Computational Linguistics and Chinese Language Processing
Vol. 12, No. 2, June 2007, pp. 107-126

Using a Generative Model for Sentiment Analysis

Yi Hu*, Ruzhan Lu*, Yuquan Chen*, and Jianyong Duan*

Abstract

This paper presents a generative model based on the language modeling approach for sentiment analysis. By characterizing the semantic orientation of documents as "favorable" (positive) or "unfavorable" (negative), this method captures the subtle information needed in text retrieval. In order to conduct this research, a language model based method is proposed to keep the dependent link between a "term" and other ordinary words in the context of a triggered language model: first, a batch of terms in a domain are identified; second, two different language models representing classifying knowledge for every term are built up from subjective sentences; last, a classifying function based on the generation of a test document is defined for the sentiment analysis. When compared with Support Vector Machine, a popular discriminative model, the language modeling approach performs better on a Chinese digital product review corpus by a 3-fold cross-validation. This result motivates one to consider finding more suitable language models for sentiment detection in future research.

Keywords: Sentiment Analysis, Subjective Sentence, Language Modeling, Supervised Learning.

1. Introduction

Traditional wisdom of document categorization lies in mapping a document to given topics that are usually sport, finance, politics, etc. Whereas, in recent years there has been a growing interest in non-topical analysis, in which characterizations are sought by the opinions and feelings depicted in documents, instead of just their themes. This method of analysis is defined to classify a document as favorable (positive) or unfavorable (negative), which is called sentiment classification. Labeling documents by their semantic orientation provides succinct summaries to readers and will have a great impact on the field of intelligent information retrieval.

* Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai, China. Tel: 86-21-3420 4591
E-mail: huyi@cs.sjtu.edu.cn

[Received August 2, 2006; Revised March 7, 2007; Accepted March 8, 2007]

In this study, the set of documents is rooted in the topic of digital product review, which will be defined in the latter part of this article. Accordingly, the documents can be classified into praising the core product or criticizing it. Obviously, a praising review corresponds to "favorable" and a criticizing one is "unfavorable" (the neutral review is not considered in this study).

Most research for document categorization adopts the "bag of words" representing model that treats words as independent features. On the other hand, utilizing such a representing mechanism may be imprecise for sentiment analysis. Take a simple sentence in Chinese as an example: "柯达 P712 内部处理器作了升级，处理速度应该更快了。(The processor inside Kodak P712 has been upgraded, so its processing speed ought to be faster.)" The term "柯达 (Kodak)" is very helpful for determining its theme of "digital product review", but words "升级(update)" and "快(fast)" corresponding to "处理器(processor)" and "处理速度(processing speed)" ought to be the important clues for semantic orientation (praise the product). Inversely, see another sentence in Chinese: "这样电池损耗就很快。(So, the battery was used up quickly.)" The words "损耗 (use up)" and "快 (fast)" become unfavorable features of the term "电池 (battery)". That is to say, these words probably contribute less to the sentiment classification if they are dispersed into the document vector, because the direct/indirect relationships between ordinary words and the terms within the sentence are lost. Unfortunately, traditional n-gram features cannot easily deal with these long-distance dependencies.

Sentiment classification is a complex semantic problem [Pang *et al.* 2002; Turney 2002] that needs knowledge for decision-making. The researchers, here, explore a new idea-based language model for the sentiment classification of sentences rather than full document, in which the terms such as "处理器 (processor)", "处理速度 (processing speed)" are target objects to be evaluated in the context. They are mostly the nouns or noun phrases: "屏幕 (Screen)", "分辨率 (Resolution)", "颜色 (Color)", etc. If the sentiment classifying knowledge on how to comment on these terms can be obtained by the training data in advance, the goal of sentiment analysis can be achieved by matching the terms in the test documents. Thus, the classifying task for the full document is changed to recognizing the semantic orientation of all terms in accordance with their sentence-level contexts. This can also be considered a positive/negative word counting method for sentiment analysis.

In this study, the authors construct two language models for each term to capture the difference of sentiment context for that term. In these language models, sentences are divided into terms and their contexts. Sentences without the defined terms are ignored since they make no contribution to the document level sentiment classification; hence, they are omitted from training and test documents. This idea of grouping a document under subjective and objective portions is similar to Pang's work [Pang and Lee 2004].

