ATLANTIS PRESS

*Advances in Computer Science Research, volume 92*

International Conference on Mathematics, Big Data Analysis and Simulation and Modeling (MBDASM 2019)

# Sentiment Analysis Augmented by Emoticons

Linyu Li

School of Software Engineering, Shandong University, Jinan 250101, China

***Abstract*—Social media platforms are the main resources to collect people's sentiments and opinions. We can extract quantities of useful information from the social network. Weibo is the most popular social networking application in China. In this paper, we'll describe our attempts at producing a state-of-art Weibo sentiment classifier using CNN, LSTM and existence of emoticons in users' microblogs. The experiments carried out on standard datasets including 120,000 microblogs and then group them into positive and negative sentiments. The models include character-based classifier and emoticon-based classifier. To boost performances, we assembled character-based classifier and emoticon-based classifier together to realize a compound classifier. We also implemented necessary experiments to measure the accuracy. The final results prove that emoticons in microblogs can improve the performance of traditional sentiment classifiers.**

***Keywords—Weibo sentiment analysis; social network; emoticons; classification***

## I. INTRODUCTION

With the rapid development of science and technology, quantities of people have been blessed with the convenience of the Internet. More and more social media platform come into people's eyes. Social media is an online media where people can express their sentiments or opinions about the services they have received or even the government affairs. In China, one of the best-known social media platform is Weibo which is also known as microblog.

Weibo allows its users to express viewpoints on different kinds of topical subjects and discuss issues that occur, which is now considered as a valuable online source for opinions. As we all know, a piece of microblog has only 140 characters which makes it easy for both senders and readers to share their feelings and communicate with each other anywhere and anytime in the world. Weibo is a "what's - happening - right - now" social media platform, thus we can extract immediate sentiments and opinions around the whole world[1].

We could use quantities of microblogs to extract useful information in a variety of fields. The public's sentiments concerning a particular topic is one of the most significant information that can be acquired by analyzing the microblog data. Problems can be evaluated for the further improvement if we can understand people's sentiments. Hence, extracting the potential sentiments under the surface of microblogs is of great benefit[2].

As it is known to us all, a piece of microblog is consist of some characters, punctuation and some emoticons. It is easier and more convenient for users to express their sentiments or feelings with the assistance of emoticons. Let's take "I am so happy today![smile]" as an instance. In fact, the emoticon "[smile]" are more likely to express the emotion "happy" than the prosy word "happy". For example, "My friends think I am happy, but actually I am sad.[cry]". In this microblog, there are 2 words which can express subjective emotions: "happy" and "sad". Humans can easily distinguish which word is the predominant one in sentiment analysis, however, computers may be confused when faced with two or more than two words which can express subjective feelings. Thus, emoticons can play a significant role in sentiment analysis.

In this paper, we tried to figure out whether and how the existence of emoticons in microblog can influence the performance of sentiment mining system. This paper is structured as follows: Section 2 discusses preliminaries and related work, Section 3 describes several methods in detail which we've come up with based on the classic classification model, Section 4 presents the results of our models and further experiments to measure the rationality of our proposals. Conclusions and future work directions are depicted in Section 5.

## II. PRELIMINARIES AND RELATED WORK

### *A. LSTM (Long Short-Term Memory)*

Let us now describe the architecture of the LSTM system. Long short-term memory (LSTM) is a recurrent neural network (RNN) architecture (an artificial neural network) published in 1997 by Sepp Hochreiter and Jürgen Schmidhuber[3]. Like most RNNs, an LSTM network is universal in the sense that given enough network units it can compute anything a conventional computer can compute, provided it has proper weight matrix, which may be viewed as its program. Unlike traditional RNNs[4][5], an LSTM network is well-suited to learn from experience to classify, process and predict time series when there are very long time lags of unknown size between important events. This is one of the main reason why LSTM outperforms alternative RNNs and Hidden Markov Models[6] and other sequence learning methods in numerous applications.

