Cogn Comput (2015) 7:226–240
DOI 10.1007/s12559-015-9319-y

Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification

Ruifeng Xu · Tao Chen · Yunqing Xia · Qin Lu · Bin Liu · Xuan Wang

Received: 31 August 2014 / Accepted: 23 January 2015 / Published online: 3 February 2015

Abstract Text classification often faces the problem of imbalanced training data. This is true in sentiment analysis and particularly prominent in emotion classification where multiple emotion categories are very likely to produce naturally skewed training data. Different sampling methods have been proposed to improve classification performance by reducing the imbalance ratio between training classes. However, data sparseness and the small disjunct problem remain obstacles in generating new samples for minority classes when the data are skewed and limited. Methods to produce meaningful samples for smaller classes rather than simple duplication are essential in overcoming this problem. In this paper, we present an oversampling method based on word embedding compositionality which produces meaningful balanced training data. We first use a large corpus to train a continuous skip-gram model to form a word embedding model maintaining the syntactic and semantic integrity of the word features. Then, a compositional algorithm based on recursive neural tensor networks is used to construct sentence vectors based on the word embedding model. Finally, we use the SMOTE algorithm as an oversampling method to generate samples for the minority classes and produce a fully balanced training set. Evaluation results on two quite different tasks show that the feature composition method and the oversampling method are both important in obtaining improved classification results. Our method effectively addresses the data imbalance issue and consequently achieves improved results for both sentiment and emotion classification.

Keywords Sentiment analysis · Emotion classification · Imbalanced training · Word embedding · Semantic compositionality

R. Xu · T. Chen · B. Liu · X. Wang
Shenzhen Engineering Laboratory of Digital Stage Performance Robot, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, Guangdong, China
e-mail: chentao1999@gmail.com

Y. Xia
Research Institute of Information Technology, Tsinghua University, Beijing, China
e-mail: yqxia@tsinghua.edu.cn

Q. Lu (✉)
Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
e-mail: csluqin@comp.polyu.edu.hk

Introduction

With the growing popularity of media-sharing services in what is known as Web 2.0, millions of people share their opinions, sentiments, and emotions over the web. These data provide valuable information for social analysis, commercial promotion, and many other applications. This motivates our research on text-based sentiment and emotion analysis presented here.

The current mainstream algorithms for sentiment and emotion classification are machine learning based classification methods, especially supervised learning algorithms. However, imbalanced training data are obstacles for supervised learning. Skewed training data for different classes lead to class predictions dominated by the larger groups and increased misclassification for underrepresented classes [29, 43]. Many real-world machine learning systems are faced with data imbalance, as it naturally occurs in data. The data imbalance problem is particularly serious in sentiment and emotion classification. Two widely used Chinese emotion corpora serve as typical

examples. The Ren-CECps and NLPCC2013 datasets[1,2] have eight and seven emotion categories, and the training samples corresponding to the largest emotion category are about 10 and 11 times the size of the smallest category, respectively [18]. A similar imbalance occurs in most available sentiment corpora. Imbalanced training data are major obstacles to further improve the performance of sentiment and emotion classification.

Different methods have been proposed to address this problem. Generally speaking, these methods can be divided into three major groups [29]: *algorithmic modification* [42], *cost-sensitive learning* [41], and *data sampling* [17]. The algorithmic modification approach uses an adaptive approach built into the learning methods to address imbalance issues [29]. The cost-sensitive learning approach adds a training penalty cost for majority classes with respect to minority classes. The data sampling approach aims to produce balanced data class distribution by adjusting training data before training. Thus, this final approach is largely independent from the learning methods. Typical data sampling methods include under-sampling, oversampling and hybrid methods [29]. Under-sampling methods eliminate some instances in the majority classes [29], whereas oversampling methods generate new instances for minority classes. When the training dataset is small (as is typical for emotion classification), under-sampling methods further reduce the number of samples, which can be detrimental to the classification performance. Thus, under-sampling is inappropriate in emotion classification. The Synthetic Minority Oversampling TEchnique (SMOTE) [15] is the most renowned oversampling method. SMOTE has a number of variants including Borderline-SMOTE [24], Safe-Level-SMOTE [7], and DBSMOTE [8]. Hybrid methods make combined use of oversampling and undersampling methods [16] which is also not commonly used in opinion analysis.

Among existing solutions to address data imbalance, data sampling is the most practical and applicable to a wider range of applications. However, for many natural language text-based classification tasks, especially for text sentiment and emotion classification, the performance improvement from using oversampling is minimal when features are based on surface forms such as the Bag-of-Words (BOW) representation. This is because BOW can generate thousands of unigram and/or bigram features and the resulting high-dimensional feature space may be sparse and have small disjuncts [27, 41]. Data sparseness is a major hindrance to machine learning classification [5]. Small disjuncts lead to more specific decision regions following oversampling which can lead to over-fitting [8].

[1] http://tcci.ccf.org.cn/conference/2013/dldoc/evsam02.zip.

[2] http://tcci.ccf.org.cn/conference/2013/dldoc/evdata02.zip.

Unlike the BOW representation, word embedding is a distributed representation of words [5]. It uses learning vector representations based on a neural probabilistic language model [32]. Word embedding and its compositional extensions to sentences and documents are expected to alleviate data sparseness and the small disjunct problem since the resulting data are low dimensional, dense, and continuous. In this paper, we propose a novel method based on Word Embedding Compositionality with Minority Oversampling TEchnique (WEC-MOTE), which uses an oversampling approach based on word embedding compositionality to address data imbalance in sentiment and emotion classification. We first use a large corpus to train a continuous skip-gram model [30] to construct word embedding maps. Skip-grams contain word sequence information and can thus better capture semantic information contained in longer lexical units than isolated words. Next, sentence vectors are constructed by a compositional algorithm based on recursive neural tensor network (RNTN) [39]. This models the training sentences using word embedding as features. The resulting sentence vectors are lower in dimension, and thus dense, and consequently are better representing the semantic information of the original sentences. This is particularly significant for capturing sentiment and emotion information. Sentence vectors are then used as the input for oversampling. New sentence vectors corresponding to minority classes are iteratively generated using the SMOTE algorithm. The algorithm takes the size of the largest majority class as a parameter to produces a fully balanced training dataset for all classes.

The proposed WEC-MOTE algorithm is evaluated on two datasets. The first dataset is generated from the Stanford sentiment treebank containing only polarity labels at the sentence level. We purposely selected an imbalanced subset to evaluate our proposed approach. The second set is the NLPCC2013 Chinese micro-blog data with seven discrete emotion labels at the sentence level. The machine learning algorithms we experimented with are the Naive Bayes method and the support vector machine (SVM) method. The baseline system uses BOW as features with no sampling. We also test linear word embedding and compositional word embedding using different oversampling methods. Our proposed method shows a significant improvement over the baseline system. The combined use of word embedding compositionality and SMOTE oversampling is the most effective.

The rest of this paper is organized as follows. "Related Work" section reviews related work. "Word Embedding Compositionality-Based Oversampling" section presents the WEC-MOTE algorithm. "Performance Evaluation" section provides the evaluation results and discussion. "Conclusion and Future Directions" section provides the conclusion and future research directions.

Related Work

Sentiment and Emotion Classification

Cambria et al. divide current sentiment and emotion classification techniques into four major categories: *keyword spotting*, *lexical affinity*, *concept-based*, and *statistic and machine learning* [14].

Keyword spotting classifies text by sentiment categories based on the presence of unambiguous sentiment words. It relies heavily on surface form of word features and is not fully robust when handling negated expressions.

Lexical affinity also detects keywords but goes beyond the aforementioned method by assigning arbitrary words a probable "affinity" to a particular sentiment or emotion. The affinity value and the unambiguous sentiment words are used for classification. The main drawback of this method is that it is domain dependent and thus not typically reusable.

