Developing a Neural Network based Index for Sentiment Classification

Long-Sheng Chen* and Hui-Ju Chiu

Abstract—**Recognizing emotion is extremely important for some text-based communication tools such as blogs. On commercial blogs, bloggers' negative comments or evaluations of products spread quickly in the cyber space. These negative comments are often harmful to enterprises and might result in great damage. Recently, researchers have paid much attention on sentiment classification to efficiently identify customers' negative emotions for helping companies to carefully response customers' comments. Following this trend, this study proposed a Neural Network (NN) based index which combines the advantages of machine learning techniques and information retrieval (semantic orientation indexes) to help companies detecting harmfully negative bloggers' comments quickly and effectively. Experimental results indicated that our proposed NN based index outperforms traditional approaches, including Back-Propagation neural network (BPN) and several semantic orientation indexes.**

Index Terms—**Semantic Orientation, Neural Networks, Sentiment Classification, Affective Computing, Blogs**

I. INTRODUCTION

Nowadays, the amount of blogs grows rapidly. It was reported that over 600 million blogs have been created [30]. Moreover, the number is still increasing in the speed of 50,000~70,000 per day [31]. Nowadays, blogs have been considered as one of the fastest growing sections of the Internet and are emerging as an important communication mechanism that is used by an increasing number of people [22, 32].

Recognizing emotion is extremely important for some text-based communication tools such as blogs that uses only text input and output. Bloggers (blogs users) often make a record of their lives and express their opinions, feelings, and emotions through writing blogs [23]. One of the most important features in blogs is the ability for any reader to write a comment on a blog entry. This ability has facilitated the interaction between bloggers and their readers [24]. Therefore, bloggers' negative comments or evaluations spread quickly in the cyber space. For commercial blogs, these negative comments toward products are often harmful to enterprises and might cause great damage. Therefore, to identify customers' emotion effectively on blogs is very important, especially for whom those utilize blogs as marketing channels [32]. The additional advantages of recognizing the users' emotional states could be: (1) to enable the dialog system to change the response and answer types [8]. (2) In net conferencing, the ability to identify the users' emotional states can reduce the translation data size and further increase the fluency of the conferencing program [9].

The most direct way of recognizing a bloggers' emotions is to use emotional keywords from text input [19]. Although there are many multimedia data in blogs, text is still the main communication tool on blogs. Moreover, other reasons includes (1) textual data is the most popular medium and it needs merely small storage requirements; (2) textual data is the most appropriate medium for network transmissions; (3) the variety and complexity of textual data makes it possible for people to exchange ideas, opinions, and emotions using text only [4]. Therefore, this study employs textual data to classify sentiments.

Recently, researchers have paid much attention on sentiment classification or affective computing [4, 10] to efficiently identify customers' negative emotions for helping companies to carefully response customers' comments. In related literatures, two popular approaches, machine learning methods and information retrieval (IR) techniques (semantic orientation index), have been employed to address this problem [3]. Readers can find the concepts of these two major methodologies in Figure 1. In the group of machine learning, several approaches have been developed. For example, Abbasi et al. [25] proposed support vector regression correlation ensemble (SVRCE) approach to analyze emotional states. Pang et al [14] investigate several supervised machine learning methods to semantically classify movie reviews. Turney [2] employs a specific unsupervised learning method for the review semantic orientation classification. Dave et al [15] develop a method for automatically classifying positive and negative reviews and experiment several methods related to feature selections and scoring. In the work of Chaovalit and Zhou [3], machine learning methods and semantic orientation index have been presented to classify movie reviewers' comments. The experimental results indicated that machine learning techniques have better performance, but they need additional time to be trained.

The classification accuracy of semantic orientation index is not very high; however, we can obtain the results of sentiment classification very soon. In the group of IR, association, Pointwise Mutual Information (PMI), and Latent Semantic Analysis (LSA) have been employed to measure the similarity between words to classify sentiment. Several works reported that SO index is still a good tool for sentiment classification. For example, Turney and littman [1] employed association to realize semantic orientation. Devillers et al. [6] found the most appropriate emotional

Manuscript received January 7, 2009. This work was supported in part by the National Science Council of Taiwan (Grant No. NSC 96-2416-H-324 -003 -MY2).

