Reducing Over-Weighting in Supervised Term Weighting for Sentiment Analysis

Haibing Wu
Department of Electronic Engineering
Fudan University
Shanghai, China
haibingwu13@fudan.edu.cn

Xiaodong Gu
Department of Electronic Engineering
Fudan University
Shanghai, China
xdgu@fudan.edu.cn

Abstract

Recently the research on supervised term weighting has attracted growing attention in the field of Traditional Text Categorization (TTC) and Sentiment Analysis (SA). Despite their impressive achievements, we show that existing methods more or less suffer from the problem of *over-weighting*. Overlooked by prior studies, over-weighting is a new concept proposed in this paper. To address this problem, two regularization techniques, *singular term cutting* and *bias term*, are integrated into our framework of supervised term weighting schemes. Using the concepts of over-weighting and regularization, we provide new insights into existing methods and present their regularized versions. Moreover, under the guidance of our framework, we develop a novel supervised term weighting scheme, regularized entropy (*re*). The proposed framework is evaluated on three datasets widely used in SA. The experimental results indicate that our *re* enjoys the best results in comparisons with existing methods, and regularization techniques can significantly improve the performances of existing supervised weighting methods.

1 Introduction

Sentiment Analysis (SA), also known as opinion mining, has enjoyed a burst of research interest with growing avenues (e.g., social networks and e-commerce websites) for people to express their sentiments on the Internet. A typical sentiment-analysis application mainly involves three key subtasks, namely holder detection, target extraction and sentiment classification (Liu, 2012; Hu and Liu, 2004). A simple and most extensively studied case of sentiment classification is sentiment polarity classification, which is the binary classification task of labelling the polarity of a sentiment-oriented document as positive or negative. Sentiment classification can be performed at the document, sentence, phase or word level. In this paper, we focus on sentiment polarity classification at document level.

Just like Information Retrieval (IR) and TTC, in sentiment classification, the content of an opinion-orientated document can be represented as a vector of terms in light of Vector Space Model (VSM). In VSM, each dimension of the vector corresponds to a term and different terms have different weights, thus the term weight represents the contribution of the term to the sentiment of a document in sentiment classification. Term weighting is the task of assigning appropriate weights to terms according to their correlations with the category concept. Term weighting schemes fall into two categories (Lan et al., 2009; Debole and Sebastiani, 2003). The first one, known as *unsupervised* term weighting method, does not take category information into account. The second one referred to as *supervised* term weighting method embraces the category label information of training documents in the categorization tasks. Although most term weighting approaches to text categorization, including sentiment classification, are borrowed from IR, recently several new supervised term weighting schemes have been studied and achieved significant successes in TTC and SA (Lan et al., 2009; Martineau and Finin, 2009; Paltoglou and Thelwall, 2010).

Despite the impressive achievements in the current field of supervised term weighting for TTC and SA, we indentify that existing supervised methods, more or less, suffer from over-weighting problem and thus develop a robust framework to address this problem. Over-weighting, overlooked by prior studies, is a new concept introduced in this paper. It would occur due to the presence of many noisy

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1322–1330, Dublin, Ireland, August 23-29 2014.

words and the unreasonably too large ratios between weights of different terms. Thus, it could result in poor representations of sentiments containing in documents. In order to reduce over-weighting problem for supervised term weighting, two regularization techniques called *singular term cutting* and *bias term* are proposed and integrated into our framework of supervised term weighting schemes. Singular term cutting is introduced to cut down the weights of noisy or unusual terms, and bias term is added to shrink the ratios between weights of different terms.

Using the concepts of over-weighting and regularization, we provide new insights into existing supervised weighting methods and then present their regularized versions. We also propose a novel term weighting scheme called *regularized entropy* (*re*) under the guidance of our framework. The formulation of *re* bases on entropy, which is used to measure the distribution of terms over different categories, and the terms with smaller entropy value have larger weights.

After presenting our framework, the regularized versions of existing methods and *re* in detail, experiments are conducted on three publicly available datasets widely used in SA. In our experiments, *re* is compared against many existing methods appearing in IR, TTC and SA. We also compare the performances of existing supervised weighting methods against their regularized versions. The results of comparative experiments indicate that *re* clearly outperform existing methods, the introduction of regularization techniques significantly improves the performances of existing supervised weighting methods.