The concept-based approach uses web ontology or semantic networks (e.g., SenticNet [13]) as the knowledge base to analyze the conceptual and affective information associated with opinions in text [20, 22]. The concept-based approach relies on the depth and breadth of the knowledge bases used. Cambria et al. [12] built a knowledge base called SenticSpace which merges commonsense knowledge and affective knowledge to obtain a multi-dimensional vector space. The concept-based approach incorporates commonsense reasoning, which significantly enhanced the emotional intelligence of computer systems [11]. Cambria et al. [10] provided an overview of the past, present, and future efforts of the AI community to endow computers with the capacity for commonsense reasoning.

Statistical and machine learning based approaches prevail in sentiment and emotion classification. They make use of features such as unigrams, bigrams, parts of speech, affect words, information gain (IG), term frequency–inverse document frequency (TF-IDF), mutual information (MI), and the Chi-square statistic (CHI). Classifiers such as support vector machines (SVM) [35], conditional random fields (CRF), hidden Markov models (HMM), Gaussian mixture models (GMM) [37], and neural networks [40] are used for sentiment and emotion classification for text of different forms (e.g., blogs, news, movie reviews, and social media). Generally speaking, statistical methods only work well when sufficient and balanced training data are provided [14]. Furthermore, they are not ideal for semantic representations or text containing implicit sentiment or emotion information.

Data Sampling for Imbalanced Training

Data sampling for imbalanced training data has attracted considerable research interest because its adjustments to the data are largely independent of the machine learning algorithms used and can be applied to a wide variety of domains. Typical data sampling methods include under-sampling, oversampling, and hybrid methods [29].

Under-sampling the majority classes can produce balanced data such that minority classes are not underrepresented [15]. However, it is not commonly used in opinion analysis and emotion classification because the training data are relatively small. Discarding some of this annotated data further reduces available training data, potentially degrading the classification performance.

The simplest oversampling method, random duplication of the minority class samples, is very easy to implement but is generally ineffective [15]. Chawla et al. [15] proposed a synthetic oversampling method, referred to as SMOTE, which generates synthetic samples along a line segment joining an existing sample to its nearest neighbor. Borderline-SMOTE and Safe-Level-SMOTE are two varieties of SMOTE. Borderline-SMOTE only generates new samples among the borderline instances of a minority class [24], whereas Safe-Level-SMOTE only generates new samples for the central instances of a minority class [7]. Since all SMOTE methods use the k-nearest neighbor (kNN) algorithm to find the k-nearest data points in the creation of a new sample, the time complexity of SMOTE is much higher than that of random duplication.

DBSMOTE [8] is a density-based clustering method based on SMOTE. It combines the DBSCAN clustering algorithm [21] and SMOTE to generate a density-reachable graph before performing oversampling [8]. Majority Weighted Minority Oversampling TEchnique (MWMOTE) identifies and weights the hard-to-learn informative minority class samples according to their Euclidean distance to the nearest majority class samples before oversampling [2]. Adaptive synthetic sampling approach for imbalanced learning (ADASYN) [25] uses a weighted distribution for different minority class examples according to their learning difficulty. This adaptively shifts the classification decision boundary toward difficult examples [16].

As an ad hoc method, Nitesh et al. proposed a wrapper-based algorithm to define the best ratio to perform both under-sampling and oversampling [16]. Cai et al. [9] proposed a hybrid learning model using a modified self-organizing maps algorithm. This method assigns a winner neuron based on an energy function minimizing local error in the competitive learning phase.

Learning Distributed Word Representations and Sentence Vector

The distributed representation proposed by Hinton et al. [26] is a low-dimensional float vector for text representation. Distributed representation for words is often referred

to as word representation or word embedding. This kind of representation is effective for capturing a large number of precise syntactic and semantic word relationships [30].

Word embedding is typically induced by neural language models, which use neural networks as the underlying predictive model [3]. Bengio et al. [4] proposed a feed-forward neural network with a linear projection layer and a nonlinear hidden layer to construct a neural language model. This model predicts the current word when the previous $n-1$ words are given. Experimental results show that word embedding decreases ambiguity by 10–20 % compared with smoothed trigram models. The Collobert and Weston (C&W) model [19] is another neural language model based on the syntactic context of words. It substitutes the center word of a sentence by a random word to generate a corrupted sentence as a negative sample. The training objective is to minimize the loss function so that the original sentence can obtain a higher score than the corrupted sentence.

The main drawback of neural probabilistic language models is that both training and testing are time consuming. The hierarchical log-bilinear model introduced by Mnih and Hinton is a fast hierarchical language model which uses a simple feature-based algorithm to automatically construct word trees from the data [33]. In feed-forward networks, the context of a word is limited to a window of n words. Mikolov et al. proposed a recurrent neural network-based language model (RNNLM) [31] in which the context of a word is represented by neurons with recurrent connections such that there is no limit on the context window.

Given a dictionary of word embedding $v = (v_{w_1}, v_{w_2}, \ldots, v_{w_n})$, there are typically two ways to learn the sentence vectors: *linear combination* and *semantic compositionality*. Linear combination, as described in [18], sums all selected word embedding combinations to construct sentence vectors. Since linear combination is concise and efficient, it is used in many applications. However, it cannot capture the generally recursive structure and word order in natural language text.

Semantic compositionality is a linguistic concept. The principle of semantic compositionality is that the meaning of an expression is a function of the meanings of its parts together with a method by which those parts are combined [1]. Compositionality has been extremely influential throughout the history of formal semantics. In fact, recent studies related to cognitive science use compositionality as a guiding principle [36].Various semantic compositionality models have been proposed to use word meanings acquired using co-occurrence statistics of a word and its neighboring words to obtain vectors for longer phrases [6]. One of these models, referred to as the RNTN by Socher et al. [39], is based on neural networks. The RNTN model for semantic compositionality over a sentiment Treebank improved sentence-based polarity classification from 80 to 85.4 %.

Word Embedding Compositionality-Based Oversampling

System Framework Design

In this work, we investigate a data sampling approach using word embedding compositionality to form sentence vectors. As discussed before, the available annotated sentiment and emotion training data are small and under-sampling the majority classes will further reduce training data. We are thus inclined to use oversampling since it increases the training data for the minority classes. In contrast to most existing research on oversampling, which directly generates new samples based on existing samples using a BOW representation, we generate new samples based on word embedding compositionality. Sentence vectors produced through word embedding compositionality are expected to overcome the data sparseness and small disjunct problems encountered when using imbalanced training data with the BOW representation.

The system framework has three main components as shown in Fig. 1. The first component, the *word embedding construction module*, takes a large collection of raw text to train the word embedding model and produce word embedding list. Based on this list, the second component, the *word embedding compositionality algorithm*, takes the training data (presumably imbalanced) to construct the corresponding sentence vectors based on the RNTN model. The third component, the *WEC-MOTE algorithm*, generates a balanced training dataset. The machine learning based classifier for either sentiment or emotion classification can be trained accordingly.

Normally, a sentence vector based on BOW will form a binary feature space (e.g., $[0, 1, \ldots, 0, 1, 1, 0, \ldots, 0, 0]$). Let N be the size of the vocabulary. The dimension of BOW using unigrams is linear with respect to N. The dimension of BOW using bigrams is proportional to N^2. Most of the values in the vector are zero rendering the data extremely sparse. In contrast, a word embedding-based sentence vector forms a real-valued feature space (e.g., $[0.022506, -0.077435, \ldots, 0.014368, -0.185020]$). The dimension of the sentence vectors depends on the application (and is controllable) and is typically between 25 and 300, and zeros are not expected for any of the features, making this data much denser. The use of word embedding also captures more semantic information from a sentence compared with the BOW representation.

