

Multi-label Chinese Microblog Emotion Classification via Convolutional Neural Network

Yaqi Wang¹, Shi Feng^{1,2(✉)}, Daling Wang^{1,2}, Ge Yu^{1,2},
and Yifei Zhang^{1,2}

¹ School of Computer Science and Engineering, Northeastern University,
Shenyang, China

wyq20100695@163.com, {fengshi, wangdaling, yuge,
zhangyifei}@cse.neu.edu.cn

² Key Laboratory of Medical Image Computing, Ministry of Education,
Northeastern University, Shenyang 110819, China

Abstract. Recently, analyzing people's sentiments in microblogs has attracted more and more attentions from both academic and industrial communities. The traditional methods usually treat the sentiment analysis as a kind of single-label supervised learning problem that classifies the microblog according to sentiment orientation or single-labeled emotion. However, in fact multiple fine-grained emotions may be coexisting in just one tweet or even one sentence of the microblog. In this paper, we regard the emotion detection in microblogs as a multi-label classification problem. We leverage the skip-gram language model to learn distributed word representations as input features, and utilize a Convolutional Neural Network (CNN) based method to solve multi-label emotion classification problem in the Chinese microblog sentences without any manually designed features. Extensive experiments are conducted on two public short text datasets. The experimental results demonstrate that the proposed method outperforms strong baselines by a large margin and achieves excellent performance in terms of multi-label classification metrics.

Keywords: Sentiment analysis · Emotion classification · Multi-label classification · Convolutional Neural Network

1 Introduction

With the fast development of social media, more and more people are willing to express their feelings and emotions via microblogging platform. The microblog has aggregated huge number of tweets that containing people's opinions about personal lives, celebrities, social events and so on. Therefore, analyzing people's sentiments in microblogs has attracted more and more attentions from both academic and industrial communities.

The traditional sentiment analysis problems usually classify the microblog into positive and negative orientations or identify people's fine-grained emotions in microblogs, such as *happiness*, *sadness*, *like*, *anger*, *disgust*, *fear*, *surprise* and so on.

Table 1. The examples of coexisting multiple emotions in microblogs

Microblogs	Emotions
The movie tonight is fantastic, but the dinner sucks!	<i>happiness, like, disgust, anger</i>
The ending of this novel is so miserable and terrible, and I think I won't read it anymore.	<i>disgust, fear</i>
What an unbelievable farewell performance by Kobe Bryant! It's a pity that I can't see him in NBA anymore!	<i>surprise, like, sadness</i>

In these methods, each microblog is associated with a single sentiment orientation or emotion label. This is commonly referred as a single-label classification problem. However, in fact multiple emotions may be coexisting in just one tweet or even one sentence of the microblog, as shown in the following examples of Table 1.

Different from the traditional single-label sentiment classification problem, there are critical new challenges for detecting the emotions in microblogs. Firstly, the microblog usually has a length limitation of 140 characters, which leads to extremely sparse vectors for the learning algorithms. The free and informal writing styles of users also set obstacles for the emotion detection in microblogs. Secondly, although the text is short, multiple emotions may be coexisting in one microblog. As shown in Table 1, the second tweet expresses a *disgust* emotion as well as a *fear* emotion simultaneously. Therefore, for the emotion detection task, each microblog can be associated with multiple fine-grained emotion labels, which is rarely studied in the previous literature.

To tackle these challenges, in this paper, we regard the emotion detection in microblogs as a multi-label classification problem. We utilize word distributed representation to alleviate the sparse vector problem and leverage a Convolutional Neural Network (CNN) based method to detect the coexisting mixed emotions in the short text of microblogs.

Multi-label classification has been extensively studied in many applications [1]. However, only limited literature are available for analyzing the coexisting emotions in text. Ye et al. employed multi-label classification algorithm to classify the input online news articles into categories according to different readers' emotions [2]. Bhowmick et al. presented a novel method for classifying news into multiple emotion categories using an ensemble based multi-label classification technique called RAKEL [3]. Read et al. used restricted Boltzmann machines to develop better feature-space representations which could make labels less interdependent and easier to predict for multi-label classification [4]. Although the above two studies have achieved promising results, both of them focused on the online long news articles with dense vector space. Liu et al. gave detailed empirical study of different classic multi-label learning methods for the multiple emotion sentiment classification task of Chinese microblogs [5]. The goal of Liu's work is similar to ours. However, they conducted the experiments on two very small datasets that only had a few hundreds of microblogs and the classic learning methods they used were heavily dependent on the quality of the lexicons and manually designed features.