### *B. CNN (Convolutional Neural Network)*

Let us now describe the architecture of the CNN we worked with. Convolutional neural network (CNN) is an efficient recognition method which has been developed in recent years and attracted wide attention. In the 1960s, Hubel and Wiesel found that their unique network structure could effectively

reduce the complexity of the feedback neural network when they studied the neurons used for local sensitivity and direction selection in the cat's cerebral cortex. Convolutional Neural Networks (CNN) was proposed. Nowadays, CNN[7] has become one of the research hotspots in many scientific fields, especially in the field of pattern classification. Because the network avoids the complex pre-processing of images and can input the original images directly, it has been widely used. K. Fukushima's[8] new recognition machine in 1980 is the first implementation network of convolutional neural network. Subsequently, more researchers improved the network.

### *C. Related Work*

Existing theses and researches on sentiment analysis and opinion mining are mentioned below. Roman Bartusiak and his teammates[9] in their research proposed Transfer Learning approach for sentiment analysis which means learning on one dataset and testing on another. After the extracting the knowledge from the dataset where the classifier was trained, the classifier predicts the sentiment for the specific textual dataset. This approach was proved to be effective. Yequan Wang and his team[10] reveal that the sentiment polarity of a sentence is closely relevant to the concerned aspect. To carry out aspect-level sentiment classification, they eventually proposed an Attention-based Long Short-Term Memory Network which can achieve a state-of-art performance.

Furthermore, in Hamid Bagheri and Md Johirul Islam[11]'s experiments, they realized that the neutral sentiment for text are significantly high which clearly shows the limitations of the current works. The research by Juncai Guo and Xue Chen[12] focused on the word bias in Weibo which includes objective bias and subjective bias. For the purpose of dealing with the relations of topic, bias, and sentiment appropriately, they proposed an integrated classification model named Bias-Sentiment-Topic (BST) model which has a major improvement in sentiment classification.

## III. METHODS

In this paper, with the aim of figuring out whether and how the existence of emoticons in microblog can influence the performance of sentiment mining system, we totally designed 4 models and carried out 4 experiments to compare the performances among them. sentiment analysis of each model is obtained through several processes which I will describe in details below:

### *A. Simple-Character-Based Model*

For the first model, we simply used the characters in microblogs as the classification criterion and removed all emoticons in the microblogs to reduce the noise. The processes include text preprocessing, building index dictionary and word vector dictionary for the training dataset and LSTM classification. Figure 1 shows the complete process.

- Text Preprocessing: Text preprocessing aims to convert the original text data into concise data and eliminate the noise and then we can obtain more optimal calculation results. This stage includes emoticon removal, stopword removal, cleansing and tokenizing.
- Building Index Dictionary And Word Vector Dictionary: Index dictionary and word vector dictionary are used in the LSTM classification. The text corpus are converted into word vector by the word2vec.

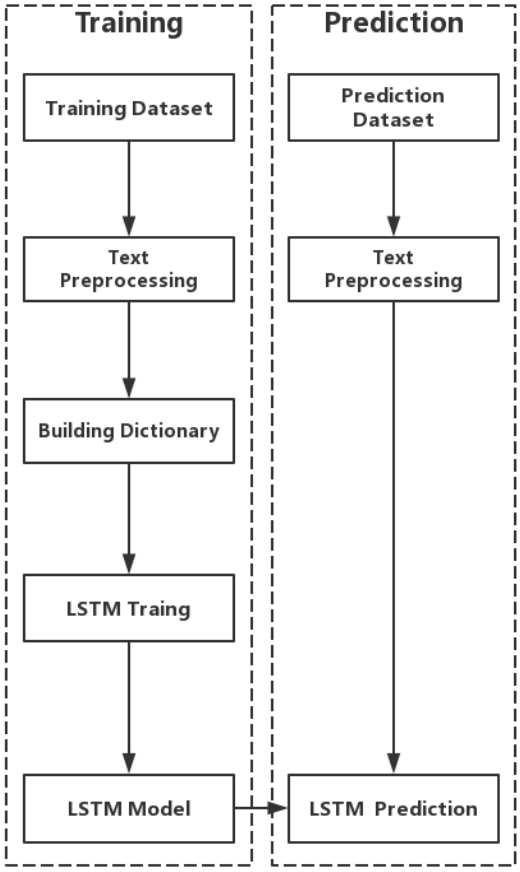


FIGURE I. SIMPLE-CHARACTER-BASED MODEL

- LSTM Classification: Long Short-Term Memory(LSTM) in this research is used in both training and prediction. This study used Keras which is a library that supports LSTM functions such as training and classification.