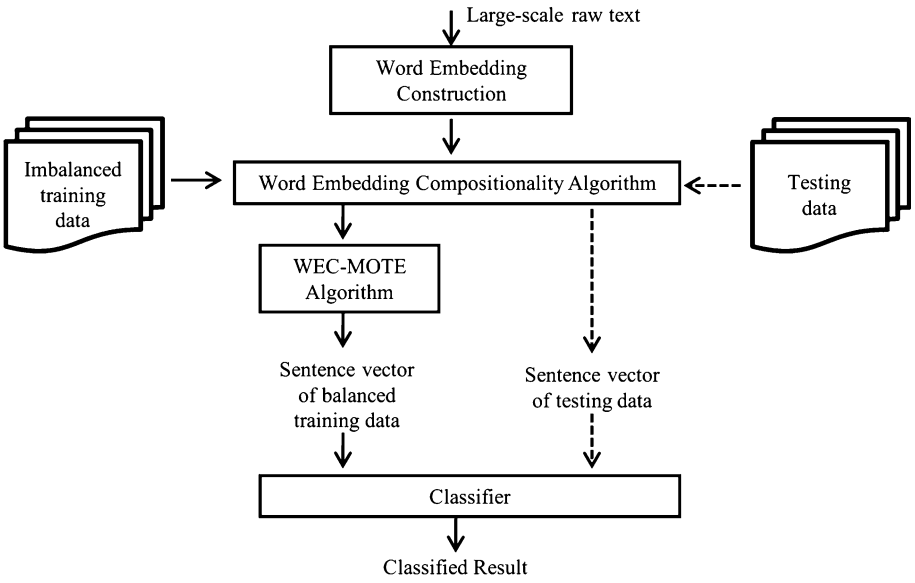

Fig. 1 Framework of our approach

RNTN-Based Word Embedding Composition Algorithm for Sentence Vector Construction

Mikolov et al. [30] introduced the continuous skip-gram model to learn vector representations capturing a large number of syntactic and semantic word relationships from unstructured text data. The training objective is to find word representations to predict the surrounding words in a sentence or a document. Given a sequence of training words $w_1, w_2, w_3 \ldots w_T$, the training objective is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq i\leq c, i\neq 0}\log p(w_{t+i}|w_t) \tag{1}$$

where c is the size of the training context, w_t is the center word, and $\log p(w_{t+i}|w_c)$ is the conditional log probability of w_{t+i} given the center word w_t. The *hierarchical softmax* process is used to reduce computational complexity. It uses a binary tree representation of the output layer with the words as leaves. Accordingly, $p(w_{t+i}|w_t)$ is defined as:

$$p(w_{t+i}|w_t) = \prod_{j=1}^{L(w_{t+i})-1}\sigma(\| n(w_{t+i}, j+1) = \mathrm{ch}(n(w_{t+i}, j)) \| \upsilon'_{n(w_{t+i},j)}{}^{\upsilon_{w_t}}) \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$, $\| x \| = \begin{cases} 1, & \text{if } x \text{ is true} \\ -1, & \text{else} \end{cases}$. $n(w,j)$ is the jth node on the path from the root to w. $L(w)$ is the length of this path. $ch(n)$ is an arbitrary fixed child of n. v'_n is the representation of inner node n. v_w is the representation of word w.

Since a Chinese word having different part of speech (POS) tags serve different lexical functions, we use the word/POS pair as the basic lexical unit in word embedding. For English word embedding, many pre-trained word embedding resources are already available. Collobert and Weston embedding are provided by senna[3] and several kinds of word embedding are provided on the word2vec Web site. Thus, there is no need to generate the data by ourselves. The large free text used to obtain the Chinese word embedding is a large-scale micro-blogging corpus in Chinese. The word2vec[4] toolkit accessible from Google is use to build the skip-grams. The dimension of each word vector is a system parameter set for specific applications and ranges from tens to thousands in previous work [30]. We use the X-d notation where X denotes the dimension size (200-d means a dimensionality of 200).

The RNTN proposed by Socher et al. [39] takes a sentence as input. It represents a sentence through word embedding and a parse tree. The vectors for higher nodes in the parse tree are computed by using the same tensor-based composition function.

As shown in Fig. 2, when a sentence is fed to the RNTN model, it is parsed into a binary tree where each leaf node is a word embedding corresponding to a word in the sentence. Let us assume the dimension of a vector, denoted by d, is fixed. Then $v_{w_1}, v_{w_2}, v_{w_3}, v_{w_4}$ should be represented by d-dimensional vectors using word embedding. The RNTN model computes parent vectors y_i from the bottom up using compositionality functions f. Here, the tangent function $f = tanh$ is used a common choice for element-wise nonlinearity. y_i has the same dimension d. $[v_{w_i}; v_{w_j}]$ is the concatenation of two leaf node word embeddings. $W \in \Re^{d\times 2d}$

[3] http://ml.nec-labs.com/senna/.

[4] https://code.google.com/p/word2vec/.

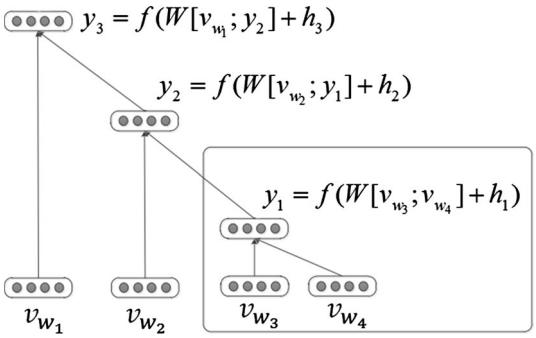

Fig. 2 An illustration of the recursive neural tensor network

is the main parameter for the neural network to learn. Here, the tensor product $h \in \Re^d$ is defined as follows:

$$h_k = [v_{w_i}; v_{w_j}]^T V^{[1:d]} [v_{w_i}; v_{w_j}] \tag{3}$$

where $V^{[1:d]} \in \Re^{2d \times 2d \times d}$ is the tensor that defines multiple bilinear forms. RNTN uses the following equation to compute y_1:

$$y_1 = f(W[v_{w_3}; v_{w4}] + [v_{w_3}; v_{w_4}]^T V^{[1:d]} [v_{w_3}; v_{w_4}]) \tag{4}$$

The next parent vector y_2 is then computed as follows:

$$y_2 = f(W[v_{w_2}; y_1] + [v_{w_2}; y_1]^T V^{[1:d]} [v_{w_2}; y_1]) \tag{5}$$

Each node in the binary tree is computed recursively until the root node vector representing the meaning of the given sentence is obtained.

A softmax classifier is trained on each of the RNTN node vectors to predict a given target vector. The back-propagation algorithm is used to train the RNTN model. Each node's error is back-propagated to the recursively used weights V and W [39]. The full derivative for V and W is the sum of the derivatives at each of the nodes.

The sentence vectors for a training corpus are constructed by using a RNTN-based composition algorithm. The results using real data show that word embedding-based sentence vectors reduce data sparseness, cluster better between classes, and have fewer small disjuncts within classes when compared to the BOW representation.

Figures 3 and 4 show the two-dimensional principal components analysis (PCA) projection of the word embedding sentence-vector distributions and the BOW distributions using 10 % random sampling, respectively. The training set is the Stanford sentiment treebank [39]. Figure 3 shows a fairly even spread of the data points across the projection space. Data points from different classes are partitioned naturally such that the negative class (circles) is mainly distributed in the left part of the projection space, the positive class (stars) largely in the right part, and the neutral class (dash) mainly in the center. In contrast, the BOW projection in Fig. 4 results in clusters rather than even spreading. Furthermore, the clusters are not aligned well to the different clusters. Figure 5 gives a micro-view of the top left part of Fig. 4. It shows that the negative samples, positive samples, and neutral samples are all mixed together. This entails that most of the nearest neighbors for any sample may belong to different classes when using BOW.

WEC-MOTE for Sentiment and Emotion Classification

Our WEC-MOTE algorithm is derived from the SMOTE method to generate new samples for the training data for minority classes. The main idea of WEC-MOTE is to interpolate several nearby minority class instances to create new examples for minority classes [29] producing a fully balanced training set across all classes.

