Received September 30, 2017, accepted November 7, 2017, date of publication January 1, 2018, date of current version May 16, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2776930

Deep Convolution Neural Networks for Twitter Sentiment Analysis

ZHAO JIANQIANG[1,2], GUI XIAOLIN[1,2], AND ZHANG XUEJUN[2,3]
[1]School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[2]Key Laboratory of Computer Network, Xi'an Jiaotong University, Xi'an 710049, China
[3]School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

Corresponding author: Gui Xiaolin (xlgui@mail.xjtu.edu.cn)

This work was supported in part by the NSFC Projects under Grant 61472316, Grant 61172090, and Grant 61762058, and in part by the Science and Technology Project of Shaanxi Province under Grant 2016ZDJC-05 and Grant 2017ZDXM-GY-011.

ABSTRACT Twitter sentiment analysis technology provides the methods to survey public emotion about the events or products related to them. Most of the current researches are focusing on obtaining sentiment features by analyzing lexical and syntactic features. These features are expressed explicitly through sentiment words, emoticons, exclamation marks, and so on. In this paper, we introduce a word embeddings method obtained by unsupervised learning based on large twitter corpora, this method using latent contextual semantic relationships and co-occurrence statistical characteristics between words in tweets. These word embeddings are combined with n-grams features and word sentiment polarity score features to form a sentiment feature set of tweets. The feature set is integrated into a deep convolution neural network for training and predicting sentiment classification labels. We experimentally compare the performance of our model with the baseline model that is a word n-grams model on five Twitter data sets, the results indicate that our model performs better on the accuracy and F1-measure for twitter sentiment classification.

INDEX TERMS Twitter, sentiment analysis, word embeddings, convolution neural network.

I. INTRODUCTION

Twitter, with over 319 million[1] monthly active users, has now become a goldmine for organizations and individuals who have a strong political, social and economic interest in maintaining and enhancing their clout and reputation. Sentiment analysis provides these organizations with the ability to surveying various social media sites in real time.

Text Sentiment analysis is an automatic process to determining whether a text segment contains objective or opinionated content, and it can furthermore determine the text's sentiment polarity. The goal of Twitter sentiment classification is to automatically determine whether a tweet's sentiment polarity is negative or positive.

Most of the existing methods of Twitter sentiment classification follow the method proposed by Pang *et al.* [1] and apply machine learning algorithms to build a classifier from tweets with manually annotated sentiment polarity label. In recent years, there has been a growing interest in using deep learning techniques, which can enhance classification accuracy.

[1]https://about.twitter.com/company

In this article, we apply convolution algorithm on Twitter sentiment analysis to train deep neural network, in order to improve the accuracy and analysis speed. First we learn global vectors for word representation (Pennington *et al.* [2]) by unsupervised learning on large Twitter corpora, which expresses the word sentiment information as the words embeddings. Afterwards, we concatenate these word representation with the prior polarity score feature and state-of-the-art features as sentiment feature set. These feature sets is combined and fed into an deep convolution neural networks to train and predict the sentiment classification labels of the tweet. A model called GloVe-DCNN is presented which implements the binary task of classifying the tweet into negative or positive sentiment categories. The experimental results indicate that our approach has a good classification performance.

The rest of this paper is organized as follows. Section II discussed related work on this topic. We proposed a model called GloVe-DCNN for tweet sentiment classification in Section III. The process of the experiment is shown in Section IV. The Section V discuss experiment results. Finally, Section VI concludes the paper.

II. RELATED WORK

Most existing studies to Twitter sentiment analysis can be divided into supervised methods [3]–[8] and lexicon-based methods [9]–[11]. Supervised methods are based on training classifiers (such as Naive Bayes, Support Vector Machine, Random Forest) using various combinations of features such as Part-Of-Speech (POS) tags, word N-grams, and tweet context information features, such as hashtags, retweets, emoticon, capital words etc. Lexicon-based methods determine the overall sentiment tendency of a given text by utilizing pre-established lexicons of words weighted with their sentiment orientations, such as SentiWordNet [12].

These methods rely on the presence of lexical or syntactical features that explicitly express the sentiment information. Though, in a lot of cases, the sentiment of a tweet is implicitly associated with th semantics of its context. In this work, we present semantic feature for sentiment analysis, which is word vector contextual representation of a word in tweet, which can capture the deep and implicit semantic relation information in the words of tweets.

