Journal of Informetrics 5 (2011) 313–322

Contents lists available at ScienceDirect

Journal of Informetrics

journal homepage: www.elsevier.com/locate/joi

A neural network based approach for sentiment classification in the blogosphere

Long-Sheng Chen[a,*], Cheng-Hsiang Liu[b], Hui-Ju Chiu[a]

[a] *Department of Information Management, Chaoyang University of Technology, 168 Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan, ROC*
[b] *Department of Industrial Management, National Pingtung University of Science and Technology, 1 Shuefu Road, Neipu, Pingtung, Taiwan, ROC*

ARTICLE INFO

Article history:
Received 16 July 2009
Received in revised form 6 January 2011
Accepted 6 January 2011

Keywords:
Semantic orientation
Neural networks
Sentiment classification
Blogs

ABSTRACT

Recognizing emotion is extremely important for a text-based communication tool such as a blog. On commercial blogs, the evaluation comments by bloggers of a product can spread at an explosive rate in cyberspace, and negative comments could be very harmful to an enterprise. Lately, researchers have been paying much attention to sentiment classification. The goal is to efficiently identify the emotions of their customers to allow companies to respond in the appropriate manner to what customers have to say. Semantic orientation indexes and machine learning methods are usually employed to achieve this goal. Semantic orientation indexes do not have good performance, but they return results quickly. Machine learning techniques provide better classification accuracy, but require a lot of training time. In order to combine the advantages of these two methods, this study proposed a neural-network based approach. It uses semantic orientation indexes as inputs for the neural networks to determine the sentiments of the bloggers quickly and effectively. Several actual blogs are used to evaluate the effectiveness of our approach. The experimental results indicate that the proposed approach outperforms traditional approaches including other neural networks and several semantic orientation indexes.

1. Introduction

Blogs are one of the fastest growing sections of the emerging communication mechanisms (Cohen & Krishnamurthy, 2006; Lambiotte, Ausloos, & Thelwall, 2007; Singh, Veron-Jackson, & Cullinane, 2008; Tang, Tan, & Cheng, 2009). Bloggers record the daily events in their lives and express their opinions, feelings, and emotions in an on-line journal, or blog (Nardi, Schiano, Gumbrecht, & Swartz, 2004). One of the features of most blogs is the fact that readers can comment on-line on whatever the blogger wrote in his/her journal. This ability has facilitated interaction between bloggers and those that read their blog (Chau & Xu, 2007). In addition, many of these blogs contain reviews on many products, issues, etc. (Martin, 2005; Murphy, 2006; Tang et al., 2009). Some of these reviews or evaluations found in a blogger's log may be negative, and sometimes they spread like wildfire through cyberspace. Negative comments about a product can be harmful to an enterprise. It is therefore important to effectively recognize the sentiment of bloggers, especially for those enterprises that use blogs as a marketing channel (Singh et al., 2008).

Lately, researchers have been paying much attention to sentiment classification and analysis (Subasic & Huettner, 2001; Wu, Chuang, & Lin, 2006) which identifies the emotions of their customers to allow companies to respond in the appro-

* Corresponding author. Tel.: +886 4 2332 3000x7752; fax: +886 4 2330 4902.
E-mail addresses: lschen@cyut.edu.tw, lschen0227@gmail.com (L.-S. Chen).

1751-1577/$ – see front matter
doi:10.1016/j.joi.2011.01.003

priate manner to what customers have to say. Nowadays, sentiment analysis became an important subfield of the field of information management (Tang et al., 2009) and can provide commercial bloggers with a tool to estimate the extent of product acceptance and to determine strategies that might improve a product's quality (Prabowo & Thelwall, 2009). Another advantage of recognizing the emotional state of bloggers is that it enables the companies to adjust their type of response to bloggers with negative comments (Lee, Narayanan, & Pieraccini, 2002). Another researcher indicated that the ability to identify the emotional state can reduce translation data size thereby increasing the fluency of net conferencing (Boucouvalas, 2002). Sentiment classification has also been widely applied in areas such as product comparisons, opinion summarizing, and reason mining (Tang et al., 2009; Yanaru, 1995).

The most direct method for determining a blogger's emotions is to use emotional keywords from the blogger's blog (Zhuang, Jing, & Zhu, 2006). Although a blog often contains multimedia data, simple text remains its main communication tool (Wu et al., 2006). In the literature, two popular approaches, machine learning (ML) methods and information retrieval (IR) (Hearst, 1992) techniques (semantic orientation index, SO index) (Tang et al., 2009), are employed to address the issue (Chaovalit & Zhou, 2005). In ML, several approaches have been developed. For examples, Abbasi, Chen, Thoms, and Fu (2008) proposed the support vector regression correlation ensemble (SVRCE) approach to analyze emotional states. Pang, Lee, and Vaithyanathan (2002) investigated several supervised ML methods to semantically classify movie reviews. Turney (2002) employs a specific unsupervised learning method for the semantic orientation of a review classification. Dave, Lawrence, and Pennock (2003) developed a method for automatically classifying positive and negative reviews and experimented with several methods related to the selection of features and scoring. In the work of Chaovalit and Zhou (2005), they used ML methods and the SO index to classify the comments of movie reviewers. Prabowo and Thelwall (2009) used multiple classifiers in a hybrid manner which may be more effective than any one of the classifiers separately. However, they did not consider the issue of training time required. These experimental results indicate that ML techniques have a better performance than the IR methods, but they require more time to be trained.