### *B. Simple-Emoticon-Based Model*

For the second model, the emoticons in microblogs such as "[smile]" are the key factors and the classification criterion. This classifier simply uses emoticons as classification criterion for the reason that almost every piece of microblog contains one or more emoticons which include more information about sentiments. In addition, words or characters sometimes may cause ambiguity. For instance, "My friends think I am happy, but actually I am sad.[cry]". In this microblog, there are 2 words which can express subjective emotions: "happy" and "sad". Humans can easily distinguish which word is the

predominant one in sentiment mining, however, computer may be confused when faced with two or more than two words which can express subjective feelings. Hence, emoticons can outperform characters in this perspective.

The processes contain data filtering, manual labeling for the training dataset, CNN classification that is shown in Figure 2.

- Data Filtering: This stage aims to delete the microblogs which do not contain emoticons to create a new dataset where every piece of microblog contains one or more emoticons.
- CNN classification: Like the Simple-Character-based Model, the Simple-Emoticon-based Model also needs two classes: positive class and negative class. The CNN model divides the microblogs into two classes according to the categories of the emoticons in the microblogs. For instance, on one hand, there are emoticons like "[smile]" or "[happy]" in a microblog, this microblog tends to be a positive one. While on the other hand, emoticons such as "[angry]" or "[cry]" may make the microblog possess a sense of negative to great extent. The CNN process includes Imageinput, Convolution, Maxpooling and Fully-ConnectedNeural Network Calculation. Furthermore, TensorFlow is an essential library in this stage which will support CNN functions like training and classification.

### *C. Character-Emoticon-Mixed Model*

After possessing the foundation of the Simple-Character-based Model and the Simple-Emoticon-based Model, it is of vital significance to realize the idea of mixing the superior feature of characters and emoticons.

For the third model, we mixed the characters and emoticons in a simple and easy manner which is converting the pictures of emoticons into characters. For instance, the little picture of a face with a bright smile is the emoticon "[smile]" or "[happy]" in Weibo. Hence, we converted all emoticons into the character form to mix characters and emoticons and created a new dataset with pure text eventually.

With the pure-text dataset, we could carry out the model training with the Simple-Character-based Model. The processes contain converting emoticons, text preprocessing, building index dictionary and word vector dictionary for the training dataset and LSTM classification that is shown in Figure 3. Unlike the first model's process, this model added converting emoticons.