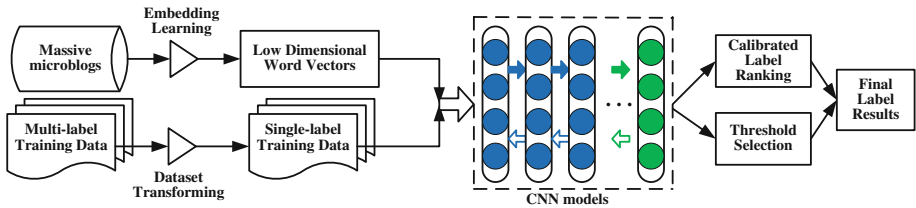


Fig. 1. The details of our proposed framework

In this paper, we utilize a Convolutional Neural Network (CNN) based method to solve multi-label emotion classification problem without any manually designed features in the Chinese microblogs. The details of our proposed framework are shown in Fig. 1. (1) We preprocess the microblogs and transform the multi-label training dataset into single-label dataset. (2) We utilize skip-gram language model and extremely large corpus to train the distributed representation of each word in the single-label dataset. The dense vector representations of the microblog sentences are generated by the word vector with appropriate semantic composition method. (3) We use a CNN based method to train one multi-class classifier and k binary classifiers to get the probability values of the k emotion labels. (4) We leverage a multi-label learning method based on Calibrated Label Ranking (CLR) to get the final emotion labels of each microblog. As a powerful deep learning algorithm, CNN has achieved remarkable performance in computer vision and speech recognition. However, as far as we know, CNN has never before been reported for solving the multi-label text classification task.

The main contributions of this paper are summarized as follows: (1) To the best of our knowledge, this is the first attempt to explore the feasibility and effectiveness of CNN model with the help of word embedding to detect the coexisting emotions in the microblog short text. (2) We utilize skip-gram language model to get word distributed representation instead of the TFIDF based extremely sparse vectors. Meanwhile, the input features are learned by CNN automatically rather than being designed manually. (3) We conduct extensive experiments on two public short text datasets. The experimental results demonstrate that the proposed method achieves excellent performance in terms of multi-label classification metrics.

2 Related Work

Analyzing the sentiments and emotions in microblogs has attracted increasing research interest in recent years. The methods used for microblog sentiment analysis generally can be classified into lexicon-based [6] and learning-based [7] categories. The performance of many existing learning based methods are heavily dependent on the choice of feature representation of tweets. For example, Mohammad et al. designed a number of hand-crafted features to implement a twitter sentiment classification system that achieved the best performance of SemEval 2013 task [8]. Unlike the previous studies, on one hand the CNN model used in this paper could automatically learn discriminative features for the classification task. On the other hand, we utilize the multi-label learning

based framework to detect the coexisting multiple emotions in tweets, which is rarely studied in previous literature.

The goal of multi-label classification is to associate a sentence with a set of labels. There are two main types of methods for multi-label classification: algorithm adaptation methods and problem transformation methods [9]. Algorithm adaptation methods generalize existing classification algorithms to handle multi-label datasets. Zhang et al. proposed a multi-label lazy learning approach named ML-KNN to solve the multi-label problem [10]. Problem transformation methods transform a multi-label classification problem into single-label problems. Wang et al. proposed a calibrated label ranking based framework for detecting the multi-label fine-grained emotions in the Chinese microblogs [11]. According to the detailed empirical study of different multi-label classification methods by [5] and our previous work, we utilize the Calibrated Label Ranking (CLR) as the learning method for the multi-label microblog emotion classification task.