In IR, association, Pointwise Mutual Information (PMI), and Latent Semantic Analysis (LSA) have been employed to measure the similarity between words to classify sentiments. Several works reported that IR techniques are still good tools for sentiment classification. For examples, Turney and Littman (2003) employed association to determine the semantic orientation. Devillers, Luniel, and Vasilescu (2003) measured the emotional state by computing the conditional probability of keywords and related emotional states. Tao and Tan (2004) evaluated emotional conditions by utilizing emotive function words instead of emotive keywords. Hu and Liu (2004) used the adjective synonym sets and antonym sets in WordNet to judge the semantic orientation of adjectives. The classification accuracy of IR techniques is not very high (Pang et al., 2002); however, it allows us to quickly obtain the results of sentiment classification. In addition to the IR methods and ML techniques, there are various other ways to classify sentiments. Subasic and Huettner (2001) use fuzzy logic to manually construct a lexicon, based on which fuzzy technique was applied to which fuzzy set to analyze the affect of a document. Natural language processing (NLP) techniques are also used in this area (Tang et al., 2009).

The above methods either require a certain amount of manual construction or they must rely on externally structured information sources (Wang, Lu, & Zhang, 2005). Although ML techniques have better classification abilities, they require additional learning time. Besides, the information of classes which has to be determined by domain experts must be provided before training (Pang et al., 2002; Su, Chen, & Chiang, 2006). Therefore, so as to avoid the problems of both these methods but keep their strengths, this study proposed a neural network (NN) based approach which combines the advantages of the ML techniques and the IR techniques. In our proposed method, the back-propagation neural network (BPN) (Rumelhart & McClelland, 1986) has been selected as the basic learner based on its strength of fault tolerance. Our method uses the results of the SO indexes as the inputs for the BPN. Several cases collected from real world blogs or databases are provided to demonstrate the effectiveness of our method. The experimental results indicate that our method can efficiently increase the performance of sentiment classification and save a substantial amount of training time compared with traditional IR and ML techniques, respectively.

2. Neural networks and semantic orientation indexes

The proposed approach uses 4 different types of SO indexes as the input neurons. A brief introduction regarding BPN and the 4 different types of SO indexes is provided in the following subsections.

2.1. Back-propagation neural networks

Neural networks offer advantages such as adaptive learning, parallelism, fault tolerance, and generalization. In general, neural nets can be classified into two categories, feed-forward and feedback networks. In this study, the feed-forward network was employed because of its superior classification ability.

Among the feed-forward networks, BPN is the best known networks and it remains one of the most useful ones. This iterative gradient algorithm is designed to minimize the mean square error between the actual output of a multilayer feed-forward perceptron and the desired output. Based on the rule of thumb and reports of available published papers (Chen, Su, & Chen, 2009; Su, Hsu, & Tsai, 2002; Su, Yang, & Ke, 2002), the number of hidden layers could be one or two. The back-propagation algorithm includes a forward pass and a backward pass. The purpose of the forward pass is to obtain the activation value, and the backward pass is to adjust the weights and biases based on the difference between the desired and the actual network outputs. These two passes will continue iteratively until the network converges. The feed-forward network training by back-propagation pseudo-code algorithm can be summarized as follows (Chen, Hsu, & Chen, 2009b).

While error is too large

Step 1. For each training pattern (presented in random order):
  Step 1.1. Apply the inputs to the network.
  Step 1.2. Calculate the output for every neuron from the input layer, through the hidden layer(s), to the output layer.
  Step 1.3. Calculate the error at the outputs.
  Step 1.4. Use the output error to compute error signals for pre-output layers.
  Step 1.5. Use the error signals to compute weight adjustments.
  Step 1.6. Apply the weight adjustments.
Step 2. Periodically evaluate the network performance.

2.2. Semantic orientation indexes

The general SO index is used to infer semantic orientation from the semantic association (SO-A). In the SO-A index defined in Eq. (1), we should first construct a positive and a negative paradigm set first. Take a movie review for example. The positive and negative paradigm sets could be {feel good, good movie, good performance, great deal} and {long time, main problem, old fashioned}, respectively. The semantic orientation of a given word is then calculated from the strength of its association with a positive paradigm, a set of positive words (*Pword*), minus the strength of its association with a negative paradigm, a set of negative words (*Nword*). A word, *word*, is classified as having a positive (negative) semantic orientation when the SO-A(*word*) is positive (negative). The magnitude (absolute value) of the SO-A(*word*) can be considered as the strength of the semantic orientation:

$$\text{SO-A}(word) = \sum_{pword \in Pword} A(word, pword) - \sum_{nword \in Nword} A(word, nword) \tag{1}$$

