2008

An ontology-based sentiment classification methodology for online consumer reviews

Jantima Polpinij
University of Wollongong, jp989@uow.edu.au

Aditya K. Ghose
University of Wollongong, aditya@uow.edu.au

An ontology-based sentiment classification methodology for online consumer reviews

Abstract

This paper presents a method of ontology-based sentiment classification to classify and analyse online product reviews of consumers. We implement and experiment with a support vector machines text classification approach based on a lexical variable ontology. After testing, it could be demonstrated that the proposed method can provide more effectiveness for sentiment classification based on text content.

Disciplines

Physical Sciences and Mathematics

Publication Details

Polpinij, J. & Ghose, A. K. (2008). An ontology-based sentiment classification methodology for online consumer reviews. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (pp. 518-524). IEEE Explore Online: ieeexplore.ieee.org: Institute of Electrical and Electronics Engineers (IEEE).

2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

An Ontology-based Sentiment Classification Methodology for Online Consumer Reviews

Jantima Polpinij and Aditya K. Ghose
Decision Systems Laboratory
School of Computer Science and Software Engineering, Faculty of Informatics,
University of Wollongong, Wollongong 2500 NSW, Australia
{*jp989*, *aditya*}*@uow.edu.au*

Abstract

This paper presents a method of ontology-based sentiment classification to classify and analyse online product reviews of consumers. We implement and experiment with a support vector machines text classification approach based on a lexical variable ontology. After testing, it could be demonstrated that the proposed method can provide more effectiveness for sentiment classification based on text content.

Keywords**:** *support vector machines, lexical variable ontology, online product reviews, sentiment classification.*

1. Introduction

Modern business has been strongly geared towards customers through the WWW. Therefore, with the growth of technology, electronic commerce (e-commerce) appears in society and rapidly grows. Based on this, in particular business-to-consumer (B2C) has been explosive during the last few years because it has an influence on the success of business. At present, many companies have been moving their markets to the Internet because it is a new way to directly and any more easily present information to the customers and improve organizational effectiveness. In Simon's research [1], he referred to www.CIO.com that its information reported that Internet shoppers will spend between $21 and $57 billion for on-line sales in the year 2000 and it is increased to over $380 billion by the year 2003.

At present, sentiment classification [2, 3] has become a significant research area for e-commerce system. This is because many web sites have emerged that offer reviews of items like books, cars, snow tires, vacation destinations, etc. They describe the items in some detail and evaluate them as good/bad, preferred/not preferred. Thus, there is motivation to categorize these reviews in an automated way by a property other than topic, namely, by what is called their 'sentiment' or 'polarity'. That is, whether they recommend or do not recommend a particular item. One speaks of a review as having positive or negative polarity. Therefore, if automatic categorization by sentiment system works effectively, they may have many approached. The first, it can help users quickly to classify and organize such as on-line reviews of goods and services, political commentaries, etc. The second, categorization by sentiment can help businesses to handle 'form free' customer feed-back. They can use it to classify and tabulate such feedback automatically and can thereby determine, for instance, the percentage of happy clientele without having actually to read any customer input. Not only businesses but governments and non-profit organizations might benefit from such an application. The third, categorization by sentiment can also be used to filter email and other messages. A mail program may use it to eliminate so-called 'flames'. In final, this idea is suitable motivation to look at the possibility of automated categorization by sentiment.

An early study can be found in [4, 5]. The movie and product reviews have been the main focus of many of the recent studies in this area. Typically, these reviews are classified at the document level, and the class labels are "positive" and "negative". The expansion of the label set is also motivated by real world concerns; while it is a given that review text expresses positive or negative sentiment, in many cases it is necessary to also identify the cases that do not carry strong expressions of sentiment at all. Pang et al [4, 5] limits the domain to documents that humans have classified as clearly positive or negative. It does not attempt to rank documents on a spectrum. The methods include two probabilistic approaches, both

