Joint Emoji Classification and Embedding Learning

Xiang Li[1,⋆], Rui Yan[2,3], Ming Zhang[1]

[1]School of EECS, Peking University, China
{lixiang.eecs,mzhang_cs}@pku.edu.cn
[2]Institute of Computer Science and Technology, Peking University, China
ruiyan@pku.edu.cn
[3]Beijing Institute of Big Data Research, China

Abstract. Under conversation scenarios, emoji is widely used to express humans' feelings, which greatly enriches the representation of plain text. Plentiful utterances with emoji are produced by humans manually in social media platforms every day, which make emoji great influence on the human life. For the academic community, researchers are always with the help of utterances including emoji as annotated data to work on sentiment analysis, yet lack of adequate attention to emoji itself. The challenges lie in how to discriminate so many different kinds of emoji, especially for those with similar meanings, which make this problem quite different from traditional sentiment analysis. In this paper, in order to gain an insight into emoji, we propose a matching architecture using deep neural networks to jointly learn emoji embeddings and make classification. In particular, we use a convolutional neural network to get the embedding of the utterance and match it with the embedding of the corresponding emoji, to obtain its best classification, and otherwise also train the emoji embeddings. Experiments based on a massive dataset demonstrate the effectiveness of our proposed approach better than traditional softmax methods in terms of p@1, p@5 and MRR evaluation metrics. Then a test of human experience shows the performance could meet the requirement of practice systems.

Keywords: Emoji classification, Embedding learning, Deep learning, Neural networks

1 Introduction

Conversation is one of the most important activities for humans, which could communicate their thought and feelings. For face-to-face conversation, humans use expression to indicate their emotion. Recently with the prosperity of Web 2.0, more and more conversation occurs on web platforms like Facebook and Twitter, or using chat tools, which could make humans communicate with each other overcoming distance. For those scenarios, plain text is used instead of face-to-face talking, and emoji is used as the expression on human's face.

Emoji is a kind of symbols to present one's expression, for instance, 🙂 and 😕, which express happy and sad respectively. Emoji is important in humans' daily interaction, like social networks and conversation, through uttering their feelings, which

⋆ Corresponding author: Ming Zhang (mzhang_cs@pku.edu.cn).

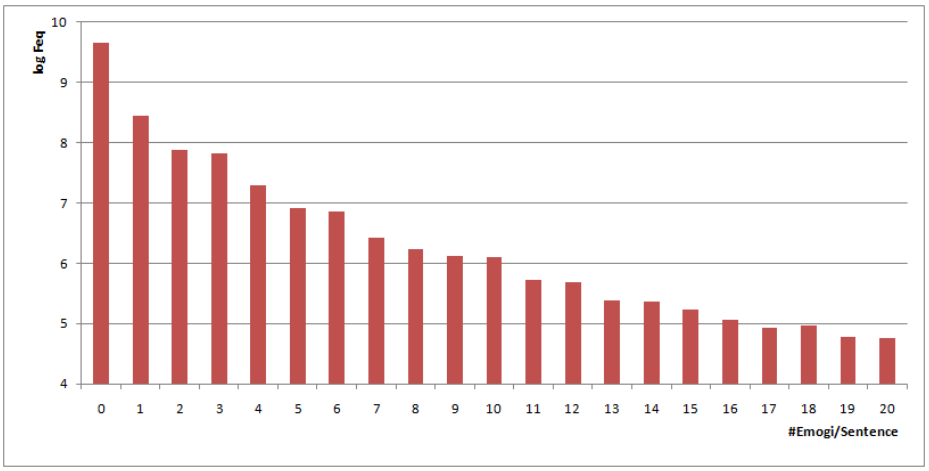

Fig. 1. For utterances manually input by humans, we plot the frequency of utterances including different amount of emoji. The range of emoji amount is from 0 to 20, where 0 means no emoji in the utterance, and the frequency has been in the logarithm. Generally speaking, 9.18% utterances include at least one kind of emoji, and some proportion of them consist of different kinds, which indicates that people often use emoji in their everyday life.

could make the expression more interesting and lively. Thus, platforms like Facebook and Twitter are adding increasing numbers of emoji sets to improve user experience, and Unicode, as an international encoding standard, extends its emoji set continuously, including more than 7000 kinds in its recent 8.0 version.

The importance of emoji is increasingly realized by the academic community of artificial intelligence. For the task of sentiment analysis, researchers often use emoji as a kind of distant supervision [4]. Since large amount of annotation data is needed, they regard emoji as the ground truth, which means that, some emoji expressing happy feelings such as 🙂 indicates a positive polarity, and sad expressions like 😟 indicates a passive polarity. However, emoji is just used as an assistant way, and research work for emoji itself is really lack.