Deep learning methods are now well established in machine learning [13], [14]. They have been especially successful for image and voice p4rocessing tasks. Recently, such methods have begun to overtake traditional sparse, linear models for nature language processing. Kalchbrenner *et al.* [15] proposed a convolution neural network architecture with multiple convolution layers, positing latent, dense and low-dimensional word vectors (initialized to random values) as inputs. They uses a dynamic convolution neural network for sentiment classification of movie reviews and Twitter. Experiments show that dynamic convolution neural networks are better than those based on unigram and bigram models. dos Santos *et al.* [16] introduced a new deep convolution neural network that utilizes from character-level to sentence-level information to implement sentiment classification of short texts. For the Stanford Twitter Sentiment corpus, the method obtains a prediction accuracy of 86.4%. Kim [17] studied the performance of convolution neural network for sentence level sentiment classification tasks on a pre-trained word vector model. Experiments on different datasets show that convolution neural networks can improve classification performance. Johnson and Zhang [18] apply convolutional neural network to high dimensional text for text classification and obtain several state-of-the-art performances on some benchmark data sets for sentiment categorization, but the model are more complex and expensive to train. Johnson and Zhang [19] proposed a similar model, but swapped in high dimensional 'one-hot' vector representations of words as CNN inputs. Their focus was on classification of longer texts, rather than sentences. Tang *et al.* [20], [21] proposed a method of continuous speech using neural network automatic acquisition of word emotion information representation, building micro-blog text emotion feature vector, excellent performance in SemEval2013 classification tasks. Socher *et al.* [22] introduced the recursive neural tensor network for sentiment detection, which represents a phrase through word vectors and a parse tree and then compute vectors for higher nodes in the tree using the same tensor-based composition function. Li *et al.* [23] benchmark recursive neural models against sequential recurrent neural models on for nature language processing tasks including sentiment classification at the sentence level and phrase level.

III. DEEP CONVOLUTION NEURAL NETWORKS FOR SENTIMENT ANALYSIS

A. FEATURE REPRESENTATION

• **Word N-grams features**

The word N-grams feature model is one of most simple and effective representation model for natural language analysis and Twitter sentiment analysis. Some studies have shown state-of-the-art performance for sentiment classification on Twitter data using a unigram model [13], [24]. In this paper, we use unigram and bigram features (referred to BoW feature) as the baseline feature models.

• **Twitter specific features**

The number of hashtags, emoticons, negation, POS and the presence of capitalized words are used as features.

• **Word sentiment polarity score features**

The word sentiment polarity score is a lexicon-based sentiment feature, and some approaches [9], [10] commonly use it as a sentiment feature for tweet sentiment analysis. We used the AFINN [25] lexicon and extended it using Senti-Wordnet [12] to obtain the tweet sentiment polarity score. The sentiment polarity score of a tweet is the sum of the sentiment polarity score of each word in the tweet. The sentiment score of each word is computed by measuring the *PMI* (point-wise mutual information) between the word and the negative or positive sentiment classification of the tweet using the formula:

$$SenScore(w) = PMI(w, pos) - PMI(w, neg)$$

Where w is a word in the lexicon, $PMI\ (w, pos)$ is the PMI score between w and the positive category, and $PMI\ (w, neg)$ is the PMI score between w and the negative category. Therefore, a positive $SenScore\ (w)$ indicates that there is a stronger relationship between the word w with positive sentiment and vice versa.

• **Word representation features**

Learning word vector representations from a large number of unannotated text corpora has recently been used in various natural language processing tasks. The word vector representations from unsupervised learning in massive corpora can capture grammatical and semantic characteristics of words. Recent studies have shown that the use of pre-trained word embeddings can substantially improve the performance of the model [22], [26]–[28]. In our study, we use the GloVe model to implement unsupervised learning of word-level embeddings. The GloVe (Global Vectors for word representation) was by Pennington *et al.* [2] 2014. The GloVe model is global