2 Review of Term Weighting Schemes in IR, TTC and SA

In IR, TTC and SA, one of the main issues is the representation of documents. VSM provides a simplifying representation by representing documents as vector of terms. Term weighting aims to evaluate the relative importance of different terms in VSM. There are three components in a term weighting scheme, namely local weight, global weight and normalization factor (Salton and Buckley, 1988; Lan et al., 2009). Final term weight is the product of the three components:

$$t_{ij} = l_{ij} \times g_i \times n_j \, , \tag{1}$$

where t_{ij} is the final weight of i_{th} term in the j_{th} document, l_{ij} is the local weight of i_{th} term in the j_{th} document, g_i is the global weight of the i_{th} term, and n_j is the normalization factor for the j_{th} document.

2.1 Local Term Weighting Schemes

Local weight component is derived only from frequencies within the document. Table 1 lists three common local weighting methods, namely raw term frequency (*tf*), term presence (*tp*) and augmented term frequency (*atf*). In IR and TTC, the most widely used local weight is *tf* , but pioneering research

Local weight	Notation	Description
tf	*tf*	Raw term frequency.
$\begin{cases} 1, \text{ if } tf > 0 \\ 0, \text{ otherwise} \end{cases}$	*tp*	Term presence, 1 for presence and 0 for absence.
$k + (1-k)\dfrac{tf}{\max_t(tf)}$	*atf*	Augmented term frequency, $\max_t(tf)$ is the maximum frequency of any term in the document, k is set to 0.5 for short documents (Salton and Buckley, 1988).

Table 1: Local term weighting schemes.

Notation	Description
a	Positive document frequency, i.e., number of documents in positive category containing term t_i.
b	Number of documents in positive category which do not contain term t_i.
c	Negative document frequency, i.e., number of documents in negative category containing term t_i.
d	Number of documents in negative category which do not contain term t_i.
N	Total number of documents in document collection, $N = a + b + c + d$.
N^+, N^-	N^+ is number of documents in positive category, and N^- is number of documents in negative category. $N^+ = a+b$, $N^-=c+d$.

Table 2: Notations used to formulate global term weighting schemes.

Global weight	Notation	Description
$\log_2 \frac{N}{a+c}$	*idf*	Inverse document frequency (Jones, 1972)
$\log_2 \left(\frac{N}{a+c} - 1 \right)$	*pidf*	Probabilistic *idf* (Wu and Salton, 1981)
$\log_2 \frac{b+d+0.5}{a+c+0.5}$	*bidf*	BM 25 *idf* (Jones et al., 2000)
$\frac{a}{N} \log_2 \frac{aN}{(a+b)(a+c)} + \frac{b}{N} \log_2 \frac{bN}{(a+b)(b+d)} +$ $\frac{c}{N} \log_2 \frac{cN}{(a+c)(c+d)} + \frac{d}{N} \log_2 \frac{dN}{(b+d)(c+d)}$	*ig*	Information gain
$\log_2 \left(\max(\frac{aN}{(a+c)N^+}, \frac{cN}{(a+c)N^-}) \right)$	*mi*	Mutual information
$\log_2 \frac{N^- a}{N^+ c}$	*didf*	Delta *idf* (Martineau and Finin, 2009)
$\log_2 \frac{N^- a + 0.5}{N^+ c + 0.5}$	*dsidf*	Delta smoothed *idf* (Paltoglou and Thelwall, 2010)
$\log_2 \frac{N^- (a+0.5)}{N^+ (c+0.5)}$	*dsidf'*	Another version of *dsidf*
$\log_2 \frac{(N^- - c + 0.5)a + 0.5}{(N^+ - a + 0.5)c + 0.5}$	*dbidf*	Delta BM25 *idf* (Paltoglou and Thelwall, 2010)
$\log_2 \frac{(N^- - c + 0.5)(a + 0.5)}{(N^+ - a + 0.5)(c + 0.5)}$	*dbidf'*	Another version of *dbidf*
$\log_2 \left(2 + \frac{a}{\max(1, c)} \right)$	*rf*	Relevance frequency (Lan et al., 2009)

Table 3: Global term weighting schemes.

on SA by Pang et al. (2002) showed that much better performance was achieved by using *tp*, not *tf*. This conclusion for SA was opposite to TTC, so *tp* was preferred in subsequent SA research.