For a given minority class where each real sample S corresponds to a sentence vector V_S, a new synthetic sample V_{new} can be generated based on the formula given in Eq. (6). V_{new} is derived from V_S and V_{S_N}, which is one of the k -nearest neighbors of V_S [15].

$$V_{\text{new}} = V_S + R_{0-1} \times V_{s_N} \tag{6}$$

where R_{0-1} is a random number between zero and one. The value k is an algorithm parameter specifying the over-sampling rate. Since R_{0-1} is a real number between zero and one, the algorithm can generate as many different V_{new} as needed. Let M denote the number of training samples of the largest class. For any class c, let N denote its number of training samples where N is less than M. Then, sentence vectors V_s in the safe region [24]: where less than half of the k -nearest neighbors are negative samples, see "Comparison Methodology" section) of class c can be randomly selected. New samples can be randomly generated according to Eq. (6) for class c until there are M samples in total. k is defined as the imbalance ratio of the size of class c and that the largest class: $k = M/N$.

Figure 6a is the vector illustration of three training sentences which have the same distance between each other using the BOW representation. Figure 6b shows the vectors resulting from word embedding. Figure 6a illustrates that the neighbors of "I hate it" belong to c'. Thus, "I hate it" is regarded as a small disjunction in class c. The new sample derived from "I love it" and "I like it" for the BOW representation is closer to the training data of "I hate it" than "I like it". This is because in BOW, data of the same class do not cluster well. Conversely, word embedding results in a new sample which is closer to the samples that generated it due to better class clustering.

The proposed WEC-MOTE algorithm is applicable to many imbalanced text classification systems whether they involve single-label or multi-label classification or sentiment or emotion classification. Data imbalances are quite common in sentiment corpora since there are often more

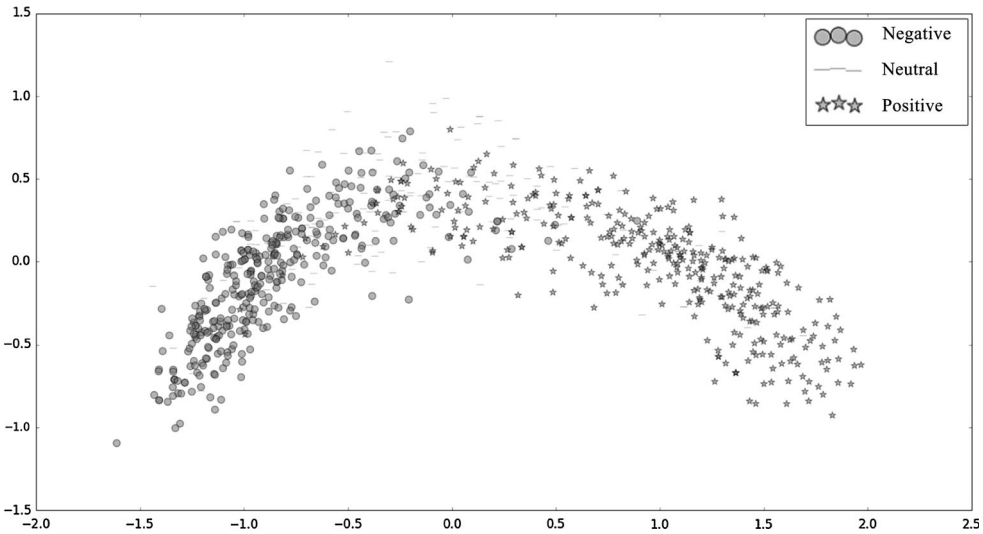

Fig. 3 PCA projection of word embedding sentence vectors trained on Stanford sentiment treebank

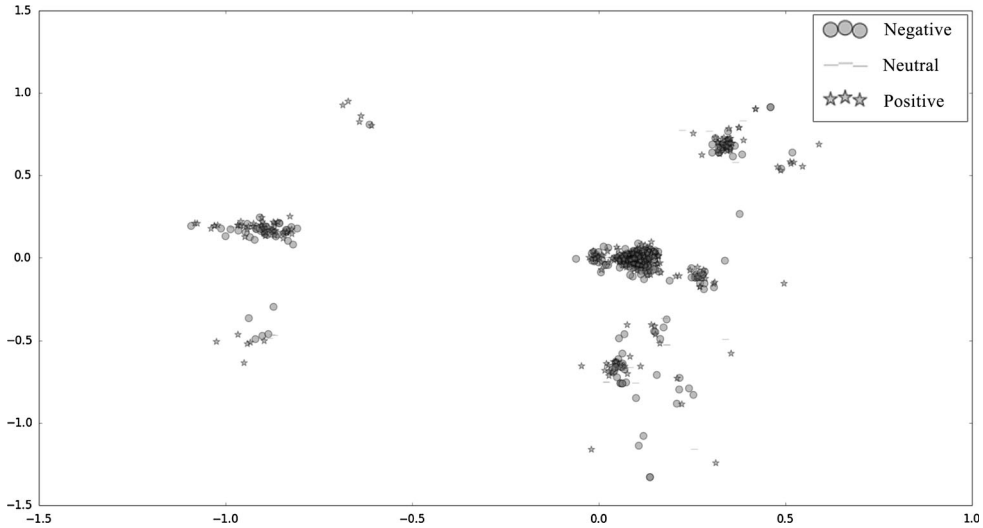

Fig. 4 PCA projection of BOW representation trained on Stanford sentiment treebank

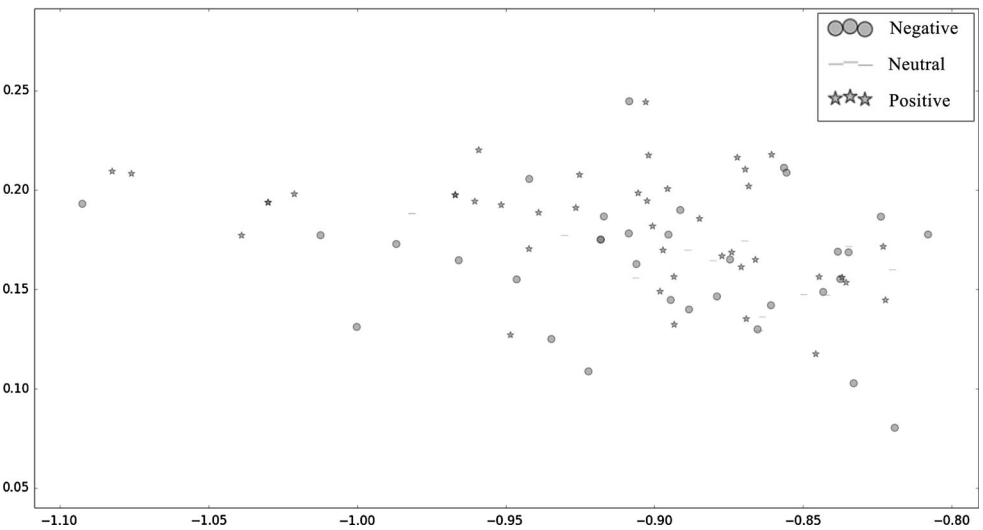

Fig. 5 Enlarged view of the top left portion of PCA projection of BOW representation (Fig. 4)

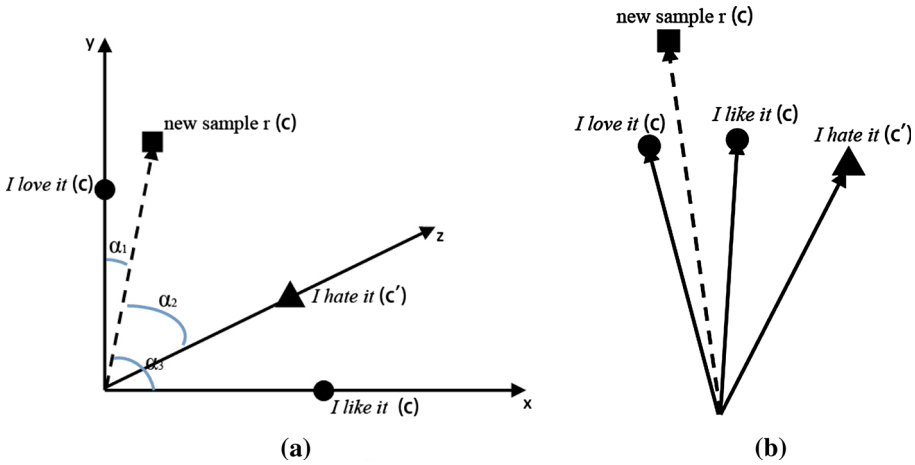

Fig. 6 An illustration of a new sample generated by the SMOTE algorithm using **a** BOW representation and **b** word embedding

positive sentences than negative sentences and *like* and *disgust* samples are much more common than *fear* and *surprise* samples. Using the WEC-MOTE algorithm on these corpora before applying a machine learning method can improve the classification performance significantly.