Researches for emoji itself is very significative, and could support many applications. Besides as a kind of human annotation for sentiment analysis, it is convenient for humans to know when and what emoji he may use during inputting the utterance, if there is a function of recommending emoji for an utterance on the social media platform. Moreover, in automatic human-computer conversation systems, it could make the computer side more proactive through adding emoji behind the reply, just like a human expressing his feeling, considering that in human-human conversation on social network platforms, emoji is often used as a part of humans' utterances to indicate their current emotion.

In practice utterances generated by humans, the use of emoji is widely. The phenomenon is supported by the statistics of utterances collected from an online forum[1],

[1] http://www.weibo.com

which is shown in Figure 1, consisting of over 5 billion items of data manually input by humans. Our observation is that, people often use emoji in their everyday life, in order to make the communication more vivid and make it easier to deliver their feelings. Therefore, many applications like automatic human-computer conversation systems, which need to make the computer side be more like a human, especially for commercial products, should add the function of automatic emoji expression in order to attract more people.

Emoji is quite different to traditional sentiment or emotion analysis, which only need to discriminate polarities or several pre-defined kinds of emotion, since there are many kinds of emoji, and meanings of some emoji are so similar that it is difficult to distinguish them well for traditional classification methods using softmax. For example, 😂 and 🙂, which means laugh and smile respectively, both express emotion of happiness, with only a little difference on degree. Therefore, we propose a joint architecture to learn emoji embeddings and classification through matching them in multi-modal vector space. To be specific, based on the layer of word embeddings, we get the embedding of the whole utterance by a convolutional neural network (CNN), and then a HingeLoss function is used to match it with the emoji embeddings, which should be also trained. Comparing to the traditional softmax function, our approach could better distinguish emoji with similar meanings.

To sum up, the main contributions of this paper are as follows.

- We conduct scientific experiments to analyze the problem of emoji classification and embedding learning in conversation scenarios[2].
- We propose a matching approach using deep neural networks by utilizing emoji embeddings and observe that the performance of emoji classification is better than traditional softmax methods.
- Empirical experiments demonstrate the effectiveness of both our embedding learning and emoji classification, and the analysis shows a good human experience in practice.

2 Related Work

Traditional sentiment or emotion analysis is a significant research task which has attracted many researchers in the domain of natural language processing. Research work on sentiment analysis often focuses on classifying the polarities of positive and negative [5, 6], or extends to the third polarity of neutral [7, 8], or sometimes adds fine-grained classes like a spectrum such as very positive and very negative [9–11]. Pre-defined kinds of emotion are also involved into some work on sentiment analysis, such as happy, sad, and so on [12–14], while sometimes the emotion classification could be multi-label [45].

With the development of natural language processing, many theories and technologies have been used to deal with traditional sentiment analysis. Lexicon-based models

[2] We notice a piece of parallel work [2], which is an application named Dango on Android platform, and also suggest emoji for conversation between humans. However, we are the first to conduct scientific experiments, showing the effectiveness of matching.

using sentiment dictionaries are an effective series of approaches to deal with sentiment analysis [15, 16], since some words have clear trends of sentiment polarities. Feature-based models using traditional classifiers are another kind of methods with high performance, which is called distant supervision by leverage utterances with emoticons as annotated data [3, 17]. Other theories like statistical machine translation [18], graph-based approach [19] and topic model [20] are also used to analyze sentiment.

Recent years, with the development of word embeddings and neural networks, research work appears continuously using these technologies to improve the performance of sentiment analysis. Since word embeddings could well represent its semantic features and also latent information [35, 36], it is natural to add sentiment-specific information into the word embeddings while training by neural networks [8, 21, 23]. Another series of approaches is to propose novel structures of neural networks [1, 22, 24, 46], which means adapting the theory of deep learning to sentiment analysis. Furthermore, under some specific scenarios, especially on social networks, context of human interaction are considered to improve sentiment analysis [25–27].

Besides using emoji as a kind of distant supervision, emoji or emoticons themselves are also related to sentiment expression. Emoticons could indicate sentiment polarities in plain-text computer-mediated communication [44] and a sentiment map for several hundred kinds of most frequently used emoji is established [37], both in order to improve the performance of sentiment analysis.

Although there has been research work which proposes a multi-modal approach to generate emoji labels for an image [28], it is still lack of effort to match emoji with plain text. Thus, we adjust the problem of emoji classification and embedding learning, which is more complicated than traditional sentiment or emotion analysis, and then propose a match approach to obtain better performance than softmax classifiers.

3 Approach

3.1 Task Definition

Given an utterance set $Y = \{y_1, y_2, ..., y_n\}$ and a emoji set $X = \{x_1, x_2, ..., x_k\}$, our aim is to train a classification model which could predict the correct emoji $g(y) \in X$ for an utterance y, meanwhile get a vector set $E = \{e_1, e_2, ..., e_k\}$ after training, and each vector e_i is the embedding of emoji x_i in X, as a distributed representation indicating its latent semantic information.

