3. Pooling layer

For each tweet, we obtain different number vectors after the convolution layer because every tweet contains different number of words. To form a vector with fixed dimensions, the pooling techniques are exploited to merge the varying number of features from the convolution layer. Commonly used pooling technique includes *max, min* and *average* poolings, which have also been used for many NLP tasks [20,37], achieving competitive performance. The most commonly used technique is the *max* function, choosing the highest value on each dimension from a set of vectors. In this paper, we use all the three pooling techniques to fully capture sarcastic evidence, concatenating them together as a new hidden layer h^2.

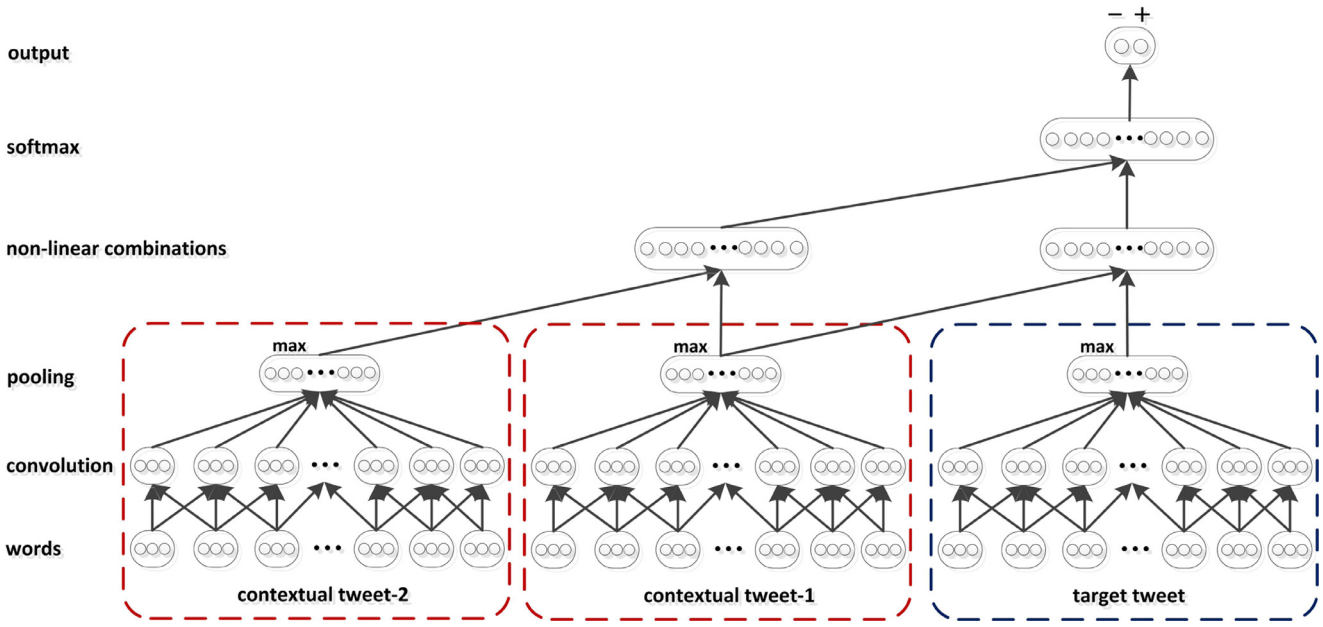

Fig. 2. The context-augmented neural network model CANN-ALL for twitter sarcasm detection.

3.1.4. Hidden layer

Based on the above pooling techniques, different types of features are obtained. In order to fully utilize these sources of information, a non-linear hidden layer is used for automatically combining these pooling features. The non-linear hidden layer is computed as:

$$h^3 = tanh\left(\mathbf{W}^2 \cdot \begin{bmatrix} h^2 \\ 1 \end{bmatrix}\right),$$

where $\mathbf{W}^2 \in R^{\mathrm{H}\times(3\mathrm{C}+1)}$ is a model parameter, and H is the dimension of this non-linear hidden layer.

3.1.5. Output layer

After the above four layers, an output layer is used to score all category labels according to the features in the last hidden layer. The output can be obtained with a linear transformation based on the following equation:

$$o = \mathbf{W}^3 \cdot h^3,$$

where, $\mathbf{W}^3 \in R^{3\times\mathrm{H}}$ is a model parameter.