As a powerful deep learning algorithm, Convolutional Neural Network could decrease the number of features by capturing local semantic features and is to use end-to-end automatically trainable systems which do not rely on human-designed heuristics. CNN has been successfully used in image recognition, speech recognition [12], and machine translation [13]. Wei et al. propose a flexible deep CNN infrastructure, called Hypotheses-CNN-Pooling (HCP), to solve the multi-label problem of images [14]. In recent years, CNN has achieved remarkable performances for various challenging NLP tasks, such as part-of-speech tagging, chunking, named entity recognition and semantic role labeling [15]. Santos et al. proposed a new deep CNN that exploited from character- to sentence-level information to perform sentiment analysis of short text [16]. Kim et al. proposed a simple CNN architecture to allow for the use of both task-specific and static vectors for sentence-level classification tasks [17]. Although a lot of papers have been published for CNN based text classification, no literature has been reported for exploiting the feasibility and effectiveness of CNN model for the multi-label microblog emotion classification task. In this paper, we introduce a CNN framework to solve multi-label emotion classification problem in the Chinese microblogs without any manually designed features.

3 Microblog Modeling Using Convolutional Neural Network

In this section, we propose a CNN framework to solve multi-label sentiment classification problem in the Chinese microblogs. Suppose $Y = \{y_1, y_2, \dots, y_k\}$ represents the emotion label set. Giving the training microblog dataset $D = \{(x_i, Y_i) | 1 \leq i \leq p\}$ where p is the number of sentences in the D , x_i is a sentence, Y_i represents the emotion label subset, $Y_i \subseteq Y$. Our aim is to use a CNN classifier to predict the emotion label set $h(x) \subseteq Y$ of a sentence x in test dataset T .

3.1 Preprocessing of Datasets and Training Dataset Transformation

The raw microblog data is inconsistency and redundancy. We remove all the URLs, @ symbols, stop words and English tweets. Then, we use Jieba¹ python package for Chinese text segmentation which break the microblogs into words.

Because we use Calibrated Label Ranking (CLR) as method for multi-label classification, we need to transform the multi-label training microblog dataset into single-label dataset. The goal of problem transformation method is to transform the multi-label problem into other well-established learning scenario. For the given training dataset, each sentence in the microblog may contain one or more relevant emotion labels, therefore we need to transform the multi-label training dataset into the single-label train dataset so that we can use the CNN learning algorithm to address the multi-label learning problem. The dataset transformation method is shown in Algorithm 1.

Algorithm 1. Training dataset transformation

Input: The multi-label training microblog sentence dataset D ;

Output: The single-label training dataset D' after transformed;

Description:

1. FOR every microblog sentence x_i , where Y_i is the relevant emotion label set of the sentence x_i and $D = \{(x_i, Y_i) | 1 \leq i \leq p\}$, where p is the number of microblog sentences in dataset D .

2. FOR every label y_j in the label set Y , where $Y = \{like, happiness, sadness, fear, surprise, disgust, anger\}$ or $Y = \{anger, anxiety, expect, hate, joy, love, sorrow, surprise\}$.

3. If $y_j \in Y_i$, put the microblog sentence (x_i, y_j) into D' .

In Algorithm 1, we associate each microblog sentence with only one emotion label, which means that a sentence x_i in D may appear multiple times in D' .

3.2 Convolutional Neural Network Model

The CNN model is a variant of neural network, which includes the input layer, convolutional layer, pooling layer and fully connected layer. The CNN model used in this paper for modeling the microblog sentences is shown in Fig. 2 below.

The neurons in CNN are usually arranged in three dimensions: depth, width, and height. The size of input layer, convolutional layer, pooling layer and output layer are depth \times width \times height. For example, when a sentence has 70 words and each word is represented by 300 dimensional vector, the size of input layer is $1 \times 70 \times 300$.

Input Layer. This layer contains the input data. Let $s_i \in R^k$ be the k -dimensional word vector corresponding to the i -th word in the sentence. The max sentence length in the datasets is n . If the length is less than n , the sentence is padded necessarily. Therefore, every sentence is represented as

¹ <https://github.com/fxsjy/jieba>.