Performance Evaluation

In this section, the performance of our proposed WEC-MOTE algorithm is evaluated on two datasets, namely an English sentiment analysis dataset (single label) and a Chinese emotion classification (multi-label), respectively.

Dataset

The English Sentiment Corpus

The single-label dataset is from the Stanford sentiment treebank proposed by Socher et al. [39]. The treebank provides fully labeled parse trees based on the dataset introduced by Pang and Lee [34]. This dataset is quite balanced with 11,855 single sentences extracted from movie reviews comprised of three subsets: the training data, the development data, and the testing data. To obtain the imbalanced dataset for evaluation purpose, we combine the training data and the development data. We then discard the neutral sentences. Finally, we randomly remove negative sentences to achieve an imbalance ratio of 3.5. For our evaluation, the training data has 4,054 positive sentences and 1,158 negative sentences. The test dataset has 909 positive sentences and 912 negative sentences.

The Chinese Emotion Corpus

The NLPCC2013 Chinese micro-blog emotion classification dataset is used as the multi-label classification dataset. The dataset includes seven emotion categories: *like*, *disgust*, *happiness*, *anger*, *sadness*, *surprise*, and *fear*. Since natural language text may express complex emotions in a sentence, each sentence in this dataset is labeled by a primary and a secondary emotion category. Details of the distribution of the NLPCC2013 dataset are listed in Table 1. Note that in the training set, the data distribution is quite skewed. The majority class, *like*, is about four times the size of the *surprise* class and 11 times the size of *fear*.

Parameter Settings

Performance Metric

When data show a high degree of imbalance, traditional empirical measures such as accuracy rate are no longer appropriate [29] since correctly classifying all examples corresponding to the majority class will achieve a high accuracy rate despite misclassifying minority classes. For sentiment analysis, the *geometric mean*, G_{mean}, as defined in Eq. (7), is a more appropriate performance measure.

$$G_{\text{mean}} = \sqrt{\text{TP}_{\text{rate}} \times \text{TN}_{\text{rate}}} \tag{7}$$

where TP_{rate} is the true positive rate signifying the percentage of correctly classified positive examples and TN_{rate} is the true negative rate signifying the percentage of correctly classified negative examples.

Since there are more than two classes in emotion classification, geometric mean is not applicable. The common measure *average precision* is used in NLPCC2013's multi-class multi-labeled emotion classification evaluation.[5] Average precision takes the weighted precision for different classes of data. Since NLPCC2013 data label sentences by both primary and secondary classes, NLPCC2013 requires

[5] http://tcci.ccf.org.cn/conference/2013/dldoc/evres02.

Table 1 Emotional class distribution in NLPCC2013 dataset

Class	Training set				Testing set			
	Primary emotion		Secondary emotion		Primary emotion		Secondary emotion	
Like	1,226	24.8%	138	21.6%	2,888	27.6%	204	26.1%
Disgust	1,008	20.4%	187	29.2%	2,073	19.8%	212	27.1%
Happiness	729	14.7%	95	14.8%	2,145	20.5%	138	17.6%
Anger	716	14.5%	129	20.2%	1,147	10.9%	82	10.5%
Sadness	847	17.1%	45	7.0%	1,565	14.9%	84	10.7%
Surprise	309	6.2%	32	5.0%	473	4.5%	43	5.5%
Fear	114	2.3%	14	2.2%	186	1.8%	20	2.6%

two additional measures depending on the importance of the secondary class. The *loose measure* tracks the classification of the primary and secondary classes equally such that one point is awarded for each correct class label whether it corresponds to the primary or secondary class. The *strict measure* introduces a weighting such that a correct secondary label receives half the score of a correct primary label.

Pre-trained Word Embedding for RNTN Model

For sentiment analysis, a 25-d English word embedding is constructed from a snapshot of Wikipedia in April 2010 provided by the Westbury Lab[6] [38]. This snapshot contains about 990 million words in over 2 million documents. The sentence vector for the English training set is obtained using the RNTN model. The Stanford sentiment treebank, obtained manually through Amazon Mechanical Turk, 215,154 phrases labeled 10,662 sentences. The Chinese word embedding is obtained from a 4.29-billion word Chinese micro-blog corpus from weibo.com using the continuous skip-gram model. The word/POS pair is used as the basic lexical unit. Empirically, the maximum skip length between words is set to five and the threshold for word occurrences is set to 10^{-5}. The minimum word frequency is set to four, the initial learning rate is set to 0.025, and the number of negative examples is set to five. The vector dimension is set to 200-*d* and 25-*d* for the RNTN linear model and the RNTN compositional model, respectively. 399,059 word embedding models are thus obtained.

To train the RNTN model on the NLPCC2013 dataset, a Chinese sentiment treebank is required. Since a Chinese sentiment/emotion treebank similar to Stanford's English sentiment treebank is not available, we prepared the data ourselves. We first parsed each of the training sentences into a binary tree using the Stanford Chinese PCFG parser [28] and the sentiment model in Stanford CoreNLP tools.[7] We then labeled each node of the binary tree with the emotion class describing the sentence. Eight classes are used to build the RNTN, and the learning rate is 0.01.

Classifiers

For sentiment analysis, we explore two base classifiers: Naive Bayes (NB) and support vector machines (SVM). For NB, we use the default parameters in the Weka software tool [23]. For SVM, we use the radial basis function: $e^{-\gamma|u-v|^2}$. We use a cost parameter of 1.0 for C-SVC. The full parameter list is "-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1", again, the default used in Weka.

For the multi-label multi-class task of emotion classification, we train a multi-label *k*-nearest Neighbor (ML-*k*NN) classifier using the original and balanced training data. We test the hyper-parameter k with every odd number from 1 to 99 and find $k = 41$ brings the best classified result. So empirically, $k = 41$ is used in this evaluation. The nearest-neighbor similarity between two sentences is estimated by the cosine of the angle between their corresponding sentence vectors:

$$\text{similarity}(S_i, S_j) = \cos(\theta) = \frac{V_{S_i} \cdot V_{S_j}}{||V_{S_i}|| \times ||V_{S_j}||} \tag{8}$$

where V_{S_i} and V_{S_j} are sentence vector of sentence S_i and S_j, respectively.

Comparison Methodology

We evaluate the proposed WEC-MOTE algorithm with different sentence representations and oversampling methods. For sentence representation, we use unigram BOW as the baseline. We also use the linear combination word embedding algorithm (WE linear) proposed by [18] for comparison. We use the *duplicate instances method* as the baseline oversampling method. *Safe-Level-SMOTE* and *Borderline-SMOTE* are also used for comparison. In this study, we use the definition of safe samples given by Han et al. [24]. The minority samples are grouped into three

[6] http://www.psych.ualberta.ca/~westburylab/downloads/westburylab.wikicorp.download.html.

[7] http://nlp.stanford.edu/software/corenlp.shtml.

