Co-training Based on Multi-type Text Features

Wenting Liu[1,2(✉)], Xiaojun Jing[1,2], Yaqin Chen[1,2], and Jia Li[3]

[1] School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications, Beijing, China
liuwentinghb@l26.com
[2] Key Laboratory of Trustworthy Distributed Computing and Service (BUPT),
Ministry of Education, Beijing University of Posts and Telecommunications,
Beijing, China
[3] School of Engineering and Computer Science, Oakland University,
Rochester, USA

Abstract. Sentiment classification is intended to classify the sentiment color categories expressed by the text. This paper illustrates the sentiment classification method based on the semi-supervised algorithm that aims to improve performance by using unlabeled data. This paper proposes a novel co-training style semi-supervised learning algorithm in order to improve semi-supervised learning ability. In our algorithm, there are three classifiers trained on the original labeled data, where the text representation for each classifier is unigram, bigram, and word2vec, respectively. And then these classifiers can use unlabeled data to update themselves. In detail, any of two classifiers have the same label, then add the new labeled data to a training set of the third classifier. By combining different types of features, our algorithm can extract text information from multiple views which contribute to sentiment classification. In addition, this algorithm doesn't require redundant and sufficient perspectives. Experiments show that our algorithm is superior to traditional co-training algorithm and partial semi-supervised learning algorithm.

Keywords: Sentiment classification · Semi-supervised learning
Co-training · Multi-type text features

1 Introduction

In recent decades, user-generated subjective texts quickly emerged, which contains a large quantity of useful information. In order to analyze and mine valuable opinions of the texts, sentiment analysis came into being. Sentiment classification divides the target text into positive or negative through analyzing the subjective texts [5].

At present, many researchers interest in supervised learning of sentiment classification. However, supervised learning depends on massive labeled data. To solve this problem, semi-supervised learning is applied to sentiment analysis, it takes advantage of both labeled and unlabeled data.

Blum and Mitchell (1998) proposed a high-performance semi-supervised learning algorithm called co-training [1]. The algorithm uses labeled data to train classifiers on two different views, and then adds the new labeled data which was predicted by the

S. Sun et al. (eds.), *Signal and Information Processing, Networking and Computers*, Lecture Notes in Electrical Engineering 473,
https://doi.org/10.1007/978-981-10-7521-6_26

other classifier to training set of each classifier so that the classifier can use the new labeled data to update itself. The algorithm requires two sufficient and conditional independent views, "sufficient" means that each view contains enough information to generate a strong classifier, "conditional independent" means that the two views should be independent. However, it is difficult to meet both sufficient and independent conditions in most practical applications. Goldman and Zhou (2000) put forward an improved co-training algorithm that does not need to satisfy multi-views condition. However, it needs two different supervised learning algorithms which divide the example space into multiple equivalence classes, and then labeling data through cross validation technique. The negative impact of the extensive apply cross validation to the algorithm include a high time complexity.

In this paper, we proposed a novel co-training algorithm for sentiment classification which does need to meet sufficient and independent conditions, nor does it need to use two totally different supervised learning algorithms which divide the example space into multiple equivalence classes. Thus it is more adaptable. Compared with the above-mentioned algorithms, our algorithm made several changes to attain a better performance. The innovation of our approach includes following two aspects: (1) our approach adopts three classifiers compared with co-training. (2) the text features of each classifier are different.

The remaining part of this paper is organized as follows. Section 2 makes a brief introduction of semi-supervised sentiment analysis. Section 3 illustrates our novel co-training approach for sentiment classification, Sect. 4 evaluates our approach based on the experimental results, Sect. 5 draws the conclusion.

2 Related Work

Supervised sentiment classification is the current mainstream method, it was first introduced into the sentiment classification task by Pang et al., and has achieved good classification performances. A substantial number of follow-up studies have focused on enhancing the performance of supervised learning.