L.-S. Chen is an assistant professor of the Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan (e-mail: lschen@cyut.edu.tw).

H. J. Chiu is a graduate student of the Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan (e-mail: s9614621@cyut.edu.tw).

state by calculating the conditional probability between the emotional keywords and the emotional states. Tao and Tan [7] used emotional function words instead of emotional keywords to evaluate emotional states. Hu and Liu [12] use the adjective synonym sets and antonym sets in WordNet to judge semantic orientations of adjectives. Besides IR methods, there are other various ways to classify sentiment. Subasic and Huettner [10] use fuzzy logic to manually construct a lexicon, based on which fuzzy techniques applicable to fuzzy sets are used to analyze the affect of documents.

To sum up, the above mentioned methods either require certain amounts of manual constructions or rely on external structured information sources [11]. In practice, the semantic orientation indexes have poor classification performance. Although machine learning techniques have better classification abilities, they need additional learning time and the information of classes (class labels) which should be determined by domain experts have to be provided before training. Therefore, to avoid these problems and keep their strengths in the same time, this study proposes a Neural Network (NN) based index which combines the advantages of machine learning techniques and semantic orientation indexes to effectively classify sentiment. In our proposed method, BPN has been selected as the basic learner due to its strength of fault tolerance. That means even one of these input SO indexes cannot be obtained, our proposed NN index still can classify the sentiment. Finally a movie reviews data set has been provided to evaluate the effectiveness of our proposed method. Experimental results will indicate that our proposed method can efficiently detect the negative/harmful comments which will bring great damage to enterprises.

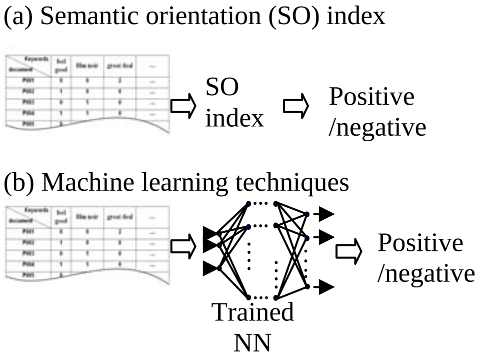

Figure 1 The conventional approaches for sentiment classification

II. Literature Review

In this work, Back-Propagation neural Network (BPN) [26] is employed as the basic learner due to its superior classification ability which has been reported in lots of related works. Besides, in order to combine the advantages of both BPN and SO indexes, the proposed NN based index use 4 different types of SO indexes as the input neurons. The brief introduction regarding BPN and SO indexes can be found in following subsections.

A. Back-Propagation Neural Networks

Neural networks offer exciting 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, shown as Figure 2, was employed because of their superior ability of classification.

Among feed-forward networks, BPN is the best known networks and still one of the most useful. 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. According to the rule of thumb and reports of available published papers, the number of hidden layers should 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 weights and biases according to the difference between the desired and actual network outputs. These two passes will go through iteratively until the network converges. The feed-forward network training by back-propagation can be summarized as the following steps:

Step 1: Select an architecture

Step 2: Randomly initialize weights

Step 3: While error is too large

For each training pattern (presented in random order)

Step 3.1: Select training pattern and feedforward to find actual network output

Step 3.1.1: Apply the inputs to the network

Step 3.1.2: Calculate the output for every neuron from the input layer, through the hidden layer(s), to the output layer

The output from neuron *j* for pattern *p* is O_{pj} where

$$O_{pj}(net_j) = \frac{1}{1+e^{-net_j}} \tag{1}$$

and

$$net_j = bias + \sum_k O_{pk} W_{jk} \tag{2}$$

k ranges over the input indices and W_{jk} is the weight on the connection from input *k* to neuron *j*.

