Advanced Combined LSTM-CNN Model for Twitter Sentiment Analysis

Nan Chen, Peikang Wang

University of Science and Technology of China, Hefei 230022, China
nan022@mail.ustc.edu.cn, wangpk@ustc.edu.cn

Abstract: In this paper, we proposed an advanced model which is based on the LSTM-CNN model presented by Pedro M. Sosa for twitter sentiment analysis. We combined the encoder-decoder framework with the regular LSTM-CNN framework. In this model, LSTM can 'remember' forward information of the sequence and multilayer CNN can catch and learn local information sufficiently. Meanwhile, the multilayer CNN is also regarded as an encoder and a two-layer deconvolution part is the corresponding decoder. This encoder-decoder framework is used to reconstruct the input matrix, this process of the reconstruction of input matrix by decoder makes the features learning in CNN much more intrinsic and effective. As the result, the more effective the feature learning is, the higher accuracy rate the classifier will achieve. Furthermore, our framework can also be used for other classification issues besides sentiment analysis. This work will make sense in fields such as machine learning and natural language processing.

Keywords: neural network; sentiment analysis; machine learning; feature extraction

1 Introduction

In recent years, social media service is getting more and more popular and is becoming an indispensable part in our life. Most of people are familiar with twitter, face book or micro-blog. Social work has been widely used on communication, information, advertising and social events [1]. As the most widely used social media platform, twitter is a valuable and effective source of information, twitter has nearly 350 million active users posting around 500 million tweets per day [2] reflecting users' daily life, comments about issues and so on. Every tweet posted expressed the user's sentiment which can be happy, exciting or angry, sorrow. In this paper, we investigate the sentiment of twitters expressed.

Sentiment analysis has grown to be one of the most active research topics in natural language processing (NLP). Sentiment analysis is studying people's sentiment, attitude or emotion toward some issue according to the texts they write [3]. Sentiment analysis can be expressed as three levels: sentence, aspect and document level [4]. In this paper, we investigate the framework for sentiment analysis in sentence level identifying the polarity of a sentence. There are many different ways in dealing with sentiment analysis. In lexicon based approach, sentiment lexicon is a lexicon consists of positive sentiment words and negative sentiment words which can be used to analysis sentiment. But there are many challenges relying on sentiment lexicon [4]. Experimental studies showed that corpus-based approach performs much better than lexicon-based approach [5]. In machine learning based approaches, a text is usually transformed into features by using bag of words, and then features are fed into classifier which is usually based on Naive Bayes (NB) or Support Vector Machine (SVM) or Decision Tree (DT) [6]. It's showed that machine-learning based ways outperform human generated classifications.

In recent years, deep-learning method is catching more and more attention which can achieve state-of-art performance in many field of NLP, for example, in sentiment analysis. Many approached has been proposed. Socher proposed semi-supervised recursive auto encoders for predicting sentiment distributions in 2011. Kalchbrenner [7] proposed a CNN models for sentiment classification which can deal with varying-length input sentences. Irsoy and Cardie [8] presented a deep recursive neural network (DRNN) with stacking multiple recursive layers on sentiment classification tasks. Tai, Socher [9] proposed a long short-term memory (LSTM) framework on sentiment classification which improves the semantic representations and outperforms many existing systems.

In this paper, we focus on improving the performance of sentiment analysis on twitter data by combining deep learning framework with a decoder part. Our goal is trying best to classify a twitter into 'positive' or 'negative'. This can be used to analysis whether some aspect of a product or issue is good or not.

2 Related work

Our work is based on a combined LSTM-CNN framework proposed by Sosa P M. [10]. We would introduce related neural network here.

2.1 Convolutional neural network (CNN)

Nowadays, CNN has been widely used in image recognition, nature language processing as well as other fields which has the ability to recognize local features with convolution kernel and then learn features automatically to do classification. It has achieved good performance in many different fields. In this paper, we used CNN for sentence classification [10] as shown in figure 1.