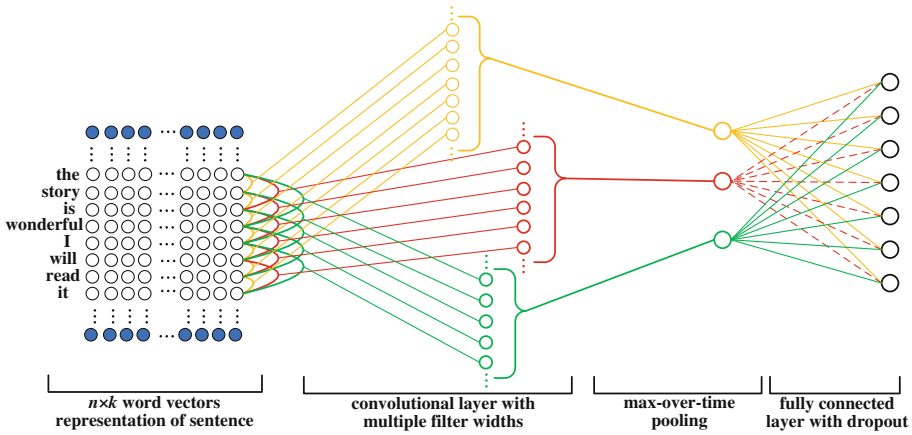


Fig. 2. Convolutional approach to sentence-level classification with different filter widths

$$s_{1:n} = s_1 \oplus s_2 \oplus \dots \oplus s_n \tag{1}$$

where \oplus is the concatenation operator.

Convolutional Layer. In this layer, local semantic features can be captured and the connection between two connected neuron represents the layer of convolution and input. Here a convolution operation involves a filter $w \in \mathbb{R}^{hk}$, which is applied to a window of h words to produce a new feature. The filter w can also be considered as the weight matrix in the convolution layer. We can calculate the value of h words to generate new feature z_i :

$$z_i = f(w \cdot s_{i:i+h-1} + b) \tag{2}$$

where f is an activation function such as rectified linear units and $b \in \mathbb{R}$ is a bias term. This filter is applied to the sentence $\{s_{1:h}, s_{2:h+1}, \dots, s_{n-h+1:n}\}$ to produce a new feature map $z = [z_1, z_2, \dots, z_{n-h+1}]$, with $z \in \mathbb{R}^{n-h+1}$.

Pooling Layer. In this layer, the max-over-time pooling layer selects global semantic features and attempts to capture the most important feature with the highest value for each feature map. The pooling layer is aimed at sampling with common statistics methods, such as mean value, max value and L2-norm. By doing this, the problem of overfitting is addressed to a certain degree. We then apply max-over-time pooling operation over the new feature map by convolution operation and take maximum value $z_{max} = \max(z)$ as the feature corresponding to the filter. Because the model employs several parallel convolutional layers with different window sizes of words, we obtain multiple features so that we can get the input vectors for classifier.

Fully Connected Layer. In this layer, the dropout regularization method is used to summarize the features. The neurons in fully connected layers have full connections with all neurons in the previous layer. The structure of a fully connected layer is the same as that of the layer in the ordinary neural network. In order to avoid overfitting and approximate a way to solve the exponentially neural network computational architectures, we apply dropout on the fully connected layers. When training the model, the neurons which is dropped out has a probability p to be temporarily removed from the network, as shown in Fig. 2. When calculating input and output both in the forward propagation and the back propagation, the dropout neurons are ignored. That means the dropout will prevent neurons from co-adapting too much.

Softmax Layer. Finally, these features are passed to the softmax layer and the output of softmax layer is the probability distribution over labels.

Input Word Vectors. The input of the CNN model is the distributed representation of the words, namely word vectors, in the microblog sentences. These vectors could be initialized randomly during CNN model training. But a popular method to improve the performance of CNN is to initialize word vectors by unsupervised neural language model. In this paper, we utilize the skip-gram model [18] to pre-train word vectors from extremely large corpus that has more than 30 million Chinese microblogs. Throughout the training process of CNN, one strategy is to keep the pre-trained word vectors static, another strategy is to fine-tune the vectors via back propagation.

4 Calibrated Label Ranking Based Multi-label Classification

Based on the transformed single label microblog dataset and the CNN classification algorithm, in this section we leverage the calibrated label ranking method to detect the multiple emotions in the microblog short text.

