ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Shanghai, 19-21 June, 2020, pp. 62-69
doi: 10.18178/wcse.2020.06.011

CNN-BiLSTM with Attention Model for Emoji Recommendation

Yan Wang [1 +], Yixian Di [2]

[1] School of Software Engineering, University of Science and Technology of China, China
[2] Viterbi School of Engineering, University of Southern California, United States

Abstract. Emojis are ideograms which are naturally combined with plain text to visually complement or emphasis the meaning of a message. In social communication, if plain text sentences are used in combination with different emojis, the meaning can be very different. Emoji recommendations are designed to solve the time-consuming issue of choosing an emoji among multiple possibilities on social platforms. The emoji recommendation task is defined as predicting the most likely emoji from a given text, as known as emoji prediction. Although emoji has been widely used on social platforms, the relationship between words and emojis is rarely researched by natural language processing (NLP). Also, the existing studies about emoji prediction are rarely based on Chinese corpus. For this purpose, we study the novel task of emoji prediction based on Chinese Weibo corpus. In this paper, we proposed a CNN-BiLSTM with attention model (CBLA) on the emoji prediction of Chinese social media. Our model is based on bidirectional long short-term memory (BiLSTM) layer with context-aware self-attention mechanism and convolutional layer (CNN). Experimental results show that our method achieved a good result and outperforms other prediction models.

Keywords: emoji recommendation, long short-term memory, attention mechanism.

1. Introduction

In the last few years, various forms of visual expression such as emoticon, emojis and stickers have become very popular in social platforms and gradually changed the way people communicate on social media. These graphic symbols can be used in non-verbal communication of the Internet to express feelings and emotions and is indispensable as an auxiliary tool for enhancing emotions. From the perspective of natural language processing, they provide extra information about the semantics of sentences.

With the divergence of social platform, the number of emojis has also increased. Nowadays, users often have to open the emoji keyboard and scroll up and down to select the most suitable one from hundreds of emojis, which seriously affects the user experience and wastes users' time. However, with the help of NLP, most of the time we can easily predict what users want to use according to what they have typed. In order to solve this problem, we propose an emoji prediction method focusing on investigating the relationship between emojis and words. It's not an easy task. there are still some problems that we need to take care of. First, every emoji has unique emotional meaning in a different context. For example, some emojis have very clear emotional meanings, such as 😁 indicating positive sentiment, 😭 indicating negative sentiment. However, 🙂 in Chinese social media sometimes can be used to express friendly smile, the other times can express dissatisfaction and irony. For example, "Scored 59 points, I really don't care. 🙂 ", it more likely means "I'm so angry that I almost passed 😡 ". Also, the meaning of emojis are not always the same in different cultures, it is challenging to predict emojis in natural language processing. Another problem of current language model used for emoji prediction is, the incomplete feature extraction of the contextual information due to ignoring the relationship between each word. Which doesn't make sense, since the prediction of the emoji should be determined by all the context. To solve the proposed problems, we employed a novel model which contains a

[+] Corresponding author. Tel.: + 18896533832.
E-mail address: sa517333@mail.ustc.edu.cn.

BiLSTM network, context-aware self-attention mechanism, and the convolution layer. The convolution layer means to extract the higher-level phrase representations from the word embedding vectors. BiLSTM is used to produce both the preceding and succeeding context representations of entire text. Attention mechanism is used to give different distribution of the attention weights for each part of text. Finally, softmax classifier help us classify the processed context information.

The remainder of this paper is organized as follows. First, we summarize the related work (Section 2). Then we introduce our method in detail (Section 3), and at the end we analyse the experimental results (Section 4.1).