2.2 Global Term Weighting Schemes

In contrast to local weight, global weight depends on the whole document collection. To formulate different global weighting schemes, some notations are first introduced in table 2. By using these notations, table 3 presents several representative global weighting schemes in IR, TTC and SA, including inverse document frequency (*idf*), probabilistic *idf* (*pidf*), BM25 *idf* (*bidf*), information gain (*ig*), delta *idf* (*didf*), *dsidf'*, delta BM25 *idf* (*dbidf*), *dbidf'* and relevance frequency (*rf*). Among these global weighting methods, *idf*, *pidf* and *bidf* are unsupervised methods because they do not utilize the category label information of document collection. The common idea behind them is that a term that occurs rarely is good at discriminating between documents.

Other global weighting schemes in table 3 are supervised term weighting methods. Among these supervised factors, feature selection methods, *ig* and *mi* are studied earliest. In TTC field, Debole and Sebastiani (2003) replaced *idf* with *ig* and other feature selection methods, *gr* and *chi*, for global term weighting. They concluded that these feature selection methods did not give a consistent superiority over the standard *idf*. In SA field, Deng et al. (2013) also employed several feature selection methods, including *ig* and *mi*, to learn the global weight of each term from training documents with category labels. The experimental results showed that compared with *bidf*, *mi* produced better accuracy on two of three datasets but *ig* provided very poor results.

For the rest of supervised term weighting schemes in table 3, *rf* is published in TTC literature, *didf* and *dbidf* are published in SA literature. The intuitive consideration of *rf* is that the more concentrated a high frequency term is in the positive category than in the negative category, the more contributions it makes in selecting the positive samples from the negative samples. Driven by this intuition, *rf* was proposed to capture this basic idea. The experimental results showed that when combined with the local component *tf*, *rf* consistently and significantly outperformed other term weighting methods, including *idf* and *ig*. Due to the asymmetry of *rf*, it only boosts the weights of terms that appear more frequently in the positive category. In other words, *rf* discriminates against terms appearing more frequently in negative category. The asymmetry of *rf* is reasonable for TTC because it only cares whether a document belongs to a topic or not and a single document can concentrate on different topics. However, it is not the case for binary sentiment classification since terms appear in positive or negative reviews are of the same importance.

In SA field, The first published supervised term weighing scheme, introduced by Martineau and Finin (2009), is called delta *idf*. Instead of only using *tf* as term weights, the authors assigned term weights for a document by calculating the difference of that term's *idf* values in the positive and negative training documents. Obviously, *didf* boosts the importance of terms that are unevenly distributed between the positive and negative categories and discounts evenly distributed words. It is known that the distribution of sentimental words is more uneven than stop words, as a result, *didf* assign much greater weights to sentimental words than stop words. The produced results showed that *didf* provided higher classification accuracy than the simple *tf* or the binary weighting scheme *tp*. Nonetheless, *didf* is susceptible to the errors caused by the case that $a = 0$ or $c = 0$, and the authors did not provide any detail that how they deal with this problem. Following the idea of *didf* and to rectify the problem of *didf*, Paltoglou and Thelwall (2010) presented a smoothed version of *didf*, delta smoothed *idf* (*dsidf*), and explored other more sophisticated global term weighting methods originated from IR including BM25 *idf* (*bidf*) and delta BM25 *idf* (*dbidf*). The formulas of these schemes are also presented in table 3. They showed that these variants of the classic *tf-idf* scheme provided significant increases over the best term weighting methods for SA in terms of accuracy. The idea of introducing smoothness technique is wonderful and can indeed avoid the computational errors in *didf*, but due to the unsuitable implementation, the smoothed version of *didf* provided by Paltoglou and Thelwall (2010) severely encounters the problem of over-weighting. We provide another version of *dsidf*, namely *dsidf'*. Besides *dsidf*, over-weighting is also severely encountered by *dbidf*, and our versions of it is denoted as *dbidf'*.