Step 3.2: Calculate errors and backpropagate error signals

Step 3.2.1: Calculate the error at the outputs

The output neuron error signal δ_{pj} is given by

$$\delta_{\mathrm{pj}} = (\mathrm{T_{pj}} - \mathrm{O_{pj}}) \times \mathrm{O_{pj}} \times (1 - \mathrm{O_{pj}}) \tag{3}$$

where T_{pj} is the target value of output neuron *j* for pattern *p* and O_{pj} is the actual output value of output neuron *j* for pattern *p*.

Step 3.2.2: Use the output error to compute error signals for pre-output layers

The hidden neuron error signal δ_{pj} is given by

$$\delta_{pj} = O_{pj}(1 - O_{pj}) \sum_k \delta_{pk} W_{kj}) \tag{4}$$

where δ_{pk} is the error signal of a post-synaptic neuron k and W_{kj} is the weight of the connection from hidden neuron j to the post-synaptic neuron k.

Step 3.3: Adjust weights

Step 3.3.1: Use the error signals to compute weight adjustments

Compute weight adjustments ΔW_{ji} at time t by equation (5) which is defined as bellow.

$$\Delta W_{ji}(t) = \eta \times \delta_{pj} \times O_{pi} + \alpha \times \Delta W_{ji}(t-1) \quad (5)$$

where η is the learning rate and α is the momentum coefficient ($\alpha \in [0,1]$).

Step 3.3.2: Apply the weight adjustments

Apply weight adjustments according to equation (6).

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}(t) \quad (6)$$

Step 4: Evaluate performance using the test data set.

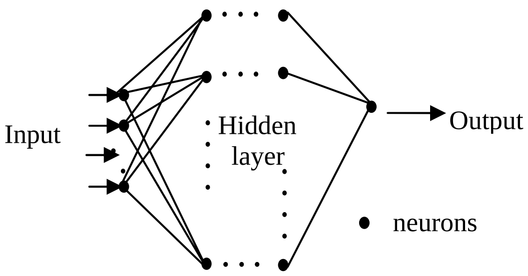

Figure 2 The back-propagation neural network structure

B. Semantic Orientation from Association

The general SO index is to infer semantic orientation from semantic association. The semantic orientation of a given word is calculated from the strength of its association with a set of positive words, minus the strength of its association with a set of negative words:

Pwords= a set of words with positive semantic orientation (7)

Nwords= a set of words with negative semantic orientation (8)

A(word1, word2)= a measure of association between word1 and word2 (9)

$$SO - A(word) = \sum_{pword \in Pwords} A(word, Pword) - \sum_{nword = Nwords} A(word, Nword) \quad (10)$$

There, when A(*word*1, *word*2) is positive, the words tend to be associated with each other. Larger values correspond to stronger associations. When A(*word*1, *word*2) is negative, the presence of one word makes it likely that the other is absent. A word, *word*, is classified as having a positive semantic orientation when SO-A(*word*) is positive and a negative orientation when SO-A(*word*) is negative. The magnitude (absolute value) of SO-A(*word*) can be considered the strength of the semantic orientation.

C. Semantic Orientation from PMI

The PMI-IR algorithm is employed to estimate the semantic orientation of a phrase [2]. PMI-IR uses Pointwise Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of pairs of words or phrases. The PMI between two words, *word*1 and *word*2, is defined as equation (11) [20]:

$$\text{PMI}(\text{wrod}_1, \text{word}_2) = \log_2 \left(\frac{\text{P}(\text{wrod}_1 \text{ Near word}_2)}{\text{P}(\text{wrod}_1)\,(\text{word}_2)} \right) \quad (11)$$

Where, p(*word*1 & *word*2) represents the probability 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(*word*1) p(*word*2). The ratio between p(*word*1 & *word*2) and p(*word*1) p(*word*2) is thus a measure of the degree of statistical dependence between the 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:

$$SO - PMI(word) = \sum_{pword \in Pwords} PMI(wrod, pword) - \sum_{nword \in Nwords} PMI(word, nword) \quad (12)$$

PMI-IR estimates PMI by issuing queries to a search engine (hence the IR in PMI-IR) and noting the number of hits (matching documents). Our study uses the AltaVista Advanced Search engine 5, which indexes approximately 350 million web pages. In addition, except AltaVista NEAR operator, the AND has been employed in this work [1].