Table 2 Evaluation results for English sentiment dataset

Method	Geometric mean (BOW unigram)		Geometric mean (WE linear)		Geometric mean (WE composition)	
	NB	SVM	NB	SVM	NB	SVM
Original training set	0.361	0.618	0.422	0.486	0.824	0.827
Duplicating instances	0.606	0.650	0.636	0.650	0.828	0.828
Borderline-SMOTE	0.460	0.657	0.484	0.640	0.807	0.814
Safe-Level-SMOTE	0.512	0.645	0.486	0.639	0.829	0.829
WEC-MOTE	0.480	0.658	0.491	0.644	0.830	**0.833**

Bold value indicates the best performance of all the experiment results

regions: borderline, safe, and noise regions by considering the number of negative samples n in k -nearest neighbors: $k/2 \leq n < k$ for borderline samples, $0 \leq n < k/2$ for safe samples, and $n = k$ for noise samples.

Evaluation for Single-Labeled English Sentiment Data

Table 2 shows the result for the selected two-class imbalanced Stanford sentiment treebank corpus. WEC-MOTE achieves the best classification performance for both the NB classifier and the SVM classifier. We first examine the different data models without considering any sampling methods (first row in Table 2). Even without sampling, data representation accounts for significant performance variations. Compared with the BOW baseline data representation, both word embedding (WE) composition and WE linear achieve superior performance. Furthermore, all four oversampling methods used with WE composition lead to superior performance compared with both the BOW unigram representation and the WE linear representation. For example, WEC-MOTE paired with WE composition achieves geometric mean classification accuracy values of 0.830 (NB) and 0.833 (SVM). WEC-MOTE paired with the BOW unigram representation achieves only a 0.480 (NB) and 0.658 (SVM) geometric mean accuracy. This translates to a relative improvement of 72.9 and 26.6 % for NB and SVM, respectively, just by using the WE composition representation rather than BOW unigram.

Comparing different oversampling methods, the duplication method is most effective when paired with the BOW unigram or WE linear representations. Still, the overall best performance when using this oversampling method is only 0.65. In contrast, WE composition leads to an overall accuracy rate of at least 0.807 regardless of the sampling method. In other words, the results in Table 2 indicate that word embedding with composition is most effective for document representation. The performance improvements achieved by the different oversampling methods are minor compared to the effect of changing the data representation.

Table 3 gives a more detailed analysis of the different sampling algorithms using only WE composition. The first row shows the results obtained without oversampling, and the rest of the rows are the results with different oversampling methods. The results show that compared with SVM, NB is less effective for true positive classification and generally more effective for true negatives. Overall, WEC-MOTE paired with the SVM classifier achieves the highest weighted average accuracy rate.

Figuers 7 and 8 show the PCA projection (using 10 % random sampling) of the sentence vectors without and with WEC-MOTE oversampling, respectively. The samples that are generated through oversampling are marked by stars in Fig. 8. We can see from both figures that WE composition leads to well-defined clusters and a very good partition between the two classes with the positive samples mainly scattered over the right region and the negative ones over the left region. Figure 8 shows that the new negative samples generated by WEC-MOTE oversampling are usually close to the original samples. This indicates that WEC-MOTE generally maintains the same distribution as the original samples while rendering denser data to resolve the small disjuncts problem.

Evaluation for Chinese Multi-label Emotion Data

Table 4 shows the multi-class emotion classification performance results for Chinese micro-blog text. This set of results shows that WEC-MOTE achieves the highest average precision of 0.520 (loose) and 0.497 (strict). Compared with the top performers of the 19 submitted systems in the NLPCC2013 evaluation, which achieved 0.365 (loose) and 0.348 (strict)[8] average precisions, our approach achieves a very significant 42.5 % (loose) and 42.8 % (strict) relative improvement.

Regardless of the sampling method, word embedding with both the linear and the composition methods leads to an improvement of more than 100 % over BOW. WE composition does not show a significant advantage over the linear RNTN model. The significant improvement gained from using word embedding instead of BOW is due to BOW unigram features having 25,229 dimensions and word embedding features having only 25 dimensions.

[8] http://tcci.ccf.org.cn/conference/2013/dldoc/evres02.

Table 3 Performance for each sentiment class using WE composition

Method	TP_{rate}		TN_{rate}		Weighted average	
	NB	SVM	NB	SVM	NB	SVM
Original training set	0.794	0.890	0.855	0.768	0.825	0.829
Duplicating instances	0.779	0.810	0.880	0.846	0.830	0.828
Borderline-SMOTE	0.705	0.743	0.924	0.892	0.815	0.817
Safe-Level-SMOTE	0.778	0.836	0.884	0.823	0.831	0.830
WEC-MOTE	0.817	0.812	0.843	0.855	0.830	**0.834**

Bold value indicates the best performance of all the experiment results

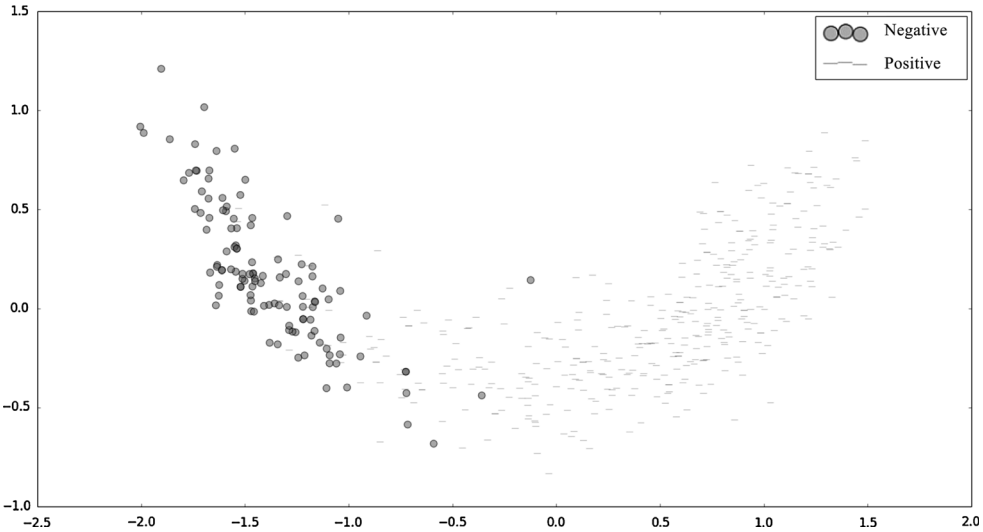

Fig. 7 PCA projection without oversampling for Stanford sentiment treebank

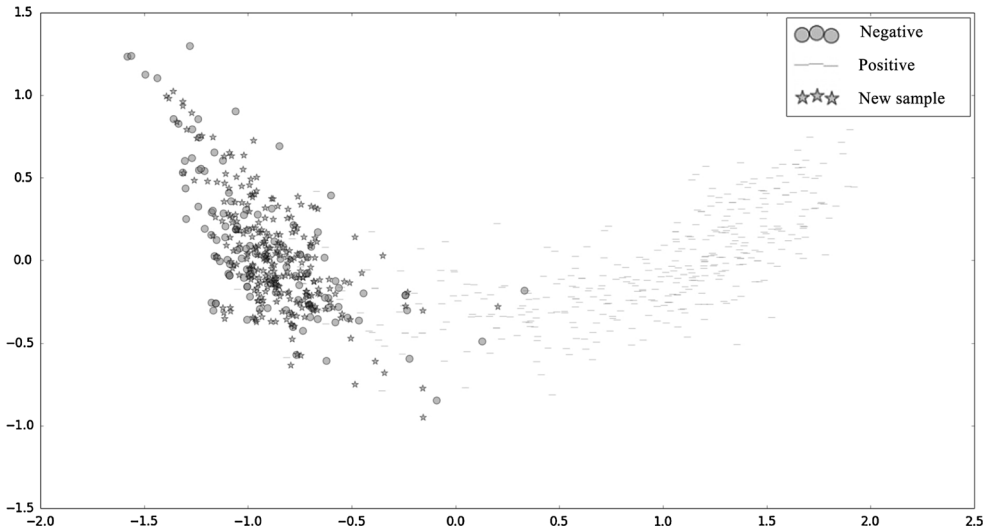

Fig. 8 PCA projection with oversampling for Stanford sentiment treebank

Since there are only 4,949 emotional sentences in the training set, using 25,229 dimensions as features in BOW makes the training data too sparse. Word embedding has a much lower dimensionality and the data show improved clustering effect when using RNTN.