The second index calculates the semantic orientation from the PMI, called the SO-PMI index. This index is extended from SO-A and it is widely applied in practice (Abbasi et al., 2008; Chaovalit & Zhou, 2005; Turney, 2002; Turney & Littman, 2003). Unlike the SO-A, the SO-PMI uses the PMI-IR (Pointwise Mutual Information and Information Retrieval) to estimate the semantic orientation of a phrase (Church & Hanks, 1989; Turney, 2002). The PMI between two words, *word*1 and *word*2, is defined as

$$PMI(word1, word2) = \log_2\left(\frac{p(word1 \quad \text{AND} \quad word2)}{p(word1)\,p(word2)}\right) \tag{2}$$

where $p(word1 \text{ AND } word2)$ represents the probability in which *word*1 and *word*2 co-occur. If the words are statistically independent, then the probability that they co-occur is given by the product $p(word1)\,p(word2)$. The ratio between $p(word1 \text{ AND } word2)$ and $p(word1)\,p(word2)$ is thus a measure of the degree of statistical dependence between these words. The log of this ratio is the amount of information that we acquire about the presence of one of the words when we observe the other. Finally, the SO-PMI can be calculated as follows:

$$\text{SO-PMI}(word) = \sum_{pword \in Pword} \text{PMI}(word, pword) - \sum_{nword \in Nword} \text{PMI}(word, nword) \tag{3}$$

Besides, PMI-IR estimates PMI by issuing queries to a search engine (hence the IR in PMI-IR) and by noting the number of hits (matching documents). Our study uses the AltaVista Advanced Search engine 5, which indexes approximately 350 million web pages. Moreover, in addition to the AND operator, the NEAR operator is also being employed in this work (Turney & Littman, 2003). Thus, using 2 different operators, we have two SO-PMI indexes, SO-PMI(AND) and SO-PMI(NEAR) in this study.

The last index is SO-LSA which calculates the strength of the semantic association between words using LSA (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990; Landauer & Dumais, 1997). LSA uses the Singular Value Decomposition (SVD) shown in Fig. 1 to analyze the statistical relationships among words in a corpus. The first step in implementing SVD is to use the text to construct a document-term matrix *A*. Each cell represents the weight of the corresponding word in the corresponding chunk of text. The weight is typically the TF-IDF score (Term Frequency times Inverse Document Frequency) for the word in that chunk (Van Rijsbergen, 1979). Next, we apply the SVD to decompose *A* into a product (USV^T) of three matrices, U, S, and V^T, where U and V^T are in column orthonormal form and S is a diagonal matrix of singular values. If A is of rank r, then S is also of rank r. Let S_k, where $k < r$, be the diagonal matrix formed from the top k singular values, and let U_k and V_k^T be the matrices produced by selecting the corresponding columns from U and V^T. The matrix $U_k S_k V_k^T$ is the matrix of rank k which best approximates the original matrix A. LSA works by measuring the similarity of words using $U_k S_k V_k^T$ instead of the original matrix A. The similarity of two words is measured by the cosine of the angle between their corresponding row vector s of U_k (Bartell, Cottrell, & Belew, 1992; Deerwester et al., 1990; Landauer & Dumais, 1997). The semantic orientation

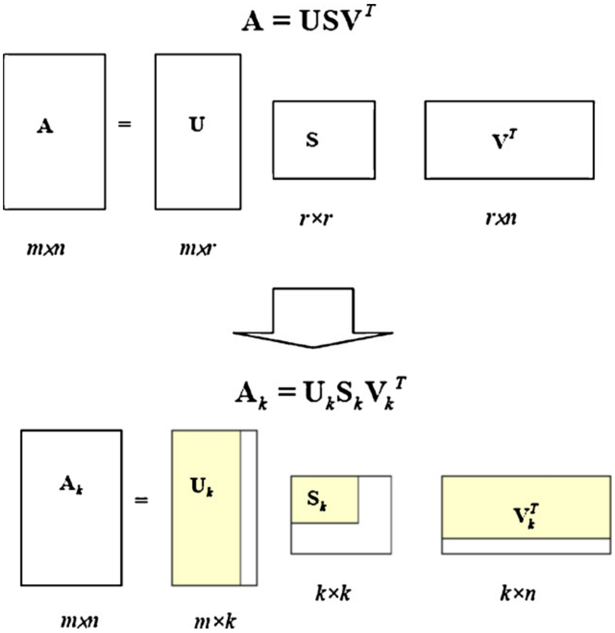

Fig. 1. The process of singular value decomposition (Deerwester et al., 1990).

of a word, *word*, is calculated by SO-LSA as follows:

$$\text{SO-LSA}(word) = \sum_{pword \in Pword} \text{LSA}(word, pword) - \sum_{nword \in Nword} \text{LSA}(word, nword) \tag{4}$$

3. Proposed neural network based methodology

This section will introduce the proposed NN based approach. As shown in Fig. 2, the implementation of our method can be divided into 4 steps. These four steps can be demonstrated as follows.

3.1. Step 1: prepare data

In this step, we need to segment words to construct a term-document matrix (TDM) for further analysis. Not all the words in a sentence are useful for classifying semantic orientations or related tasks. As Hu and Liu (2004) mentioned, nouns and noun phrases in sentences are likely to be the features that customers comment on, while adjectives are often used to express opinions and feelings. Therefore, following the works in Chaovalit and Zhou (2005), Turney (2002) and Wang et al. (2005), we use part-of-speech (POS) tagging to distinguish adjectives and adverbs in the sentences as candidate features that indicate semantic orientations. Table 1 shows the examples of POS used in the extraction of *n*-gram keywords. These extracted keywords can then be utilized to describe our experimental data. We record the occurrence frequency of every single keyword in each comment (document). Finally, a TDM has been completed for further analysis.