D. Semantic Orientation from LSI

SO-LSI applies Latent Semantic Analysis (LSA) [21] to calculate the strength of the semantic association between words [27]. LSA uses the Singular Value Decomposition (SVD) to analyze the statistical relationships among words in a corpus.

First step 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 the chunk [28]. Next, we apply SVD to decompose *A* into a product of three matrices, USV^T shown in Figure 3. Next, we apply SVD to decompose X into a product of three matrices USV, where *U* and *V* are in column orthonormal forms 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 be the matrices produced by selecting the corresponding columns from *U* and *V*. The matrix $U_k S_k V_k^T$ is the matrix of rank *k* which the best approximates of 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* [21, 27, 29]. The semantic orientation of a word, *word*, is calculated by SO-LSI as follows:

$$SO - LSI(word) = \sum_{pword \in Pwords} LSA(word, pword) - \sum_{nword \in Nwords} LSA(word, nword) \quad (13)$$

As with SO-PMI, a word, *word*, is classified as having a positive semantic orientation when SO-LSA(*word*) is positive and a negative orientation when SO-LSA(*word*) is negative. The magnitude of SO-LSA(*word*) represents the strength of the semantic orientation.

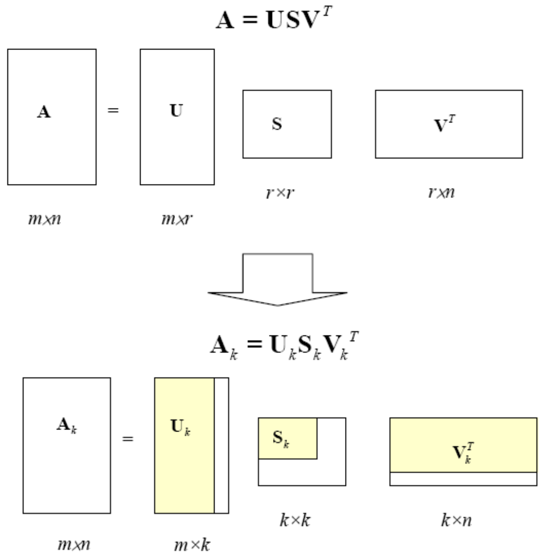

Figure 3 The singular value decomposition [21]

III. THE PROPOSED NN BASED INDEX

A. *Proposed NN based index*

In this section, we will introduce the proposed NN based index for sentiment classification. As shown in Figure 4, the implementation of NN based SO index can be divided into 4 steps. These four steps can be demonstrated as bellow.