3.2 Structure Overview

Our proposed approach is shown in Figure 2, which is a matching structure based on neural networks, and consists of two parts. The left component is a sub convolutional neural network to get a sentence embedding which could represent the utterance, while the right one is the embeddings of emoji that should also be trained, and finally joint the two parts through a matching score. Our intuition to use a matching structure is that the embeddings in continuous vector space could well represent emoji, and perform better than discrete softmax classifiers, since meanings of some emoji are amphibolous and difficult to distinguish.

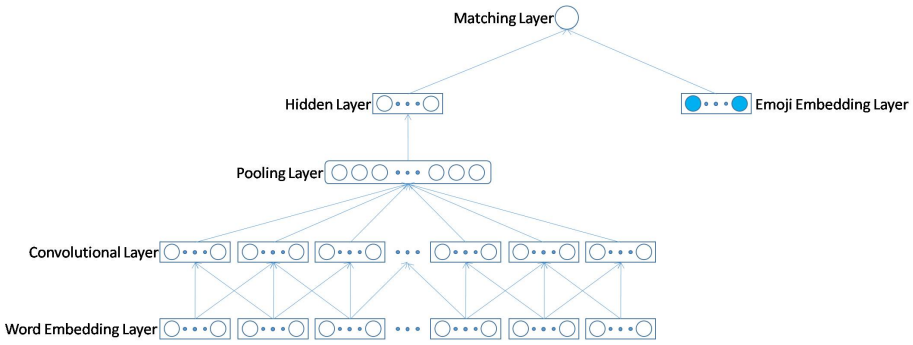

Fig. 2. The whole structure of our matching approach based on neural networks.

3.3 Layers

Word Embedding Layer This word embedding layer is at the bottom of the left CNN part, which aims to get the distributed representation of each word. The embedding of a word is a vector, and each element in the vector is a real number to represent one dimension feature of the word. Thus, the embedding vector could be regarded as a feature set of the word in a low-dimension space, and could indicate latent information of the word both semantically and syntactically, with wonderful performance [36] for some tasks in the domain of natural language processing. Instead of manually designed as feature engineering methods, embeddings of words are often trained by neural networks or calculated by matrix decomposition.

For a plain text utterance, it could be regarded as a sequence of words, and for each word w_i, we first represent it as a one-hot vector of dictionary dimension, with one 1 on the corresponding word bit and other bits 0. Then using an embedding matrix $\mathbf{E}_1 \in R^{D \times V}$, where D is the dimension of word embeddings and V is the dimension of the word dictionary, we could obtain the embedding of the word $\mathbf{e}_1(w_i)$. Thus, we get each word embedding after this layer, and the matrix $\mathbf{E}_1$ consists of all the embeddings $\mathbf{e}_1(w_i)$ for words in the dictionary, which is randomly initialized and trained during the training process. In practice, we have over 30 thousand words in the dictionary and choose the parameter D equal to 128.

Convolutional Layer The convolutional structure of neural networks is believed suitable to synthesize lexicon n-gram information of a sequence, especially for short text [29]. Different from full connection layers, CNN uses the concept of sliding windows, which is like a local feature extractor, to get information from word embeddings. If the window size is t, and the corresponding words embeddings are $\mathbf{e}_1(w_1), \mathbf{e}_1(w_2), \cdots, \mathbf{e}_1(w_t)$, then we have:

$$\mathbf{y}_1 = f(\mathbf{W}_1[\mathbf{e}_1(w_1); \mathbf{e}_1(w_2); \cdots; \mathbf{e}_1(w_t)] + \mathbf{b}_1) \tag{1}$$

where $\mathbf{b}_1$ is the bias vector, f is the non-linear activation function, and $\mathbf{W}_1$ is the parameter matrix which needs trained. We choose a window size of 3 in practice.

Pooling Layer After the convolutional layer, we get a series continuous representations of local features. Since we need to synthesize these local embeddings into one vector as the distributed representation of the whole utterance, we use the theory of dynamic pooling [31]. To be specific, let $\mathbf{y}_1^1, \mathbf{y}_1^2, \cdots, \mathbf{y}_1^l$ be the output vectors from the convolutional layer, and then we have:

$$\mathbf{y}_2[i] = max\{\mathbf{y}_1^1[i], \mathbf{y}_1^2[i], \cdots, \mathbf{y}_1^l[i]\} \tag{2}$$

which means we use max pooling actually, and then obtain $\mathbf{y}_2$ as the sentence embedding of the utterance, which is also 128-dimension in practice. Obviously, a sentence embedding is also a vector to represent the utterance in low-dimension space, which could indicate latent information, and generally be used in the domain of natural language processing, just like word embeddings.