3.2. Step 2: calculate the SO indexes

In this study, we use four SO indexes including SO-A, SO-PMI(AND), SO-PMI(NEAR), SO-LSA as the input neurons of BPN. Therefore, the second step of our method is to calculate these SO indexes.

Table 1
The patterns of part-of-speech.

	First word	Second word
a.	Adjective	Noun
b.	Adverb	Adjective
c.	Adjective	Adjective
d.	Noun	Adjective
e.	Adverb	Verb

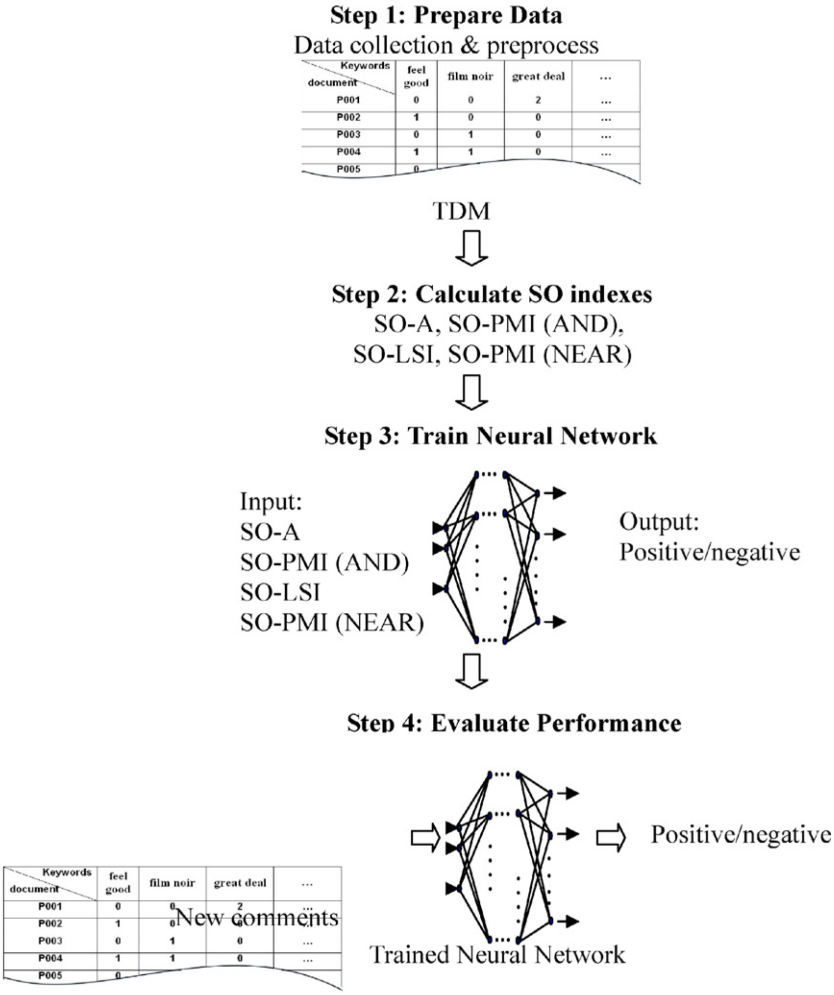

Fig. 2. The implemental procedure of the proposed NN based approach.

3.3. Step 3: train the neural network

The experimental data set is divided into training and test sets. First, we random arrange the collected samples. Next, in order the find the best performance, we systematically tried a different proportion (50–90%) of all examples to be the training data set, with the rest of the samples as the test set. This method is similar to a cross validation experiment. Then, we begin the training process of the BPN using the training data set. All optimal settings of the BPN including parameters and structure such as learning rate, training iterations, number of hidden neurons, and so on are obtained by a trial-and-error experiment. After the experiments, we picked the best performance.

3.4. Step 4: performance evaluation

In this step, we use the test data to evaluate the performance of our NN based approach, the BPN, and the four SO indexes.

4. Implementations

4.1. Data preparation

Table 2 provides the brief background of these employed data. In addition to movie review database, following the works of Leshed and Kaye (2006) and Na et al. (2005), we use two websites “LiveJournal (www.livejournal.com/stats/)” and “Review Centre (www.reviewcentre.com)” which provide bloggers’ comments and product reviews to be our experiment corpuses. The first data set comes from “LiveJournal”. Without considering topics of issues, we randomly and manually selected 353 bloggers’ comments after filtering some comments whose mood tags cannot be identified as positive or negative, or that

Table 2
The employed data sets.

No.	Data set	Source	No. of keywords	Data size	Class distribution
1	Blog	http://www.livejournal.com	35	353	Positive: 157 Negative: 186
2	MP3	http://www.reviewcentre.com	48	579	Positive: 235 Negative: 344
3	EC		66	386	Positive: 249 Negative: 137
4	Movie-1	http://www.cs.cornell.edu/people/pabo/movie-review-data/	81	500	Positive: 250 Negative: 250
5	Movie-2		78	1000	Positive: 500 Negative: 500

have no tags. Then, we use the "mood tags" of each comment to be its class label. Therefore, the first data set called "BLOG" contains 157 positive and 186 negative comments.