4.1 Ranking Function

A single label multi-class classifier $f(\cdot)$ is built, so that we can get the confidence of the emotion labels for a given microblog sentence. We use $f(x_i, y_j)$ represents the confidence that sentence x_i has the probability of emotion label y_j . The $f(x_i, y_j)$ can be transformed to a rank function $rank_f(x_i, y_j)$ subsequently and we use the Formula 3 to get the rank value.

$$rank_f(x_i, y_j) = \sum_{k=1, j \neq k}^q \llbracket f(x_i, y_j) > f(x_i, y_k) \rrbracket \quad (3)$$

where $y_j \in Y$, Y is the label set, q is the number of labels, $\delta = 1$ if the predicate δ holds, and 0 otherwise.

4.2 Threshold Selection and Label Set Result

After we get the rank values of all labels by the rank function, the key issue is to gain a threshold to divide the rank value list into relevant label set and irrelevant label set. In order to determine the threshold, the processing procedure is as follows:

- (1) Transform the multi-class single label training dataset into q separate datasets and each dataset is associated with a binary label (with/without an emotion label).
- (2) For q binary labels training datasets, we construct q binary CNN classifiers $g_j(x_i)$, $j \in \{1, 2, \dots, q\}$ and x_i represents a sentence. We use the Formula 4 to construct the training dataset of the q binary classifiers.

$$D_j = \{(x_i, \phi(Y_i, y_j)) | 1 \leq i \leq p\}$$

$$\phi(Y_i, y_j) = \begin{cases} +1, & y_j \in Y_i \\ -1, & \text{otherwise} \end{cases} \quad (4)$$

where D_j is the train dataset, $j \in \{1, 2, \dots, q\}$, y_j represent the emotion label, p is the number of sentences in the training dataset and Y_i is the relevant label set of the sentence x_i . If the label y_j in the labels set Y_i , the $\phi(Y_i, y_j) = +1$, otherwise $\phi(Y_i, y_j) = -1$.

The q binary classifiers $g_j(x_i)$ trained by the separate training datasets classify every sentence in the testing dataset. If x_i is corresponding with y_j , then $g_j(x_i) = +1$, else $g_j(x_i) = -1$.

- (3) Apply the output results of the q binary classifiers to product the threshold and the rank value for every microblog sentence in the testing dataset. The rank value $rank_f(x_i, y_j)$ of label y_j , and the threshold value $rank_f^*(x_i, y_v)$ is updated as follow.

$$rank_f^*(x_i, y_j) = rank_f(x_i, y_j) + \llbracket [g_j(x_i) > 0] \rrbracket \quad (5)$$

$$rank_f^*(x_i, y_v) = \sum_{j=1}^q \llbracket [g_j(x_i) < 0] \rrbracket \quad (6)$$

Finally, we compare the rank value with the threshold value so that we can get the relevant label set and irrelevant label set. If the rank value of a label is bigger than the threshold, namely $rank_f^*(x_i, y_v)$, we put the label into the relevant label set. Otherwise we put it into the irrelevant label set. We regard the relevant label set as the final result of multi-label classification. The formula of multi-label emotion classifier $h(x_i)$ is:

$$h(x_i) = \{y_j | rank_f^*(x_i, y_j) > rank_f^*(x_i, y_v), y_j \in Y\} \quad (7)$$

In summary, for a sentence in test dataset, the process of predicting its multi-label emotion set is shown as Algorithm 2.

Algorithm 2. Predicting Multi-label Emotion Set for a Sentence in Test Dataset**Input:** a sentence x , word vectors, dataset D' from Algorithm 1;**Output:** emotion set of x ;**Description:**

1. Transform x and D' into the feature representation of word vectors.
2. Put the word vectors of x and D' into CNN model.
3. Get multi-class probability of emotion label y_j and multiple binary labels.
4. Apply Calibrated Label Ranking algorithm on the multi-class probability of emotion label y_j to get the rank value:

$$rank_f(x_i, y_j) = \sum_{k=1, j \neq k}^q \mathbb{I}[f(x_i, y_j) > f(x_i, y_k)]$$