Hidden Layer Then we use a hidden layer of full connection to exert a non-linear transformation to the sentence embedding, which could be calculated by:

$$\mathbf{y}_3 = f(\mathbf{W}_2\mathbf{y}_2 + \mathbf{b}_2) \tag{3}$$

where $\mathbf{b}_2$ is the bias vector, f is the non-linear activation function, and finally we get another 128-dimension vector to represent the utterance.

Emoji Embedding Layer Since we have dealt with the plain text side through the CNN which is shown as the left part of our proposed matching structure, next we need to embed the emoji, aiming to learn continuous representations of emoji in vector space, just like word embeddings. Thus, in a similar way like the word embedding layer, we could also represent each emoji x_i as a one-hot vector of K-dimension, where K is equal to the amount of emoji, and then use a matrix $\mathbf{E}_2 \in R^{D\times K}$ to obtain its embedding $\mathbf{e}_2(x_i)$. The matrix $\mathbf{E}_2$ includes all the emoji embeddings, and each element is one parameter of the neural network, which is randomly initialized and trained during the training process.

Matching Layer In this layer, we plan to match the embedding of the plain text $\mathbf{y}_3$ from the non-linear hidden layer with the emoji embedding $\mathbf{e}_2(x_i)$, and then obtain a score as the final result which could indicate their matching degree. We choose the cosine similarity as the measurement of matching and then have:

$$score(\mathbf{y}_3, \mathbf{e}_2(x_i)) = \frac{< \mathbf{y}_3, \mathbf{e}_2(x_i) >}{||\mathbf{y}_3|| \cdot ||\mathbf{e}_2(x_i)||} \tag{4}$$

where $< \cdot, \cdot >$ means inner product of two vectors and $|| \cdot ||$ means the length of a vector. Since a higher score indicates more matching for the utterance embedding and the emoji embedding, we choose $argmax_{x_i}\ score(\mathbf{y}_3, \mathbf{e}_2(x_i))$ as the final emoji which should be added into the plain text.

Algorithm 1: Process of Training One Sample

Input: One utterance, its correct emoji x_t, and the whole emoji set X
Output: Updating model parameters
Description:
foreach *Negative emoji* x_j **do**
  Forward propagation to calculate the matching score between the given utterance and the correct emoji x_t
  Forward propagation to calculate the matching score between the given utterance and the negative emoji x_j
  Calculate the error between the two scores using the HingeLoss function
  Backward propagation to update model parameters

3.4 Training

In the training process, we use the HingeLoss function, which is convex with wonderful properties, through punishing negative matching to optimize parameters. Different from the softmax function, the HingeLoss function exerts pairwise comparisons between the positive matching and each negative one, in order to distinguish them, especially for similar kinds of emoji. For each plain text $\mathbf{y}^i$ in the training set, let $\mathbf{y}_3^i$ denote the sentence embedding of $\mathbf{y}^i$ from the non-linear hidden layer, and our optimal objective is

$$Obj(\mathbf{y}^i) = min \sum_{j \neq t} max(0, \alpha + score(\mathbf{y}_3^i, \mathbf{e}_2(x_j)) - score(\mathbf{y}_3^i, \mathbf{e}_2(x_t))) \quad (5)$$

where x_t is the correct emoji for training samples $\mathbf{y}^i$ according to the ground truth, and α is the margin of the HingeLoss function. The process of training one sample is depicted in Algorithm 1.

Before the training process, we calculate the frequency of each emoji, and select top 20, 50 and 100 respectively, as three emoji sets. We use the theory of stochastic gradient descent to train our proposed neural networks, and adjust the learning rate on the set consisting of 20 kinds of emoji. Finally we get a learning rate of 10^{-6} and adapt it to the other two emoji sets.

4 Experiments

4.1 Datasets

We collect massive conversation resources of human interaction from microblog websites including Sina Weibo[3], which contains over 170 thousand kinds of emoji in all. Then we extract tens of millions of utterances with top 20, 50 and 100 most frequent kinds of emoji respectively, as our three datasets. The datasets are randomly divided into training sets, validation sets and test sets, and the details are summarized in Table 1.

[3] `http://www.weibo.com`

Table 1. Statistics of the Datasets

#Emoji	#Training	#Validation	#Test
20classes	11.6M	50K	56K
50classes	14.4M	50K	75K
100classes	15.9M	50K	83K