This work can be divided into three main parts: first, some terms are extracted from a Chinese digital product review corpus [Chen *et al*. 2005]; second, two language models representing positive and negative classifying knowledge for each term are determined from training a subjective sentence set; third, the two models are applied to the test set and then compared with a popular discriminative classifier, SVM. The experiments demonstrate the better performance of the language modeling approach.

The rest of this paper is structured as follows. Section 2 briefly reviews the related works. Section 3 provides short introductions to SVM and language model. Section 4 describes the model in detail. Section 5 presents the method of estimating model parameters, in which a smoothing technique is utilized. Section 6 shows some experiments to exemplify the availability of the language modeling approach. In section 7, conclusions are given.

2. Related Works

A considerable amount of research has been done about document categorization other than topic-based classification in recent years. For example, Biber [Biber 1988] concentrated on sorting documents in terms of their source or source style with stylistic variation such as author, publisher, and native-language background. Sentiment classification for documents, though, has attracted tremendous attention for its broad applications in various domains such as movie reviews and customer feedback reviews [Gamon 2004; Pang *et al*. 2002; Pang and Lee 2004; Turney and Littman 2003]. Many research projects have used positive or negative term counting methods, which automatically determine the positive or negative orientation of a term [Turney and Littman 2002]. Other projects have focused on machine learning algorithms, such as Bayesian Classifier and SVMs, to classify entire reviews in a manner similar to a pattern recognition task.

Some related works focus on categorizing the semantic orientation of individual words or phrases by employing linguistic heuristics [Hatzivassiloglou and McKeown 1997; Hatzivassiloglou and Wiebe 2000; Turney and Littman 2002]. The word's semantic orientation refers to a real number measure of the positive or negative sentiment expressed by a word or a phrase [Hatzivassiloglou and McKeown 1997]. In previous works, the approach taken by Turney [Turney and Littman 2002] is used to derive such values for selected phrases in the document. The semantic orientation of a phrase is determined based on the phrase's Pointwise Mutual Information (PMI) with the words "excellent" and "poor". PMI is defined by Church and Hanks [Church and Hanks 1989] as follows:

$$PMI(w_1 \& w_2) = \log_2\left(\frac{p(w_1 \& w_2)}{p(w_1)p(w_2)}\right), \tag{1}$$

where $p(w_1\&w_2)$ is the probability that w_1 and w_2 co-occur. The orientation for a phrase is the difference between its PMI with the word "excellent" and the PMI with the word "poor". The final orientation is:

$$SO(phrase) = PMI(phrase, "excellent") - PMI(phrase, "poor"). \quad (2)$$

This yields values above zero for phrases having greater PMI with the word "excellent" and below zero for greater PMI with "poor". An *SO* value of zero denotes a neutral semantic orientation. This approach is simple but effective. Moreover, it is neither restricted to words of a particular part of speech (*e.g*. adjectives), nor restricted to a single word, but can be applied to multiple-word phrases. The semantic orientation of phrases can be used to determine the sentiment of complete sentences and reviews. In Turney's work, 410 reviews were taken and the accuracy of classifying the documents was found when computing the polarity of phrases for different kinds of reviews. Results ranged from 84% for automobile reviews to as low as 66% for movie reviews.

Another method of classifying documents into positive and negative is to use a learning algorithm to classify the documents. Several algorithms were compared in [Pang *et al*. 2002], where it was found that SVMs generally give better results. Unigrams, bigrams, part of speech information, and the position of the terms in the text are used as features, where using only unigrams is found to produce the best results. Pang *et al*. further analyzed the problem to discover how difficult sentiment analysis is. Their findings indicate that, generally, these algorithms are not able to generate accuracy in the sentiment classification problem in comparison with the standard topic-based categorization. As a method to determine the sentiment of a document, Bayesian belief networks are used to represent a Markov Blanket [Bai 2004], which is a directed acyclic graph where each vertex represents a word and the edges are dependencies between the words.