The 2nd and 3rd datasets come from "Review Centre" which has millions of product reviews by consumers. In addition, these reviews have been classified into several categories. We choose "E-Commerce and on-line shopping (EC)" and "MP3 player (MP3)" to be keywords for searching related product reviews. Then we manually collect and copy these textual comments, and then use QDA package software to transfer them into TDM. Consequently, MP3 and EC data sets have 579 reviews (235 positive and 344 negative) and 386 reviews (249 positive and 137 negative). Furthermore, we used the 5-star rating system of "Review Centre" to define its class. If one comment was ranked above 4-star (or below 2-star), then this comment was labeled as positive (or negative) class. We ignore the reviews whose rank is 3-star. Last two data sets are from a movie reviews database. We randomly selected 250 and 500 comments from both positive and negative groups, respectively. Therefore, we have 500 and 1000 examples as our experimental corpuses. In this work, we use "Movie-1" and "Movie-2" to represent the last two data sets.

Additionally, we employed the shareware Rubryx (http://www.sowsoft.com/rubryx) to segment words. Rubryx segments words are based on n-gram (unigram, bigrams, and tri-grams) features. Before extracting *n*-gram key words, some frequently used stop words must be removed. Each comment is converted into a vector of terms (keywords) with TF-IDF weights which is defined by the following formula (Na et al., 2005; Van Rijsbergen, 1979):

$$\mathrm{TFIDF} = \mathrm{TF} \times \log\left(\frac{N}{\mathrm{DF}}\right) \tag{5}$$

where TF is the number of times the term occurs in the current comment (document), N denotes the number of comments in the training set, and DF represents the document frequency (the number of comments in the training set containing the term).

Because our NN based approach uses 4 SO indexes to be the input neurons, we should consider the normalization problem. Because the different scales of values of the SO indexes may affect the performance of neural networks, we use two representations, quantitative & qualitative, in our proposed method. In the quantitative representation, all values of attributes are normalized to the interval [0,1] by employing a min–max normalization equation, as shown by Eq. (6). In this equation, $\max_i$ is the maximum, and $\min_i$ is the minimum of the ith SO index (attribute) values, and v_{ij} is the value of the ith SO index (attribute) of the jth example and v'_{ij} is the normalized value:

$$v'_{ij} = \frac{v_{ij} - \min_i}{\max_i - \min_i} \tag{6}$$

In the qualitative representation, we use the discrete values, +1, 0, −1, to denote all inputs of the NN based approach:

$$v''_{ij} = \begin{cases} +1, & \text{if } v_{ij} > 0 \\ 0, & \text{if } v_{ij} = 0 \\ -1, & \text{if } v_{ij} < 0 \end{cases} \tag{7}$$

4.2. Performance evaluation

The performance evaluation matrices, overall accuracy (OA) and $F1$ have been used. In short, the common way for evaluating the performance of classifiers is based on the confusion matrix shown in Table 3.

Table 3
The confusion matrix.

	Predicted positives	Predicted negatives
Actual positive examples	The number of True Positive examples (TP)	The number of False Negative examples (FN)
Actual negative examples	The number of False Positive examples (FP)	The number of True Negative examples (TN)

Table 4
The results of four SO indexes.

SO index	Data sets				
	Movie-1	Movie-2	EC	MP3	Blog
OA					
SO-A	11.1%	11%	6%	2.5%	21%
SO-PMI(NEAR)	50.5%	50%	32%	42.5%	57%
SO-PMI(AND)	51%	**51%**	34%	46%	55%
SO-LSA	**52.2%**	45%	**53%**	**54.5%**	**58%**
*F*1					
SO-A	0.598	0.102	0.092	0.045	0.202
SO-PMI(NEAR)	0.673	**0.673**	0.076	0.073	0.566
SO-PMI(AND)	**0.673**	0.671	0.251	0.296	0.211
SO-LSA	0.518	0.574	**0.574**	**0.583**	**0.611**

In general, the performance of a sentiment classifier is evaluated by the OA compared to the number of test cases. OA can be defined by Eq. (8):

$$\text{overall accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{8}$$

Another popular index is *F*1 whose formula comes from the combination of Precision and Recall. *F*1, Precision, and Recall are defined by Eqs. (9)–(11).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} \times \text{Recall}} \tag{11}$$

4.3. Experimental results

This section provides the results of the implementations. First, we attempted to compare the effectiveness of SO indexes, SO-A, SO-PMI(NEAR), SO-PMI(AND), and SO-LSA. Table 4 summarizes the results of these four indexes. In SO-A, some examples could not be identified. This might be due to insufficient data size, which is a weakness of SO-A.

When considering OA, SO-LSA has the best classification performance in 4 of 5 data sets. If we take *F*1 into consideration, SO-LSA also has the best performance in EC, MP3, and the Blog data sets. In addition, the average OA and *F*1 of SO-LSA are 52.54% and 0.572, respectively. They are better than SO-PMI (AND) (OA: 47.4%, *F*1: 0.420), SO-PMI (NEAR) (OA: 46.4%, *F*1: 0.412), and SO-A (OA: 10.32%, *F*1: 0.208). However, this performance of SO-LSA is not good enough. Therefore, next, we implemented BPN and our method.