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.68

more involved than that presented here, and a support vector approach that creates vectors describing training documents and finds a hyperplane that best separates them. The best accuracy reported by these authors is 82.9% correctly classified. Turney [6] has worked on a similar task, tries an interesting method: using a Web search engine to find associations between various words and the words "poor" and "excellent," classifying words that co-occur frequently with "poor" and infrequently with "excellent" to be negative sentiment terms, and vice versa. Although this research achieves impressive 84.0% accuracy on automotive reviews, his attempt at classifying movie reviews logged a lackluster 65.8% accuracy. This mentions that "descriptions of unpleasant scenes" could be hampering the movie review results. This is not surprising, because his sentiment data is gleaned from a web search of general documents, where words might be used very differently than in movie reviews – not to mention the dubious choice of the word "poor" as the flag for negative sentiment, when the word is frequently used in the economic sense. In addition, Na Jin Cheon et al [7] have proposed the sentiment classification to classify product reviews into "recommended" or "not recommended" (downloaded from Review Center at http://www.reviewcentre.com/). They have used the several text features investigated such as baseline (unigram), selected words (such as verb, adjective, and adverb), words labeled with Part of Speech (POS) Tags, and Negation Phrases to group data of product reviews based on Support Vector Machines (SVMs). Finally, the negation phrase approaches report the highest accuracy, since they are separated negative phrase approaches to two models: unigram with negation phrase and DF 3, and unigram with negation phrase and DF 1. These models achieve impressive 78.33% accuracy and 79.33% accuracy respectively on automotive reviews.

However, although text classification techniques can be applied to sentiment classification, it may be argued [8] that this technique is not ripe enough to be used in the specification problem because the domain knowledge of text classification strongly depends on the particular task. It is hard to transfer the same knowledge to a variety of domains of interest.

A possible solution to solve this problem is to use ontology. Many researchers believe ontology-based [9, 10] can bring about an improvement in this case because an ontology represents a shared understanding of the domain of interest. This can enhance the performance of information processing systems.

Therefore, this paper presents a method of ontology-based sentiment classification to classify and analyse online product reviews. The results can help to understand why some people choose the products. We implement and experiment our assumption with Support Vector Machine based on the lexical variation ontology.

This paper is organized as follows. Section 2 is a lexical variation ontology acquisition. Text classification method and ontology-driven in text classification is presented in Section 3. Afterwards, the experimental results are given in Section 4. Finally, a conclusion is drawn in Section 5.

2. A Lexical Variation Ontology

Traditional grammar classifies words based on eight parts of speech: the verb, the noun, the pronoun, the adjective, the adverb, the preposition, the conjunction, and the interjection. Then, the original form of the noun and the verb parts of speech can be changed. For example, the word 'block' can be changed into the word 'blocks'. This can lead to unclear analysis in a text classifier, resulting in the accuracy of the text classifier model being decreased. Thus, this part aims to present a method of lexical ontology acquisition that concentrates on the variation of the noun and the verb. It is called the *lexical variation ontology*. In the experiment, we use three resources for ontology construction: dictionary, irregular verb, and raw texts. Moreover, we used datasets of 20 newsgroups and 5000 web pages gathered from the WWW. The method can be shown in Figure 1 and can be explained as follows:

2.1. Tokenization

The method is started with the tokenization (or word segmentation) process. It is the first and obligatory task in natural language processing because word is a basic unit in linguistics. In English, it can be delimited by white space or punctuation.

2.2. Morphological Analysis

This process can start with considering words as primitive units. This process uses two statistical methods. A shortest matching algorithm is applied to extract the sequence of minimum-length sub-words for each word in the training corpus. The shortest match directive forces the multicharacter expression to match the shortest possible string that is in a root form of each word. Let c be a character in each word, by the chain rule of probability, we can write the probability of any sequence as

$$P(c_1 c_2 ... c_T) = \prod_{i=1}^{T} P(c_i \mid c_1 ... c_{i-1}) \quad (1)$$

As this, it is to estimate a probabilistic alignment between inflected forms and root forms. In this case, we employ the inflected word based on the longest matching algorithm (or greedy matching) [11]. This algorithm starts with a word span that could be another word. The method scans an input word from left to right and selects the longest match with a basic word entry at each point. In case the selected match cannot lead the algorithms to find the rest of words, the algorithm will *backtrack* to find the next longest one and continue finding the rest and so on. After string matching between forms and root forms, the surplus character of each word is a suffix of each original word. The dictionary and Irregular verbs must be used in the learning process.