5. Apply threshold selection on multiple binary labels to update rank value and get threshold value:

$$rank_f^*(x_i, y_j) = rank_f(x_i, y_j) + \mathbb{I}[g_j(x_i) > 0]$$

$$rank_f^*(x_i, y_v) = \sum_{j=1}^q \mathbb{I}[g_j(x_i) < 0]$$

6. Get final result of multi-label classification, that is emotion set of x :

$$h(x_i) = \{y_j | rank_f^*(x_i, y_j) > rank_f^*(x_i, y_v), y_j \in Y\}$$

7. Return the emotion set of x .

5 Experiment

5.1 Experiment Setup and Datasets

We conduct our experiment on the NLPCC 2014 Emotion Analysis in Chinese Weibo Text (EACWT) task² and Ren_CECps [19] dataset respectively. In the EACWT, the training dataset contains 15,677 microblog sentences, the testing dataset contains 3,816 microblog sentences, the fine-grained emotion label set contains seven basic emotions, i.e. $Y = \{like, happiness, sadness, fear, surprise, disgust, anger\}$ and every sentence has less than two emotion labels. In the Ren_CECps, the training dataset contains 26,000 Chinese blog sentences, the testing dataset contains 6,153 Chinese blog sentences, the fine-grained emotion label set contains eight basic emotions, i.e. $Y = \{anger, anxiety, expect, hate, joy, love, sorrow, surprise\}$ and every sentence has less than three emotion label. Note that Ren_CECps is a Chinese blog dataset which contains documents with different topics. However, the average length of the sentences is similar to that of the microblog dataset EACWT. Therefore, we can utilize the sentences in Ren_CECps as well as EACWT to evaluate the performance of our proposed short text multi-label emotion classification algorithm.

Table 2 show the statistics information of the number of sentences in the datasets with different number of emotion labels. In Table 2, the Ren_CECps dataset has much more sentences with two emotion labels. We argue that using these two datasets with different emotion label distributions, we can better evaluate the feasibility and effectiveness of our proposed model and method.

² http://tcci.ccf.org.cn/conference/2014/pages/page04_eva.html.

Table 2. The number of sentences in the datasets with different number of emotion labels

# of emotion labels in sentence	# of sentences in the dataset	
	EACWT	Ren_CECps
1	17,545	18,812
2	1,948	11,416
3	0	1,824

5.2 Hyper-parameters Settings

For the CNN model we use: rectified linear units, filter windows (h) of 2, 3, 4 with 100 feature maps each, dropout rate (p) of 0.5, l_2 constraint (s) of 3, mini-batch size of 50 based on stochastic gradient descent. The Adadelta update rule is used. The max sentence length n is 70 and 90 in the EACWT and Ren_CECps respectively.

We utilize the well-known word embedding tool word2vec to train the word vectors on a crawled 30 million Chinese microblog corpus. The word2vec hyper-parameters of our pre-trained word vectors includes: the selected algorithm is skip-gram, the size of dimensionality is 50, 100, 200 and 300 respectively, the size of window is 8, the value of sample is $1e-4$, and the times of iteration is 15.

5.3 Model Variations and Baselines

In this paper, we conduct the experiments with several variants of the CNN model. The SVM classifier that has achieved excellent performance in many previous sentiment classification tasks is viewed as the baseline model.

SVM-tfidf: We use the tfidf to generate the unigram feature vectors, and apply SVM classifier with CLR algorithm to solve the microblog multi-label classification problem. In the previous work, the SVM-tfidf method has achieved the best performance in the EACWT evaluation task [11]. We regard it as a strong baseline.

SVM-word2vec: We use the vector semantic composition based degeneration algorithm for modeling the microblog sentence. The vectors of words in the sentence are added together to form the distributed representation of the microblog sentence.

CNN-static: The inputs of CNN are the pre-trained vectors of words in microblog sentences from word2vec. If the word is not in the set of pre-trained word vectors, the unknown one is randomly initialized. Here “-static” means that the word vectors are kept static during training and only other parameters of the model are learned.

CNN-non-static: It is the same as the CNN-static but the pre-trained vectors are fine-tuned during training for different datasets.

CNN-rand: The input of all words vector are randomly initialized and modified during training.

Evaluation metrics. The classic multi-label learning evaluation metrics Ranking Loss (RL), Hamming Loss (HL), One-Error (OE), Subset Accuracy (SA), macro F1 (maF), micro F1 (miF) and Average Precision (AP) are selected for evaluating the

performance of the proposed model and algorithm. The seven metrics can be grouped into label-based metrics, example-based metrics and ranking-based metrics. For more details about these metrics, please refer to Madjarov et al. [20] and Zhang et al. [9].