An increasing amount of researchers focus on semi-supervised sentiment classification for the past few years. Wan (2009) proposed an algorithm based on co-training which employs English and Chinese as different views for sentiment classification, English and Chinese have significant logical expression difference [3]. Li et al. (2010a) proposed a co-training approach which exploits personal view and non-personal view for sentiment classification [4]. Dasgupta and Ng (2009) combine several technologies including active learning, spectral clustering, transductive learning and ensemble learning to sentiment classification [10]. However, the accuracy rate of the experiment was low in [2], an improved co-training algorithm which exploits two different supervised learning algorithm for co-training was proposed by Goldman and Zhou. In [7], the two authors adopt an algorithm which uses three classifiers for co-training. Like the above algorithms, our algorithm is also based on co-training, it uses original labeled data set to train three classifiers on different text representation models [11].

3 Improved Co-training Algorithm

3.1 Algorithm Principle

The traditional co-training algorithm needs to satisfy the sufficient and independent conditions. In the actual scenarios, two sufficient and conditional independent views are difficult to find. In order to tackle the problem, our approach use three classifiers. Zhou and Li (2005) have proved that using three classifiers neither needs to meet the sufficient and independent conditions nor needs to use different supervised learning algorithms. In addition, these three classifiers in our algorithm should have greater differences. If the three classifiers are all the same, the results of labeled data obtained by any two classifiers are consistent with the result of the third classifier. Under these circumstances, our algorithm degenerate into self-training algorithm. In the co-training, the diversity of the classifiers can be ensured by satisfying independent and sufficient conditions. In [6], Goldman and Zhou (2000) put forward an algorithm which does not need to meet redundant and sufficient conditions, using two different supervised learning algorithm enable the two classifiers to be diverse. Zhou (2007) proposed a tri-training algorithm uses bootstrap sampling technique to gain three diverse classifiers [7]. Our approach adopts different feature representation models to achieve the diversity of three classifiers. In detail, extracting three different types of feature from original labeled data, and using them to train three classifiers, these classifiers then update through exploiting unlabeled data. each text representation model has its own unique advantages for sentiment classification, considering some of them are complementary and interrelated to some extent. Our algorithm fully takes advantage of different types of features which contribute to a better performance of sentiment classification, thus our algorithm is more advantageous than above-mentioned algorithms.

The main procedure of our algorithm is as follows:

a. Generating three classifiers from original labeled training set that uses different types of features.
b. During the training process, put new labeled data to the third classifier's training set if the other two classifiers have the same prediction.
c. Using updated labeled data set to train classifiers.
d. Continuing to iterate until a certain stop condition is reached.

The pseudocode of the algorithm is as follows.

3.2 Text Feature

This paper chooses three different types of features, which are bigram, unigram and word2vec.

N-gram

Unigram is individual word tokens separated by a punctuation mark or a whitespace, bigram is pairs of adjacent word tokens. For instance, consider the following sentence:

"I love this new phone, and its music experience is great".

Firstly, remove the stop words, then the sentence becomes the following:
'love phone, music experience great'.

The features (Unigram, Bigram) are shown in Table 1. Individual features are separated by square brackets "[]".

The pseudocode of our algorithm

Input: Original labeled data L
Unlabeled data U
Learning algorithm
Output: New classifier C
Procedure:
for $i \in \{1, 2, 3\}$ do
extract text feature F_i from L
use L to train a classifier h_i that text feature is F_i
end for
set L_i equals to ϕ ($i \in \{1, 2, 3\}$)
repeat until none of L_i ($i \in \{1, 2, 3\}$) changes
set S_i equals to ϕ ($i \in \{1, 2, 3\}$)
for every x in U do
use classifier h_i to predict label of x
if $h_j = h_k$ ($j \neq i$ and $k \neq i$)
then $S_i = S_i \cup \{(x, h_j(x))\}$
end for
for $i \in \{1, 2, 3\}$ do
$L_i = L_i \cup S_i$
remove duplicate sample for L_i
use $L_i \cup L$ to learn a classifier h_i
end repeat

Table 1. Text feature from the sample data

Feature set	Text features
Unigram	[love][phone][music][experience][great]
Bigram	[love-phone][phone-music][music-experience][experience-great]

word2vec

Word2vec is an open source and efficient tool published by Google in 2013, and each word is characterized as a numerical vector. The distributed characterizations of each word are obtained by training neural network with one hidden layer [8].