2. Related Work and Background

In recent years, we have witnessed the widespread use of emoji in social text communication. Emoji can better simplify emotional expression and convey emotions that text cannot express, making communication more fluid. Experimental results show that users are more satisfied with the communication of emoji [1]. Emoji does help users relate emotions and rich information of messages [2]. As the public and business are increasingly interested in the commercial value of social media, the analysis of the semantics of emoji has become an important research aspect of NLP researchers. In order to use the semantics of emojis to enhance any upstream and downstream NLP tasks, Eisner et al. [3] trained a Unicode description of emojis to obtain the embedding of the emojis. Wijeratneet al. [4] used emoji descriptions and emotion definitions to improve the emoji embedding model. However, these approaches cannot reveal the relationship between words and emojis. In order to fill this gap, Barbieri et al. [5] proposed a novel task to predict emojis most associated with given tweet. This model is based on LSTM [6] with both standard lookup word representations and character-based representation of tokens. Wang and Pedersen [7] realized multi-channel CNN network model to predict emoji by improving word embedding method. While, Çöltekin and Rama [8] use linear classification model SVM [9] with n-gram to represent sentence feature and the experiments show that SVM performs better than RNNs at emoji prediction. Effrosynidis et al. [10] employed TF-IDF vector context to train prediction model combined with linear SVC classification algorithm. Chen et al. [11] proposed a method based on vector similarity to generate a vector for tweet, and then used cosine similarity method to find the most appropriate emoji symbols. All the above studies are based on English corpus. Jiang F et al. [12] proposed a sentiment classifier based on mapping blogs to the emoji space of Chinese Weibo. Xie et al. [13] predict the top 10 most frequently used emojis based on a given Weibo dialog, use a hierarchical LSTM network to encode context information in the dialog, and then express the emojis in the conversation based on the dialog they learned. However, there is still not much research on emoji prediction based on Chinese social media, and the effect of the proposed model is not particularly good.

The above research work focuses on exploring the relationship between emojis and words, which depends on extracting text features related to emoji. In Weibo, emoji is labelled by the way of "[]". Such as ❤️ is defined as "[heart]" in the text. The word "heart" is similar to the emotion expressed by "love" in the Internet. Therefore, the definition of emoji and its related words are of great significance for understanding the relationship between words and emojis. The main entry point of this paper is to effectively represent the relationship between emojis and words to improve the performance of the model. Therefore, our model utilize BiLSTM combined with CNN layers to capture information features to learn Weibo representation. Furthermore, attention mechanism [14] is adopted to select important components. Experimental results prove that for CBLA, all components are useful for the final results.

3. Method

3.1. Overall Architecture

Emoji recommendation aims to recommend appropriate emoji based on contextual information. Fig.1 provides an overview of our approach, which includes four main steps: 1) pre-processing the dataset; 2) get word embedding of weibo text; 3) training deep learning model; 4) prediction layer, get the most likely emoji of this weibo text.

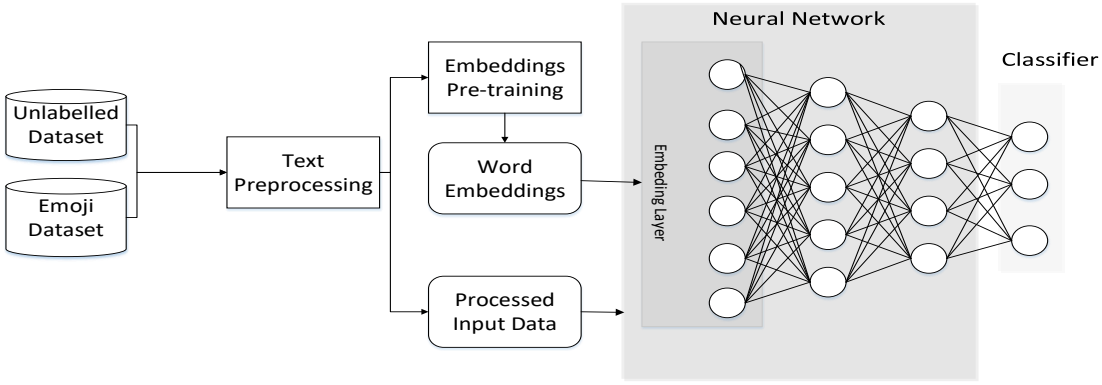

Fig. 1: Overview of our approach.