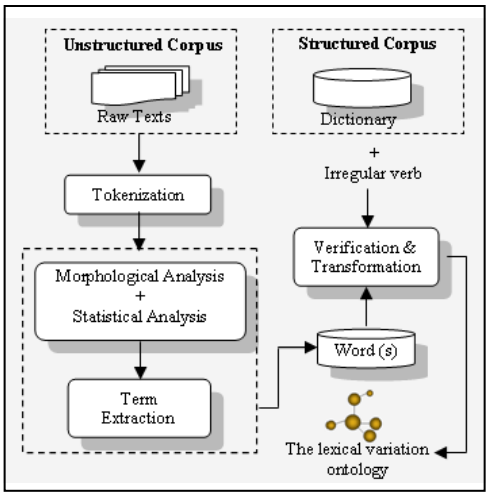

Figure 1. Acquisition of the Lexical Variation Ontology

2.3. Term Extraction

We employ the Apriori association algorithm to tagging the term co-occurring between original words and their suffixes. The Apriori algorithm [12] is a technique in association rule mining. The major steps in association rule mining are: (1) Frequent Itemset generation and (2) Rules derivation. An itemset is a collection of one or more items. A rule can be generated by the following:

$$\text{Association rule: } X \rightarrow Y \quad (2)$$

The Association Rules technique needs two parameters, minimum Support Threshold (minSupp) and minimum Confident Threshold (minConf). The minSupp is used for generating frequent itemsets and minConf is used for rule derivation. MinSupp and minConf can be expressed as follows:

$$\text{Support } (X \rightarrow Y) = P(X \cap Y) \quad (3)$$

$$P(X \cap Y) = \frac{\#(\text{words which are a sequence of word } X \text{ and suffix } Y)}{\#\{\text{words}\}}$$

$$\text{Confidence } (X \rightarrow Y) = P(Y \mid X) \quad (4)$$

$$P(Y \mid X) = \frac{\#(\text{word which is a sequence of word } X \text{ and suffix } Y)}{\#\{\text{word } X\}}$$

The Apriori algorithm uses the downward closure property to prune unnecessary branches for further consideration. This work uses the confidence value at 0.9, while the support value is 0.05.

2.4. Verification and Transformation

This stage is to verify each word that is extracted in the corpus. Afterwards, these words are transformed to XML format. The current structure of the lexical entry of the lexical variation ontology can be decomposed into three information parts: morphological, syntactic, and semantic information. The morphological information indicates a pattern of word composition, while syntactic information contains information of grammatical classification and word variation. Semantic information provides a set of *logical constraints* which can be referred to a word or any class. The logical constraints are capable of dealing with the absence of relatedness of word meanings. There are three types of logical constraints: ISA (a kind-of), EQU (synonyms), and NEQ (antonyms). ISA is a conceptual class of a given word, while EQU is a set of words which have a similar meaning to a given word. NEQ is a set of words which have the opposite meaning to a given word. Finally, our ontology contains a set of distinct and identified concepts C that relates with a set of relations R. Suppose that our dictionary D_L is an association of ontology concepts C with vocabularies set W_L that is concerned with a language L. We denote this by D_L: $C \rightarrow W$. Indeed, the concept C is labeled by a set of vocabularies $w_1, w_2, .., w_n$ in the language L. That means, $D_L(c) = \{w_1, w_2, .., w_n\}$. In addition, we determine the mutual relation R_L: $W_L \rightarrow C$ by $S_L(w) = \{c \in C \mid w \in D_L(c)\}$. Finally, the word w indicates the concepts $c_1, c_2, .., c_n$. We also denote $R_L(w) = \{c_1, c_2, .., c_n\}$. We adapt the notation from [13] to express the structure of the representation in the form following.

$$h\,[c_1 \rightarrow \{w_{11}, w_{12}, ...\}; c_2 \rightarrow \{w_{21}, w_{22}, ...\}] \quad (5)$$