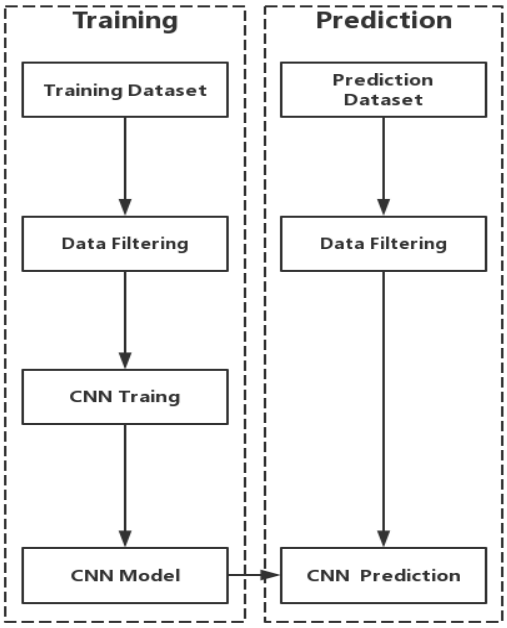


FIGURE II. SIMPLE-EMOTICON-BASED MODEL

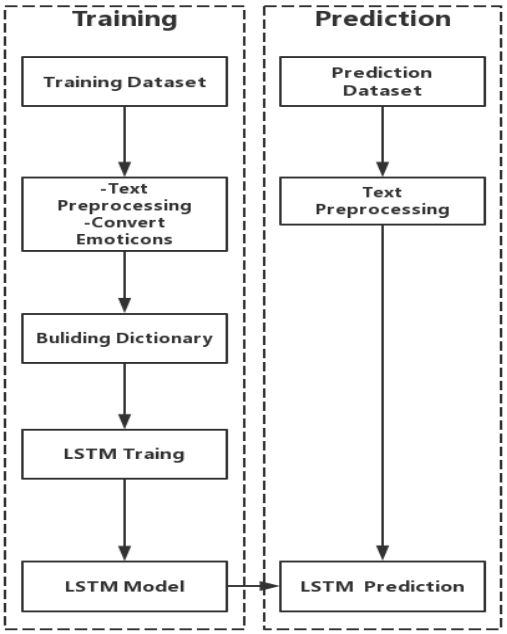


FIGURE III. CHARACTER-EMOTICON-MIXED MODEL

### *D. Character-Emoticon-Integrated Model*

The Character-Emoticon-Mixed Model is an immature model to some extent for the reason that it just combined the features of character and emoticon mechanically, not in an organic way.

For the forth model, we integrated characters and emoticons more reasonably. The Character-Emoticon-Integrated Model which has a higher integration than the Character-Emoticon-Mixed Model is consist of Simple-Character-based Model and

Simple-Emoticon-based Model. The key idea of this model is as follows:

- Using the training dataset to train the Simple-Emoticon-based Model.
- Using another dataset as the prediction dataset for the trained Simple-Emoticon-based Model and then dividing the results as two classes: positive class and negative class. Saving the results as two new datasets.
- Using above two new datasets as the training sets for the Simple-Character-based Model. Doing some essential preprocessing work and then training the Simple-Character-based Model.
- The trained Simple-Character-based Model is the final Character-Emoticon-Integrated Model. The whole processes of the Character-Emoticon-Integrated Model is shown in Figure 4.

## IV. Results and Discussion

In order to evaluate the performance of each model, we carried out four experiments.

The standard dataset we depended on contains 120,000 microblogs including 60,000 positive microblogs and 60,000 negative ones. After the data filtering process, we picked out 101,983 pieces of microblogs which contain emoticons. And the size of our dataset is 23.6 MB.

The measure of the performance for judgment is the classification accuracy. The results are shown in Table 1. In this research, we used the confusion matrix to test the accuracy of each model. The confusion matrix includes four conditions: "True Positive (TP)", "True Negative (TN)", "False Positive (FP)", and "False Negative (FN)" which are shown in Table 2.

The computational formulas are as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

According to the table 1, the Model 4 has the best performance, however, the Model 1 ranks the last. It is clear that emoticons in microblogs play a significant role in improving the classification model.

The results of the experiment 1 and 2 could prove that emoticons have higher classification performance.