3.1.6. Contextual sub network

This sub network aims to explore how to extract sarcastic clues from contextual information. Here, we take the history-based contexts for example to explain how to extract keywords. First, all words in contextual tweets are sorted based on their *tf-idf* values, by modeling all contextual tweets as one document. So all tweets in the dataset will totally generate a number of documents. Second, we choose the most important keywords (the highest *tf-idf* values) as the input of this sub network.

Contextual sub network is illustrated in the right part of Fig. 1. Like local sub network, each word is represented by word embeddings, and all word representation is obtained by the lookup table. Later, we also use *max, min* and *average* pooling techniques to extract features from the input words. Based on this pooling neural network, we can integrate the output of this pooling layer into the output of the pooling layer of the local neural model before feeding them to the non-linear hidden layer. In this way, the hidden layer automatically combines the features from the target tweet and contextual tweets. Note that unlike local sub network, contextual sub network does not use any convolutional functions, because contextual information are a set of salient words, which does not contain the n-grams information.

3.2. The model CANN-ALL

In the above model CANN-KEY, we explore how to capture important information from contextual tweets. In this section, we will discuss how to use complete tweets in the neural network models for improving the detection performance. The proposed model CANN-ALL is illustrated in Fig. 2. The proposed model CANN-ALL consists of six layers, which is named as input layer, convolution layer, pooling layer, non-linear combination layer, softmax layer and output layer, respectively. Note that input layer of this neural network includes three tweets (arranged in chronological order), and the last tweet is the target tweet. Here, input layer, convolution layer and pooling layer are similar to local sub network of the first model CANN-KEY.

Here, we take the conversation-based tweets for example to explain how to utilize the contextual information in the model CANN-ALL. Formally, for the target tweet t_i and its conversation-based tweets t_{i-2} and t_{i-1}, we can obtain the vector representation s_i, s_{i-1} and s_{i-2} by using input layer, convolution layer and *max* pooling layer. Based on the vector representation of the tweet s_i, s_{i-1} and s_{i-2}, we mainly introduce the non-linear combination layer and softmax layer, which are different from the model CANN-KEY. For the non-linear combination layer, this layer takes s_i, s_{i-1} and s_{i-2} as inputs, and outputs the sequence f_{i-1} and f_i, which is computed as:

$$f_i = tanh(\mathbf{W}_1 \cdot s_i + \mathbf{W}_2 \cdot s_{i-1} + b_1),$$
$$f_{i-1} = tanh(\mathbf{W}_3 \cdot s_{i-2} + \mathbf{W}_2 \cdot s_{i-1} + b_2),$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3 \in R^{K\times C}$ are the weight matrices, and b_1, $b_2 \in R^K$ is the bias vector. $f_i \in R^K$ is the class representation, and K is the category number of twitter sarcasm detection task.

Similar to the non-linear combination layer, the softmax layer takes f_{i-1} and f_i as inputs, and outputs $o_i \in R^K$, which is computed as:

$$o_i = softmax(\mathbf{W}_4 \cdot f_i + \mathbf{W}_5 \cdot f_{i-1} + b_3),$$

Table 1
Statistical information of all target and its contextual tweets. Here, #Basic represents the tweet number in basic dataset, and #History represents the tweet number in the history-based contexts. #Conversation represents the tweet number in the conversation-based contexts.

Category	#Basic	#History	#Conversation
Negative	500	2224	73
Sarcastic	500	2321	267
Positive	500	2229	113
total	1500	6774	453

where, $\mathbf{W}_4, \mathbf{W}_5 \in R^{K\times K}$ are the weight matrices, and $b_3 \in R^K$ is the bias vector. Based on the output vector o_i, we can obtain the category label of the target tweet.

3.3. Training

The training objective is to minimize the cross-entropy loss over a set of training examples $(x_i, y_i)|_{i=1}^N$, plus a l_2-regularization term. Follow previous work, the online AdaGrad is used to minimize the objective function [38]. At step j, parameters are updated by:

$$\theta_{j,i} = \theta_{j-1,i} - \frac{\alpha}{\sqrt{\sum_{j'=1}^{j} g_{j',i}^2}} g_{j,i},$$

where, α is the initial learning rate, and $g_{j,i}$ is the gradient of the ith dimension at step j.