Word	02034 Block	# Identify concept ID to linking # Word
Morphological information	Single	# Word formation – Single word
Syntactic information	Class{V} Suffix{s, ed}	# Grammatical classification of word # Word variation
Semantic information	ISA{prevent} EQU{stop} NEQ -	# A conceptual class of a given word # A word that has the similar meaning of a given word # A word that has the opposite meaning of a given word

Figure 2. An example entry of the word "block"

3. The Research Methodology

This paper applies text classification as sentiment classifier based on ontology to analyse the products' reviews. The research method can be expressed as follows:

3.1. Reviews Document Representation

Before sentiment classifier is built, the training set must be tokenized based on our ontology and represented in a structured "*bag of words* (BOW)" (also known as vector space model format). We obtain $w = (w_1, w_2,\ldots,w_k,\ldots,w_v)$, where ν is the number of unique words within the collection. In the BOW, a product review document d_i is composed of a sequence of words, with $di = (w_{i1}, w_{i2},\ldots, w_{ik}, \ldots,w_{iv})$, where w_{ik} is the frequency of the *k*-th word in the product reviews document d_i. After parsing the product reviews collection to extract unique words, stopwords and a word that occurs only one are removed. Stopwords are words which are considered as non–descriptive within a BOW approach. They typically comprise prepositions, articles, etc. These words usually have very high frequency in the total corpus, and are removed prior to classification. Following common practice, we removed stopwords by using a standard list with 571 stopwords[1].

Finally, each word is weighted by *TF-IDF* [14]. It is used for providing a pre-defined set of features for exchanging information.

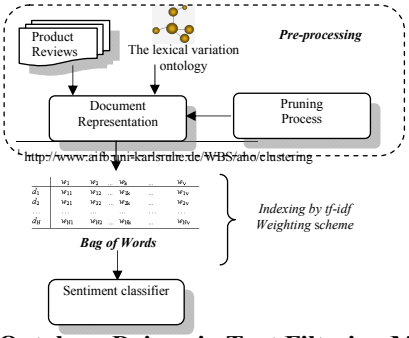

Figure 3. Ontology-Driven in Text Filtering Method

For *TF-IDF*, each unique word w_i corresponds to a feature with *TF* (w_i, d_i), the number of times word w_i occurs in the document d_i, as its value. Refining the document representation, it has been shown that scaling the dimensions of the feature vector with their *inverse document frequency IDF* (w_i) leads to improved performance. *IDF* (w_i) can be calculated from the document frequency *DF* (w_i), which is the number of documents in which word w_i occurs. It is described as follows.

$$IDF(w_i) = 1 + log(|D| / DF(w_i)) \quad (6)$$

3.2. Pruning a BOW Size

Sentiment classification traditionally focuses on improving the learning capabilities of classifiers. Nevertheless, the effectiveness of classification is limited by the suitability of document representation. Intuitively, the more features that are used in representation, the more comprehensive that documents are represented. However, if a representation contains too many irrelevant features, the classifier would suffer from not only the curse of high dimensionality, but also over-fitting. Therefore, if some words occur very rarely and cannot be regarded as statistical evidence, they can be removed prior to classification as *rare words*. For a pre-defined threshold δ, a term word *t* is discarded from the representation, if $tf\text{-}idf(t) < \delta$.

This work applied the Mixed Min and Max model (MMM) to find the threshold δ that is a minimum term word weighting of the BOW. This technique is based on the concept of fuzzy set proposed by Zadeh [15]. The MMM has been developed by Fox and Sharat [15]. There are two operations in the MMM: union and intersection. The union operation is used for finding a minimum, while the intersection operation is to find a maximum. Let *T* be the BOW and *t* be the term words that are weighted based on *tf-idf*. It can be determined that $t \in T$. The degree of membership for union and intersection are defined as follows:

$$T_{t_1 \cup t_2 \cup \ldots t_n} = \max(t_1, t_2, \ldots, t_n) \quad (7)$$

$$T_{t_1 \cap t_2 \cap \ldots t_n} = \min(t_1, t_2, \ldots, t_n) \quad (8)$$

The MMM algorithm attempts to soften the Boolean operation by considering the range of terms weight as a linear combination of the minimum and maximum term weighting. In this work, we interest only the minimum of the MMM. They can be computed as follows:

$$Min\ (\delta) = C_{or1} * max\ (tf\text{-}idf) + C_{or2} * min\ (tf\text{-}idf) \quad (9)$$

$$Max\ (\delta) = C_{and1} * max\ (tf\text{-}idf) + C_{and2} * min\ (tf\text{-}idf) \quad (10)$$

where C_{or1}, C_{or2} are "soften" coefficients of "*or*"operator, and C_{and1}, C_{and2} are softness coefficients of "*and*" operator. To give the maximum of the document weight more importance while considering "*or*" query and the minimum of the document more importance while considering "*and*" query. In general, they have $C_{or1} > C_{or2}$ and $C_{and1} > C_{and2}$. For simplicity, it is generally assumed that $C_{or1} = 1 - C_{or2}$ and $C_{and1} = 1 - C_{and2}$. The best performance usually occurs with C_{and1} in the range [0.5, 0.8] and $C_{or1} > 0.2$ [16]. In this our experiment, we select 0.5 of coefficients. Finally, we can get the threshold δ.

3.3. Text Classifier as Sentiment Classifier

Sentiment classification is closely related to categorization and clustering of text. Traditional automatic text classification [17, 18] systems are used for simple text and so may be applied to text filtering. The basic concept of text categorization may be formalized as the task of approximating the unknown target function Φ: D x $C \rightarrow \{T, F\}$ by means of a function Φ: D x $C \rightarrow \{T, F\}$ - called the classifier - where $C = \{c_1, c_2, \ldots, c_{|C|}\}$ is a predefined set of categories, and D is a set of documents. If $\Phi(d_i, c_j) = T$, then d_i is called a positive member of c_j, while if $\Phi(d_i, c_j) = F$, it is called a negative member of c_j. The majority approaches to text classification are machine learning algorithms [18] and the support vector machines algorithm is applied in this work. The basic concept of SVM [17] is to build a function that takes the value +1 in a "relevant" region capturing most of the data points, and -1 elsewhere. In addition, let Φ: $\Re^N \rightarrow$ F be a nonlinear mapping that maps the training data from $\Re$N to a feature space F. Therefore, the dataset can be separated by the following primal optimization problem:

$$Minimize: \quad \nu(w, \xi, \rho) = \frac{\|w\|^2}{2} + \frac{1}{vl} \sum_{i=1}^{l} \xi_i - \upsilon \quad (11)$$

$$Subject\ to: \quad (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \quad (12)$$

where $\nu \in \{0,1\}$ is a parameter which lets one control the number of support vectors and errors, ξ is a measure of the mis-categorization errors, and ρ is the margin. When we solve the problem, we can obtain w and ρ. Given a new data point x to classify, a label is assigned according to the decision function that can be expressed as follows:

$$f(x) = sign\ ((w \cdot \Phi\ (x_i) - \rho) \quad (13)$$

where α_i are Lagrange multipliers and we apply the Kuhn Tucker condition. We can set the derivatives with respect to the primal variables equal to zero, and then we can get:

$$W = \Sigma \alpha_i \cdot \Phi(x_i) \quad (14)$$

There is only a subset of points x_i that lies closest to the hyperplane and has nonzero values α_i. These points are called support vectors. Instead of solving the primal optimization problem directly, the dual optimization problem is given by:

$$Minimize: \quad W(\alpha) = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \quad (15)$$

$$Subject\ to: \quad 0 \leq \alpha_i \leq \frac{1}{vl}, \sum_i \alpha_i = 1 \quad (16)$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = (\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j))$ are the kernels functions performing the non-linear mapping into the feature space based on dot products between mapped pairs of input points. They allow much more general decision functions when the data are nonlinearly separable and the hyperplane can be represented in a feature space. The kernels frequency used is polynomial kernels $K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i \cdot \mathbf{x}_j)+1)^d$, Gaussian or RBF (radial-basis function) kernels $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$. We can eventually write the decision from equation (16) and (17) and the equation can be illustrated as follow:

$$f(x) = sign\ (\sum \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho) \quad (17)$$

For SVM implementation, we use and modify *LIBSVM* tools from the National Taiwan University [19] in our experiments, since we select the RBF kernels for model building.