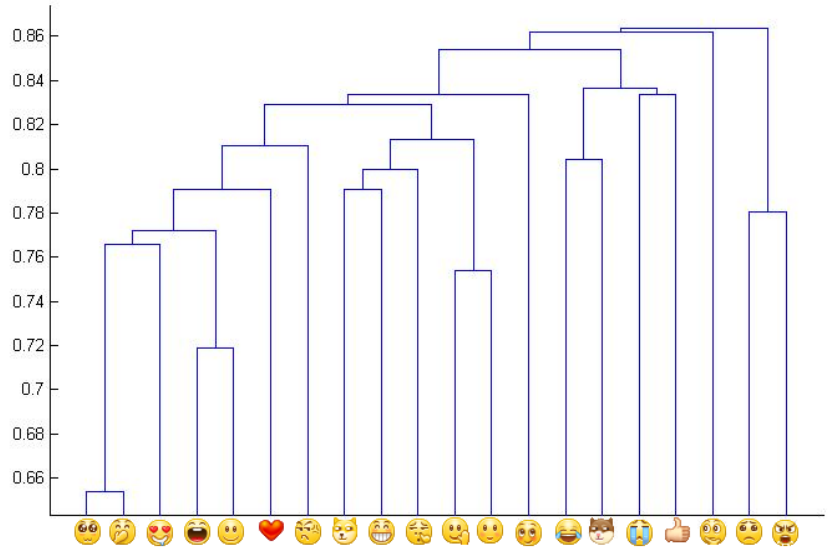

Fig. 3. The result of the hierarchical clustering of top 20 kinds of emoji.

To be specific, each item of data is a pair of $\langle utterance, emoji \rangle$, where emoji comes from the utterance itself and is regarded as the ground truth produced by humans. Since we only need to put one utterance into one class, any expression including more than one kind of emoji is filtered, aiming to produce a clear training set. Moreover, different people may give different emoji to the same plain text, so in order to avoid this kind of confusion, we also filter any expression having different emoji for all its appearance. Besides, since most expression has no emoji, we should not give emoji to all utterances, so finally, we filter the utterances with a max-majority of non-emoji for all its appearance.

4.2 Qualitative Analysis of Embeddings

Since we regard the emoji embedding layer as a continuous representation of emoji, we firstly have a qualitative analysis of these embeddings. As mentioned before, we aim to get a meaningful set of emoji embeddings, so similar kinds of emoji should have shorter distances than others. Thus, a hierarchical clustering is applied on the embeddings of the most commonly used 20 kinds of emoji (Figure 3), with a cosine similarity as the metric of distance and using nearest neighbourhood. Our observation is that, the

embeddings could represent semantic information of the emoji in some degree, due to some phenomenon like "laugh" close to "smile" and the six kinds of emoji on the left denoting the positive polarity while the two on the right representing the negative polarity. Yet there are also confusing aspects like unclear clustering especially for clusters with low similarity.

4.3 Quantitative Analysis of Emoji Classification

In this section, we have a quantitative analysis of emoji classification directly, with lots of other experimental analysis such as parameter sensitivity and case studies.

Baseline Algorithms We include the following methods as baselines to compare with our proposed approach. Since our approach is a matching architecture based on neural networks, besides the method based on textural similarity, we mainly use some basic neural network structures to make the comparison, with a traditional softmax function as the objective. For fairness, we conduct the same data cleaning and layer dimensions in neural networks for all algorithms. Besides, we also adjust the learning rate for baseline neural networks just the same as the process of our proposed approach, and finally get a learning rate of 10^{-3}.

Textural Similarity. This method ranks candidates according to textual similarity, which is a basic way to calculate relevance between queries and documents in the domain of information retrieve (IR). Here we regards each emoji as a document consisting of all utterances including the particular emoji, while the utterance as a query, and each word is weighed by tf-idf.

CBOW. Since an utterance consists of words, bag-of-words is a natural way to model the utterance and widely used [36]. The basic thought is to regard the utterance as a set of words, so from the word embedding layer, a summation or average operation is simply done to get the embedding of the whole utterance. The bag-of-words method could get the utterance representation quickly and concisely, however lose the sequentiality of natural languages.

CNN. Convolutional neural networks is a kind of structure that could extract local features, and is believed to get better performance on images [38] or short text [29, 30]. Instead of the way of full-connection, a convolutional layer has a sliding window which means that a neuron could only have particular numbers of connections from the last layer. After the convolutional operation, there is often a pooling layer to integrate the information, and here we also use max pooling for fairness.

RNN. Due to the sequentiality of natural languages, recurrent neural networks and its variants are also widely used to represent an utterance, especially for the generation process [39, 40, 42, 43]. For each hidden layer, the inputs are the current word embedding as well as the last hidden layer, until the end of the utterance, and the final hidden layer is regarded as the embedding of the whole utterance, which could represent all the sequential information. In practice, due to the increasing sparsity with the propagation going on, the Long Short-term Memory (LSTM) [41] or the Gated Recurrent Unit (GRU) [32] is often used to improve its performance. Here we use the GRU version.