Methods for extracting subjective expressions from collections are presented in [Pang and Lee 2004]. Subjectivity clues include low-frequency words, collocations, and adjectives and verbs identified using distribution similarity. In [Riloff and Wiebe 2003], a bootstrapping process learns linguistically rich extraction patterns for subjective expressions. Classifiers define unlabeled data to automatically create a large training set, which is then given to an extraction pattern learning algorithm. The learned patterns are then used to identify more subjective sentences. A method to distinguish objective statements from subjective statements is also presented in [Pang and Lee 2004]. This method is based on the assumption that objective and subjective sentences are more possibly to appear in groups. First, each sentence is given a score indicating if the sentence is more likely to be subjective or objective using a Naive Bayes classifier trained on a subjectivity data set. The system then adjusts the subjectivity of a sentence based on how close it is to other subjective or objective sentences.

This method obtains amazing results with up to 86% accuracy on the movie review set. A similar experiment is presented in [Yu and Hatzivassiloglou 2003].

Past works on sentiment-based categorization of entire texts also involve using cognitive linguistics [Hearst 1992; Sack 1994] or manually constructing discriminated lexicons [Das and Chen 2001; Tong 2001]. These works enlighten researchers on the research on learning sentiment models for terms in the given domain.

It is worth referring to an interesting study conducted by Koji Eguchi and Victor Lavrenko [Eguchi and Lavrenko 2006]. In their contribution, they do not pay more attention to sentiment classification itself, but propose several sentiment retrieval models in the framework of generative modeling approach for ranking. Their research assumes that the polarity of sentiment interest is specified in the users' need in some manner, where the topic dependence of the sentiment is considered.

3. SVMs and Language Model

3.1 SVMs

Support Vector Machine (SVM) is highly effective on traditional document categorization [Joachims 1998], and its basic idea is to find the hyper-plane that separates two classes of training examples with the largest margin [Burges 1998]. It is expected that the larger the margin, the better the generalization of the classifier.

The hyper-plane is in a higher dimensional space called feature space and is mapped from the original space. The mapping is done through kernel functions that allow one to compute inner products in the feature space. The key idea in mapping to a higher space is that, in a sufficiently high dimension, data from two categories can always be separated by a hyper-plane. In order to implement the sentiment classification task, these two categories are designated positive and negative. Accordingly, if d is the vector of a document, then the discriminant function is given by:

$$f(d) = w \cdot \phi(d) + b\,. \tag{3}$$

Here, w is the weight vector in feature space that is obtained by the SVM from the training examples. The "·" denotes the inner product and b is a constant. The function ϕ is the mapping function. The equation $w \cdot \phi(d) + b = 0$ represents the hyper-plane in the higher space. Its value $f(d)$ for a document d is proportional to the perpendicular distance of the document's augmented feature vector $\phi(d)$ from the separating hyper-plane. The SVM is trained such that $f(d) \geq 1$ for positive (favorable) examples and $f(x) \leq -1$ for negative (unfavorable) examples.

Joachim's SVM^{light} package [Joachims 1999] was used for training and testing. For more details on SVM, the reader is referred to Cristiani and Shawe-Tailor's tutorial [Cristianini and

Shawe-Taylor 2000] and Roberto Basili's paper [Basili 2003].

3.2 Language Models

A statistical language model is a probability distribution over all possible word sequences in a language [Rosenfeld 2000]. Generally, the task of language modeling handles the problem: how likely would the i^{th} word occur in a sequence given the history of the preceding *i-1* words? In most applications of language modeling, such as speech recognition and information retrieval, the probability of a word sequence is decomposed into a product of n-gram probabilities. Let one assume that L denotes a specified sequence of k words,

$$L = w_1 w_2 ... w_k \ . \tag{4}$$

An n-gram language model considers the sequence L to be a Markov process with probability

$$p(L) = \prod_{i=1}^{k} p(w_i \mid w_{i-n+1}^{i-1}) \ . \tag{5}$$

When n is 1, it is a unigram language model which uses only estimates of the probabilities of individual words, and when n is equal to 2, it is the bigram model which is estimated using information about the co-occurrence of pairs of words. On the other hand, the value of *n-1* is also called the order of the Markov process.

To establish the n-gram language model, probability estimates are typically derived from frequencies of n-gram patterns in the training data. It is common that many possible n-gram patterns would not appear in the actual data used for estimation, even if the size of the data is huge. As a consequence, for a rare or unseen n-gram, the likelihood estimates that are directly based on counts may become problematic. This is often referred to as data sparseness. Smoothing is used to address this problem and has been an important part of various language models.