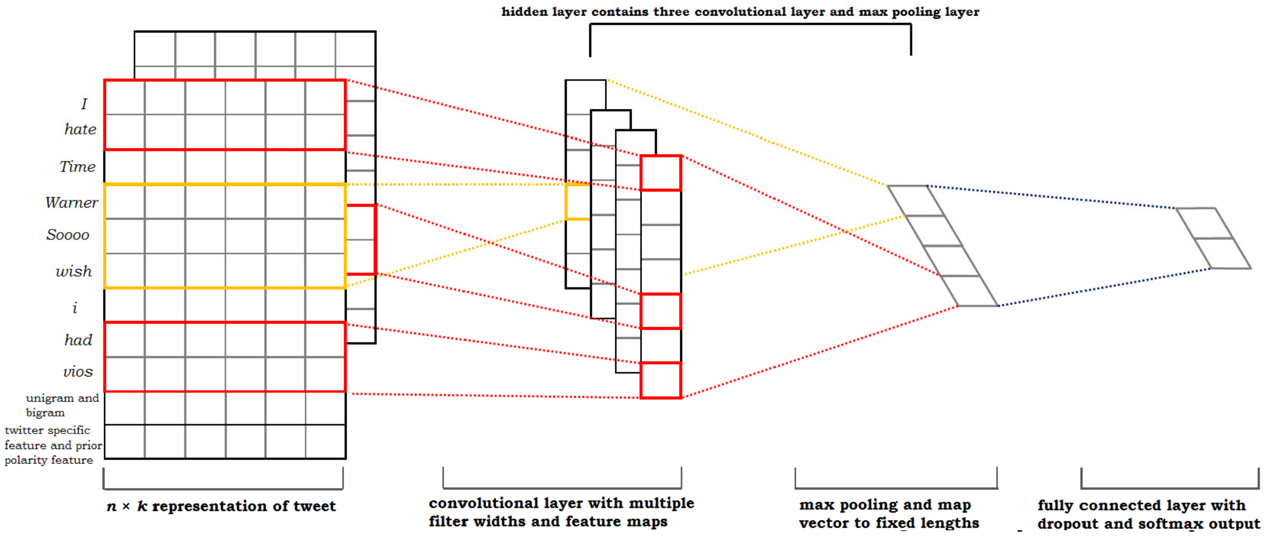

FIGURE 1. GloVe-DCNN model architecture for an example tweet.

log bilinear regression model that combines the advantages of the two major model families in the literature: local context window and global matrix factorization methods. The model efficiently utilizes statistical information by training the nonzero elements in a word-word co-occurrence matrix only, rather than on the entire sparse matrix or on individual context windows in a large corpus. The model learns word vector with ratios of co-occurrence probability rather than the probability itself. Let's consider words $w_i$and $w_j$that exhibit a particular aspect of interest, specifically, suppose we are interested in the concept of thermodynamic phase, for which we might take $w_i = ice$ and $w_j = steam$. The relationship of these words can be examined by studying the ratio of their co-occurrence probabilities with various probe words w_k. Lets Pij is the probability that word j appear in the context of word w_i. For words k related to *ice* but not *steam*, say $w_k = solid$, we expect the ratio Pik/Pjk will be large. Similarly, for words w_k related to *steam* but not *ice*, say $w_k = gas$, the ratio should be small. For words w_k like *water* or *fashion*, that are either related to both ice and steam, or to neither, the ratio should be close to one. Because Synonyms and similar paragraphs usually have similar context, they are mapped to feature vectors that are close to each other.

After training by the GloVe model, the word vectors can be represented as semantic features of the tweet. The vectors can be concatenated as tweet semantic sentiment features.

The GloVe tools [29] is used to train tweet word vectors in Twitter corpus. The detailed algorithms are described in in the literature Pennington *et al.* [2]. We trained the GloVe model using a 20 billion twitter corpus, and induced 200-dimensional word vectors. Words that are not pre-trained set are initialized randomly

B. TWEETS PREPROCESSING

Tweets are usually composed of incomplete expression, a variety of noise and poorly structured sentences because of the frequent presence of acronym, irregular grammar, ill-formed words and non-dictionary terms. Noise and unstructured Twitter data will affect the performance of tweet sentiment classification [30], [31]. Prior to feature selection, a series of preprocessing are performed to tweets to reduce the noise in the micro-blog text. The preprocessing is:

-Removal all non-ASCII and non-English characters in the tweets.

-Removal all URL links. The URLs do not contain the sentiment information of tweet, so there will be deleted from tweets. First, we expand Twitter's short URLs to URLs and tokenize it. Then, remove the URL matching the tokens from tweets in order to refine the tweet content.

-Removal numbers. The numbers generally do not contain sentiment information, so it are useless when measuring sentiment and are deleted from tweets in order to refine the tweet content.

-Replace negative references. Tweets contain various notions of negation. Generally speaking, negation plays an important role in determining the sentiment polarity of the tweet. Here, the process of negation is transforming "won't", "can't", and "n't" into "will not", "cannot", and "not", separately.

-Expand acronyms and slang to their full words form. Acronyms and slang are common in tweets, but are ill-formed

TABLE 1. The number of tweet sentiment all datasets.

Dataset	Num of Tweets	Num of Negative	Num of Positive
STSTd	359	177	182
SE2014	5892	1650	4242
STSGd	2034	1402	632
SSTd	3326	1037	2289
SED	2648	990	1658

TABLE 2. Accuracy using GloVe-DCNN and baseline on all datasets. BoW refer to uni- and bi-gram features. GloVe refer to concatenate BoW vectors with the average GloVe representations, word sentiment polarity feature and twitter specific feature. DCNN refers to deep convolution neural network. BoW-SVM represents the use of the SVM classifier and the BOW features vector.