3 Research Design

Based on our review of term weighting schemes above, we believe that supervised term weighting can, but not always, boost the performances of text categorization. Actually, the somewhat successful ones, such as *rf*, *didf* and *dsidf*, follow the same intuition that the more imbalanced a term's distribution is across different categories, the more contribution it makes in discriminating between the positive and negative documents. The only difference between them lies in the quantification of the imbalance of a term's distribution. However, existing methods more or less suffer from the problem of over-weighting. We argue that a successful supervised weighting method should satisfy the following two criteria and develop a robust framework of supervised term weighting schemes.

Criterion 1: Assign large weights to terms that unevenly distribute across different categories.

Criterion 2: Avoid the over-weighting problem.

3.1 Our Framework

Over-weighting is somewhat like over-fitting in statistical machine learning, so we name it over-weighting. It is known that over-fitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Similarly, over-weighting could occur in supervised term weighting. In practice we indentify that over-weighting is caused by the presence of noisy terms and the unsuitable quantification of the degree of the imbalance of a term's distribution.

The presence of noisy terms would lead to the problem of over-weighting. To illustrate this phenomenon, suppose that the training document collection contains 10,000 documents and evenly distributes over the positive and negative category, the number of documents containing the strange term "leoni" belonging to positive category is 5, i.e., $a = 5$, and no document belonging to negative category contains "leoni", i.e., $c = 0$, according to the formulation of most existing supervised methods such as dsidf, the weight of "leoni" should be large since "leoni" unevenly distributes over different categories. However, since the total number of documents containing "leoni" is so trivial compared to the size of training collection, “leoni” could be an unusual word. We call the terms like “leoni” singular terms. Statistically, singular terms account for a great part of the whole terms in the dictionary constructed based on the training documents even if we filter out low frequency words. As singular terms do not embody any sentiment and the weights of them are supposed to be small, we formulate the global weight of term t_i as

$$g_i = \begin{cases} 0, \text{if } a = 0 \text{ (or } c = 0) \text{ and } (a+c)/N < \alpha \\ r, \text{otherwise} \end{cases} \tag{2}$$

where r is a variable quantifying the imbalance of a term’s distribution across different categories and its value ranges from 0 to 1, α is a very small number, here we set α to 0.005. As formula (2) cuts down the weights of singular terms, we name the first regularization technique *singular term cutting*.

Also, an unsuitable quantification of a term’s distribution would lead to unreasonably too large ratios between different weights and thus results in over-weighting, although the term weight calculated by (2) is no more than 1. This finding leads us to introduce the second regularization technique, bias term, to the weight of term t_i, so our framework of supervised term weighting schemes is modified as

$$g_i = \begin{cases} b_0, \text{if } a = 0 \text{ (or } c = 0) \text{ and } (a+c)/N < \alpha \\ b_0 + r, \text{otherwise} \end{cases} \tag{3}$$

where b_0 is the bias term, it shrinks the ratios between different weights of terms, the value of it controls the trade-off between weighting the terms freely and preventing over-weighting. If b_0 is too large, supervised term weighting would make no difference and under-weighting would occur. If b_0 is too small, over-weighting would occur. The optimal value of b_0 can be obtained via cross-validation, a model selection technique widely used in machine learning.

3.2 Regularized Versions of Existing Methods

As mentioned before, the somewhat successful ones of existing supervised weighting methods try to quantify the imbalance of a term’s distribution. Recall that in our framework, r is just right a variable sharing this purpose, so we can make improvement on existing supervised weighting methods by replacing r with them. Ahead of the improvement of existing methods, we first provide new insights into existing methods using the concepts of over-weighting and regularization.

Because r quantifies the degree of the imbalance of a term’s distribution across different categories, existing methods are required to satisfy Criterion 1. It has been clear that *didf*, *dsidf*, *dsidf’*, *dbidf*, *dbidf’*, *mi* and *rf* satisfy Criterion 1 via the review of existing methods in section 2. Another property shared by them is that the formulations of them base on logarithmic function. It is known that logarithmic function plays the role of shrinking the ratios between different term weights, so they implicitly satisfy Criterion 2 and in some degree reduce the over-weighting problem. In actuality, *dsidf*, *dsidf’* and *rf* can be treated as the further regularized versions of *didf* since the constant 2+ in *rf* and the smoothness in *dsidf* and *dsidf’* can be treated as regularization techniques. We have pointed out in section 2 that due to the unreasonable implementation of smoothness, *dsidf* and *dbidf* do not reduce, but aggravate over-weighting. As to *dsidf’* and *dbidf’*, they limit over-weighting in a very great degree via the introduction of smoothness technique and logarithmic function, but over-weighting is still not overcome completely, experimental results in section 4 will show that the performances of them can be further enhanced by cutting the weights of singular terms and adding a bias term.