4. A Generative Model for Sentiment Classification

In this section, a language modeling approach to detect semantic orientation of document is proposed. This approach is very simple: one must observe the usage of language in contexts of terms appearing in positive and negative documents. "Favorable" and "unfavorable" language models are likely to be substantially different: they are prone to different language habits. This divergence in the language models is exploited to effectively classify a test document as positive or negative.

4.1 Two Assumptions

Models usually have their own basic assumptions as foundation of reasoning and calculating, which support their further applications. The researchers also propose two assumptions in this study, and, based on them, employ a language modeling approach to deal with the sentiment classification problem. As mentioned above, ordinary words in a sentence might have correlation with the term in the same sentence. Therefore, this method follows the idea of learning positive and negative language models for each term within sentences. After this, the sentiment classification is transferred into calculating the generation probability of all subjective sentences in a test document by these sentiment models. The following two assumptions are presented:

A_1. A subjective sentence contains at least one sentiment term and is assumed to have obvious semantic orientation.

A_2. A subjective sentence is the processing unit for sentiment analysis.

The first assumption (A_1) gives the definition of subjective sentence, and it means a significant sentence for training or testing should contain at least one term. In contrast, a sentence without any term is regarded as an objective sentence because of its "no contribution" to sentiment. It also assumes that a subjective sentence has complete sentiment information to characterize its own orientation.

The second assumption (A_2) allows one to handle the classification problem of sentence-level processing. Therefore, the authors pay more attention to construct models within the given sentence in terms of this assumption. A_2 is an intuitive idea in many cases.

Previous work has rarely integrated sentence-level subjectivity detection with document-level sentiment polarity. Yu and Hatzivassiloglou [Yu and Hatzivassiloglou 2003] provide methods for sentence-level analysis and for determining whether a sentence is subjective or not, but do not consider document polarity classification. The motivation behind the single sentence selection method of Beineke *et al.* [Beineke *et al.* 2004] is to reveal a document's sentiment polarity, but they do not evaluate the polarity-classification accuracy of results.

4.2 Document Representation

Based on these two assumptions, a document d is naturally reorganized into subjective sentences, and the objective sentences are omitted from d. That is to say, the original d is reduced to:

$$d \triangleq \{s \mid \exists t \in s\}. \tag{6}$$

Furthermore, a subjective sentence can be traditionally represented by a Chinese word sequence as follows,

$$w_1\ w_2 \ldots w_{l-1}\ t_{i,l}\ w_{l+1}\ \ldots w_{l+2}\ w_n\ . \tag{7}$$

In this, "$t_{i,l}$" indicates one term t_i appears in the sentence s_i, which is usually denoted as the serial number 'l' in the sequence. Moreover, the subsequence from w_1 to w_{l-1} is the group of ordinary words on the left side of t_i, and the subsequence from w_{l+1} to w_n is the group of ordinary words on the right. In (7), ordinary words in this sentence consist of t_i's context ($\boldsymbol{Cx_i}$). So, a subjective sentence s_i is simplified to:

$$s_i \triangleq < t_i, Cx_i > . \tag{8}$$

The authors now focus on a special form, by which a document is represented. Let d be defined again,

$$d \triangleq \{< t_i, Cx_i >\} . \tag{9}$$

Definition (9) means that there also exists an independent assumption between sentences and every word has certain correlation with the term within a sentence. Each sentence has semantic orientation and makes a contribution to the global polarity.

Note that it is possible for there to exist more than one term in a sentence. However, when investigating one of them, the others are to be treated as ordinary words. Each term can create a <*t, Cx*> structure. That is to say, one sentence may create more than one such structure.