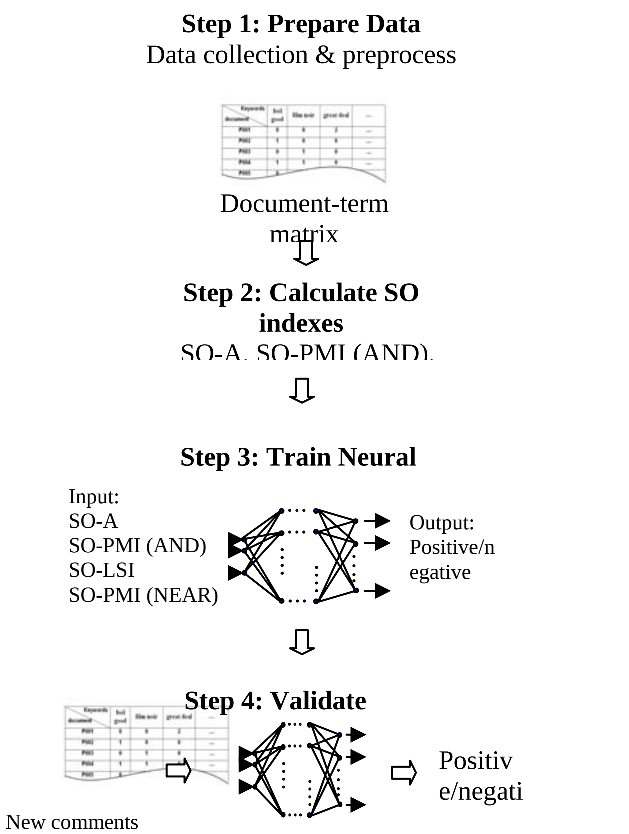

Figure 4 The implemental procedure of the proposed NN based index

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

Keywords / document	feel good	film noir	great deal	…
P001	0	0	2	…
P002	1	0	0	…
P003	0	1	0	…
P004	1	1	0	…
P005	0			

Figure 5 A part of the document-term matrix

Step 1: Prepare data

In this step, we need to segment words to construct a document-term matrix for further analysis. Not all the words in sentences are useful for classifying semantic orientations or related tasks. As Hu and Liu [12] mentioned, nouns and noun phrases in the 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 [2, 3, 11], 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. Then, these extracted keywords can be utilized to describe our experimental data. As shown in Figure 5, we record the occurrence frequency of every single keyword in each comment (document). Finally, a document-term matrix has been completed to be analyzed further.

Step 2: Calculate SO Indexes

In this study, we use four SO indexes including SO-A, SO-PMI(AND), SO-PMI(NEAR), SO-LSI as the input neurons of BPN. Therefore, the first step of our method is to calculate these SO indexes by following the equations described in section 2.2.

Step 3: Train Neural Network

The experimental data set is divided into training and test sets. This step begins training process of BPN by using training data set. All optimal setting of BPN includes parameters and structure such as learning rate, training iterations, number of hidden neurons and so on are obtained by a trial-and-error experiment.

Step 4: Validate the results

In general, we need to use the test data to evaluate the performance of BPN. However, in order to compare with the results of 4 SO indexes, we employ all experimental data to validate the built BPN.

IV. IMPLEMENTATIONS

A. *Data Preparation*

A movie reviews database available at http://www.cs.cornell.edu/people/pabo/movie-review-data/

is employed to evaluate our proposed index. This database totally involves 2,000 text files. 1,000 of them are labeled as positive comments and the rests are negative reviews. For the purpose of being simple, we merely select 100 comments at random to be our experimental corpus, in which 69 are positive examples and 31 negative reviews.

In addition, we employed the shareware Rubryx which can be downloaded at the website (http://www.sowsoft.com/rubryx) to segment words in this study. Rubryx segments words based on n-gram (unigram, bigrams, and tri-grams) features. Before extracting n-gram key words, some frequently used stop words should be removed. Readers can find a useful stop word list at http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_ words. Finally, we extracted 33 keywords to describe the collected data. A 100×33 document-term matrix can be constructed after data preparation phase.

B. Experimental Results

Table 2 summarizes the results of 4 SO indexes. In SO-A, some examples can be identified due the insufficient data size. That's also the weakness of SO-A. Besides, from Table 2, SO-LSI has the best classification ability compared with results of rests. However, the accuracy is 57% which is not good enough.

Table 2 Results of four SO indexes

Performance / Index	Accuracy (%)	Error rate (%)	Unidentified (%)
SO-Association	22	16	62
SO-PMI-And	36	64	-
SO-PMI-Near	42	58	-
SO-LSI	57	43	-

Table 3 The parameter setting and structure of the proposed NN based index

% of Training set	Structure	Learning rate	Iterations
50%	4-4-1	0.2	100
60%	4-3-1	0.1	100
70%	4-4-1	0.01	100
80%	4-4-1	0.01	100
90%	4-4-1	0.01	100

Table 4 Results of the proposed Neural Network based Semantic Orientation Index.