A preliminary finding in this experiment suggests that sufficient training of the RNTN model is imperative when using word embedding trained with a continuous skip-gram. If training is insufficient, a usable vector (i.e., a vector which is float valued and does not suffer from the vanishing gradient problem) in the root node of a sentence parse tree is not guaranteed. This is especially true for long sentences. Thus, we use the mean vector of two usable top-node vectors instead. When the RNTN model is not fully

Table 4 Emotion classification results with different sentence representation and oversampling methods

Method	Average precision (BOW unigram)		Average precision (WE linear)		Average precision (WE composition)	
	Loose	Strict	Loose	Strict	Loose	Strict
Original training set	0.144	0.141	0.334	0.325	0.347	0.339
Duplicating instances	0.158	0.154	0.383	0.369	0.398	0.380
Borderline-SMOTE	0.175	0.167	0.416	0.397	0.433	0.409
Safe-Level-SMOTE	0.275	0.264	0.434	0.420	0.449	0.437
WEC-MOTE	0.185	0.177	0.501	0.478	**0.520**	**0.497**

Bold values indicate the best performance of all the experiment results

Table 5 Performance for each emotion class before/after oversampling

Class	Average precision		Imbalance ratio	Improvement ratio
	Original	Balanced		
Like	0.424	0.649	1.000	0.531
Disgust	0.256	0.552	1.216	1.156
Happiness	0.280	0.373	1.682	0.332
Anger	0.208	0.355	1.712	0.707
Sadness	0.413	0.612	1.447	0.482
Surprise	0.009	0.136	3.967	14.111
Fear	0.024	0.283	10.754	10.797

trained, the classification performance using WE composition degrades to equal the WE linear performance.

When the BOW unigram method is used to obtain the features, Safe-Level-SMOTE oversampling leads to the best performance. In fact, for this feature set, it is considerably better than WEC-MOTE and Borderline-SMOTE. However, WEC-MOTE leads to superior performance when word embedding is used. Generally speaking, borderline samples contain more negative instances than safe samples. This implies that there are more small disjuncts for borderline samples than for safe samples. So, there must be more small disjuncts in BOW features than in word embedding features. The newly generated samples from WEC-MOTE reduce small disjuncts such that classification performance is significantly improved.

While sentence representation is crucial to performance improvement as seen from the original data without any oversampling, It is important to note that the use of oversampling is more significant for multi-label emotion classification compared with binary sentiment classification. This is because the number of training samples for the emotion classification is much smaller than for sentiment classification and thus oversampling plays a more important role in the overall performance improvement.

Table 5 shows the class-by-class classification results using the original training data as well as the balanced training data generated by WEC-MOTE. It is obvious that the performance for every class is improved using the fully balanced training data. Yet, the important contribution of the algorithm is that the improvement for the minority classes is more significant. As an example, the two classes *surprise* and *fear* show improvement factors of about 15 and 12 times, respectively, whereas the improvement for the majority class *like* is 53.1 %. Table 5 is a clear indication that the proposed oversampling method is very effective when data are more imbalanced.

Figure 9 shows the LDA projection of sentence vectors from the largest class *like* and the smallest class *fear* without WEC-MOTE oversampling. In contrast, Fig. 10 shows the LDA projection of sentence vectors from the largest class *like* and the smallest class *fear* with WEC-MOTE oversampling for the minority class *fear*. The newly generated samples are denoted by stars. Similar to the distribution of data in Figs. 7 and 8 for sentiment analysis, the two classes are clustered well and the additional samples introduced in the *fear* class create more density and decrease small disjuncts within that class.

Discussion

Our proposed method shows a greater performance improvement for the multi-class emotion classification than for the binary sentiment classification. This is because multi-class emotion data are more skewed and thus oversampling is more effective. In terms of performance in absolute values, the binary classification task yields much better results as the overall performance levels are over 80 %, yet the best performance for the multi-class data is

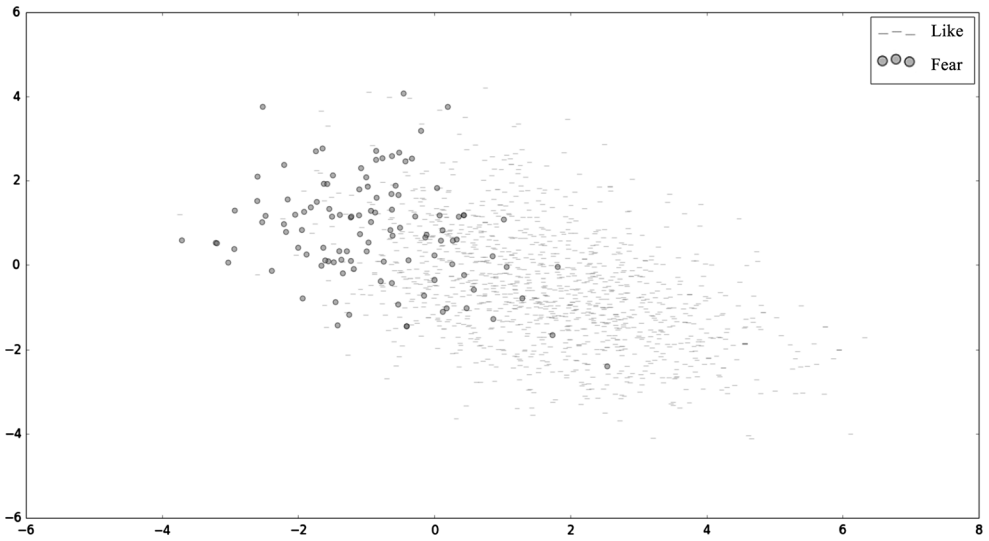

Fig. 9 LDA projection of *like* and *fear* classes without oversampling

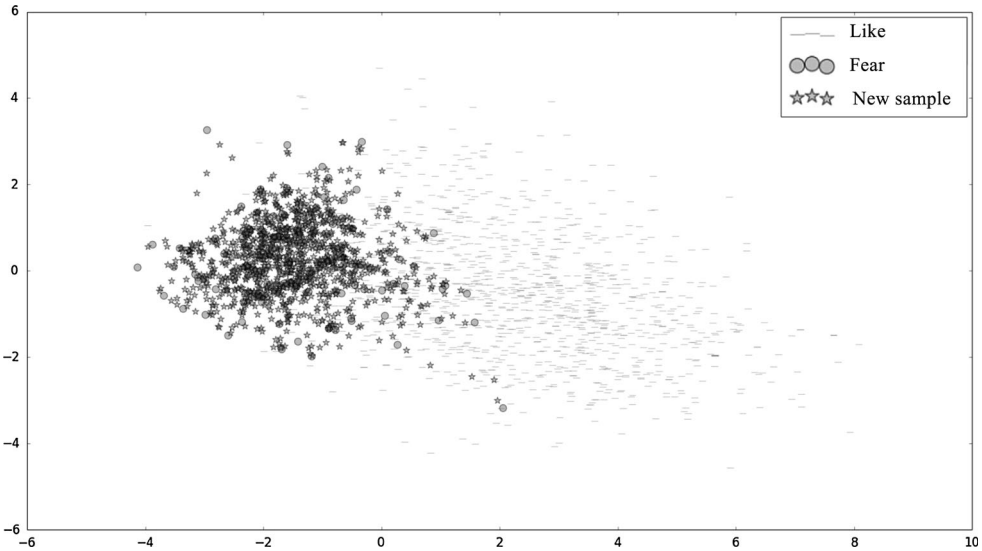

Fig. 10 LDA projection of *like* and *fear* classes after WEC-MOTE oversampling

only about 50 %. This is because there are many more training instances in the binary data.