Method	Accuracy /%					
	SSTd	SED	STSGd	SE2014	STSTd	Average
BoW-SVM	59.11	62.51	68.79	73.21	68.81	66.49
BoW-LR	69.64	80.1	75.86	73.98	71.04	74.12
GloVe-SVM	79.16	85.87	80.07	83.06	81.61	81.95
GloVe-LR	77.11	86.72	80.17	82.55	81.62	81.63
GloVe-DCNN	**81.36**	**87.39**	**85.97**	**85.82**	**87.62**	**85.63**
SentiStrength[46]	-	-	81.32	-	-	-
MaxEnt[38]	-	-	-	-	83.0	-
Hbd[45]	-	-	80.33	-	-	-
Updated+Expanded[47]	-	-	82.3	-	-	-
LProp[44]	-	-	-	-	84.7	-
CharSCNN[16]	-	-	-	-	86.4	-

words. It is essential to expand them to their original complete words form for sentiment analysis. In this paper, the acronyms and slang were expanded to their original the acronyms and slang to their original words form utilizing the acronym dictionary Internet Slang Dict [32]. The internet acronyms and slang are created by internet users in order to save typewriting. Terms have originated from various sources, including Bulletin Boards, Email, AIM, MySpace, Yahoo, Twitter, Facebook, IRC, Chat Rooms, and Cell Phone Text Messaging. Each acronym corresponds to an explanation. Example, "idts" is "i don't think so", "ICYMI" is "In Case you Missed It".

-Removal stopwords. Stop words usually refer to the most common words in a language, such as "tbe", "an", and "than". The classic method is based on removing the stop-words obtained from precomplied lists. There are multiple stopwords lists existing in the literature. In our study, we utilized the Van stoplist [33].

-Replace emoticons and emoji. The emoticon and emoji are a writer's moods expression in the form of icons in the tweet. We replace the emoticons and emoji with their origin text form by looking up the emoticon dictionary [34].

-Tokenization using the Tweet-NLP [35].

C. DEEP CONVOLUTION NEURAL NETWORKS MODEL

This section, we propose a deep convolution neural network model to classifying tweet as negative or positive sentiment.

Consider a tweet t with m tokens as an example. First, each tokens in tweets was mapping to the corresponding word vectors by looking up word vector table $L \in R^{n\times|V|}$ generated by GloVe modelčwhere V is a vocabulary of the words, n is the dimension of the word vectors. Each word w is mapping to a vector $w_i \in R^n$. After the mapping, a tweet is expressed as a vector of the word embeddings concatenation.

Then the unigram and bigram features vector, twitter-specific features vector and word sentiment polarity features vector can be concatenated with the word embeddings vector as the feature vector v of the tweet t.

$$v = w_1 \oplus w_2 \oplus w_3 \oplus \ldots \oplus w_m \oplus w_{m+1} \oplus w_{m+2} \oplus w_{m+3} \tag{1}$$

where $\oplus$ is the concatenation operator of vector, $w_{m+1} \in R$ is word sentiment polarity features vector of the tweet t, $w_{m+2} \in \{0, 1\}^l$ is unigram and bigram features vector of the tweet t, $w_{m+3} \in R^{l'}$ is twitter-specific features vector of the tweet t.

To unify the matrix representation of tweets in different length, the maximum length of all tweets in the dataset is used as the fixed length for tweet matrices. For shorter tweets, zero vector was padded at the back of a tweet matrix

In the first convolution layer, convolution calculation are performed using employ multiple filters with variable window size h, and generate local sentiment feature vector x_i for each possible word window size. And the bias term $b \in R$ and transition matrix $W \in R^{h_u\times hn}$ are generated for each filter, where h_u is the amount of hidden units in the convolution layer. Each convolution operation generates a new context local feature vector x_i in a word window h.

$$x_i = f(W \cdot v_{i:i+h-1} + b) \tag{2}$$

where f is non-linear active function and $v_{i:i+h-1}$ is the the local vector from position i to position $i + h - 1$ in the vector v.

The convolution filter generates a local feature mapping vector for each possible word window in the tweet, which is followed by the completion of the convolution operation to generate a new vector that can be expressed as:

$$x = [x_1, x_2, \ldots, x_{n-h+1}] \tag{3}$$

Afterward the convolution operation, a k-max pooling operation is employed on the new feature vector x generated

TABLE 3. The results of Precision,Recall and F1-measureusing GloVe-DCNN and baseline. BoW refer to uni- and bi-gram features. GloVe refer to concatenate BoW vectors with the average GloVe representations,word snetiment polarity feature and twitter-specific feature. DCNN refers to deep convolution neural network. BoW-SVM represents the use of the SVM classifier and the BoW features vector.