5.4 Experimental Results

We show the best result of every model in the Table 3 and Table 4. “↓” means the smaller values are better. “↑” means the larger values are better. Meanwhile, the SVM-word2vec-300, CNN-static-300, CNN-non-static-300 and CNN-rand-300, “-300” represent that the input word vectors have the dimensionality of 300 in the corresponding models. We can clearly see that in EACWT dataset the model CNN-non-static-300 has achieved the best performances under the every evaluation metric comparing with the other models in the Table 3. In Ren_CECps dataset, CNN-static-300 is better than other baselines and CNN models under all the metrics except Hamming Loss in the Table 4.

Based on the observations from above two tables, we have following discussions.

- (1) With appropriate parameter settings, the CNN based models outperform the SVM algorithms. We argue that the CNN models can automatically learn the higher level of feature representation, so it is the better than the SVM based methods.
- (2) The performance of CNN models can be significantly improved by using pre-trained word vectors. This is because that compared with randomly initialized vectors, pre-trained vectors could learn the semantic information of the word from extremely large corpus. Moreover, pre-trained vectors can be fine-tuned on the dataset in “non-static” model, so that vectors indeed reflect the word semantic in the dataset. We can see that CNN-non-static model has achieved best performances in EACWT dataset.
- (3) The “non-static” model does not perform best as we expected in the Ren_CECps dataset. This may be because that Ren_CECps contains various topics and has more scattered emotion labels. Different documents have the different topics. So it is more difficult for learning the effective feature representations and semantic vectors. The CNN-static model has achieved the best performances in Ren_CECps dataset. A similar observation of “static” model is reported for a single-label text classification task in [17]. We think the performance of “static” and “non-static” CNN models are related to the topic and label distribution for the text classification.

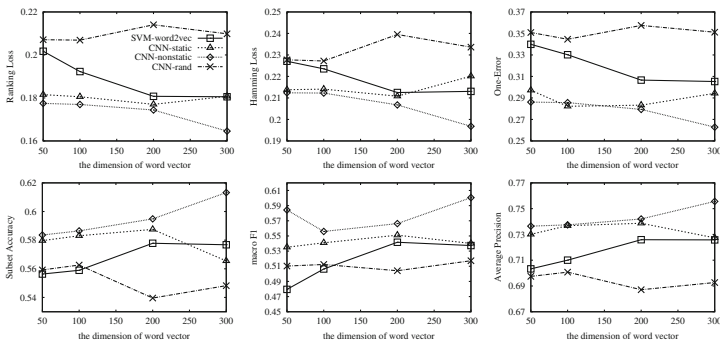
Finally, we demonstrate the effects of word vector dimension for EACWT dataset in Fig. 3. Generally, we find that when the word dimension is bigger, a better performance for the CNN-non-static model could be achieved. Similar performances can be observed in Ren_CECps dataset. We omit the results of micro F1 metric in EACWT and all the metrics in Ren_CECps due to length limitation.

Table 3. Microblog multi-label emotion classification results on the EACWT dataset

Algorithm	RL↓	HL↓	OE↓	SA↑	maF↑	miF↑	AP↑
SVM-tfidf	0.1879	0.2045	0.3160	0.6014	0.5532	0.6014	0.7250
SVM-word2vec-300	0.1805	0.2130	0.3052	0.5767	0.5373	0.5767	0.7257
CNN-static-300	0.1807	0.2201	0.2942	0.5655	0.5401	0.5655	0.7273
CNN-non-static-300	0.1645	0.1968	0.2628	0.6132	0.6005	0.6132	0.7556
CNN-rand-300	0.2098	0.2336	0.3511	0.5482	0.5173	0.5482	0.6926

Table 4. Microblog multi-label emotion classification results on the Ren_CECps dataset

Algorithm	RL↓	HL↓	OE↓	SA↑	maF↑	miF↑	AP↑
SVM-tfidf	0.1914	0.2888	0.4965	0.2665	0.3095	0.2665	0.6125
SVM-word2vec-300	0.1917	0.3043	0.4687	0.2790	0.3491	0.2790	0.6226
CNN-static-300	0.1882	0.3164	0.4352	0.2889	0.3868	0.2889	0.6320
CNN-non-static-300	0.1893	0.3269	0.4493	0.2668	0.3682	0.2668	0.6219
CNN-rand-300	0.1996	0.3778	0.5264	0.2192	0.2780	0.2192	0.5782