Method	Regularized version
didf *dsidf* *dsidf’* *rf*	$\begin{cases} b_0, \text{if } a = 0 \text{ (or } c = 0) \text{ and } (a+c)/N < \alpha \\ b_0 + \dfrac{\log_2 \max(a,c)/\min(a,c)}{\max_t\{\log_2 \max(a,c)/\min(a,c)\}}, \text{otherwise} \end{cases}$
dbidf *dbidf’*	$\begin{cases} b_0, \text{if } a = 0 \text{ (or } c = 0) \text{ and } (a+c)/N < \alpha \\ b_0 + \dfrac{\log_2 (N^- - \min(a,c))/\max(a,c)}{\max_t\{\log_2 (N^- - \max(a,c))/\min(a,c)\}}, \text{otherwise} \end{cases}$
mi	$\begin{cases} b_0, \text{if } a = 0 \text{ (or } c = 0) \text{ and } (a+c)/N < \alpha \\ b_0 + \log_2 \dfrac{mi}{\max_t\{mi\}}, \text{otherwise} \end{cases}$

Table 4: Regularized versions of existing supervised term weighting schemes.

Up to present we have known that existing supervised methods encounter over-weighting in different degree. In order to make improvements on existing methods and under the guidance of our framework, we present the regularized versions of *didf*, *dsidf*, *dsidf'*, *dbidf*, *dbidf'* and *mi* in table 4. These methods are selected to improve due to their typical representations and diversities.

Note that the regularized versions of *didf*, *dsidf*, *dsidf'* and *rf* and are the same one due to the fact that *dsidf*, *dsidf'* and *rf* are same as *didf* if there is no smoothness or constant in them. For the same reason, *dbidf* and *dbidf'* are grouped together.

3.3 Regularized Entropy

Inspired by the derivation of our framework for supervised term weighting, we propose a novel supervised term weighting scheme called regularized entropy (*re*). For *re*, entropy is exploited to measure the degree of the imbalance of a term's distribution across different categories. According to information theory (Shannon, 1948), for a random variable *X* with *m* outcomes $\{x_1,\ldots, x_m\}$, the entropy, a measure of uncertainty and denoted by *H*(*X*), is defined as

$$H(X) = -\sum_{i=1}^{m} p(x_i)\log_2 p(x_i), \tag{4}$$

where p(x_i) is the probability that *X* equals to x_i. Let p^+ and p^- denote the probability of documents where term t_i occurs and belonging to positive and negative category respectively, then p^+ and p^- can be estimated as

$$p^+ \approx \frac{a}{a+c}, p^- \approx \frac{c}{a+c}. \tag{5}$$

According to formula (4), if term t_i occurs in a document, the degree of uncertainty of this document belonging to a category is

$$h = -p^+\log_2 p^+ - p^-\log_2 p^- = -\frac{a}{a+c}\log_2\frac{a}{a+c} - \frac{c}{a+c}\log_2\frac{c}{a+c}. \tag{6}$$

Obviously, if the documents containing term t_i distribute evenly over different categories, the entropy *h* will be large. In contrast, if the documents containing term t_i distribute unevenly over different categories, the entropy *h* will be relatively small. However, we hope that the more uneven the distribution of documents where term t_i occurs, the larger the weight of t_i is. And that the entropy *h* is between 0 and 1, so the original formula of the weight of term t_i is

$$g_i = 1 - h. \tag{7}$$

We call the scheme formulated by the (7) *nature entropy* (*ne*). It seems that *ne* can be used as the weights of terms directly and will perform well. Unfortunately, *ne* suffers from the same problem with existing methods. Under the guidance of our framework, the regularized version of *ne* is formulated as

$$g_i = \begin{cases} b_0, \text{ if } a = 0 \text{ (or } c = 0) \text{ and} (a+c)/N < \alpha \\ b_0 + (1-h), \text{otherwise} \end{cases}. \tag{8}$$

We name the proposed method formulated by (8) *regularized entropy* (*re*), which literally indicates the idea behind the scheme.