Table 2. Performance of the Emoji Classification

#Emoji	20classes			50classes			100classes		
Metrics	p@1	p@5	MRR	p@1	p@5	MRR	p@1	p@5	MRR
Textural Similarity	15.61%	57.69%	32.38%	12.37%	47.91%	24.83%	11.60%	43.85%	20.93%
CBOW	22.22%	60.03%	39.74%	18.10%	49.62%	33.24%	16.41%	44.43%	30.12%
CNN	22.42%	60.31%	39.96%	18.62%	50.06%	33.67%	16.74%	44.78%	30.50%
RNN	22.21%	59.64%	39.61%	17.90%	48.96%	32.86%	16.23%	43.78%	29.75%
Matching	**24.30%**	**63.01%**	**41.39%**	**20.16%**	**51.29%**	**34.74%**	**18.69%**	**46.74%**	**31.94%**

Evaluation Metrics We first evaluate the performance using p@1 metric, which could reflect the accuracy of algorithm results, and is believed to be the most direct judgment for classification tasks. Besides, for most applications possibly developed based on emoji classification, p@1 is also appropriate since the utterance should match at least one emoji when we want to add emoji behind the plain text.

Next, since the results we returned based on our approach or baseline algorithm are matching scores between the sentence and emoji, or distributions on emoji, they could be regarded as ranking lists of emoji given the plain text. So we could also evaluate the performance in terms of p@k, and here we choose k equal to 5.

Another evaluation metric is the Mean Reciprocal Rank (MRR)[4], which is also able to evaluate a ranking list:

$$MRR = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{rank_i} \tag{6}$$

Here $rank_i$ refers to the rank position of the correct emoji according to the ground truth for the i-th plain text, and T is the set of plain text.

Performance In this section, we show the performance of the proposed matching approach against other baselines. The results are summarized in Table 2, in which we report the performance of emoji classification in all the mentioned evaluation metrics. It is obvious that methods based on neural networks perform better than textual similarity, and our matching approach is the best of all. Although a softmax objective function could make a good classification, the matching structure is more appropriate to distinguish little difference between emoji especially for those kinds with similar or confusing meanings. The improvement is small yet consistent, which is similar to the conclusion of a prior work [33], and our observation is that in some degree, our matching approach seems similar to use a cross-entropy objective function instead of the traditional one-hot vector in the softmax structure.

Analysis In this section, we have some analysis on emoji classification with grouping similar kinds, parameter sensitivity of our matching approach, and human experience for possible applications.

[4] `https://en.wikipedia.org/wiki/Mean_reciprocal_rank`

Table 3. Analysis of the Grouped Emoji Classification

#Emoji	20classes			Grouped20classes		
Metrics	p@1	p@5	MRR	p@1	p@5	MRR
Textural Similarity	15.61%	57.69%	32.38%	19.89%	67.74%	38.50%
CBOW	22.22%	60.03%	39.74%	25.20%	70.47%	44.85%
CNN	22.42%	60.31%	39.96%	25.45%	70.91%	45.09%
RNN	22.21%	59.64%	39.61%	25.09%	70.41%	44.70%
Matching	24.30%	63.01%	41.39%	27.62%	72.77%	46.48%

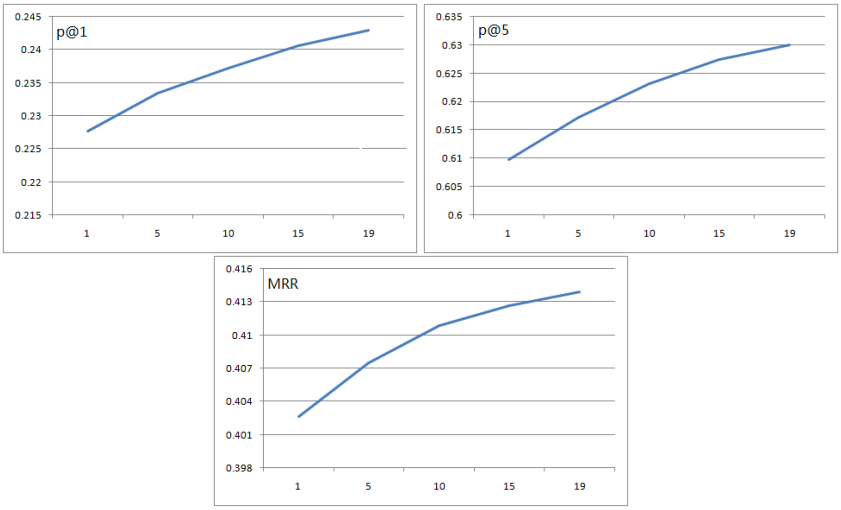

Fig. 4. The illustration of parameter influence to the results of 20 classification. The top left chart presents results of p@1 with the increasing of the amount of negative samples, the top right one is for p@5, and the bottom one is for MRR.

Grouped Classification. Since some kinds of emoji are similar and could express the same meaning, they could be merged together as one class. In our experiment, we group emoji in the data set of 100 kinds into 20 classes manually according to prior knowledge, and then use our matching approach as well as all the baselines to make the classification. The results comparing to the original classification of 20 classes are summarized in Table 3, which shows that, all the methods perform better than the original result, with a small increasing on p@1 yet a large improvement on p@5. The main reason of these phenomenon is that after grouping, the meaning of each class is clearer so that they could be distinguish better, especially for the situation of 5 candidates.