4. Experiments

4.1. Experimental settings

4.1.1. Dataset

In this paper, we use the dataset constructed by Wang et al.. Statistical information of the dataset is shown in Table 1. Based on Table 1, we know that basic dataset consists of 1500 tweets, which contains all target tweets. For all contextual tweets, Table 1 shows that the history-based context contain 6774 tweets, and the conversation-based context only contains 453 tweets. The above information tell us that the number of the conversation-based contexts is far less than the number of the history-based contexts. Note that all tweets in basic dataset have the category label, and all tweet in contextual information have no category labels.

4.1.2. Evaluation method

Ten-fold cross-validation method is used in our experiments. Typically, basic dataset is randomly split into ten equal folds, where nine folds are selected for training and the tenthfold for test. We randomly choose one section from the nine training sections as the development dataset in order to tune hyper-parameters. The classification results are reported by macro-F score.

4.1.3. Hyper-parameters

In our models, there are five hyper-parameters, which includes two parts. The first part is the network structure parameters, including the dimensions of word vectors D, the output dimensions of the convolution layer C, and the output dimension of the non-linear combination layer H. The second part is the parameters in training, including the l_2-regularization co-efficient λ and initial learning ratio α for AdaGrad. According to previous work [2,18], the parameters values are set based on Table 2.

Table 2
Hyper-parameter values in neural model.

Type	#parameters
Network structure	D = 50, C = 100, H = 50
Training	$\lambda = 10^{-8}$, $\alpha = 0.01$

Table 3
Final results of our proposed model.

Model	macro-F Score (%)
The Proposed Model	
MODEL-KEY(local)	56.37
MODEL-KEY(local + conversation-based context)	57.96
MODEL-KEY(local + history-based context)	63.28
MODEL-ALL(conversation-based context)	58.46
MODEL-ALL(history-based context)	62.05
Other Context-Based Models	
$SVM^{multi\text{-}class}$	54.54
SVM^{HMM}(conversation-based context)	56.72
SVM^{HMM}(history-based context)	60.32
Discrete Model	
LMS	55.31

4.1.4. Embeddings

To learn the pre-trained word embeddings, we crawl 20 million tweets by using Twitter API. Based on this large-scale twitter corpus, we use the *word2vec* tool[1] to train the word embeddings. For the proposed model CANN-KEY and CANN-ALL, we use the learned word embeddings to initialize each word. Note that unknown words are represented by using the averaged vector from the pre-trained word embeddings.

4.2. Baseline methods

In order to verify the effectiveness of our proposed model, the following sarcasm detection systems are re-implemented as the baseline methods:

- LMS: Barbieri and Saggion uses rich linguistically-motivated features from the target tweet for twitter sarcasm detection [12].
- SVM^{HMM}: Wang et al. model the sarcasm detection problem as a sequential classification task over streams of tweets. SVM^{HMM} is used to assign the category labels for entire sequences [13].

4.3. Experimental results

Table 3 shows the final results of our proposed models. Based on the result of the model MODEL-KEY(local), we can know that neural models give better F-score compared to the corresponding discrete model LMS. Using only the target tweet, the neural model achieves a 56.37% macro-F score, higher than the 55.31% macro-F score from the best discrete model LMS. In Table 3, $SVM^{multi\text{-}class}$ is also a discrete model, and this model represents the baseline system of Wang et al.' work, which does not utilize any contextual information. Compared with the model MODEL-KEY(local), it achieves lower performance. The above analysis demonstrates the power of neural model in capturing semantic features. Note that neural models do not require any hand-crafted features, but discrete models need a large amount of manual features.

Shown in Table 3, our proposed context-augmented neural network models (MODEL-KEY and MODEL-ALL) can both give better performance compared to the current context-based discrete model, which is proposed by Wang et al. Taking the model MODEL-KEY for example, our model achieve the 57.96% macro-F

[1] https://code.google.com/p/word2vec.

score by using the conversation-based contexts, and the SVM^{HMM} achieves the 56.72% macro-F score by using same contexts. For the history-based contexts, the same trend can be found. Wang et al., regard the sarcasm detection as a sequential classification task over the target tweet and its contextual tweets. The above analysis demonstrate that the sequential assumption of SVM^{HMM} between one tweet and its contextual tweets is relatively weak. Because the influence of one tweet is not limited to a few tweets in the timeline, and coreferences between tweets can form a complex graph structure. Different from the model of Wang et al., we show that significant improvements can be achieved by modeling the context tweets of a given tweet as a set, using neural pooling functions to extract the most useful features from tweets automatically. So better performance can be obtained by modeling a complex graph structure (context-augmented neural network model) for twitter sentiment classification. Compared to the context-based discrete model, the above analysis demonstrates the effectiveness of our context-augmented neural model.