To find the best performance of BPN and our method, we systematically tried a different proportion (50–90%) of all examples to be the training data set, with the rest of the samples as the test set. After the experiments, we picked the best performance. Table 5 provides the results of our NN based method (involving qualitative and quantitative representations) and the original BPN (which uses a TDM as input). From Fig. 3 and Table 5, we found that the proposed method, including quantitative and qualitative representation, has the best OAs in Movie-1, Movie-2, EC and Blog data sets. Compared with the original BPN, the NN based method can increase the classification performance by 4–6% in these 4 data sets. But, in the MP3 data set, the performance (OA) of the NN based method (quantitative: 60.0%; qualitative: 61.0%) drops slightly compared with the BPN (61.7%). On average, the NN based method can increase the OA by 3.06% and 0.46%, for the quantitative and qualitative representation, respectively.

When considering *F*1, our method had the best performance in all data sets. The *F*1 results including Movie-2 (*F*1: 0.678, qualitative), Movie-1 (*F*1: 0.750, quantitative), EC (*F*1: 0.812, qualitative & quantitative), MP3 (*F*1: 0.735, quantitative), Blog (*F*1: 0.875, quantitative) of our method were better than the original BPN (Movie-2, *F*1: 0.542; Movie-1, *F*1: 0.610; EC, *F*1: 0.778; MP3, *F*1: 0.683; Blog, *F*1:0.698). In addition, from the OA and *F*1 results in Table 5, it is evident that the quantitative representation has a better performance using our NN based approach.

Next, we wanted to know if our method could reduce training time. Table 6 summarizes the average processing time of the BPN and NN based methods. From this table it is evident that our method reduced the processing time substantially. On average, compared to the BPN, our method reduced the percentage of processing time by 57.2–73.7%. This significant reduction is very crucial for processing textual data. This is because the size of textual data from the Internet grows rapidly. The huge data size and high dimensionality of textual data will degrade the performance of classifiers and lead to long training time. From results of Table 6, this data size problem might be solved by our proposed method.

Table 5
Results of the proposed neural network based approach.

Data/method	Performance					
	OA			Precision	Recall	*F*1
	Max	Mean	SD			
Movie-2						
BPN	58.0%	53.46%	3.60%	0.500	0.591	0.542
Our method (quantitative)	62%	58.07%	2.72%	0.375	0.709	0.491
Our method (qualitative)	64%	57.78%	3.84%	0.750	0.619	0.678
Movie-1						
BPN	60.0%	55.7%	2.9%	0.585	0.637	0.610
Our method (quantitative)	64.0%	56.8%	4.7%	0.600	1.000	0.750
Our method (qualitative)	58.0%	54.2%	3.8%	0.565	0.693	0.623
EC						
BPN	75.0%	70.8%	3.7%	0.804	0.753	0.778
Our method (quantitative)	79.0%	72.4%	5.5%	0.800	0.824	0.812
Our method (qualitative)	79.0%	73.2%	5.3%	0.800	0.824	0.812
MP3						
BPN	61.7%	58.8%	3.2%	0.692	0.675	0.683
Our method (quantitative)	60.0%	55.4%	3.9%	0.582	0.997	0.735
Our method (qualitative)	61.0%	58.5%	2.3%	0.583	0.673	0.625
Blog						
BPN	75.0%	65.8%	6.2%	0.625	0.789	0.698
Our method (quantitative)	80.0%	64.4%	10.3%	1	0.778	0.875
Our method (qualitative)	70.0%	59.0%	6.2%	1	0.633	0.775

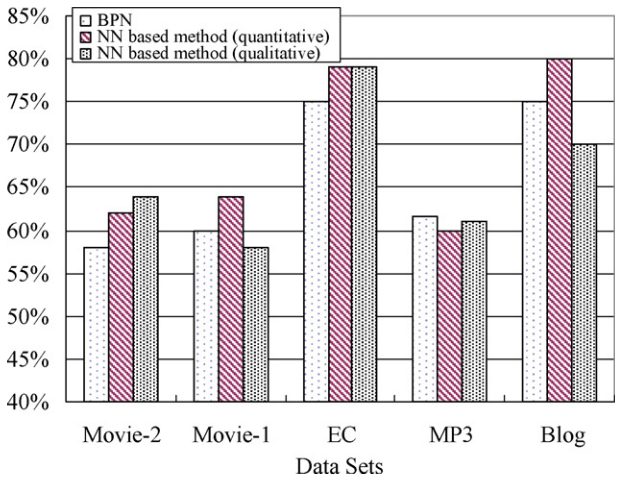

Fig. 3. The comparison between the BPN and the proposed NN based method when considering OA.

Table 6
Summary of the processing time.

Method	Data sets				
	Movie-1 Time (s)	Movie-2 Time (s)	EC Time (s)	MP3 Time (s)	Blog Time (s)
BPN	193.2	172.8	97.9	98.8	92.0
Our method (quantitative)	50.8 (73.7%↓)	53.49 (69.0%↓)	39.87 (59.3%↓)	38.3 (61.2%↓)	30.98 (66.3%↓)
Our method (qualitative)	52.42 (72.9%↓)	43.05 (75.1%↓)	41.88 (57.2%↓)	41.11 (58.4%↓)	36.98 (59.8%↓)

Note: (73.7%↓) denotes the % reduction of the processing time compared with BPN.