4 Experimental Results

We conduct sentiment classification experiments on three document-level datasets. The first one is Cornell movie review dataset introduced by Pang and Lee (2004). This sentiment polarity dataset consists of 1,000 positive and 1,000 negative movie reviews. The second dataset is taken from Multi-Domain Sentiment Dataset (MDSD) of product reviews (Blitzer et al., 2007). MDSD is initially released for the research on sentiment domain adaption but can also be used for sentiment polarity classification. It contains Amazon product reviews for different product types, we select camera reviews and thus refer the second corpus as Amazon camera review dataset. Also, it consists of 1,000 positive and 1,000 negative camera reviews.

For the above two datasets, the results are based on the standard 10-fold cross validation. Term weighting is performed on the 1,800 training reviews for each fold and the remaining 200 are used to evaluate the predicting accuracy. The overall classification accuracy is the average accuracy across 10 folds.

We also use the Stanford large movie review dataset developed by Mass et al. (2011). It contains 50,000 movie reviews, split equally into 25,000 training and 25,000 testing set. For this dataset, due to the original

Cornell movie review				**Amazon camera review**				**Stanford movie review**			
	tf	*tp*	*atf*		*tf*	*tp*	*atf*		*tf*	*tp*	*atf*
no	85.20	88.05	88.15	*no*	86.80	87.25	87.50	*no*	88.38	88.72	88.71
idf	84.15	84.90	85.10	*idf*	85.70	85.75	86.10	*idf*	88.30	88.24	88.26
ig	86.40	87.65	87.90	*ig*	87.25	87.85	87.65	*ig*	88.71	88.40	88.45
mi	86.90	88.85	88.85	*mi*	88.95	89.05	89.15	*mi*	89.23	89.45	89.52
dsidf	80.25	80.20	80.10	*dsidf*	83.15	82.80	83.30	*dsidf*	86.72	86.89	86.77
dsidf’	86.65	88.20	88.15	*dsidf’*	88.20	88.95	89.10	*dsidf’*	89.23	89.25	89.32
dbidf	81.20	81.10	81.10	*dbidf*	86.60	87.00	86.90	*dbidf*	86.80	86.73	86.78
dbidf’	87.30	88.30	88.40	*dbidf’*	88.85	89.10	89.00	*dbidf’*	89.41	89.39	89.52
rf	85.10	88.00	87.75	*rf*	86.95	87.35	87.85	*rf*	87.84	88.36	88.46
re	87.85	89.60	89.65	*re*	89.15	89.45	89.50	*re*	89.53	89.81	89.80

Table 5: Classification accuracy of local and global weighting methods.

split, no cross validation is used. Term weighting is only implemented on the training set, and the classification accuracy is reported based on the testing set.

We only use unigrams as the features. Support Vector Machine (SVM) is used as the classifier. Specially, we adopt the L2-regularized L2-loss linear SVM and the implementation software is LIBLINEAR (Fan et al., 2008). In all our experiments, cross-validation is performed on training document collection to obtain optimal value of b_0. On Cornell and Stanford movie review dataset, b_0 is set to 0.1 for *re*, 0.05 for the improved versions of *didf*, *dsidf*, *dsidf’* and *rf*, 0.02 for that of *mi*, and 0.01 for those of *dbidf* and *dbidf’*. On Amazon camera review dataset, b_0 is set to 0.05 for *re* 0.03 for the improved versions of *didf*, *dsidf*, *dsidf’* and *rf*, 0.02 for that of *mi*, and 0.01 for those of *dbidf* and *dbidf’*.

4.1 Experiment 1: Comparisons of *re* Against Existing Methods

Table 5 reports the classification accuracies of *re* and other term weighting schemes. On the Cornell movie review dataset, the local weighting method *tp* outperforms *tf* significantly in general except the case that *dbidf* and *dsidf* are used as the global weighting methods. There is no distinct difference between *tp* and *atf*, neither of them consistently performs better than each other when combined with various global weighting methods.