3.2. Dataset

Since few studies focuses on Chinese-emoji prediction, there is no public dataset for emoji prediction based on Chinese text. We build our dataset based on NLPCC 2014 and 2017 (The CCF International Conference on Natural Language Processing and Chinese Computing) contest public Sina Weibo dataset to ensure the reliability of data. In order to be used for emoji prediction, we applied necessary processing strategies on initial dataset. Here are the pre-processing flow we did on the initial dataset: 1) remove weibo text which doesn't include emojis; 2) remove hyperlinks in each weibo text; 3) remove unrecognized characters in weibo text; 4) split weibo text with emojis, if there are several distinct emojis for one weibo text, duplicate the weibo text and mapping one emoji to one weibo text copy; 5) do word segmentation on weibo text, remove stop words; 6) choose top 20 frequent emojis and its corresponding weibo text; 7) extract the label of each weibo text and set it as the label. The Table. 1 is the format of two datapoints.

Table. 1: The format of dataset.

The initial weibo text	The pre-processed weibo text	Label
谢谢你帮我带来书包！我爱你！ Thank you for bringing my bag back! I love you!	谢谢 带来 书包 我 爱 你	❤️
一辆路过的汽车溅水到了我的裙子上，伤心！ A passing car splashed water on my dress, crying!	一 辆 路过 汽车 溅 水 裙子 伤心	😭

Overall, we have 316,395 weibo text and its corresponding emoji labels. We split this dataset to train set (189,837), validation set (63,279) and test set (63,279). The Table. 2 shows the top 20 weibo emojis we have on our dataset and its corresponding distribution.

Table. 2: The distribution of the 20 most frequent emojis that we use in our experiments.

17.34%	12.37%	9.57%	8.08%	5.17%	4.9%	4.76%	4.75%	4.26%	3.2%
2.9%	2.77%	2.73%	2.65%	2.64%	2.59%	2.58%	2.45%	2.25%	2.04

3.3. Word Embeddings

In order to predict emoji based on weibo text. The first step is find a good way to represent weibo text. In this section, we present embeddings we used to express the weibo text.

Traditional one-hot vector strategy usually be used to represent word embeddings. There are two main problems: loss of word order and oversize. In contrast, the distributed representation of word embeddings is more suitable and more powerful because it can easily use existing word embedding matrices that are already online. In this paper, our method uses the word2vec method proposed by Mikolve [15] for word embedding. The task in the word2vec method uses a skip-gram model. The model is trained by a skip-gram model by maximizing the average log probability of all words. The skip-gram model trains semantic embedding by

predicting the target word based on the context of the target word, and the skip-gram model can also capture the semantic relationship between words. In this project, we chose Chinese word vectors trained on weibo domain. The dimension of each word vector is 300, it only contains words, no n-gram included in this dictionary.

3.4. CNN-BiLSTM with Attention Model

We use a CNN-BiLSTM with attention (CBLA) model in this task. In CBLA model, the convolutional layer extracts n-gram features from the text for sentence modelling. And then BiLSTM accesses both the preceding and succeeding contextual features by combining a forward hidden layer and a backward hidden layer. The attention mechanism for the single word representation pays more attention to the words related to the sentiment of the text and it can help to understand the sentence semantics. The architecture of our model is illustrated in Fig. 2. Next, we will go through each layer in the model in detail from bottom to top.