Method	Pos. /%			Neg. /%			Avg./%		
	P	R	F1	P	R	F1	P	R	F1
SED									
BoW-SVM	66.40	96.39	78.52	83.86	27.04	40.54	75.13	61.71	59.53
BoW-LR	86.92	83.05	84.89	76.11	81.45	78.60	81.51	82.25	81.74
GloVe-SVM	86.82	90.22	88.42	84.30	79.61	81.74	85.56	84.92	85.08
GloVe-LR	88.61	89.42	88.96	83.72	83.89	83.20	86.16	86.15	86.08
GloVe-DCNN	87.29	92.39	89.77	88.71	82.25	85.357	**88**	**87.32**	**87.66**
SSTd									
BoW-SVM	64.86	88.37	74.77	66.20	32.43	43.50	65.53	60.40	59.13
BoW-LR	77.49	74.79	75.58	66.59	68.48	66.34	72.04	71.64	70.96
GloVe-SVM	78.89	87.88	83.07	79.53	67.25	72.73	79.21	**77.57**	77.90
GloVe-LR	79.41	82.27	80.72	73.52	70.22	71.64	76.46	76.24	76.18
GloVe-DCNN	85.03	83.46	84.23	76.19	78.26	77.21	**80.61**	**80.86**	**80.72**
STSGd									
BoW-SVM	80.00	11.88	8.04	69.35	99.64	80.06	74.68	55.76	44.05
BoW-LR	67.38	55.59	47.25	77.34	93.70	83.42	72.36	74.64	65.34
GloVe-SVM	70.74	61.59	53.21	79.43	94.57	85.21	75.09	78.08	69.21
GloVe-LR	63.76	70.00	56.60	82.05	89.21	84.71	72.91	79.61	70.66
GloVe-DCNN	75.35	74.85	75.06	90.15	90.34	90.24	**82.75**	**82.61**	**82.65**
Hbd[39]	-	-	-	-	-	-	-	-	77.52
SentiStrength[46]	-	-	-	-	-	-	-	-	78.56
Updated+Expanded[47]	-	-	-	-	-	-	-	-	79.62
SE2014									
BoW-SVM	74.65	97.55	84.58	61.64	10.61	18.11	68.15	54.08	51.34
BoW-LR	79.26	91.87	85.10	61.57	35.14	44.74	70.42	63.51	64.92
GloVe-SVM	86.68	91.85	89.19	73.80	61.94	67.35	80.24	76.89	78.27
GloVe-LR	86.74	88.26	87.49	66.75	63.59	65.13	76.74	75.92	76.31
GloVe-DCNN	75.93	71.93	73.87	89.19	91.03	90.10	**83.56**	**81.48**	**81.99**
Splusplus[43]									72.80
STSTd									
SVM- BoW	76.81	59.33	65.93	65.32	81.86	71.75	71.07	70.60	68.84
LR- BoW	75.91	65.35	69.39	68.42	79.14	72.64	72.17	72.25	71.01
GloVe-SVM	82.18	83.30	81.70	81.62	82.20	80.93	81.90	82.75	81.32
GloVe-LR	82.10	82.68	81.60	81.08	82.60	81.06	81.59	82.64	81.33
GloVe-DCNN	87.98	89.47	88.71	87.22	85.42	86.29	**87.60**	**87.45**	**87.50**

by the convolution layer. K-Max pooling mapped the vector x to a fixed length vector. The length of the vector is a hyperparameter to be determined by the user and corresponds to the number of hidden units in the convolution layer. The local sentence features are integrated into all the features. Pooling operation commonly used are the mean pooling and k-max pooling. In this work, k-max pooling was used. For sentiment classification, the most decisive word or phrase is often only a few, but not uniformly scattered throughout the text. The k-max pooling is just some of the most discriminative language fragments. The k-max pooling select the top k number of features corresponding to multiple hidden layers, so that the most important sentiment feature information can be retained. At the same time, the sequence of words and the context information of each word are also taken into consideration in the pooling operation. This solves the problem that the traditional method can't express the negative words in a tweet and affect the feeling of the tweet.

$$v' = \max\{x_1, x_2, \ldots, x_{n-h+1}\} \quad (4)$$

In order to obtain better feature information, we fed the fixed length vectors created by the k-max pooling to a convolution layer for obtaining a new vector again. In the model, we select the hidden layers to contain three convolution layers and three k-max pooling layers.

The output layer of the architecture is a softmax layer that generates probability value of positive or negative sentiment. The output layer uses a fully connected softmax layer to adjust the sentiment characteristics of the input layer, and gives a probability distribution of the sentiment classification labels.

$$y^{(j)} = W^{(j)}y^{(j-1)} + b^{(j)} \tag{5}$$

where $y^{(j)}$ is output vector of softmax layer, $y^{(j-1)}$ is output vector of pooling layer, $W^{(j)}$ is transition matrix of softmax layer, $b^{(j)}$ is bias factor of softmax layer.