3.4. Ontology-driven in Sentiment Classification Model

In this work, sentiment classifier building and testing utilizes the lexical variations and synonyms in the ontology. In this way, these have identical weights. For instance, suppose word-1 is a synonym or variation of word-2, and that the weight of word-2 has been calculated. The weight of word-1 can be obtained from the weight of word-2. Thus, although the online product reviews are written using different words which may share the same meaning, sentiment classifier can still be analyzed.

4. The Experimental Results

This section firstly presents the experimental results of the SVM sentiment classifier. We evaluated the results of the experiments for sentiment classifier by using the information retrieval standard [20]. Common performance measures for system evaluation are *precision* (*P*), *recall* (*R*), and *F-measure* (*F*). Recall is an estimator for the "*degree of how many documents of a class are classified correctly*" and precision is an estimator for the "*degree that if a document is assigned to the class*, *this assignment will be correct*". They are described as follows.

$$Precision = \frac{\#\ Classes\ found\ and\ correct}{Total\ classes\ found} \quad (18)$$

$$Recall = \frac{\#\ Classes\ found\ and\ correct}{Total\ classes\ correct} \quad (19)$$

The F-measure [18] is a harmonic mean and it is a combination of precision and recall. It can be calculated as follow:

$$F\text{-}measure = \frac{2 \times P \times R}{P + R} \quad (20)$$

Finally, performance measures in classification can be defined in Table 1 when we run experiments with the online product reviews dataset that is gathered from http://www.reviewcentre.com/. We randomly select 20,000 the online product review documents for training the sentiment classifier models and 6,000 documents for testing. Finally, the results are estimated by precision, recall, and F-measure can be presented as follows.

Table 1. The results of the SVM sentiment classifier.

ALGORITHMS	Precision	Recall	F-MEASURE
SVM	0.93	0.97	0.949

Using the sentiment classifier models, we were able to classify with good accuracy after testing by F-measure. In this way, it could be said that numbers of feature word having the highest accuracy with the class variables are retained.

After the initially collected the online product review documents are classified into various clusters, we also tested the SVM sentiment classifier as sentiment classification model to analyze in each online product review document. Suppose that a sentence in each document is equal to one document. Therefore, if a sentence in the document is in the non-relevant class, it must be removed from the extraction analysis domain. Based on this, it helps to reduce the domain size of content in a requirement specification document. In addition, we also estimated the common evaluation of the SVM sentiment classifier by using accuracy rates. Let *FP* be a *false positive* or α *error* (also known as Type I error) and *FN* be a *false negative* or β error (also known as Type II error). *TP* is a *true positive* and *TN* is a *true negative*. Then, a *FP* normally means that a test claims something to be positive, when that is not the case, while *FN* is the error of failing to observe a difference when in truth there is one. The accuracy can be calculated as follow:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (21)$$

The results can be presented in Table 2.

Table 2. The accuracy of the SVM sentiment classifier.

ALGORITHMS	Accuracy (%)
SVM	96.00

As the results, The SVM sentiment classifier also shows a satisfactory accuracy because the solution of the SVM method gives an optimal hyperplane, which is a decision boundary between non-relevant and relevant information. The effectiveness of the SVM sentiment classifier model can be increased with a small bag of words that consists of suitable features.

5. Conclusion

This paper presents a method of ontology-based sentiment classification to classify and analyze the online product reviews. We implement and experiment our assumption with Support Vector Machine based on the lexical variation ontology. This work applied text classifier as the sentiment classifier. Then the sentiment classifier building and testing utilizes the lexical variations and synonyms in the ontology. In this way, these have identical weights. For instance, suppose word-1 is a synonym or variation of word-2, and that the weight of word-2 has been calculated. The weight of word-1 can be obtained from the weight of word-2. Thus, although the online product reviews are written using different words which may share the same meaning, text classifiers can still be analyzed. Then, we built sentiment classifier based on SVM algorithm. After testing, the SVM classifier (as sentiment classifier) also shows a satisfactory accuracy because the solution of the SVM method gives an optimal hyperplane, which is a decision boundary

between non-relevant and relevant information. The effectiveness of the SVM sentiment classifier model can be increased with a small bag of words that consists of suitable features. As the results, this would demonstrate that our method can achieve substantial improvements.