Due to the lack of training data in the form of a Chinese emotion Treebank to train the RNTN model, we have to label every node of the binary tree with the same emotion as the root node of the tree (the class corresponding to the entire sentence). Experimental results show that compared with the word embedding linear representation, word embedding composition trained on NLPCC2013 dataset shows an only 3.79 % improvement (for WEC-MOTE loose). This is in contrast to the 29.3 % improvement (for WEC-MOTE SVM) when training on the Stanford sentiment treebank corpus. We hope to continue further annotation of related Chinese corpora. The increased availability of such resources will greatly benefit the use of machine learning methods over other methods.

Conclusion and Future Directions

This paper presents an oversampling approach based on sentence vector learning from word embedding to improve sentiment and emotion classification for imbalanced data. Sentence vectors derived through word embedding composition are dense and continuous and have low dimensionality compared with the BOW representation. Word embedding composition is a superior data representation method for machine learning when data are sparse and skewed. This paper also proposes a recursive neural tensor network-based composition method to construct sentence vectors corresponding to training samples. To address the data imbalance issue and the lack of training data for minority classes, we use a SMOTE algorithm to oversample the minority classes to produce a fully balanced

training dataset. Machine learning classifiers can then be trained on this fully balanced training dataset with improved prediction outcomes. Evaluations on two different datasets show that the proposed method is very effective in overcoming data sparseness and small disjuncts present in imbalanced sentiment and emotion classification. Our approach improves imbalanced sentiment and emotion classification for both English and Chinese data and for both single- and multi-label data.

Due to the complexity of the RNTN, the sentence vectors in this paper are constructed with only 25 dimensions. Improvements to the RNTN model should allow us to increase the number of dimensions and further improve overall performance. Other potential future works include word embedding in terms of semantic representation and the development of other composition methods for sentence vector construction.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 61300112, 61370165, 61203378), Natural Science Foundation of Guangdong Province S2013010014475, MOE Specialized Research Fund for the Doctoral Program of Higher Education 20122302120070, Open Projects Program of National Laboratory of Pattern Recognition, Shenzhen International Co-operation Research Funding GJHZ20120613110641217, Shenzhen Development and Reform Commission Grant No.[2014]1507, Shenzhen Peacock Plan Research Grant KQCX20140521144507925 and Baidu Collaborate Research Funding.

References

1. Allan K. Linguistic meaning, vol. 2. London & New York: Routledge & Kegan Paul; 1986.
2. Barua S, Islam M, Yao X, Murase K, et al. MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans Knowl Data Eng. 2014;26(2):405–25.
3. Bengio Y. Neural net language models. Scholarpedia. 2008;3(1):3881.
4. Bengio Y, Ducharme R, Vincent P. A neural probabilistic language model. J Mach Learn Res. 2003;3:1137–55.
5. Bengio Y, Schwenk H, Senécal JS, Morin F, Gauvain JL. Neural probabilistic language models. In: Innovations in machine learning. Berlin: Springer; 2006. p. 137–86.
6. Blunsom P, Grefenstette E, Kalchbrenner N, et al. A convolutional neural network for modelling sentences. In: Proceedings of ACL. 2014.
7. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Advances in knowledge discovery and data mining. Berlin: Springer; 2009. p. 475–82.
8. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. DBSMOTE: density-based synthetic minority over-sampling technique. Appl Intell. 2012;36(3):664–84.
9. Cai Q, He H, Man H. Imbalanced evolving self-organizing learning. Neurocomputing. 2014;133:258–70.
10. Cambria E, Hussain A, Havasi C, Eckl C. Common sense computing: from the society of mind to digital intuition and beyond. In: Biometric ID management and multimodal communication. Berlin: Springer; 2009. p. 252–59.
11. Cambria E, Hussain A, Havasi C, Eckl C. Sentic computing: exploitation of common sense for the development of emotion-sensitive systems. In: Development of multimodal interfaces: active listening and synchrony. Berlin: Springer; 2010. p. 148–56.
12. Cambria E, Hussain A, Havasi C, Eckl C. SenticSpace: visualizing opinions and sentiments in a multi-dimensional vector space. In: Knowledge-based and intelligent information and engineering systems. Berlin: Springer; 2010. p. 385–93.
13. Cambria E, Olsher D, Rajagopal D. SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. AAAI. 2014;1515–21.
14. Cambria E, Schuller B, Xia Y, Havasi C. New avenues in opinion mining and sentiment analysis. IEEE Intell Syst. 2013;28(2):15–21.
15. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. Smote: synthetic minority over-sampling technique. J Artif Intell Res. 2002;16:321–57.
16. Chawla NV, Cieslak DA, Hall LO, Joshi A. Automatically countering imbalance and its empirical relationship to cost. Data Mining Knowl Discov. 2008;17(2):225–52.
17. Chawla NV, Japkowicz N, Kotcz A. Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl. 2004;6(1):1–6.
18. Chen T, Xu R, Lu Q, Liu B, Xu J, Yao L, He Z. A sentence vector based over-sampling method for imbalanced emotion classification. In: Computational linguistics and intelligent text processing. Berlin: Springer; 2014. p. 62–72.
19. Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of ICML. ACM; 2008. p. 160–7.
20. Das D, Bandyopadhyay S. Sentence-level emotion and valence tagging. Cogn Comput. 2012;4(4):420–35.
21. Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD. 1996. p. 226–31.
22. Grassi M, Cambria E, Hussain A, Piazza F. Sentic web: a new paradigm for managing social media affective information. Cogn Comput. 2011;3(3):480–9.
23. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The weka data mining software: an update. ACM SIGKDD Explor Newsl. 2009;11(1):10–8.
24. Han H, Wang WY, Mao BH. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in intelligent computing. Berlin: Springer; 2005. p. 878–87.
25. He H, Bai Y, Garcia EA, Li S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of IJCNN. IEEE; 2008. p. 1322–8.
26. Hinton GE. Learning distributed representations of concepts. In: Proceedings of CogSci, vol 1. Amherst, MA; 1986. p. 12.
27. Jo T, Japkowicz N. Class imbalances versus small disjuncts. ACM SIGKDD Explor Newsl. 2004;6(1):40–9.
28. Levy R, Manning C. Is it harder to parse chinese, or the chinese treebank?. In: Proceedings of ACL, vol 1. ACL; 2003.p. 439–46.
29. López V, Fernández A, García S, Palade V, Herrera F. An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf Sci. 2013;250:113–41.
30. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
31. Mikolov T, Karafiát M, Burget L, Cernockỳ J, Khudanpur S. Recurrent neural network based language model. In: Proceedings of INTERSPEECH. 2010. p. 1045–8.
32. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality.

In: Advances in neural information processing systems. 2013. p. 3111–9.
33. Mnih A, Hinton GE. A scalable hierarchical distributed language model. In: Advances in neural information processing systems. 2009. p. 1081–8.
34. Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics; 2005. p. 115–24.
35. Pang B, Lee L, Vaithyanathan S. Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of EMNLP, vol 10. ACL; 2002. p. 79–86.
36. Pelletier FJ. The principle of semantic compositionality. Topoi. 1994;13(1):11–24.
37. Přibil J, Přibilová A. GMM-based evaluation of emotional style transformation in czech and slovak. Cogn Comput. 2014;6(4): 928–939.
38. Shaoul C. The westbury lab wikipedia corpus. Edmonton: University of Alberta; 2010.
39. Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of EMNLP. CiteSeer; 2013. p. 1631–42.
40. Sun R. Moral judgment, human motivation, and neural networks. Cogn Comput. 2013;5(4):566–79.
41. Sun Y, Kamel MS, Wong AK, Wang Y. Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. 2007;40(12):3358–78.
42. Tang Y, Zhang YQ, Chawla NV, Krasser S. Svms modeling for highly imbalanced classification. IEEE Trans Syst Man Cybern Part B Cybern. 2009;39(1):281–8.
43. Yang Q, Wu X. 10 challenging problems in data mining research. Int J Inf Technol Decis Mak. 2006;5(04):597–604.