The probability distribution over the sentiment labels is :

$$P(i|t, \theta) = \frac{\exp(y_i^{(j)})}{\sum_{k=1}^{n} \exp(y_k^{(j)})} \tag{6}$$

Due to need to learn a lot of hyperparameters, the depth of convolution neural network suffer from over-fitting. We apply dropout regularization to the fully connected layers to eliminate the problem of a lot of hidden units and the connections between them.

IV. EXPERIMENT

We test the network on five different datasets, then relate the results of the experiments.

A. BASELINE

To provide a point of reference for the deep convolution neural network (DCNN) results, we used two methods as baseline. First, we used a RBF kernel SVM and Logistics Regression (LR) exploiting unigram and bigram features (BoW). This refer to BoW-SVM and BoW-LR. Researchers have reported state-of-the-art performance for Twitter sentiment analysis using BoW model [13]. Then we also experimented with combining the unigram, bigram, word sentiment polarity features, twitter-specific features and word vector features with a RBF kernel SVM and Logistics Regression classifier (refer to GloVe-SVM and GloVe-LR).

B. DATASETS

In this paper, we use five data sets that are been extensively applied in the related Twitter sentiment classification literatures.

1) The Stanford Twitter Sentiment Test (STSTd) data set was introduced by Go *et al.* in the literature [36]. The STSTd is consists of 182 positive and 177 negative sentiment tweets. Although the Stanford Twitter Sentiment Test set is relatively small, it has been extensively applied in the related literatures [3], [7], [8], [4], [37] for various evaluation tasks.
2) The SE2014 dataset was provided in SemEval2014 Task9 [38]. The dataset contains the tweet ids and its corresponding sentiment labels which have been annotated with positive, negative and neutral. A part of the tweets were no longer available for downloading at the time of the experiment, leaving 5892 tweets with positive and negative.
3) The Stanford Twitter Sentiment Gold (STSGd) data set was constructed by Saif *et al.* in the literature [39]. The STSGd contains 2034 tweets that were manually annotated negative or positive sentiment labels by three graduate students.
4) The Sentiment Evaluation Dataset(SED) was used in the literature by Narr *et al.* [40]. It is composed of 990 negative and 1658 positive sentiment tweets that have been human-annotated with sentiment label by three Mechanical Turk workers.
5) The Sentiment Strength Twitter dataset (SSTd) was applied in the literature [7] for evaluating SentiStrenth. It contains 1037 negative and 2289 positive sentiment tweets which have been manually annotated with positive or negative sentiment label.

Table 1 shows the distribution of positive and negative tweets in each dataset.

C. EXPERIMENTAL SETUP

In this paper, we apply 10-fold cross validation for each dataset. For all datasets, the same preprocessing steps was applied. For each experiment, we trained the convolution neural network on the training set, and obtained the highest accuracy points in the verification set, and reported the accuracy of the test set. We replicated cross validation experiments 100 times for each dataset, so that each replication was a cross validation of 10-fold. We recorded the average performance for each replication and report the mean average accuracy values observed over 100 replications of cross validation. Finally, we use the average of accuracy, F1-Measure, precision and recall of the five sets of experiments as the final evaluation metric on each dataset.

In experiments, we set batch size as 128 and learning rate to 0.001. A regularization with dropout rate 0.5 was applied on the fully connected layers in the network. We tested with different combinations of filter windows, the filter windows seven is shown to be an appropriate combination. In the experiment, we find that the activated function hyperbolic ReLu is better than the performance of rectified linear units for the convolution layer.

D. EXPERIMENTAL RESULTS

The experimental evaluation metrics is the accuracy and the average of precision, recall, F1-Measure in the classification of positive and negative tweets.

Table 2 reports the accuracy of sentiment prediction on all data sets. The highest accuracy is 87.62%, which was achieved using GloVe-DCNN on the STSTd dataset. From the average, the GloVe-DCNN achieves a maximum improvement in accuracy of 19.14% and a minimum improvement of 3.68% over the baseline method. From table 2, it is evident that the GloVe-DCNN model has better performance than the baseline method.

In Table 2, we compare the performance of the GloVe-DCNN method with the methods those presented in the relevant literature. In [44], a label propagation (LProp) method is proposed, Go *et al.* [36] use maximum entropy classifiers (MaxEnt) for the twitter sentiment classification on the STSTd dataset, dos Santos *et al.* [16] proposed CharSCNN approach on the STSTd dataset and Saif *et al.* [42] proposed Hdb approach, the method SentiStrength in [43] and the Updated + Expanded in [44]. The GloVe-DCNN outperforms the previous approaches in the prediction accuracy.

In Table 3, we present the results in precision (P), recall (R) and the F1- Measure (F1) of positive and negative sentiment prediction performance of baseline and GloVe-DCNN model. In this table, it can be observed that GloVe-DCNN performs better on average of precision, recall and F1-Measure than the baseline. Compared the other approaches proposed in the literatures [45]–[48], the GloVe-DCNN approach outperforms the approaches in the average of F1-Measure on the STSGd and SE2014 dataset, separately.