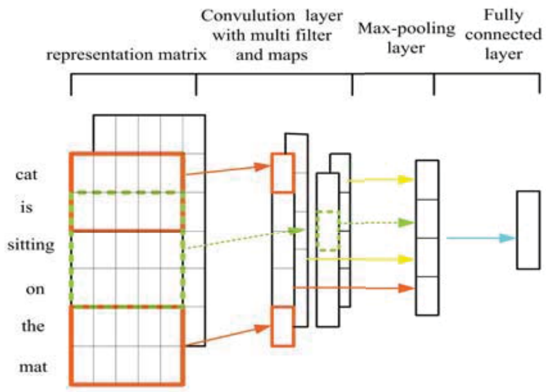

Figure 1 CNN for sentence classification

Assumed that the fed sentence S is with length n, S is represented as a matrix $\mathbf{x} = [x_1, x_2, \ldots, x_i, \ldots x_n]$, i_{th} element x_i is a k dimension vector representing the word embedding of i_{th} word in S. Then the matrix $\mathbf{x} \in R^{n \times d}$ is fed into convolution layer which is consisted of multiple filters which will capture different features. After convolution, the output then is fed to the pooling layer to be sub-sampled into smaller dimension. Finally, the output is fed to fully connected layer.

2.2 Long-short term memory neural networks (LSTM)

Long Short-Term Memory network [12] is a type of Recurrent Neural Network (RNN) which will perform better than RNN on tasks involving long time lags. As LSTM has three gates which can determine the flow should be remembered or not. The "input gate" controls if the input of new information can be memorized. The "forget gate" determine how long certain values can be held in memory. And the "output gate" controls how much the value stored in memory affects the output activation of the block. As shown in figure 2.

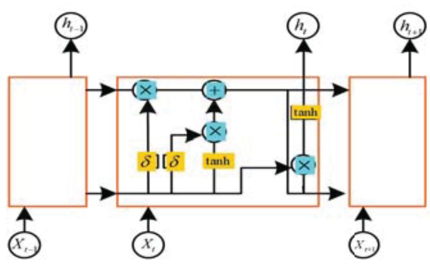

Figure 2 Unrolled LSTM network

2.3 LSTM-CNN model

Every sentence of twitters is represented as a word embedding matrix as input of the LSTM layer. LTSM layer can learn the previous information because it's ability of 'remember', then the new output generated by LSTM layer is input of CNN layer which can and be good at capturing local features. After that, the output is fed into max-pooling and fully connect layer, finally, be classified as positive or negative.

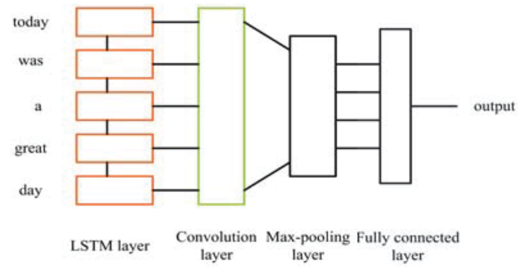

Figure 3 combined LSTM-CNN network

This model achieved around 2.7%-8.5% better than regular models as showed in the paper [10].

2.4 Encoder-decoder framework

Neural machine translation [13] is widely used approach in machine translation. Most of the proposed neural machine translation models are consists of a framework called encoder-decoder as shown in figure 4.

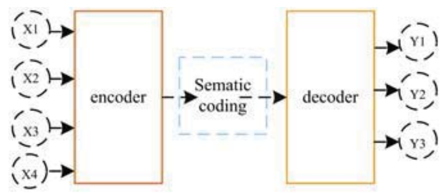

Figure 4 encoder-decoder framework

The encoder neural network reads and encodes the input sentence into a fixed-length vector then the decoder outputs a translation from the encoded vector.

3 Advanced LSTM-CNN model

We proposed the advanced LSTM-CNN model is based on the model proposed by Sosa P M. and encoder-decoder framework which is shown as follow.

3.1 Advanced LSTM-CNN model

As we know, the purpose and role of convolution layer is capturing and learning the feature of the input, the more layer of CNN is, the better the features will be learned but it will be harder for parameters learning. So, in our model, we used a four-layer CNN which do much better than single layer model.

In our model as shown in figure 5, our model named LSTM-MCNN-decoder is combined with encoder-decoder framework with multilayer LSTM-CNN model, we make use of the decoder part to reproduce the input matrix, this process will lead to a more intrinsic and effective features learning of convolution which will leads to a better final performance.