5. Discussion

In this section, we attempt to answer the question "If a different SO index set is chosen, will it influence the performance of our proposed method?" To answer this question, we implemented the feature selection technique to identify the key SO index. In our method, we used four SO indexes. We want to know which one is critical for classifying sentiment. Structure pruning of NN, one of the selection techniques, might provide a possible solution. In the training of NN, some input nodes

Table 7
The results of architectural pruning (feature selection).

Input neuron	The sum of the absolute multiplication values				
	Movie-1	Movie-2	EC	MP3	Blog
1: SO-A	6.48	2.55	1.09	0.10	0.57
2: SO-PMI(AND)	0.60	7.26[a]	3.73[a]	2.13[a]	0.71
3: SO-PMI(NEAR)	16.17[a]	2.15	1.97	0.33	0.12
4: SO-LSA	6.67	3.02	2.13	1.39	1.19[a]

[a] An important input neuron.

might be considered as irrelevant and be removed. This is common in the case of (Reed, 1993), where the input attributes are pruned rather than the hidden neurons. Su et al. (Su, Hsu, et al., 2002; Su et al., 2006) attempted to determine the important input nodes of a neural network based on the sum of the absolute multiplication values of the weights between the layers. Only the multiplication weights with large absolute values were kept and the rests were removed. The equation for calculating the sum of absolute multiplication values is thus defined as follows:

$$Node_i = \sum_j |W_{ij} \times V_{jk}| \tag{12}$$

where W_{ij} is the weight between the ith input node and the jth hidden node, and V_{jk} is the weight between the jth hidden node and the kth output node.

Table 7 summarizes the results of the calculation of Eq. (12). From this table it is found that in Movie-2, EC, and MP3, the SO-PMI(AND) is the most important input neuron in the NN based indexes. On the other hand, SO-PMI(NEAR) and SO-LSA are the important input variables in the Movie-1 and Blog data sets respectively. This finding is not consistent with the result shown in Table 4. In Table 4, the results indicate that SO-LSA has the best performance, not SO-PMI (AND).

In the results of feature selection (Table 7), we found that the key SO index is not always the same one in different training data. From the results in Table 4, which describe the classification performance when using a single SO index to classify sentiment, we can draw some conclusions. First, the performance of using a single SO index to classify the sentiment of bloggers depends heavily on the training data. Second, in our proposed method which used the results of several SO indexes as the input of the neural network, the selection of input variables (SO indexes) does not influence the performance of our method. It means that the neural network will adjust its weights. For example, if we choose SO indexes which have poor performances, a neural network will increase its weights proportionally during the iterative training process. In addition, if we choose SO indexes which have good performances, a neural network will decrease its weights accordingly. Therefore, no matter what kind of SO indexes are employed, they won't influence the performance of the NN based SO index. In other words, the kind or the number of SO indexes used is not very important in the proposed NN based SO index.

6. Conclusion

This study proposed an NN based approach to classify sentiment in blogospheres by combining the advantages of the BPN and SO indexes. Compared with traditional techniques such as BPN and SO indexes, the proposed approach shows its superiority not only in classification accuracy, but also in training time.

The reduction of input attributes is an important issue for ML techniques in sentiment classification. With the rapid growth of blogs, the data (bloggers' comments) size shows a remarkable increase over time. The dimensionality size (the number of terms) of textual data will grow exponentially with the growth of the data size. These huge amounts of feature sets and data will be a major problem for ML methods for handling the textual data, and will inevitably result in very long training processes. Therefore, our method could be a possible solution for shortening the processing time and increasing the classification performance. In addition, BPN has the advantage of fault tolerance. This means that even if one of these input SO indexes cannot be obtained, our proposed NN index will still be able to classify the sentiment correctly.

In order to obtain better or more robust results, additional experiments of using different ML approaches such as Support Vector Machines (SVM) and Naïve Bayes are necessary in future researches. In addition, it should also be noted that our proposed method is not only specific to blogs, it can be employed to classify sentiment in any text based communication tool. We just used blogs as an example in this study. Readers can apply the proposed method to any new media such as Twitter, Plurk, Facebook, and so on. Moreover, this study used five textual data sets and all results indicated the superiority of our method. But, to testify the limitations of the proposed method, future works could use different data sets or data types.

Acknowledgements

This work was supported in part by National Science Council of Taiwan (Grant No. NSC 98-2410-H-324-007-MY2).

References

Abbasi, A., Chen, H., Thoms, S., & Fu, T. (2008). Affect analysis of web forums and blogs using correlation ensembles. *IEEE Transactions on Knowledge and Data Engineering*, *20*(9), 1168–1180.

Bartell, B. T., Cottrell, G. W., & Belew, R. K. (1992). Latent semantic indexing is an optimal special case of multidimensional scaling. In *Proceedings of the fifteenth annual international ACM SIGIR conference on research and development in information retrieval* (pp. 161–167).

Boucouvalas, A. C. (2002). Real time text-to-emotion engine for expressive Internet communications. In G. Riva, G. Riva, et al. (Eds.), *Emerging communication: Studies on new technologies and practices in communication* (pp. 305–318). IOS Press.