Compared to the change of local weighting methods, global weighting methods lead to more significant difference on classification accuracy. Combined with different local weighting schemes, the proposed global weighting method, *re*, has always been shown to clearly perform better than other global weighting methods. Specially, the highest classification accuracy, 89.65%, is achieved by the combination of *re* and *atf*, i.e., *atf-re*. Compared to *no*, *re* shows apparent superiorities, the increases of accuracy are +1.55% (from 88.05% to 89.60%) and +1.50% (from 88.15% to 89.65%) respectively when the local methods are *tp* and *atf*. The most popular *idf* in IR field is not a good choice for sentiment classification. For the methods originated from TTC field, the feature selection approaches, *mi* performs well and the classification accuracies produced by it is higher than the others except *re* in apparent advantages. Unlike *mi*, *ig* is instead a disappointing performer, the accuracy 87.65%, provided by *ig* when combined with *tp*, is far lower than that of *mi*, this observation is entirely predictable due to the fact that *ig* does not follow Criterion 1 and suffers over-weighting. As for *rf*, it do not perform well, the highest accuracy provided by them is only 88.00% respectively. It is not surprising that *rf* does not even outperform no since its discrimination against the terms that appear more frequently in the negative reviews. When it comes to the approaches that recently appeared in SA literature, both *dsidf* and *dbidf* performs very poorly because of over-weighting problem caused by the unreasonable implementation. But both *dsidf’* and *dbidf’* are shown to give slightly better results than *no*.

On the Amazon camera review dataset, the performances of local weighting methods agree with those on Cornell movie review dataset. Again, *tp* and *atf* yield comparable classification accuracy and both of them outperform *tf*. The performances on this dataset produced by global weighting methods are, generally, in accordance to those on the previous dataset, but some differences deserve our attention. First, *re* outperforms *no* with greater superiorities compared to the previous dataset, the increase of accuracy is +2.20% (from 87.25% to 89.45%) and +2.00% (from 87.50% to 89.50%) respectively when the local methods are *tp* and *atf* . Another one is that *dsidf’* provides more apparent advantages over *no* compared to the previous dataset but differences between *re* and *dsidf’* become smaller.

Cornell movie review				Amazon camera review				Stanford movie review			
Method	Original version	Regularized version	Difference to original version	Method	Original version	Regularized version	Difference to original version	Method	Original version	Regularized version	Difference to original version
didf	N/A	89.50	N/A	*didf*	N/A	89.60	N/A	*didf*	N/A	89.71	N/A
dsidf	80.20	89.50	+9.30	*dsidf*	82.80	89.60	+6.80	*dsidf*	86.89	89.71	+2.82
dsidf'	88.20	89.50	+1.30	*dsidf'*	88.95	89.60	+0.65	*dsidf'*	89.25	89.71	+0.46
rf	88.00	89.50	+1.50	*rf*	87.35	89.60	+2.25	*rf*	88.36	89.71	+1.35
dbidf	81.10	89.25	+8.15	*dbidf*	87.00	89.65	+2.65	*dbidf*	86.83	89.49	+2.66
dbidf'	88.30	89.25	+0.95	*dbidf'*	89.10	89.65	+0.55	*dbidf'*	89.39	89.49	+0.10
mi	88.85	89.10	+0.25	*mi*	89.05	89.55	+0.50	*mi*	89.45	89.59	+0.14
ne	83.45	89.60	+6.15	*ne*	87.85	89.45	+1.60	*ne*	87.32	89.81	+2.49

Table 6: Classification accuracies of original versions of *ne* and some existing supervised term weighting schemes and their regularized versions under our framework.

On the Stanford large movie review dataset, differences in accuracy are smaller than those on the previous ones, but the testing set contains 25,000 documents, the variance of the performance estimate is quite low (Maas et al., 2011). Interestingly, unlike the conclusion on the Cornell movie review dataset, *tp* does not show significant advantages over *tf* and even slightly underperforms *tf* when the global methods are *idf*, *ig*, *dbidf*, and *dbidf'*. The performances of *tp* and *atf* are still comparable but *atf* reveals a slight superiority over *tp*. In spite of the smaller differences, among the global weighting methods, *re* still embraces the highest classification accuracy, 89.81%, when combined with *tp*. In accordance to the observations on the previous two datasets, *mi*, *dsidf'* and *dbidf'* yield higher classification accuracies than *no*. Again, the other global methods, *idf*, *ig*, *rf*, *dsidf* and *dbidf* still produce comparable or lower accuracies in comparison with *no*.