Parameter Sensitivity. For our matching approach, we regard the correct emoji according to the ground truth as positive samples while others as negative ones. So if the task is k classification, then the amount of negative samples is k-1. In the situation of 20 classes, we investigate the influence to the performance with changing the amount of negative samples. To be specific, besides the whole negative set of 19 samples, we randomly sample 1, 5, 10, 15 negative samples using uniform distribution. Figure 4 shows

that on all the three metrics, with the negative samples increasing, the performance become better, and the rising trend becomes slower.

Human Experience. The potential of being adapted into industrial applications could obviously demonstrate the meaning and importance of research work. Yet it seems that the performance of emoji classification is far away from practice applications, because the results of the automatic testing listed before generally do not meet the high accuracy required by commercial products, such as an automatic human-computer conversation system. However, The experience of users is very different from the automatic testing, since some kinds of emoji are similar. Even if we put an utterance into a wrong class, it may not hurt the user experience, which means that in practice applications, more than one kind of emoji could be appropriate for a given utterance.

Therefore, to investigate the performance in user experience, we use human annotation with **1 Point** meaning appropriate or **0 Point** meaning inappropriate. One utterance and the emoji given by our approach are annotated by 3 individuals in an independent and blind fashion. We regard the majority voting as the "ground truth" indicating whether the emoji is appropriate for the utterance from the vision of users. We also evaluate the kappa score: $\kappa = 0.413$, showing moderate inner-annotator agreement [34].

The test data for human annotation consists of 694 items randomly selected from the 20 classification. The performance of our matching approach shows accuracy of 73.05% for the annotation dataset, which obviously is much higher than it of the automatic testing. Moreover, we also investigate the amount of classes with accuracy lying in different ranges, and find out that there are 10 classes having accuracy over 90%. Therefore, the performance of emoji classification is good for human experience and it has large potential for commercial products.

Table 4. Five examples of Emoji Classification

笑起来最舒服(Laughing is the most comfortable)
好怕没有结果(I'm afraid of no results)
我怎么忍心拒绝你(How could I have the heart to reject you)
我先回答哪个问题好呢(Which question should I answer first)
你怎么不给老子过生日(Why don't you celebrate my birthday)

Case Study We illustrate five examples with different kinds of emoji in Table 4 obtained by our matching approach. As seen, our method could give an utterance the appropriate emoji, whatever for the positive polarity as the first case (a laughing face), the passive polarity as the second (a crying face) and fifth (an angry face) cases, or just neutral without clear emotional tendency as the third (a shy face) and fourth (a thinking face) cases. Besides, we have the ability to express much more emotion other than polarities. With emoji, the form of expression become more liberal and indeed more vivid when communicating with others.

5 Conclusion

In this paper, for the problem of emoji classification and embedding learning in conversation scenarios, we propose a matching approach and deeply analyze its performance through both qualitative and quantitative experiments. Empirical results demonstrate our approach better than traditional softmax classifiers in terms of different metrics, and the embeddings trained from our neural networks could also represent the emoji well. For the future work, one direction is to consider contextual information in the conversation process and propose more progressive models to analyze emoji better, another one is to explore the relation between emoji and sentiments in order to improve the performance of sentiment analysis.

Acknowledgements. This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Nos. 61472006 and 91646202) as well as the National Basic Research Program (973 Program No. 2014CB340405).

References

1. Tang, D., Wei, F., Qin, B., Zhou, M., Liu, T.: Building large-scale twitter-specific sentiment lexicon: a representation learning approach. In: Proceedings of the 25th International Conference on Computational Linguistics, pp. 172–182 (2014)
2. Snelgrove, X.: http://getdango.com/emoji-and-deep-learning.html
3. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: International Conference on Language Resources and Evaluation, pp. 1320–1326 (2010)
4. Yan, J. L. S., Turtle, H. R.: Exploring fine-grained emotion detection in tweets. In: Proceedings of NAACL-HLT, pp. 73–80 (2016)
5. Wu, F., Song, Y., Huang, Y.: Microblog sentiment classification with contextual knowledge regularization. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2332–2338 (2015)
6. Deng, L., Wiebe, J., Choi, Y.: Joint inference and disambiguation of implicit sentiments via implicature constraints. In: Proceedings of the 25th International Conference on Computational Linguistics, pp. 79–88 (2014)
7. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354 (2005)
8. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 1555–1565 (2014)
9. Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
10. dos Santos, C. N., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of the 25th International Conference on Computational Linguistics, pp. 69–78 (2014)
11. Mou, L., Peng, H., Li, G., Xu, Y., Zhang, L., Jin, Z. Discriminative neural sentence modeling by tree-based convolution. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 2315–2325 (2015)