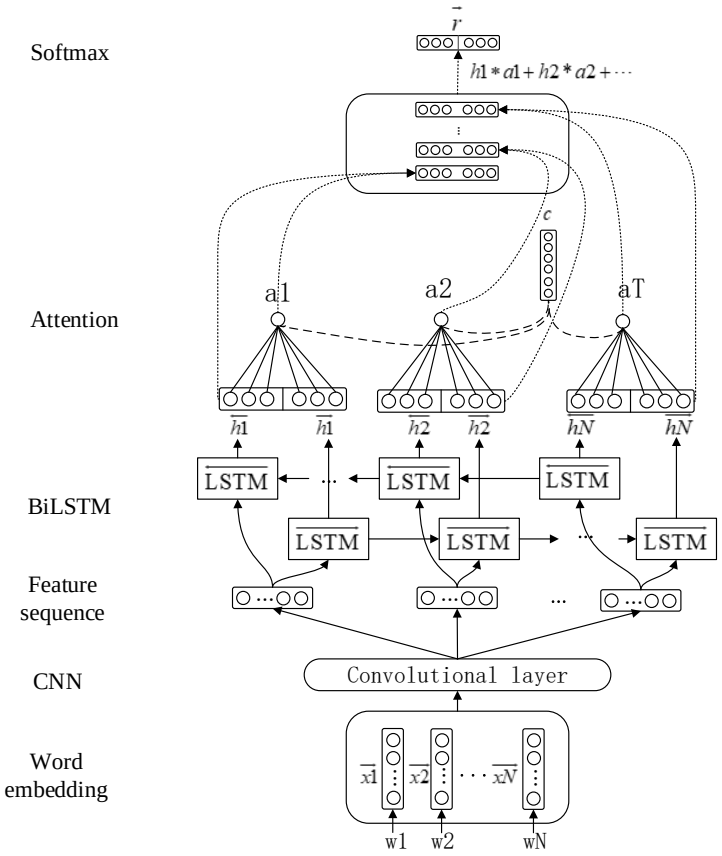

Fig. 2: Architecture of the proposed model.

Embedding Layer. The input of the model is a word sequence of weibo text. This layer is used to project the words w_1, w_2, ···, w_N into a low-dimensional vector space R^W, where the size of W is the size of the embedding layer and N is the Number of words in a text. In this paper, we employ word2vec to embedding words.

CNN Layer. The convolution layer is used to extract n-gram grammatical features from the text and reduce the dimension of the word vector generated by word embedding. In the convolutional layer, we use 100 filters with a window size of 3 to move on the text representation to obtain the feature sequence. For the convolutional layer, the dimension of the input data is 300 ×N , and the dimension of the output data is reduced to 100 ×N. Where N is the number of words in the text. and a word embedding vector $x_{i:i+m-1}$ which represents a window of m words starting from the i-th word, is used to obtain the features for the window of words in the corresponding feature sequences. Multiple filters with differently initialized weights are used to improve learning capability of the model. The n-th feature sequence S_n is generated from the window of the word $x_{i:i+m-1}$ by

$$S_n=RL\left(W_c^T x_{i:i+m-1}+b\right) \quad (1)$$

where b is a bias vector and RL(.) Represents a non-linear activation function a rectified linear unit (ReLU). ReLU is used as the nonlinear activation function because it can improve the learning dynamics of the networks and significantly reduce the number of iterations required for convergence in deep networks.

BiLSTM Layer. BiLSTM accesses both the preceding and succeeding contextual features by combining a forward hidden layer and a backward hidden layer. In previous tasks, LSTM networks are commonly used to extract text context information, because LSTM networks are an improvement on recurrent neural network (RNN) models, which can improve the long-term dependence problem of RNN models. However, the LSTM can only save information in one direction before the sequence, and produces the word annotations h_1, h_2, ···, h_N, where h_i summarize all the information of the sentence up to w_i. The bidirectional LSTM network is an optimization of the LSTM network. A bidirectional LSTM network can summarize sequence information from two directions to effectively extract the full-text context information. In CBLA, the feature sequence output by the CNN layer is the input of the BiLSTM layer. A BiLSTM consists of a forward LSTM $\vec{f}$ which reads the feature sequence from s_1 to s_{100} and a backward LSTM $\overleftarrow{f}$ which reads from s_{100} to s_1. At time step i, Formally, the output of BiLSTM is described as follows:

$$\overrightarrow{h_i}=\vec{f}(S_i),\ i\in[1,100] \tag{2}$$

$$\overleftarrow{h_i}=\overleftarrow{f}(S_i),\ i\in[100,1] \tag{3}$$

a hidden state h_i contains both previous and future context information, $h_i=\overrightarrow{h_i}\ \|\ \overleftarrow{h_i}$, $h_i\in R^{2L}$ where $\|$ represents the concatenation operation, and L represents the size of each LSTM.