There are two types of word2vec models, the continuous bag-of-words model and Skip-gram model. CBOW model predicts the center word *w(t)* when its context is known, while the skip-gram model, on the contrary, predicts its context under the condition that the center word *w(t)* is already known. Because the training procedure of CBOW model is similar to the training procedure of Skip-gram model, the following part only introduces the training of CBOW model (Fig. 1).

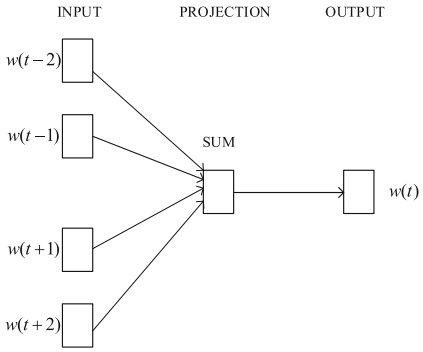

Fig. 1. The training process of the CBOW model

CBOW's network structure includes three layers, according to the data process, which are input layer, hidden layer and output layer. The objective function formula is as follows:

$$\mathrm{L} = \sum\nolimits_{c} \log P(w|context(w)), \tag{1}$$

Where c represents corpus, w represents center word, and *context(w)* represents the context of w.

Input layer: contain $2c$ words vector in *context(w)*.

Projection layer: the $2c$ vectors of the input layer are summed and accumulated, which is

$$x_w = \sum\nolimits_{i=1}^{2c} v(context(w_i)). \tag{2}$$

Output layer: output layer corresponds to a binary tree, word appear in the corpus as a leaf node, and weight is the occurrence number of the word in the corpus.

The output of the result needs to be a softmax normalized, and it's as follows:

$$p(w|context(w)) = \frac{e^{y_w, i_w}}{\sum_{i=1}^{N} e^{e^{y_w, i}}}. \quad (3)$$

When the neural network training is completed, you can find word vector of all words. Interestingly, when using a word vector to express a word, it can be found a similar law: king – "man" + "woman" = "queen". It can be seen that the word vector is very conducive to the expression of the semantic features of the word.

In our approach, the feature vector of each document is the centroid of the word embeddings of the document [12]. The formula for centroid of a document M is as follows:

$$\vec{M} = \frac{1}{|M|} \sum_{i=1}^{M} \overrightarrow{w_i}, \quad (4)$$

where $|M|$ is the number of tokens in N and $\overrightarrow{w_i}$ is the word vector of word w_i.

4 Experimentation

4.1 Experimental Settings

Data Set: We extracted the raw texts from IMDB movie reviews. The 50,000 reviews dataset was split evenly into 25,000 training sets and 25,000 test sets. The sample data are generally evenly distributed (25,000 positives, 25,000 negatives). It also includes an additional 50,000 unlabeled documents. We randomly selected 5% or 10% of the sample as the initial labeled sample, and remaining data as unlabeled data set. Word2vec model is built by the additional 50,000 unlabeled documents. Ten-fold cross validation as the final experimental result.

Features: We remove stop words if unigram is used, and don't remove them if bigram is used. For word2vec, we learn the vector representation of words through training 50000 unlabeled data, and then average all vectors of the words as the feature vector of each review.

Classification Algorithm: SVM algorithm has good performance in emotion classification, this paper uses SVM in sklearn package.

4.2 Experimental Results

In order to reflect the classification performance, the following algorithm and our algorithm for comparison:

Baseline: Supervised learning algorithm with the original labeled data only, in this paper, we use SVM classifier.

Self-training: Firstly, use the original labeled set to train the classifier, then use the classifier iteratively add the highest confidence sample to the labeled set.

Co-training: A co-training algorithm both using feature partition and language translation strategies.

Figure 2 shows the classification performance comparison of the various semi-supervised learning algorithm when the initial labeled set is 5% or 10% of the total training set. From the results we can see that our algorithm obtains the best classification effect, the classification accuracy rate is far better than baseline, compared with Self-training and Co-training, when the initial labeled set is 5% of the total training set, the accuracy rate of our algorithm has been increased by 5.9% and 2.8%, respectively. When the initial labeled set is 10% of the total training set, our algorithm attains the accuracy rate improvement by 6.7% and 2.2%.