**Fig. 3.** The experiment results for different word dimensions for the EACWT dataset

6 Conclusions and Future Work

Different from the traditional sentiment orientation classification problem, multiple fine-grained emotions can co-exist in the microblog sentences, which can be regarded as a multi-label learning problem. In this paper, we propose a Convolutional Neural Network framework using Calibrated Label Ranking to detect the multiple emotions in the Chinese microblog sentences. Extensive experiments on two dataset show that static and non-static CNN methods with the help of word embedding could achieve better performances than other strong baseline algorithms by a large margin.

In the future, on one hand we want to design new problem transformation methods for the multiple emotion detection task. On the other hand, we intend to revise the architecture of CNN models for multi-label learning problem and incorporate other deep learning models, such as RNN, to further improve the performance of the task.

Acknowledgements. The work is supported by National Natural Science Foundation of China (61370074, 61402091), the Fundamental Research Funds for the Central Universities of China under Grant N140404012.

References

1. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) *Data Mining and Knowledge Discovery Handbook*, pp. 667–685. Springer, Heidelberg (2010)
2. Ye, L., Xu, R., Xu, J.: Emotion prediction of news articles from reader’s perspective based on multi-label classification. In: *Proceedings of ICMLC*, pp. 2019–2024 (2012)
3. Bhowmick, P.K.: Reader perspective emotion analysis in text through ensemble based multi-label classification framework. *Comput. Inf. Sci. (CCSEICIS)* **2**(4), 64–74 (2009)
4. Read, J., Pérez-Cruz, F.: Deep learning for multi-label classification. *CoRR abs/1502.05988* (2015)
5. Liu, S., Chen, J.: A multi-label classification based approach for sentiment classification. *Expert Syst. Appl.* **42**(3), 1083–1093 (2015)
6. Tang, D., Wei, F., Qin, B., Zhou, M., Liu, T.: Building large-scale Twitter-specific sentiment lexicon: a representation learning approach. In: *Proceedings of COLING*, pp. 172–182 (2014)
7. Liu, K., Li, W., Guo, M.: Emoticon smoothed language models for Twitter sentiment analysis. In: *Proceedings of AAAI* (2012)
8. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: building the state-of-the-art in sentiment analysis of Tweets. *CoRR abs/1308.6242* (2013)
9. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.* **26**(8), 1819–1837 (2014)
10. Zhang, M., Zhou, Z.: ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recogn. (PR)* **40**(7), 2038–2048 (2007)
11. Wang, M., Liu, M., Feng, S., Wang, D., Zhang, Y.: A novel calibrated label ranking based method for multiple emotions detection in Chinese microblogs. In: Zong, C., Nie, J.-Y., Zhao, D., Feng, Y. (eds.) *NLPCC 2014. CCIS*, vol. 496, pp. 238–250. Springer, Heidelberg (2014)
12. Abdel-Hamid, O., Deng, L., Yu, D.: Exploring convolutional neural network structures and optimization techniques for speech recognition. In: *Proceedings of INTERSPEECH*, pp. 3366–3370 (2013)
13. Meng, F., Lu, Z., Wang, M., Li, H., Jiang, W., Liu, Q.: Encoding source language with convolutional neural network for machine translation. *ACL* (1), pp. 20–30 (2015)
14. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: CNN: single-label to multi-label. *CoRR abs/1406.5726* (2014)
15. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: *Proceedings of ICML* pp. 160–167 (2008)
16. Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of COLING* pp. 69–78 (2014)
17. Kim, Y.: Convolutional neural networks for sentence classification. In: *Proceedings of EMNLP* pp. 1746–1751 (2014)

18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS (2013)
19. Quan, C., Ren, F.: A blog emotion corpus for emotional expression analysis in Chinese. *Comput. Speech Lang. (CSL)* **24**(4), 726–749 (2010)
20. Madjarov, G., Kocev, D., Gjorgjevikj, D., Dzeroski, S.: An extensive experimental comparison of methods for multi-label learning. *Pattern Recogn. (PR)* **45**(9), 3084–3104 (2012)