Context-aware Self-Attention Layer. Because each word contributes differently to the sentiment of the context, in order to better estimate the importance of each word we use a self-attention mechanism [16] to assign different weights to the words. However, self-attention is to calculate the dependencies between representations without considering contextual information, which has proven to be useful for modelling dependencies between neural representations in various natural language tasks. Therefore, we add a "context vector" weight c to the self-attention mechanism, which is reconstructed into an aggregation of text meaning. The context vector c is taken as the average of h_i:

$$c=\frac{1}{N}\sum_{1}^{N}h_i \tag{4}$$

The context-aware annotations u_i are obtained by the concatenation of c and h_i:

$$u_i=h_i\ \|\ c \tag{5}$$

The u_i is first fed to get e_i by a layer of perceptron as a hidden representation of u_i. The e_i are formulated as follows:

$$e_i=\tanh(Wu_i+b) \tag{6}$$

where W and b are represented as the trainable weights and biases in the attention layer, tanh(.) is hyperbolic tangent function. The attention weight a_i are obtained by performing a softmax calculation on the output e_i of the attention layer. The a_i are formulated as follows:

$$a_i=\frac{\exp(e_i)}{\sum_{t=1}^{N}\exp(e_t)} \tag{7}$$

where exp(.) is exponential function. A weighted sum based on the weights a_i and the annotation h_i is taken as the final contextual representation r. r can be expressed as:

$$r=\sum_{i=1}^{N}a_ih_i,\ r\in R^{2L} \tag{8}$$

Output Layer. In CBLA, we feed the representation r to a fully connected softmax layer with L neurons, which is used to generate conditional probabilities to achieve classification. This layer outputs the probability p on each class. The p can be denoted as follows:

$$p=\frac{e^{Wr+b}}{\sum_{i\in[1,L]}(e^{w_ir+b})} \tag{9}$$

4. Experiment and Evaluation

Experiments were performed to evaluate the performance of proposed method for emoji prediction. In this section, we will introduce the experimental setup and then discuss the results.

4.1. Experimental Setup

Parameter Setting. We use a back-propagation algorithm with Adam's stochastic optimization method to optimize the network, with learning rate of 0.001 and mini batch size of 32. We use PyTorch to develop our model. Publicly available word vectors trained from Weibo text are used as pre-trained word embeddings. The size of these embeddings is 300 and the LSTM layers 300 (600 for BiLSTM). In the one-dimensional convolutional layer, the convolution window size m is set to 3, and the stride size is set to 1 and the number of filters of length 3 is set to be 100. The training batch size for all datasets is set as 256.

Baseline Methods. To further verify the validity of our model, in Table. 3 we benchmarked the proposed CNN-BiLSTM based on Attention (CBLA) model with the following baseline methods. 1) LSTM: Long short-term memory network; 2) CNN: 1d-CNN with pre-trained word embedding vector from word2vec. 3) CNN-LSTM (expressed as CL): using the combination of LSTM and CNN; 4) BiLSTM: Bidirectional Long Short-Term Memory network; 5) CNN-BiLSTM (expressed as CBL): A model combining of CNN and BiLSTM.