We also compare the performance between the model MODEL-KEY and MODEL-ALL. For the conversation-based contexts, the model MODEL-ALL achieves the 58.46% macro-F score, and outperforms the performance of the model MODEL-KEY (57.96%). This shows that model MODEL-ALL has more power in capturing the subtle clues of sarcasm for the conversation-based contexts. However, the model MODEL-KEY gives better performance compared to the model MODEL-ALL for the history-based contexts. One possible reason is that the model MODEL-ALL only takes two contextual tweets as inputs, and the model MODEL-KEY takes more contextual tweets as inputs.

Finally, we compare the usefulness of two different contexts. For the model MODEL-KEY and MODEL-ALL, we can know that the history-based contexts achieve better performance than the conversation-based contexts. Which type of context is more useful for twitter sarcasm detection in neural models? First, the conversation-based context gives a relatively small improvement compared with the local neural model. A likely reason is that the conversation-based context accounts for a very small proportion in all target tweets. Second, more than 85% target tweets have the history-based contexts, which express the views and opinions towards some events and people. Based on the above analysis, we can know that the conversation-based contexts are more effective than the history-based contexts. However, the history-based contexts can achieve better performance than the conversation-based contexts because the number of the conversation-based tweets is relative small.

5. Conclusion and Future Work

We proposed the context-augmented neural network models for twitter sarcasm detection. Compared with previous work, our neural model incorporated the features of the tweet content itself and contextual features into a single model in the form of word vectors. Experimental results showed that our proposed context-augmented neural model gave better performance compared with the state-of-the-art discrete model and context-based model, demonstrating the effectiveness of context-augmented neural network model for this task. In future work, we will explore how to improve the model MODEL-ALL in order to deal with more contextual tweets, and how to integrate the model MODEL-KEY and MODEL-KEY in order to improve the detection performance.

Acknowledgments

This work is supported by the State Key Program of National Natural Science Foundation of China (Grant No.61133012), the National Natural Science Foundation of China (Grant No.61702121, 61373108) and the National Philosophy Social Science Major Bidding Project of China (Grant No. 11&ZD189).