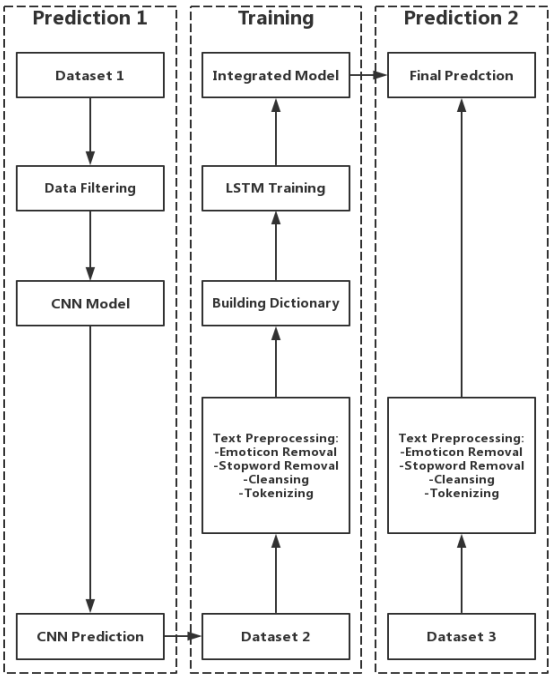


FIGURE IV. CHARACTER-EMOTICON-INTEGRATED MODEL

TABLE I. ACCURACY RESULTS

| Model | Accuracy |
|---|---|
| Simple-Character-Based Model (Model 1) | 0.886 |
| Simple-Emoticon-Based Model (Model 2) | 0.903 |
| Character-Emoticon-Mixed Model (Model 3) | 0.892 |
| Character-Emoticon-Integrated Model (Model 4 ) | 0.916 |

TABLE II. CONFUSION MATRIX

| | **Prediction (Positive)** | **Prediction (Negative)** |
|---|---|---|
| **Actual (Positive)** | TP | FN |
| **Actual (Negative)** | FP | TN |

Furthermore, the compound classifiers like the Model 3 and the Model 4 are supposed to behave better than the simple ones conceivably. But, the accuracy of the Model 3 is only 0.892 which is lower than that of the Model 2.

Then why the Model 3 behaves worse than the simpler classifier? The probable reason may be that the Model 3 just combine the character and the emoticon in an immature manner. It just converts the emoticons into characters which leads to the classifier obtain the feature of characters. But unfortunately, it lost the better feature of emoticons. Consequently, the Model 3 just have a little bit more sentiment-characters than the Model 1, which means the enhancement effect is not that high like the Model 2.

## V. Conclusion and Future Work

In this paper, to understand whether and how the existence of emoticons in microblogs can influence the performance of sentiment analysis system, we designed four models and carried out 4 experiments. The key idea of these classifiers are using the high classification accuracy of the emoticons. The experiments compared several classifiers' accuracy. And the result is emoticons can improve the performance of the traditional sentiment analysis classifiers.

In the future, with the aim of improving the accuracy, we would want to compare the classification performance using other classification method and feature. Furthermore, an interesting and possible direction may be to carry out other experiments to implement sentiment analysis in different natural languages, not just in Chinese.

## References

[1] B. Agarwal and N. Mittal, "Optimal feature selection for sentiment analysis," in Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing '13), vol. 7817, pp. 13–24,2013.

[2] J.Scott . Social network analysis: developments, advances, and prospects[J]. Social Network Analysis & Mining, 2011, 1(1):21-26.

[3] Hochreiter S, Schmidhuber, Jürgen. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[4] Z.C.Lipton. A Critical Review of Recurrent Neural Networks for Sequence Learning[J]. Computer Science, 2015.

[5] Jozefowicz, Rafal , W. Zaremba , and I. Sutskever . "An Empirical Exploration of Recurrent Network Architectures." International Conference on International Conference on Machine Learning JMLR.org, 2015.

[6] S R.Eddy. Hidden Markov models.[J]. Current Opinion in Structural Biology, 1996, 6(3):361-5.

[7] N.Ketkar. Convolutional Neural Networks[J]. 2017.

[8] Fukushima, Kunihiko , and S. Miyake . "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition.".

[9] R.Bartusiak, L.Augustyniak, T.Kajdanowicz, P.Kazienko. [IEEE 2015 Second European Network Intelligence Conference (ENIC) - Karlskrona, Sweden (2015.9.21-2015.9.22)] 2015 Second European Network Intelligence Conference - Sentiment Analysis for Polish Using Transfer Learning Approach[J]. 2015:53-59.

[10] Y.Wang, M.Huang, X.Zhu, L.Zhao. Attention-based LSTM for Aspect-level Sentiment Classification[C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[11] H.Bagheri, M.J.Islam. Sentiment analysis of twitter data[J]. 2017.

[12] J.Guo, X.Chen. Bias - Sentiment - Topic model for microblog sentiment analysis[J]. Concurrency & Computation Practice & Experience, 2018.