V. DISCUSSION

The experimental results clearly indicate that the GloVe-DCNN model can obtain a good performance of the sentiment classification. Comparing the deep convolution neural network and the baseline method for Twitter sentiment classification algorithm, the results indicate that the depth convolution neural network has obvious advantages in five datasets. This shows that deep convolution neural network can effectively construct text semantics. Compared with the BoW model used SVM classifier, convolution neural network can more effectively catch the context sentiment information in the tweet, retain the word order information, and reduce the data sparseness problem. The convolution neural network directly models context sentiment feature from text, and selects the most important features in the tweet effectively. It avoid error propagation and improve the classification performance. The pre-trained word vector representation by learning on Twitter corpus can better describe the similarity between words in Twitter, and extract the implicit semantic relation and sentiment feature information between words in the tweet.

VI. CONCLUSION

We utilize a depth convolution neural network for sentiment classification on Twitter tweets in this work. Our approach concatenates the pre-trained word embeddings feature generated using the GloVe word sentiment polarity features based sentiment lexicon and n-grams features as the sentiment features vector of the tweet, and inputs the feature sets to a deep convolution neural network. Our model captures contextual information with the recurrent structure and constructs the representation of text using a convolution neural network. We report our experimental results in five datasets. Our model performs better than the state-of-the-art approaches and baseline model. We can finally conclude that deep convolution neural network utilizing pre-trained word vectors has good performance in the task of Twitter sentiment analysis.

REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2002, pp. 79–86.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, vol. 14. 2014, pp. 1532–1543.

[3] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of twitter," in *Proc. Semantic Web-ISWC*, 2012, pp. 508–524,2012.

[4] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *J. Artif. Intell. Res.*, vol. 50, pp. 723–762, Mar. 2014.

[5] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decision Support Syst.*, vol. 66, pp. 170–179, Oct. 2014.

[6] M. Hagen, M. Potthast, M. Büüchner, and B. Stein, "Twitter sentiment detection via ensemble classification using averaged confidence scores," in *Proc. Eur. Conf. Inf. Retrieval*, 2015, pp. 741–754.

[7] Z. Jianqiang and C. Xueliang, "Combining semantic and prior polarity for boosting twitter sentiment analysis," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Dec. 2015, pp. 832–837.

[8] Z. Jianqiang, "Combing semantic and prior polarity features for boosting twitter sentiment analysis using ensemble learning," in *Proc. IEEE Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2016, pp. 709–714.

[9] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social Web," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 63, no. 1, pp. 163–173, 2012.

[10] G. Paltoglou and M. Thelwall, "Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, pp. 1–19, 2012.

[11] A. Montejo-Ráaez, E. Martínez-Cámara, M. T. Martíín-Valdivia, and L. A. Ureña-López, "A knowledge-based approach for polarity classification in Twitter," *J. Assoc. Inf. Sci. Technol.*, vol. 65, no. 2, pp. 414–425, 2014.

[12] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. 17th Conf. Int. Lang. Resource Eval.*, 2010, pp. 2200–2204.

[13] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1. Baltimore, MD, USA, Jun. 2014, pp. 655–666.

[16] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. 25th Int. Conf. Comput. Linguistics*, Dublin, Ireland, Aug. 2014, pp. 69–78,2014.

[17] Y. Kim, "Convolutional neural networks for sentence classification," presented at the Conf. Empirical Methods Natural Lang. Process. (EMNLP), Doha, Qatar, Oct. 2014, pp. 1746–1751.

[18] R. Johnson and T. Zhang. (2014). "Effective use of word order for text categorization with convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1412.1058

[19] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 919–927.

[20] D. Tang, Q. B. FuruWei, T. Liu, and M. Zhou Coooolll:, "A deep learning system for twitter sentiment classification," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, Aug. 2014, pp. 208–212.

[21] D. Tang, Y. N. FuruWei, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, Baltimore, MD, USA, Jun. 2014, pp. 1555–1565,2014.

[22] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Seattle, DC, USA, Oct. 2013, pp. 1631–1642.