4.3 Sentiment Models of Term

With respect to each term, each plays an important role in sentiment classification because the pivotal point of this work lies in learning and evaluating its context. This kind of classifying knowledge, derived from the contexts of terms in two subject-sentence collections labeled positive or negative in different contexts, would like to use words with polarity, such as "快 (Fast)" and "慢 (Slow)". A formalized depiction of classifying knowledge is shown as the following 3-tuple k_i:

$$k_i \triangleq < t_i, \theta_i^P, \theta_i^N > \ t_i \in T\ . \tag{10}$$

The character "T" denotes the list of all terms obtained from collections. With respect to t_i, its classifying knowledge is divided into two models: θ_i^P and θ_i^N which represent the positive and negative models, respectively. The model parameters are estimated from the training data. The contribution of w_j to polarity is quantified by a triggered unigram model to express the long distance dependency, which is a language modeling idea explained in next subsection.

4.4 Language Modeling Approach for Sentiment Classification

Language models applied to information retrieval [Pone and Croft 1998; Song and Croft 1999] have proven the effectiveness of this approach in an ad-hoc IR task. However, little work has been done in sentiment classification other than considering statistical language modeling. The most important idea in this study is to treat sentiment analysis of a document as the comparison of different generation probabilities in their subjective sentences. The difference is derived from the sentiment language models, $\{\theta_i^P\}$ and $\{\theta_i^N\}$, of terms.

Up to the present, the unigram model has been widely used in many applications due to its relatively small parameter space and suitability for avoiding data sparseness. The traditional unigram model takes a strict assumption that each word is independent from all others, consequently, the probability of a word sequence transfers into the product of the probabilities of individual words. In the authors' model, a triggered unigram model based on subjective sentence collection is built. Thus, the sentiment classification of a document becomes a generation process.

It is assumed that each subjective sentence has its own contribution. Therefore, the global document orientation is calculated by the differences between the probabilities of generating every subjective sentence in the document based on the sentiment language models. Thus, the logarithm decision function (11) is defined as:

$$\begin{aligned} F(d;\theta^P,\theta^N) &\triangleq \ln\left(\frac{p(d\mid\theta^P)}{p(d\mid\theta^N)}\right) \\ &= \sum_{t_i\in s_i, s_i\in d}\left(\ln p(s_i\mid t_i,\theta_i^P) - \ln p(s_i\mid t_i,\theta_i^N)\right) \end{aligned} . \tag{11}$$

Equation (11) means that, to a subjective sentence in the document, if it is more possibly generated by the positive language model of term "t_i" than by its negative language model, the sentence gives more weight to positive orientation than the negative. If the opposite is true, the sentence is regarded as more negative. The value of these probabilities is then used to classify the documents:

$$F:\begin{cases} >0 & positive \\ <0 & negative \end{cases} . \tag{12}$$

It is obvious that decision value is the semantic orientation of the whole document. Every subjective sentence will also be calculated by the multiplication of each generation probability of an ordinary word in this sentence except the term itself, *i.e.*:

$$\begin{cases} p(s_i \mid t_i, \theta_i^P) = \prod_{w_j \in Cx_i, w_j \neq t_i} p(w_j \mid t_i, \theta_i^P) \\ p(s_i \mid t_i, \theta_i^N) = \prod_{w_j \in Cx_i, w_j \neq t_i} p(w_j \mid t_i, \theta_i^N) \end{cases}. \tag{13}$$

Using the logarithm, one can rewrite (13) in its final form:

$$\begin{cases} \ln p(s_i \mid t_i, \theta_i^P) = \sum_{w_j \in Cx_i, w_j \neq t_i} \ln p(w_j \mid t_i, \theta_i^P) \\ \ln p(s_i \mid t_i, \theta_i^N) = \sum_{w_j \in Cx_i, w_j \neq t_i} \ln p(w_j \mid t_i, \theta_i^N) \end{cases}. \tag{14}$$

Equations (13) and (14) are both composed of two functions corresponding to positive and negative cases, respectively. Finally, when one substitutes Equation (14) into Equation (11), one gets a new sentiment classifying function:

$$F(d; \theta^P, \theta^N) = \sum_{s_i \in d} \sum_{w_j \in Cx_i, w_j \neq t_i} \ln\left(\frac{p(w_j \mid t_i, \theta_i^P)}{p(w_j \mid t_i, \theta_i^N)}\right). \tag{15}$$

5. Parameter Estimation

In equation (15), one has to estimate $p(w_j \mid t_i, \theta_i^P)$, and $p(w_j \mid t_i, \theta_i^N)$.