12. Beck, D., Cohn, T., Specia, L.: Joint emotion analysis via multi-task Gaussian processes. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 1798–1803 (2014)
13. Wang, Z., Lee, S. Y. M., Li, S., Zhou, G.: Emotion detection in code-switching texts via bilingual and sentimental information. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 763–768 (2015)
14. Chang, Y. C., Chen, C. C., Hsieh, Y. L., Chen, C. C., Hsu, W. L.: Linguistic template extraction for recognizing reader-emotion and emotional resonance writing assistance. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 775–780 (2015)
15. Turney, P. D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 417–424 (2002)
16. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational Linguistics, 37(2):267–307 (2011)
17. Kouloumpis, E., Wilson, T., Moore, J. D.: Twitter sentiment analysis: the good the bad and the omg!. In: Proceedings of the 15th International AAAI Conference on Web and Social Media, pp. 538–541 (2011)
18. Lambert, P.: Aspect-level cross-lingual sentiment classification with constrained SMT. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 781–787 (2015)
19. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1031–1040 (2011)
20. Yang, M., Peng, B., Chen, Z., Zhu, D., Chow, K. P.: A topic model for building fine-grained domain-specific emotion lexicon. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 421–426 (2014)
21. Zhou, H., Chen, L., Shi, F., Huang, D.: Learning bilingual sentiment word embeddings for cross-language sentiment classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 430–440 (2015)
22. Dong, L., Wei, F., Zhou, M., Xu, K.: Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 1537–1543 (2014)
23. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 3038–3044 (2016)
24. Zhang, M., Zhang, Y., Vo, D. T.: Gated neural networks for targeted sentiment analysis. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 3087–3093 (2016)
25. Vanzo, A., Croce, D., Basili, R.: A context-based model for sentiment analysis in twitter. In: Proceedings of the 25th International Conference on Computational Linguistics, pp. 2345–2354 (2014)
26. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Context-sensitive twitter sentiment classification using neural network. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 215–221 (2016)
27. Li, S., Huang, L., Wang, R., Zhou, G.: Sentence-level emotion classification with label and context dependence. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 1045–1053 (2015)
28. Cappallo, S., Mensink, T., Snoek, C. G.: Image2emoji: Zero-shot emoji prediction for visual media. In: Proceedings of the 23rd ACM international conference on Multimedia, pp. 1311–1314 (2015)

29. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Annual Conference on Neural Information Processing Systems, pp. 2042–2050 (2014)
30. Yan, R., Song, Y., Wu, H.: Learning to Respond with Deep Neural Networks for Retrieval based Human-Computer Conversation System. In: Proceedings of SIGIR, pp. 55–64 (2016)
31. Socher, R., Huang, E. H., Pennin, J., Ng, A. Y., Manning, C. D.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Annual Conference on Neural Information Processing Systems, pp. 801–809 (2011)
32. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 1724–1734 (2014)
33. Tang, Y.: Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239 (2013)
34. Fleiss, J. L.: Measuring nominal scale agreement among many raters. Psychological Bulletin, 76(5):378 (1971)
35. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of Machine Learning Research, 3(Feb): 1137-1155 (2003)
36. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
37. Novak, P. K., Smailović, J., Sluban, B., Mozetič, I.: Sentiment of emojis. PloS One, 10(12): e0144296 (2015)
38. Babu, G. S., Zhao, P., Li, X. L.: Deep convolutional neural network based regression approach for estimation of remaining useful life. In: Proceedings of the 21st International Conference on Database Systems for Advanced Applications, pp. 214–228 (2016)
39. Sutskever, I., Vinyals, O., Le, Q. V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
40. Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, pp. 1577–1586 (2015)
41. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation, 9(8): 1735-1780 (1997)
42. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J. Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. In: Proceedings of NAACL-HLT, pp. 196–205 (2015)
43. Mou, L., Song, Y., Yan, R., Li, G., Zhang, L., Jin, Z.: Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In: Proceedings of the 26th International Conference on Computational Linguistics, pp. 3349–3358 (2016)
44. Hogenboom, A., Bal, D., Frasincar, F., Bal, M., de Jong, F., Kaymak, U.: Exploiting emoticons in polarity classification of text. Journal of Web Engineering, volume 14, pp. 22-40 (2015)
45. Wang, Y., Feng, S., Wang, D., Yu, G., Zhang, Y.: Multi-label chinese microblog emotion classification via convolutional neural network. In: The 18th Asia Pacific Web Conference, Part I, LNCS 9931, pp. 567-580 (2016)
46. Zhao, Z., Liu, T., Hou, X., Li, B., Du, X.: Distributed text representation with weighting scheme guidance for sentiment analysis. In: The 18th Asia Pacific Web Conference, Part I, LNCS 9931, pp. 41-52 (2016)