Results. The comparison results are presented in Table. 3. Finally, we observe that CBLA significantly outperforms other baselines. The best results are shown in boldface. From Table. 3, the mechanism layer, convolutional layer and BiLSTM have a strong impact on the performance of CBLA. Compared with the LSTM and CNN alone, CL brings relative improvements of 2.29% and 5.24% respectively. This means that the convolutional layer helps improve the classification accuracy of the model. For CBLA, the purpose of the convolutional layer is to pre-process the input text data. Due to the ability to capture local correlations of spatial or temporal structures, the convolutional layer excels in extracting n-gram features at different locations of the text by convolutional filters in the word vector. In addition, the convolutional layer reduces the parameters of the network. The convolutional layer has a significant effect on optimizing our model. Compared to CL, CBL brings a relative improvement of 6.4%. And compared to LSTM, BiLSTM brings a relative improvement of 8.4%. Compared with LSTM, BiLSTM can access the context information forward and backward. Therefore, BiLSTM can learn the context of each word in the text more effectively. Compared with CBL, CBLA brings a relative improvement of 3.06%. It is observed that when the attention mechanism layer is removed from CBLA, the performance of the model is greatly reduced. The attention mechanism is mainly to identify the influence of each word on the sentence. It assigns attention weight to each word and can capture important parts of sentence semantics. Compared with BiLSTM, CBL brings a relative improvement of 0.29%. It means that the convolutional layer has less impact on our method than the other components. But the convolutional layer still helps to improve the classification accuracy. For CBLA, the importance of the mechanism layer or BiLSTM is higher than the importance of convolutional layer. It proves that all components are useful for the final results in CBLA.

Table. 3: Comparison against baselines (Precision, Recall, F1-score).

Model	Precision (%)	Recall (%)	F1-score (%)
LSTM	27.78	22.02	21.65
CNN	24.83	15.37	15.11
CL	30.07	23.57	23.99
BiLSTM	36.18	2715	28.71
CBL	36.47	32.06	34.12
CBLA	**39.53**	**34.79**	**37.02**

4.2. Qualitative Analysis

In Table. 4 we report Precision, Recall, F-measure of each weibo emoji on best model.

Quality of Dataset. In this project we chose 316,395 weibo and its corresponding emoji as training and evaluation dataset. Although the dataset is coming from sina weibo official. The purpose of dataset is not for

emoji prediction. We find that some of the weibo text is related to advertisement, which is not from normal weibo users. This part of data does bring some bias for the model, and it's hard to filter out.

Frequency of Emoji. The frequency of emoji seems very relevant to the prediction. The ranking of most frequent emojis is lower than the ranking of the rare emojis. This means that if an emoji is frequent, it is more likely to be on top of the possible choices even if it is a mistake. In this project we only take the largest probability label as the predict emoji for each weibo text, choosing top-3 or top-5 most likely emojis may will give us a better accuracy. On the other hand, the F-measure does not seem to depend on frequency, as the highest F-measures are scored by a mix of common and uncommon emojis.

Table. 4: Precision, Recall, F1-score of each weibo emoji on CBLA and the percentage of each emoji in test set.

Emoji	Precision(%)	Recall(%)	F1-score(%)	Percentage(%)
😂	25.72	23.05	24.31	16.97
🤔	30.27	22.92	26.08	12.84
😁	28.52	22.38	25.08	9.88
😭	24.36	23.14	23.73	8.03
❤️	24.06	6.27	9.95	5.3
👍	45.31	48.07	46.65	4.91
😤	32.02	23.25	26.94	4.66
🙏	34.05	60.12	43.47	4.94
🤓	39.11	39.73	39.42	4.27
😋	69.66	70.16	69.91	3.4
😎	31.61	32.96	32.27	2.83
🐷	53.29	50.40	51.80	2.51
😈	34.15	39.31	36.55	2.95
😍	28.61	19.02	22.85	2.52
😩	33.78	31.05	32.36	2.64
🤬	29.34	10.57	15.54	2.25
😬	46.49	27.63	34.66	2.5
🔥	38.54	15.61	22.22	2.19
🥰	36.98	17.51	23.77	2.21
🤎	24.64	12.55	16.63	2.2

Logic on Weibo Text with Multi-emojis. In the pre-processing stage, we have some logical processing of weibo text containing multiple emojis instead of simply deleting duplicate emojis. The processing step regards weibo text with multiple emojis is split weibo text with emojis, if there are several distinct emojis for one weibo text, duplicate the weibo text. An example is shown in Fig. 3.