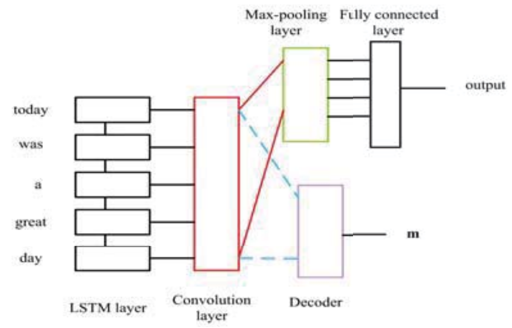

Figure 5 LSTM-MCNN-decoder Model

The model is composed of four parts as follow:

(1) Input matrix of convolution neural network. All the word vectors are concatenated into two-dimensional matrix as the input matrix of the convolution neural network. Sentence $S=\{s_1,s_2, \quad s_i, \quad ,s_n\}$ with n word is represented as matrix $\mathbf{x}=[x_1,x_2, \quad ,x_i, \quad x_n]$, $\mathbf{x}\in R^{n\times d}$ the i_{th} element x_i is a d dimension vector representing the word embedding of i_{th} word s_i in S.

(2) Convolution neural network. In our model, the convolution neural network model is constructed by the four-layer convolution layer to extract the important characteristic information in the sentence. Convolution is conducted by convolution kernel. For a kernel with length l:

$$c_i = f(\boldsymbol{\omega}\cdot\mathbf{x}_{i:i+l-1}+b) \tag{1}$$

Here, $\boldsymbol{\omega}\in R^{l\times d}$ is the weights matrix of the kernel, $\mathbf{x}_{i:i+l-1}$ is the word embedding matrix for the kernel. For the sentence with length n, then we get the feature vector:

$$\mathbf{c}=[c_1,c_2, \quad ,c_i, \quad c_n] \tag{2}$$

Here, $\mathbf{c}\in R^{n-l+1}$.

(3) Max pooling layer and fully-connected layer. We make use of max-pooling to extra the most important information of these feature and reduce the complexity of parameters, $\hat{\mathbf{c}}=\max\{\mathbf{c}\}$, for the window with m kernels, we get:

$$\hat{\mathbf{c}}=[\hat{c}_1,\hat{c}_2, \quad ,\hat{c}_m] \tag{3}$$

Here, $\hat{\mathbf{c}}\in R^{m}$ is the extracted features vector.

The utilize activation function to predict the emotional tendency:

$$\hat{y}(\mathbf{x})=g(\hat{\mathbf{c}}) \tag{4}$$

Here, g() is the activation function, $\hat{y}(\mathbf{x})$ is the predicted label (1 for positive , 0 for negative) of input sentence.

(4) The matrix reproduce layer. We utilized the decoder layer to reproduce the input matrix.

$$\mathbf{m} = deconvolution(\mathbf{c}) \tag{5}$$

Here, $\mathbf{m}\in R^{n\times d}$, is the reconstructed matrix of $\mathbf{x}$.

3.2 Model training

In our model, there are two different losses to update all parameters together because of the two outputs. Here, output is the predicted label of input sentences while the second output $\mathbf{m}$ is the reconstructed matrix of $\mathbf{x}$ by decoder.

Therefore, our loss function is:

$$\mathrm{loss} = \frac{1}{2\mathrm{n}}\sum(\left\|\mathrm{y}(\mathbf{x}) - \hat{\mathrm{y}}(\mathbf{x})\right\|^2 + \alpha * \mathrm{sim\_cos}(\mathbf{x},\mathbf{m})) \tag{6}$$

α is the parameter controlling the effect of deconvolution. $y(\mathbf{x})$ is the real label of input sentences and $y(\mathbf{x})$ is the predicted label. And $sim\_cos()$ is the function to estimate cosine similarity between input $\mathbf{x}$ and reconstructed matrix $\mathbf{m}$. Then, update parameters by back propagation algorithm.

4 Experiments and results

4.1 Dataset

The twitter dataset we used is a combination of the University of Michigan Kaggle competition dataset [14] and the "Twitter Sentiment corpus" created by Neik Sanders [15] which is consist of 1,578,627 tweets labeled as "positive" or "negative". The data we used to train and test the model is collected from the twitter set randomly which contains 100,000 twitters.

4.2 Experiments

The general parameters used in our model are shown in table 1.

Table 1 general parameters

Embedding dimension	128
Epoch	10
Dropout	0.5
Word embedding	No pretraining

The weight α in function (6) controls how much the decoder part will affect the parameters learning.

Table 2 accuracy of multilayer CNN-decoder with different α

α	**1/700**	**1/500**	**1/300**
Accuracy	0.7583	0.7724	0.76

As table 2 showed, when $\alpha = 1/500$ we get the best result.

On the other hand, the number of deconvolution layers of the decoder will affects a lot. Here is the average accuracy of CNN-decoder models with different decoder layers in which the CNN is with 4 layer.

