Sentiment Classification on Weibo Incidents Using CNN-SVM and Repost Tree

Manshu Tu$^{1,\ a}$, Shengxiang Gao$^{2,\ b}$, Zhe Ji$^{3,\ c}$, Yan Zhang$^{4,\ d}$, and Yonghong Yan$^{5,\ e}$

$^{1,4,5}$The Key Laboratory of Speech Acoustics and Content Understanding Institute of Acoustics, Chinese Academy of Sciences, Beijing, China

$^{2,3}$National Computer network Emergency Response technical Team/Coordination Center of China, Beijing

$tumanshu@hccl.ioa.ac.cn$, bgao.shengxiang@cert.org.cn, cjz@cert.org.cn, dzhangyan@hccl.ioa.ac.cn, eyanyonghong@hccl.ioa.ac.cn

Keywords: Sentiment classification. CNN-SVM. Repost tree

Abstract. Sentiment classification on weibo has recently attracted wide attention in research community. Most previous works are focused on weibo comments regarding movies or products. Our study, in contrast, is aimed at gusty incidents on weibo. Comments of the incidents are considered either positive or negative representing attitudes of users towards these incidents. Classifying users’ attitudes helps identifying the general opinion of the public. In this paper, we propose an innovative convolutional neural networks based method, termed as CNN-SVM, to classify the incident comments. In addition, according to users’ repost actions, we propose a new data structure, repost tree, for dealing with ambiguity in the comments. Extensive experiments demonstrate that the CNN-SVM method effectively improves the accuracy of incidents sentiment classification. The new data structure shows to be effective on steering the classification results towards real world sentiment tendency.

Introduction

In this paper, we are interested in analyzing people’s emotional attitudes (i.e. positive and negative sentiment bias) towards incidents. We want to improve the accuracy of sentiment classification through modifying previous methods. We also take the advantage of weibo data features to obtain the real ratio of sentiment. Pang et al. examined the effectiveness of machine learning methods for sentiment classification tasks on movie reviews. Three machine learning techniques used in their experiments are Naive Bayes, maximum entropy classification and support vector machines (SVM). SVM tends to do the best in terms of performance[1]. It is commonly used to solve small sample, nonlinear and high dimensional pattern recognition problems [2]. CNN using convolution to the local features of the filter layer was originally invented for computer vision [3]. Kim trained a simple CNN with one layer convolution on top of word vectors [4]. In contrast to conventional NLP tasks, sentiment analysis on microblogging posts is challenging due to the limited text length.

As described above, CNN and SVM are both popular methods for sentiment classification. However, to the best of our knowledge, their combined performance has not been tested in sentiment classification tasks. None of these functions above has considered microblogs reposts. In order to address these issues, we propose to combine the CNN and SVM together to build a new classification model. We further propose a novel repost tree to take context into consideration for sentiment classification tasks. The repost tree can perform logical operation on sentiment for each node to adjust its sentiment prediction result. Our experiments show that the CNN-SVM with the repost tree can get results close to the real world sentiment tendency ratio. The rest of the paper is organized as follows: Section 2 describes in details the model CNN-SVM and repost tree. Section 3 presents datasets used in our experiments and data preprocessing. Section 4 gives the conclusions of our research and future works.
Model

**CNN-SVM.** Our proposed model is shown in Fig. 1. It is inspired by Kim’s CNN model[4]. A CNN is used for feature extraction, while a SVM carries out the sentiment classification. The model works as follows. The input of this model is a matrix $S \in \mathbb{R}^{(s \times n)}$. The notation $n$ means that every word in a sentence has $n$ pre-trained vectors: $[v_1, \cdots, v_i, \cdots, v_n]$. The notations refer to every sentence consisting of $s$ words: $[w_1, \cdots, w_i, \cdots, w_s]$. The convolution layer of the CNN has a number of different types of filters $F \in \mathbb{R}^{(m \times n)}$, where $m$ is the width of the filter. The result of the $i$th matrix vectors in a sentence through the filter is computed as follows equation:

$$c_i = (S \ast F)_i = \sum_{k=1}^{K} (S_{[i-m+1:i]} \otimes F)_{ki}(1)$$

Where $k$ is the $k$-th convolution layer, while $\otimes$ is the element-wise multiplication and $S_{[i-m+1:i]}$ is a matrix slice of size $m$ along the columns. Every sentence matrix $S$ gets a feature map $c \in \mathbb{R}^{1 \times (s-m+1)}$ through one filter for convolution operations. To form a richer representation of the data, one kind filter has $p$ convolution kernels. Suppose there are $t$ types of filters, then one sentence will get $p \times t$ feature maps $C: [c_{f1}, \cdots, c_{ft}, \cdots, c_{tf}]$. Max-pooling is a pooling layer to find the maximum value in kernel size. At the max-pooling layer, every feature map $c$ returns the largest value $c_{pool}: \mathbb{R}^{1 \times (s-m+1)} \rightarrow \mathbb{R}^{1 \times 1}$. Now we get the high-dimensional features: $D: [c_{pool1}, \cdots, c_{poolt}, \cdots, c_{pooltp}]$. The output of the max-pooling layer is passed to a fully connected soft-max layer. It computes the probability of the sentence labels as follows equation, e.g.:

$$p(y=j|c_{pool}, B) = softmax_j(c_{pool}W + B) = \frac{e^{c_{pool}w_j+b_j}}{\sum_{k=1}^{K} e^{c_{pool}w_k+b_k}}(2)$$

Where $w_k$ and $b_k$ are the weight vectors and bias of $k$-th class. The softmax layer compares the predicted labels and the real labels to fine-tune the CNN model. When the CNN accuracy remains stable, the training data with high-dimensional vectors is sent to the SVM model. Then we train the SVM model until its best accuracy. Retaining these parameters we have trained, then send the test data to the model for classifying.