Chaovalit, P., & Zhou, L. (2005). Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th annual HICSS*

Chau, M., & Xu, J. (2007). Mining communities and their relationships in blogs: A study of online hate groups. *International Journal of Human – Computer Studies*, *65*(1), 57–70.

Chen, L.-F., Su, C.-T., & Chen, M.-H. (2009). A neural-network approach for defect recognition in TFT-LCD photolithography process. *IEEE Transactions on Electronics Packaging Manufacturing*, *32*(1), 1–8.

Chen, L.-S., Hsu, C.-C., & Chen, M.-C. (2009). Customer segmentation and classification from blogs by using data mining: An example of VOIP phone. *Cybernetics & Systems*, *40*(7), 608–632.

Church, K. W., & Hanks, P. (1989). Word association norms, mutual information and lexicography. In *Proceedings of the 27th annual conference of the ACL* (pp. 76–83). New Brunswick, NJ: ACL.

Cohen, E., & Krishnamurthy, B. (2006). A short walk in the Blogistan. *Computer Networks*, *50*(5), 615–630.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *The 12th WWW*.

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the Society for Information Science*, *41*(6), 391–407.

Devillers, L., Luniel, L., & Vasilescu, I. (2003). Emotion detection in task-oriented spoken dialogues. In *The international conference on multimedia and expo* Baltimore, MD, USA, (pp. 549–552).

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *The 2004 SIGKDD* (pp. 168–177).

Hearst, M. A. (1992). Direction-based text interpretation as an information access refinement. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*. Lawrence Erlbaum Associates Inc., pp. 257–274.

Lambiotte, R., Ausloos, M., & Thelwall, M. (2007). Word statistics in blogs and RSS feeds: Towards empirical universal evidence. *Journal of Informetrics*, *1*, 277–286.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Lee, C. M., Narayanan, S. S., & Pieraccini, R. (2002). Combining acoustic and language information for emotion recognition. In *The 7th international conference on spoken language processing* Denver, USA, (pp. 873–876).

Leshed, G., & Kaye, J. (2006). Understanding how bloggers feel: Recognizing affect in blog posts. In *Conference on human factors in computing systems* Montréal, Québec, Canada, (pp. 1019–1024).

Murphy, C. (2006). *Blogging: Waste of time or corporate tool?* Available at http://www.personneltoday.com/Articles/2006/03/21/34506/blogging-waste-of-time-orcorporate-tool.html

Martin, J. (2005). Blogging for dollars. *Fortune Small Business*, *15*(10), 88–92.

Na, J.-C., Khoo, C., & Wu, P. H. J. (2005). Use of negation phrases in automatic sentiment classification of product reviews. *Library Collections, Acquisitions, & Technical Service*, *29*, 180–191.

Nardi, B. A., Schiano, D. J., Gumbrecht, M., & Swartz, L. (2004). Why we blog? *Communications of the ACM*, *47*(12), 41–46.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up? Sentiment classification using machine learning techniques*. EMNLP.

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*, 143–157.

Reed, R. (1993). Pruning algorithms – A survey. *IEEE Transaction on Neural Networks*, *5*, 740–747.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing*. Cambridge: MIT Press and the PDP Research Group.

Singh, T., Veron-Jackson, L., & Cullinane, J. (2008). Blogging: A new play in your marketing game plan. *Business Horizons*, *51*, 281–292.

Su, C.-T., Hsu, J.-H., & Tsai, C.-H. (2002). Knowledge mining from trained neural network. *Journal of Computer Information Systems*, 61–70.

Su, C.-T., Chen, L.-S., & Chiang, T.-L. (2006). A neural network based information granulation approach to shorten the cellular phone test process. *Computers in Industry*, *57*(5), 412–423.

Su, C.-T., Yang, T., & Ke, C.-M. (2002). A neural-network approach for semiconductor wafer post-sawing inspection. *IEEE Transactions on Semiconductor Manufacturing*, *15*(2), 260–266.

Subasic, P., & Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems*, *9*(4), 483–496.

Tang, H., Tan, S., & Cheng, X. (2009). A survey on sentiment detection of reviews. *Expert Systems with Applications*, *36*, 10760–10773.

Tao, J., & Tan, T. (2004). Emotional Chinese talking head system. In *The 6th international conference on multimodal interface* (pp. 273–280).

Turney, P. D., & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, *21*(4), 315–346.

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting of the association for computational linguistics* Philadelphia, PA, USA, (pp. 417–424) (NRC #44946).

Van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

Wang, C., Lu, J., & Zhang, G. (2005). A semantic classification approach for online product reviews. In *The 2005 IEEE/WIC/ACM international conference on web intelligence* (pp. 276–279).

Wu, C.-H., Chuang, Z.-J., & Lin, Y.-C. (2006). Emotion recognition from text using semantic labels and separable mixture models. *ACM Transactions on Asian Language Information Processing*, *5*(2), 165–182.

Yanaru, T. (1995). An emotion processing system based on fuzzy inference and subjective observations. In *The 2nd New Zealand international two-stream conference on artificial neural networks and expert systems* New York, USA, (pp. 15–20).

Zhuang, L., Jing, F., & Zhu, X.-Y. (2006). Movie review mining and summarization. In *The 15th ACM international conference on Information and Knowledge Management* Arlington, Virginia, USA.