References

[1] X. Fu, W. Liu, Y. Xu, L. Cui, Combine hownet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis, Neurocomputing 241 (2017) 18–27.
[2] Y. Ren, R. Wang, D. Ji, A topic-enhanced word embedding for twitter sentiment classification, Information Sciences 369 (2016) 188–198.
[3] B. Liu, Sentiment analysis and opinion mining, Synthesis Lectures on Human Language Technologies 5 (1) (2012) 1–167.
[4] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and trends in information retrieval 2 (1–2) (2008) 1–135.
[5] N.F.F.D. Silva, L.F.S. Coletta, E.R. Hruschka, J.E.R. Hruschka, Using unsupervised information to improve semi-supervised tweet sentiment classification, Information Sciences 355–356 (2016) 348–365.
[6] Y. Ren, Y. Zhang, M. Zhang, D. Ji, Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 3038–3044.
[7] D. Davidov, O. Tsur, A. Rappoport, Semi-supervised recognition of sarcastic sentences in twitter and amazon, in: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, 2010, pp. 107–116.
[8] R. Gonzez-Ibez, S. Muresan, N. Wacholder, Identifying sarcasm in twitter: a closer look, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2011, pp. 581–586.
[9] A. Reyes, P. Rosso, T. Veale, A multidimensional approach for detecting irony in twitter, Language Resources and Evaluation 47 (1) (2013) 239–268.
[10] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, R. Huang, Sarcasm as contrast between a positive sentiment and negative situation., in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013, pp. 704–714.
[11] T. Ptácek, I. Habernal, J. Hong, Sarcasm detection on czech and english twitter., in: Proceedings of the 25th International Conference on Computational Linguistics, 2014, pp. 213–223.
[12] F. Barbieri, H. Saggion, Modelling irony in twitter, in: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 56–64.
[13] Z. Wang, Z. Wu, R. Wang, Y. Ren, Twitter sarcasm detection exploiting a context-based model, in: Proceedings of the International Conference on Web Information Systems Engineering, 2015, pp. 77–91.
[14] A. Rajadesingan, R. Zafarani, H. Liu, Sarcasm detection on twitter: A behavioral modeling approach, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 2015, pp. 97–106.
[15] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1642–1651.
[16] C.N. dos Santos, M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, in: Proceedings of the 25th International Conference on Computational Linguistics, 2014, pp. 69–78.
[17] Y. Ren, Y. Zhang, M. Zhang, D. Ji, Context-sensitive twitter sentiment classification using neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2016, pp. 215–221.
[18] Y. Ren, D. Ji, Neural networks for deceptive opinion spam detection: an empirical study, Information Sciences 385–386 (2017) 213–224.
[19] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, in: arXiv preprint arXiv:1404.2188, 2014.
[20] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 1555–1565.
[21] Y. Ren, Y. Zhang, Deceptive opinion spam detection using neural network, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 140–150.
[22] R.J. Kreuz, G.M. Caucci, Lexical influences on the perception of sarcasm, in: Proceedings of the Workshop on Computational Approaches to Figurative Language, 2007, pp. 1–4.
[23] P. Carvalho, L. Sarmento, M.J. Silva, E. De Oliveira, Clues for detecting irony in user-generated contents: oh...!! it's so easy;-, in: Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, 2009, pp. 53–56.
[24] T. Veale, Y. Hao, Detecting ironic intent in creative comparisons, in: Proceedings of the 19th European Conference on Artificial Intelligence, 2010, pp. 765–770.
[25] O. Tsur, D. Davidov, A. Rappoport, A great catchy name: semi-supervised recognition of sarcastic sentences in online product reviews, in: Proceedings of the International Conference on Weblogs and Social Media, 2010, pp. 162–169.
[26] B.C. Wallace, D.K. Choe, E. Charniak, Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment., in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015, pp. 1035–1044.

[27] J. Karoui, F. Benamara, V. Moriceau, N. Aussenac-Gilles, L.H. Belguith, Towards a contextual pragmatic model to detect irony in tweets, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, 2015, pp. 644–650.
[28] D. Bamman, N.A. Smith, Contextualized sarcasm detection on twitter, in: Proceedings of the Ninth International AAAI Conference on Web and Social Media, 2015.
[29] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of Machine Learning Research 12 (2011) 2493–2537.
[30] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems 26 (2013) 3111–3119.
[31] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, in: arXiv preprint arXiv:1405.4053, 2014.
[32] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, in: arXiv preprint arXiv:1412.1058, 2014.
[33] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1422–1432.
[34] D. Wang, E. Nyberg, in: A long short-term memory model for answer sentence selection in question answering, 2015, pp. 707–712.
[35] J. Li, M.-T. Luong, D. Jurafsky, A hierarchical neural autoencoder for paragraphs and documents, in: arXiv preprint arXiv:1506.01057, 2015.
[36] S.M. Mohammad, S. Kiritchenko, X. Zhu, Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets, in: Proceedings of the Second Joint Conference on Lexical and Computational Semantics, 2013, pp. 321–327.
[37] D.-T. Vo, Y. Zhang, Target-dependent twitter sentiment classification with rich automatic features, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015, pp. 1347–1353.
[38] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, The Journal of Machine Learning Research 12 (2011) 2121–2159.

Yafeng Ren, received Ph.D degree in computer school from Wuhan University, China, 2015. He was a postdoctoral research fellow with Singapore University of Technology and Design from 2015 to 2016. He is currently an associate professor with Guangdong University of Foreign Studies. His research interests include natural language processing, machine learning and data mining. He has published over 10 papers in related conferences and journals, including AAAI, EMNLP, COLING etc.

Donghong Ji, is currently a professor in Wuhan University and Guangdong University of Foreign Studies. He received his Ph.D, M.Sc. and B.Sc. Degrees from Wuhan University in 1995, 1992 and 1989 respectively. His main research interests include natural language processing and information retrieval.

Han Ren, is currently an associate professor in Guangdong University of Foreign Studies. He received his Ph.D degree in Wuhan University, China, in 2011. He was a postdoctoral research fellow with Wuhan University from 2012 to 2012. His research interests include natural language processing and machine learning.