4.2 Experiment 2: Comparisons Existing Methods Against Their Regularized Versions

We also compare the performances of some representative supervised methods, i.e., *didf*, *dsidf*, *dsidf'*, *dbidf*, *dbidf'*, *rf*, and *mi* against their regularized versions. In this experiment, we only use *tp* as the local weighting method. Table 6 records the classification accuracies of original versions of these methods and their improved versions. We can observe that the regularized versions of existing methods consistently have much better accuracy. Regularized version of *dsidf* yields the most significant improvements, the accuracy difference to original version is +9.30%, +6.80% and +2.82% on three datasets respectively. The accuracy difference between *dbidf* and its regularized version is also remarkable and significant. These observations validate our analysis in section 2 that *dsidf* and *dbidf* severely encounters over-weighting problem. Note that the improvements of the regularized versions of *dsidf'*, *dbidf'* and *mi* over their originals are trivial as they are much less subjected to over-weighting. Significance test will be included for these methods to test if the improvements are statistically reliable.

5 Conclusion and Future Work

In this study we have proposed a robust framework of supervised term weighting schemes. This framework is developed based on the techniques introduced to reduce over-weighting problem commonly suffered by existing supervised weighting methods.

Over-weighting is a new concept proposed in this paper, which is caused by the presence of many noisy words and the unreasonably too large ratios between weights of different terms. To reduce over-weighting, we have introduced two regularization techniques, singular term cutting and bias term. Singular term cutting cuts down the weights of noisy or strange words, and bias term shrinks the ratios between weights of different terms. Comparative experiments have shown that regularization techniques significantly enhance the performances of existing supervised methods.

More over, a novel supervised term weighting scheme, *re*, is proposed under our framework. The formulation of *re* bases on entropy, which is used to measure a term's distribution across different categories. The experimental results have shown that *re* not only outperforms its original version, *ne*, with great advantage but also consistently outperforms existing methods appearing in IR, TTC and SA. In the future, we would like to extend our work to other tasks such as multi-class classification and traditional text categorization.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China under grant 61371148 and Shanghai National Natural Science Foundation under grant 12ZR1402500.

References

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of ACL*, Pages 142-150.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan and Claypool Publishers.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, In *Proceedings of ACL*, pages 271-278.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques, In *Proceedings of EMNLP*, pages 79-86.

Claude E. Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27 (3):379-423.

Franca Debole and Fabrizio Sebastiani. 2003. Supervised Term Weighting for Automated Text Categorization. In *Proceedings of ACM Symposium on Applied Computing*, pages 784-788.

Georgios Paltoglou and Mike Thelwall. 2010. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In *Proceedings of ACL*, pages 1386-1395.

Gerard Salton and Christopher Buckley. 1988. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513-523.

Gerard Salton and Michael J. McGill. 1983. Introduction to Modern Information Retrieval. *McGraw Hill Book Inc.*, New York.

Harry Wu and Gerard Salton. 1981. A Comparison of Search Term Weighting: Term Relevance vs. Inverse Document Frequency. In *Proceeding of ACM SIGIR*, pages 30-39.

John Blitzer, Mark Dredze and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of ACL*, pages 440-447.

Justin Martineau and Tim Finin. 2009. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In *Proceedings of Third AAAI International Conference on Weblogs and Social Media*, pages 258-261.

Karen S. Jones, Stephen Walker and Stephen E. Robertson. 2000. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments. *Information Processing and Management,* 36(6):779-808.

Man Lan, Chew L. Tan, Jian Su and Yue Lu. 2009. Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(4):721-735.

Mingqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of ACM SIGKDD*, pages 168-177.

Rong E. Fan, Kai W. Chang, Cho J. Hsieh, Xiang R. Wang, and Chih J. Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research,* 9:1871-1874.

Sparck K. Jones. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation,* 28:11-21.

William B. Croft. 1983. Experiments with Representation in A Document Retrieval System. *Information Technology: Research and Development,* 2:1-21.

Zhi H. Deng, Kun H. Luo and Hong L. Yu. 2013. A Study of Supervised Term Weighting Scheme for Sentiment Analysis. *Expert Systems with Applications,* 41(7):3506-3513.