6. Acknowledgement

My ideas on sentiment classification were developed when I was a researcher at Nanyang Technological University (NTU) in Singapore under an ACRC fellowship. Thus, I am also grateful to this fellowship and Assoc. Prof. Christopher S.G. Khoo and Assist. Prof. Jin-Cheon Na, who advised me in this research area.

7. References

[1] S.J. Simon (2001) The Impact of Culture and Gender on the Web Sites: An Empirical Study. The DATA BASE for Advances in Information Systems, Vol. 32, No. 2, pp 18-37.

[2] A. Esuli & F. Sebastiani (2005) Determining the Semantic Orientation of Terms through Gloss Classification. ACM 14th Conference on Information and Knowledge Management (CIKM). Bremen, Germany.

[3] A. Kennedy & D. Inkpen (2005) Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations (FINEXIN).

[4] B. Pang, L. Lee & S. Vaithyanathan (2002) "Thumbs up? Sentiment Classification using Machine Learning Techniques". Proceedings of Conference on empirical methods in natural language processing (EMNLP). pp. 79-86.

[5] B. Pang & L. Lee (2004) A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Proceedings of the Association for Computational Linguistics (*ACL*). pp.217-278.

[6] P.D. Turney (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In Proceedings of the Association for Computational Linguistics (ACL). pp. 417-424. (*ISKO*). UK.

[7] J.C. Na, C. Khoo, S. Chan & Y. Zhou (2004) Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews. The 8th International Conference of the International Society for Knowledge Organization .

[8] K. Ryan (1994) The role of natural language in requirements engineering. Proceedings of IEEE International Symposium on Requirements Engineering. pp 240-242. IEEE Computer Society Press.

[9] N. Guarino (1998) Formal Ontology and Information Systems. Proceedings of International Conference on Formal Ontology in Information Systems. Maine, Italy. ACM digital library.

[10] B. Smith & C. Welty (2001) Ontology: Towards a New. Proceedings of International Conference on Formal Ontology in Information Systems. Maine, Italy. ACM digital library.

[11] S. Meknavin, P. Charoenpornsawat. & B. Kijsirikul (1997) Feature-based Thai Word segmentation. In Proceeding of the Natural Language Processing Pacific RIM Symposuim. pp 41-46.

[12] J. Han & M. Kamber (2001) Data Mining: Concept and Techniques, 2nd edn, Morgan Kaufmann Publishing, San Francisco, CA.

[13] M. Kifer, G. Lausen & L. Wu (1995) Logical Foundation of object-oreinted and frame-based languages. In Journal of the ACM (JACM), 42(4): 741-843.

[14] Y. Yang & J.O. Pederson (1997) A Comparative Study on Features selection in Text Categorization. Proceedings of the 14th international conference on Machine Learning (ICML). pp 412-420. Nashville, Tennessee.

[15] E.A. Fox & S. Sharat (1986) A comparison of two methods for soft Boolean interpretation in information retrieval. TR-86-1. Virginia Tech. Department of Computer Science.

[16] W.C Lee & E.A. Fox (1988) Experimental Comparison of Schemes for Interpreting Boolean Queries. TR-88-27. Virginia Tech M.S. Thesis Department of Computer Science.

[17] T. Joachims (1999) Transductive Inference for Text Classification using Support Vector Machines. In: Proceedings of the International Conference on Machine Learning (ICML).

[18] K. Nigram, A. K. Maccallum, S. Thrun & T. Mitchell (2000) Transductive Text Classification from Labeled and Unlabeled Document using EM. In: Machine Learning. 39(2/3). pp. 103-134.

[19] C.C. Chang & C.J. Lin (2004) LIBSVM: a Library for Support Vector Machines. Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

[20] R. Baeza-Yates & B. Ribeiro-Neto (1999) Modern information retrieval. ACM Press, New York.