Performance / Training set	Accuracy (%)	Error rate (%)
50%	69	31
60%	70	30
70%	**71**	**29**
80%	70	30
90%	**71**	**29**
Average	**70.2**	**29.8**

The NN based index is programmed in Matlab 6.1 environment. The parameter setting and structure can be found in Table 3. In order the find the best performance, we try different proportion (50%~90%) of all examples to be training data set in turn. For example, in Tables 3~5, "50%" represents that we use 50 examples (full data size is 100) to build model.

Table 4 lists the experimental results of the proposed NN based index. In 70% and 90% data sets, we have a highest accuracy 71%. The results of BPN which uses 33 original keywords as inputs can be found in Table 5. The best performance is 69%. The classification performance of NN based index (71%) is slightly better than BPN (69%). In average, the proposed NN based index (70.2%) indeed outperforms BPN (66%) without considering the length of training time.

Table 5 The results of BPN

Performance / Training set	Accuracy (%)	Error rate (%)
50%	61	39
60%	64	36
70%	68	32
80%	68	32
90%	**69**	**31**
Average	**66**	**34**

V. DISCUSSIONS AND CONCLUSIONS

There are four different SO indexes as the input neurons in our proposed method. We are interested in which one is critical for classifying sentiment. The structure pruning of NN might provide a possible solution. In the training of neural networks, some input nodes might be considered as irrelevant and then be removed. That's common in the case of [18], where the input attributes are pruned rather than the hidden neurons. Su et al. [16, 17] attempted to determine the important input nodes of a neural network based on the sum of absolute multiplication values of the weights between the layers. Only the multiplication weights with large absolute values are kept and the rests are removed. The equation for calculating the sum of absolute multiplication values is defined as follows.

$$Node_i = \sum_j \left| W_{ij} \times V_{jk} \right| \quad (14)$$

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 6 The results of architectural pruning

Input neuron	The sum of absolute multiplication values
1: SO-Association	0.2701
2: SO-PMI-AND	0.1712
3: SO-PMI-NEAR*	0.4162*
4: SO-LSI	0.0718

Note: "*" represents important input neuron

Table 6 summarizes the results of calculation. From this table, the SO-PMI-NEAR is the most important input neuron in NN based index. This finding is not consistent to the result in Table 1 (In table 1, SO-LAI has the best performance, not SO-PMI-NEAR). The reason might be that NN will adjust the weights between neurons to find an optimal solution during the iterative training process.

Therefore, no matter what kind of SO indexes is employed, they won't influence the performance of NN based SO index. In other words, using what kinds or how many SO indexes are not very important in the proposed NN based SO index.

Combining the advantages of BPN and SO indexes, this study proposes an NN based index which to classify sentiment. Compared with SO indexes, our proposed index shows its superiority in classification accuracy. In addition, because the proposed NN based index uses 4 SO indexes as inputs of BPN, it can dramatically shorten the long training time which is major disadvantage of BPN. Moreover, BPN has the advantage of fault tolerance. That means even one of these input SO indexes cannot be obtained, our proposed NN index still can classify the sentiment correctly.

In experiments, the dimensional size (the number of terms) of textual data grows greatly when we increase the data size. It will result in very long training process to machine learning techniques. Therefore, reduction of input attributes should be an important issue in machine learning techniques. In order to obtain better or more robust results, additional data sets and other experiments of using different machine learning approaches are necessary to further researches.

ACKNOWLEDGMENT

This work was supported in part by National Science Council of Taiwan (Grant No. NSC 96-2416-H-324 -003 -MY2).

REFERENCES

[1] Turney, P.D., and Littman, M.L. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS). 21, 4, 315-346.

[2] Turney, P.D. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, Pennsylvania, USA, 417-424 (NRC #44946).

[3] Chaovalit, P. and Zhou, L., 2005. Movie review mining: a comparison between supervised and unsupervised Classification Approaches. Proceedings of the 38th Annual HICSS.

[4] 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.

[5] Yanaru, T. 1995. An emotion processing system based on fuzzy inference and subjective observations. The 2nd New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems (Dunedin, N Z., Nov. 20-23). IEEE Computer Society Press, New York, 15-20.