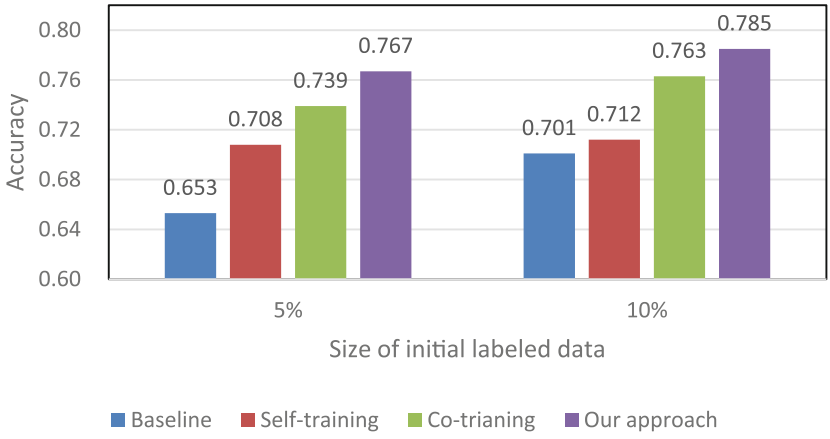

Fig. 2. Performance comparison of different algorithm

5 Conclusions

In this paper, we illustrate a novel co-training style algorithm in semi-supervised sentiment classification. The algorithm uses original training set to train three classifiers, where the text representation model for each classifier is unigram, bigram, and word2vec, respectively and then these classifiers are refined with unlabeled data. Our approach is clearly superior to self-training algorithms and innovative co-training algorithms for semi-supervised sentiment classification, according to the analysis of the experimental data.

Acknowledgments. Project 61471066 supported by NSFC.

References

1. Zhou, X., Wan, X., Xiao, J.: Cross-lingual sentiment classification with bilingual document representation learning. In: Meeting of the Association for Computational Linguistics, pp. 1403–1412 (2016)
2. Xiang, B., Zhou, L.: Improving Twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. In: Meeting of the Association for Computational Linguistics, pp. 434–439 (2014)
3. Xie, S., Wang, T.: Dividing for combination: a bootstrapping sentiment classification framework for micro-blogs. In: International Conference on Information Science and Cloud Computing, pp. 78–84. IEEE (2014)
4. Kim, Y., Zhang, O.: Credibility adjusted term frequency: a supervised term weighting scheme for sentiment analysis and text classification. In: Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 79–83 (2014)
5. Xu, S., Liang, H.Z., Baldwin, T.: UNIMELB at SemEval-2016 tasks 4A and 4B: an ensemble of neural networks and a Word2Vec based model for sentiment classification. In: International Workshop on Semantic Evaluation, pp. 183–189 (2016)
6. Jeevankumar, M., Jain, P., Chetan, M.: Opinion analysis of text on the basis of three domain classification. In: International Conference on Automatic Control and Dynamic Optimization Techniques, pp. 173–177 (2016)
7. Li, S., Huang, L., Wang, J.: Semi-stacking for semi-supervised sentiment classification. In: Meeting of the Association for Computational Linguistics and the, International Joint Conference on Natural Language Processing, pp. 27–31 (2015)
8. Le, T., Mikolov, T.: Distributed representations of words and phrases. In: Proceedings of the 31th International Conference on Machine Learning, Beijing, pp. 1188–1196 (2014)
9. Silva, N.F.D., Hruschka, E.R., Hruschka Jr., E.R.: Biocom Usp: tweet sentiment analysis with adaptive boosting ensemble. In: International Workshop on Semantic Evaluation, pp. 123–128 (2014)
10. Giorgis, S., Rousas, A.: A weighted ensemble of SVMs for Twitter sentiment analysis. In: Proceedings of SemEval-2016, pp. 96–99. Association for Computational Linguistics (2016)
11. Gao, W., Li, S., Lee, S.Y.M.: Joint learning on sentiment and emotion classification. In: ACM International Conference on Information & Knowledge Management, pp. 1505–1508 (2013)
12. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, Lille, France (2015)