Happy Father's Day! 😂 Love you forever! ❤️ ❤️ ❤️	
Happy Father's Day Love you forever	Label：😂
Happy Father's Day Love you forever	Label：❤️
Happy Father's Day Love you forever	Label：❤️
Happy Father's Day Love you forever	Label：❤️

Fig. 3: The schematic diagram of processing steps for weibo text containing multiple emojis.

By digging into the data, we found that simply remove duplicate emojis may not that reasonable. Sometimes people will use duplicate emoji signals to enhance their feeling. In Fig. 3, the initial weibo text means "Happy Father's Day! Love you forever!". In this weibo, people used ❤️ three times and 😂 once. It makes sense because people want to express his/her great love to someone, so ❤️ appears three times on the

initial weibo text. In this weibo text, ❤️ is more related to the text rather than 😂. If we simply remove duplicate, set the appearance of all emoji as one, we actually lose the strong connection between weibo text and emojis. It could be harmful for the model.

5. Conclusion

In this paper, we introduce the CNN-BiLSTM with attention model (CBLA) for Chinese emoji recommendations. In order to completely catch the semantics of weibo text and improve classification accuracy, we propose an improved LSTM method, CBLA, in which convolutional layer, BiLSTM, and attention mechanisms are used to enhance the semantic understanding of weibo text and catch the most important part to express users' feeling. The experimental results show that each part of our model have a positive impact on the final result. Compared with the five baseline methods, the experimental results verify that our model achieved better result in Chinese emoji prediction.

Our future work will focus on multi-emoji prediction and optimization of network architecture. Future work mainly includes the following parts: 1) using other attention mechanisms to further improve our method; 2) exploring the prediction of multiple emoji in the text; 3) applying our method to practical applications.

6. References

[1] Rivera K, Cooke N J, Bauhs J A. The effects of emotional icons on remote communication[C]//Conference companion on human factors in computing systems. 1996: 99-100.

[2] Yigit O T. Emoticon usage in task-oriented and socio-emotional contexts in online discussion boards[J]. 2005.

[3] Eisner B, Rocktäschel T, Augenstein I, et al. emoji2vec: Learning emoji representations from their description[J]. arXiv preprint arXiv:1609.08359, 2016.

[4] Wijeratne S, Balasuriya L, Sheth A, et al. A semantics-based measure of emoji similarity[C]//Proceedings of the International Conference on Web Intelligence. 2017: 646-653.

[5] Barbieri F, Ballesteros M, Saggion H. Are emojis predictable?[J]. arXiv preprint arXiv:1702.07285, 2017.

[6] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[7] Wang Z, Pedersen T. UMDSub at SemEval-2018 Task 2: multilingual emoji prediction multi-channel convolutional neural network on subword embedding[J]. arXiv preprint arXiv:1805.10274, 2018.

[8] Çöltekin Ç, Rama T. Tübingen-oslo at SemEval-2018 task 2: SVMs perform better than RNNs in emoji prediction[C]//Proceedings of The 12th International Workshop on Semantic Evaluation. 2018: 34-38.

[9] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural processing letters, 1999, 9(3): 293-300.

[10] Effrosynidis D, Peikos G, Symeonidis S, et al. DUTH at SemEval-2018 Task 2: emoji prediction in tweets[C]//Proceedings of The 12th International Workshop on Semantic Evaluation. 2018: 466-469.

[11] Chen J, Yang D, Li X, et al. Peperomia at SemEval-2018 Task 2: vector similarity based approach for emoji prediction[C]//Proceedings of The 12th International Workshop on Semantic Evaluation. 2018: 428-432.

[12] Jiang F, Liu Y Q, Luan H B, et al. Microblog sentiment analysis with emoticon space model[J]. Journal of Computer Science and Technology, 2015, 30(5): 1120-1129.

[13] Xie R, Liu Z, Yan R, et al. Neural emoji recommendation in dialogue systems[J]. arXiv preprint arXiv:1612.04609, 2016.

[14] Luong M T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation[J]. arXiv preprint arXiv:1508.04025, 2015.

[15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[16] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.