[6] Devillers, L., Luniel, L., and Vasilescu, I. 2003. Emotion detection in task-oriented spoken dialogues. The International Conference on Multimedia and Expo (Baltimore, MD, July 6-9). 549-552.

[7] Tao, J. and Tan, T. 2004. Emotional Chinese talking head system. The 6th International Conference on Multimodal Interface (Oct. 13-15). 273-280.

[8] Lee, C.M., Narayanan, S.S., and Pieraccini, R. 2002. Combining acoustic and language information for emotion recognition. The 7th International Conference on Spoken Language Processing (Denver, CO). 873-876.

[9] Boucouvalas, A.C. 2002. Real time text-to-emotion engine for expressive internet communications. In Emerging Communication: Studies on New Technologies and Practices in Communication. G. Riva et al. eds. IOS Press. 305-318.

[10] Subasic, P., Huettner, A., 2001, Affect analysis of text using fuzzy semantic typing, IEEE Transactions on Fuzzy Systems, Volume: 9, Issue: 4, page(s): 483-496.

[11] Wang, C., Lu, J., Zhang, G., 2005, A semantic classification approach for online product reviews, Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on 19-22 Sept. 276 – 279.

[12] Hu, M. and Liu, B. 2004. Mining and summarizing customer reviews. The 2004 SIGKDD, 168-177.

[13] 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., 257-274

[14] Pang, B., Lee, L. and Vaithyanathan, S. 2002. Thumbs up? sentiment classification using machine learning techniques. EMNLP 2002.

[15] Dave, K., Lawrence, S. and Pennock, D. M. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. The 12th WWW.

[16] Su, C.-T., Hsu, J.-H. and Tsai, C.-H. 2002. Knowledge mining from trained neural network. Journal of Computer Information Systems, 61-70.

[17] Su, C.-T., Chen, L.-S., and 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.

[18] Reed, R. 1993. Pruning algorithms- a survey. IEEE Transaction on Neural Networks. 5, 740-747.

[19] Zhuang, L., Jing, F., Zhu, X.-Y. 2006. Movie review mining and summarization, The 15th ACM international conference on Information and Knowledge Management, November 06-11, 2006, Arlington, Virginia, USA

[20] Church, K.W., and Hanks, P. 1989. Word association norms, mutual information and lexicography. Proceedings of the 27th Annual Conference of the ACL (pp. 76-83). New Brunswick, NJ: ACL.

[21] Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. 1990. Indexing by latent semantic analysis. Journal of the Society for Information Science. 41, 6, 391-407.

[22] Cohen, E., and Krishnamurthy, B. 2006. A short walk in the Blogistan. Computer Networks. 50, 5, 615-630.

[23] Nardi, B. A., Schiano, D. J., Gumbrecht, M., Swartz, L. 2004. Why we blog? Communications of the ACM. 47, 12, 41-46.

[24] Chau, M. and 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.

[25] 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.

[26] Rumelhart, D. E., and McClelland, J. L. 1986. Parallel distributed processing, MIT Press and the PDP Research Group, Cambridge.

[27] Landauer, T.K., and 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.

[28] van Rijsbergen, C.J. 1979. Information Retrieval (2nd edition), Butterworths, London.

[29] Bartell, B.T., Cottrell, G.W., and Belew, R.K. 1992. Latent semantic indexing is an optimal special case of multidimensional scaling. Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 161-167.

[30] Murphy, C. 2006. Blogging: Waste of time or corporate tool? Retrieved August 12, 2006, fromhttp://www.personneltoday.-com/Articles/2006/03/21/34506/blogging-waste-of-time-orcorporate- tool.html.

[31] Martin, J. 2005. Blogging for dollars. FSB: Fortune Small Business, 15(10), 88-92.

[32] Singh, T., Veron-Jackson, L., Cullinane, J. 2008. Blogging: A new play in your marketing game plan. Business Horizons 51, 281-292.