[23] J. Li, M.-T. Luong, D. Jurafsky, and E. Hovy. (2015). "When are tree structures necessary for deep learning of representations?" [Online]. Available: https://arxiv.org/abs/1503.00185

[24] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. LREC*, vol. 10. 2010, pp. 1320–1326.
[25] F. Å. Nielsen. (2011). *Informatics and Mathematical Modelling*. Technical Univ. Denmark. Accessed: Sep. 9, 2016. [Online]. Available: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010
[26] R. Collobert *et al.*, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.
[27] M.-T. Luong, R. Socher, and D. Christopher, "Manning. Better word representations with recursive neural networks for morphology," in *Proc. Conf. Comput. Natural Lang. Learn.*, Sofia, Bulgaria, 2013, pp. 104–113.
[28] X. Zheng, H. Chen, and T. Xu, "Deep learning for Chinese word segmentation and pos tagging," in *Proc. Conf. Empirical Methods NLP*, 2013, pp. 647–657.
[29] J. Pennington, R. Socher, and D. Christopher. (2014). *Manning*. Accessed: Dec. 12, 2016. [Online]. Available: https://nlp.stanford.edu/projects/glove
[30] Z. Jianqiang, "Pre-processing boosting Twitter sentiment analysis?" in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Dec. 2015, pp. 748–753.
[31] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, pp. 2870–2879, 2017.
[32] (2005). *Internet & Text Slang Dictionary*. Accessed: Feb. 2, 2017. [Online]. Available: https://www.noslang.com/dictionary
[33] C. J. V. Rijsbergen, *Information Retrieval*, 2nd ed. Newton, MA, USA: Butterworth-Heinemann, 1979.
[34] Wikipedia. *List of Emoticons*. Accessed: Feb. 2, 2017. [Online]. Available: http://en.wikipedia.org/wiki/List of emoticons
[35] *Brendano GitHub.Com*. Accessed: Feb. 2, 2017. [Online]. Available: https://github.com/brendano/ark-tweet-nlp/tree/master/src/cmu/arktweetnlp
[36] A. Go, R. Bhayani, and L. Huang, *Twitter Sentiment Classification Using Distant Supervision*. document CS224N, 2009.
[37] A. Bakliwal, P. Arora, S. Madhappan, N. Kapre, M. Singh, and V. Varma, "Mining sentiments from tweets," in *Proc. 3rd Workshop Comput. Approaches Subjectivity Sentiment Anal., Assoc. Comput. Linguistics*, Jeju, Korea, 2012, pp. 11–18.
[38] (2014). *SemEval-2014 Task 9, Datasets*. Accessed: Sep. 9, 2015. [Online]. Available: http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools
[39] H. Saif, M. Fern, and Y. He, "Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold," in *Proc. 1st ESSEM Workshop*, Turin, Italy, 2013, pp. 21–26.
[40] S. Narr, M. Hulfenhaus, and S. Albayrak, "Language-independent twitter sentiment analysis," in *Proc. Knowl. Discovery Mach. Learn. (KDML)*, 2012, pp. 12–14. [Online]. Available: http://data.dai-labor.de/corpus/sentiment/
[41] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph," in *Proc. 1st Workshop Unsupervised Learn. NLP*, 2011, pp. 53–63.
[42] H. Saif *et al.*, "Contextual semantics for sentiment analysis of twitter," *Inf. Process. Manage.*, vol. 52, no. 1, pp. 5–19, 2016.
[43] H. Saif *et al.*, *SentiCircles for Contextual and Conceptual Semantic Sentiment Analysis of Twitter*. Cham, Switzerland: Springer, 2014, pp. 83–98.
[44] H. Saif, Y. He, M. Fernandez, and H. Alani, "Adapting sentiment lexicons using contextual semantics for sentiment analysis of twitter," in *Proc. Eur. Semantic Web Conf.*, 2014, pp. 54–63.
[45] S. Rosenthal, P. Nakov, and S. Kiritchenko, "SemEval-2015 task 10: Sentiment analysis in twitter," in *Proc. SemEval*, 2015, p. 8.
[46] J. An, X. Gui, J. Jiang, L. Qi, and J. Yang, "Semi-supervised learning of k-nearest neighbors using a nearest-neighbor self-contained criterion in for mobile-aware service," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 27, no. 5, pp. 1–13, 2013.

ZHAO JIANQIANG is currently pursuing the Ph.D. degree with Xi'an Jiaotong University, Xi'an, China. His research interests include social network and Web mining.

GUI XIAOLIN received the B.Sc. degree in computer and the M.Sc. and Ph.D. degree in computer science from Xi'an Jiaotong University (XJTU), China, in 1993 and 2001, respectively. In 1988, he joined XJTU, where he has been an Active Researcher in network computing, network security, and wireless networks. He has been serving as the Director of the Key Laboratory of Computer Network, XJTU, China, since 2008, where he is currently a Professor. His recent research covers secure computation of open network systems, including grid, P2P, and cloud, dynamic trust management theory, development on community network.

ZHANG XUEJUN received the M.E. degree in computer science and technology from the Southeast University, Jiangsu, China, in 2008, and the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, China, in 2016. He was with the Electronic Information School, Wuhan University, in 2010, for one year as a Visiting Scholar. He is currently an Associate Professor with the Department of Computer Science, Lanzhou Jiaotong University, China. His research interests include data mining privacy, location privacy-preserving, cloud computing security, and mobile security.