Table 3 average accuracy of different CNN-decoder

Deconvolution layers	1 layer	2 layers	4 layers
Accuracy	0.779	0.7818	0.7724

Therefore, we know the best decoder is the one with 2 deconvolution layers.

4.3 Results

Being based on these best parameters, we do experiments comparing with other baseline. We compared our model with single layer CNN, multilayer CNN, LSTM-CNN and CNN-LSTM proposed by Sosa P M as well as multilayer LSTM-CNN (LSTM-MCNN) model.

Table 4 average accuracy of different models

Model	Average accuracy
CNN	75.18 %
Multilayer CNN	76.09 %
Multilayer CNN-decoder	78.18 %
CNN-LSTM	76.36 %
LSTM-CNN	76.91 %
LSTM-MCNN	77.09 %
LSTM-MCNN-decoder	78.6 %

4.4 Results analysis

As shown in table 3, the LSTM-MCNN-decoder model outperforms all other models. We can learn that Multilayer CNN perform nearly 1% better than the CNN and LSTM-MCNN performs 1.1% better than LSTM-CNN which indicates that multilayer of convolution makes feature learning more effective. In addition, Multilayer CNN-decoder performs nearly 2% better than Multilayer CNN and LSTM-MCNN-decoder performs nearly 1.6% better than LSTM-MCNN which indicate that the decoder part indeed improve the accuracy of classification because of making the features extraction and learning more intrinsic and effective.

These results showed that our initial intuition was correct the decoder part can make making the features extraction and learning more intrinsic and effective.

5 Conclusions

We proposed a new model LSTM-MCNN-decoder based on the LSTM-CNN model proposed by Sosa P M which achieve the state-of-art performance. This model makes good use of multilayer LSTM-CNN in which LSTM can 'remember' forward information of the sequence and multilayer CNN catch and learned local information sufficiently. Our model also makes use of encoder-decoder framework in which the multilayer CNN is regard as encoder and the two-layer deconvolution layer is the corresponding decoder. The process of the reconstruction of input matrix by decoder makes the features learning in CNN much more intrinsic and effective.

Our model is not restricted to sentiment analysis, it's a good model of classification which can be used in other machine learning fields. In the future, we can try to connect our model with other nature language processing technology like Part-of–Speech tagging trying to get a better result in issues of NLP.

Acknowledgements

This work is supported by University of Science and Technology of China. And thanks for the help of my supervisor and senior schoolfellow.

References

[1] Alarifi A, Alsaleh M, Al-Salman A M. Twitter turingtest:Identifying social machines. Information Sciences, 2016, 372:332-346.

[2] Crannell, W.C., Clark, E., Jones, C., James, T.A., Moore, J. A pattern-matched twitter analysis of US cancer-patient sentiments. J. Surg. Res. 2016, 206 (2):536–542.

[3] Liu B. Sentiment analysis: Mining opinions, sentiments, and emotions. Computational Linguistics, 2016, 42(3):1-4.

[4] Beaudouin V. Note de lecture : Bing Liu. Sentiment Analysis and Opinion Mining. Morgan & Claypool publishers, 2012[M]. 2012.

[5] S. Zhou, Q. Chen, X. Wang, Fuzzy deep belief networks for semi-supervised sentiment classification. Neurocomputing, 2014, 131:312–322.

[6] R. Xia, C. Zong, S. Li, Ensemble of feature sets and classification algorithms for sentiment classification. Inf. Sci, 2011, 181:1138–1152.

[7] Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences. Eprint Arxiv, 2014, 1.

[8] ICardie C. Deep recursive neural networks for compositionality in language. International Conference on Neural Information Processing Systems, MIT Press, 2014:2096-2104.

[9] Tai K S, Socher R, Manning C D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. Computer Science, 2015, 5(1):: 36.

[10] Sosa P M. Twitter Sentiment Analysis using combined LSTM-CNN Models. 2017.

[11] Kim Y. Convolutional Neural Networks for Sentence Classification. Eprint Arxiv, 2014.

[12] Sundermeyer M, Schlüter R, Ney H. LSTM Neural Networks for Language Modeling. Interspeech, 2012:601-608.

[13] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. Computer Science, 2014.

[14] University of Michigan, "Sentiment classification dataset." https://inclass.kaggle.com/c/si650winter11, 2011.

[15] N. Sanders, "Twitter sentiment corpus". "http://www.sananalytics.com/lab/twitter sentiment/, 2011.