![Fig. 1.](image)

*Fig. 1.* In this figure, there are three different filters. At the end of the convolutional layer, every column represents a sentence go through a kind of filter. Every kind feature map has $p$ feature maps. These feature maps go through max-pooling layer changing to $p$ high-dimensional features.

*Fig. 2.* The different sizes of the circles represent different hierarchies. The number inside or outside the circle is the sentiment result using a classification algorithm.

**Repost Tree.** Reposting a microblog is a common behavior in microblogging. Users repost a microblog to air their opinion about the microblog they have forwarded. The repost can express different types of opinions. For example, when an incident occurred, one user comments with a negative message. Then another user reposts this message and writes a positive sentence, expressing a supporting attitude. It means that the second user has a negative attitude towards this
incident. If we only analyze the first sentence, the classifier will classify it to be positive, but the real sentiment of this user is negative.

Weibo has a convenient feature for building a repost tree. It shows paths of each repost. The paths include nicknames and contents like “@+nickname :+content + //”. We build repost trees using this structure to adjust sentiment classification results. Repost trees contain nickname, content and timestamp. The structure of the proposed tree is shown in Fig. 2. We arrange all the nodes according to the number of their child nodes in descending order. Then the repost tree can proceed to its logical operation step. We define the status of all nodes on a branch as $B_{i_1}, \ldots, B_{i_n}$, the value of $n_i$ is either +1 or -1. The logical operation result for $Node_n$ is regarded as the nodereal sentiment tendency. It is computed as follows:

$$Node_n = \prod_{i=1}^{n} n_i (3)$$

**Experiments**

**Datasets and preprocessing.** Our model is tested in three datasets. All of them are extracted from SinaWeibo. MN is used for this data abbreviation. WZ is used for this data abbreviation. Paris for this data abbreviation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN</td>
<td>1600</td>
<td>400</td>
</tr>
<tr>
<td>WZ</td>
<td>23120</td>
<td>5780</td>
</tr>
<tr>
<td>Paris</td>
<td>480</td>
<td>120</td>
</tr>
</tbody>
</table>

**Table 1. Datasets**

We use the method described in 3.2 to label MZ and Paris data. All the data are processed using the following steps: deleting URL, word segmentation using ICTCLAS, removing stop words and sentences with number of words less than two. The details of these datasets are listed in Table 1. Training datasets and testing datasets have the same negative examples to positive examples ratio.

**Distance Supervision and Feature.** Emoticons. Users nowadays like to air their opinion with emoticons. Weibo come with many emoticons. Some of them show obvious emotional tendency, therefore we use these emoticons as labels Go et al. [5]. In Weibo, the emoticons are encoded in the form of “[Chinese word]”. We convert obvious emoticons to “+1” or “-1”, where “+1” represents positive sentiment and “-1” represents negative sentiment. We use the publicly available word2vec vectors that were trained on three billions of Weibo sentences. The vectors have a dimensionality of 200, containing 1043631 words [6].

**Other Methods for Comparing.** We use SVM as a baseline. The Deep Belief Network (DBN) and Restricted Boltzmann Machine (RBM) also be used in our experiments [7]. We implement the DBN and RBM model [8] for comparison. We also include a simple CNN model in our experiments. A five-fold cross-validation are conducted in all models except the dictionary method.

**Performance Results** Table 2 shows the accuracy of CNN-SVM in comparison with the other methods. We first compare the result of these three datasets, we find that all methods have better performance in MN dataset than that in WZ and Paris dataset. The reason is that the MN dataset is a standard dataset, all the sentences are integrated and coherent.

In these three datasets, RBM and DBN have similar poor accuracies. On one hand DBN consists of RBMs, so DBN performs poorly when RBM has weak results. One the other hand, RBM is not very suitable for nonstandard text data classification. CNN performs with good accuracies in both three datasets. It improves with 7.68% in the first dataset, 15.75% in the second datasets and 4.69% in third datasets, compared with the best performance among SVM, RBM and DBN. The proposed CNN-SVM shows higher accuracy than that of CNN. It has a 1.32% increase in the first dataset and 5.58% increase in the second dataset, 2.65% increase in the third dataset compared with CNN. The improvement in first dataset is less than second and third dataset. This might due to that CNN has shown a good performance in the first dataset already. The SVM has a better performance in the
third dataset than in other two datasets, because SVM is usually suitable for small datasets. For the repost tree, we construct experiments using the Paris dataset, which has 14470 repost microblogs. At the website, (the website reported the People’s Daily survey result of the sentiment tendency on this incident), we take the survey result as real world ratio of positive and negative sentiment of the public. As shown in Fig. 3, the prediction with CNN-SVM differs by 20% compared with the real ratio. However, after applying the repost tree, the prediction results are steered close to the real ratio. According to the result, CNN-SVM performs the best when connected to the repost tree. It differs from the real ratio by 7.46%.

**Conclusion and Feature Work**

In this paper we show that CNN-SVM is a method with better performance on weibo incident sentiment classification compared to other state-of-the-art approaches. It can improve the accuracy of incidents sentiment classification by a large margin. Besides, the new data structure, repost tree, can further adjust the prediction result of CNN-SVM towards the real world sentiment tendency. The repost tree might have ambiguity, we will consider this situation in our following work.

**Acknowledgements**

We acknowledge the support of China post doctoral science foundation 2015LH0041.

**References**