5.1 MLE for $p(w_j \mid t_i, \theta_i)$

The researchers have two available training collections labeled with “positive” and “negative”. The detailed information of this corpus will be described in Section 6.1.

Two methods are used to estimate the unigram probability: <1> the Maximum Likelihood Estimate (MLE); <2> the Dirichlet Prior Smoothing for language models. The two estimating methods are compared in sentiment classification. The language models are trained on the positive collection (C^P) and negative collection (C^N), respectively. The MLE is

$$\begin{cases} p_{mle}(w_j \mid t_i, \theta_i^P) = \dfrac{\#(< w_j, t_i > \mid w_j \in Cx_i)}{\#(< *, t_i > \mid * \in Cx_i)} & s_i \in C^P \\ p_{mle}(w_j \mid t_i, \theta_i^N) = \dfrac{\#(< w_j, t_i > \mid w_j \in Cx_i)}{\#(< *, t_i > \mid * \in Cx_i)} & s_i \in C^N \end{cases}, \tag{16}$$

where $\#(< w_j, t_i > \mid w_j \in Cx_i)$ is the number of times w_j co-occurring with t_i in same subjective sentences in positive/negative document collection C^P/C^N, while $\#(< *, t_i > \mid * \in Cx_i)$ is the total number of any word (*) co-occurring with the term t_i in the same subjective sentences in C^P/C^N.

In the probability perspective, if a word w_j often co-occurs with t_i in sentences in the training corpus with a positive view, it may mean that it contributes more to a positive orientation than negative, and vice-versa.

The training data consists of small document samples. The MLE models are inherently poor representations of the true models for unseen words that will be unreasonably assigned zero probability. Therefore, a smoothing language model is worthy of being tried to approximate their true models.

5.2 Dirichlet Prior Smoothing

Dirichlet Prior smoothing [Zhai and Lafferty 2001; Zhai and Lafferty 2002] is a general smoothing method for the problem of zero probabilities and is suitable for unigram smoothing. It belongs to a type of linearly interpolated method. The purpose of the Dirichlet Prior smoothing is to address the estimation bias due to the fact that a document collection has a relatively small amount of data used to estimate a unigram model. More specifically, it is designed to discount the *MLE* appropriately and assign non-zero probabilities to n-gram, which are not observed in the collection. This is the normal role of language model smoothing.

The sentence generation is now taken into account. The basic models are the unigram models $\{\theta_i\}$ (includes $\{\theta_i^P\}$ and $\{\theta_i^N\}$, respectively), which will result in models with the Dirichlet Prior smoothing. That is,

$$p_{dir}(w \mid t_i, \theta_i) = \begin{cases} p_\gamma(w \mid t_i, \theta_i) & w \in \{Cx_i\} \\ \alpha\, p_{mle}(w \mid C) & otherwise \end{cases}, \tag{17}$$

where $p_\gamma(w \mid t_i, \theta_i)$ indicates the smoothed probability of w seen in the positive/negative subjective sentence collection of t_i. The probability $p_{mle}(w \mid C)$ denotes the whole corpus (C) language model based on *MLE*, and α is a coefficient controlling the probability mass assigned to unseen words, so that all probabilities sum to one. In general, α may depend on all $p_\gamma(w \mid t_i, \theta_i)$. In this study, the authors exploit the following smoothing formalizations:

$$p_\gamma(w \mid t_i, \theta_i) = \begin{cases} \dfrac{\#(< w, t_i > \mid w \in Cx_i, s_i \in C^P) + \mu\, p_{mle}(w \mid C)}{\#(< *, t_i > \mid * \in Cx_i, s_i \in C^P) + \mu} & to\ \theta_i^P \\ \dfrac{\#(< w, t_i > \mid w \in Cx_i, s_i \in C^N) + \mu\, p_{mle}(w \mid C)}{\#(< *, t_i > \mid * \in Cx_i, s_i \in C^N) + \mu} & to\ \theta_i^N \end{cases}, \tag{18}$$

and

$$\alpha = \frac{\mu}{\mu + \mid C \mid}, \tag{19}$$

where μ is a controlling parameter that needs to be set empirically.

In particular, Dirichlet Prior smoothing may play two different roles in the sentence likelihood generation method. One is to improve the accuracy of the estimated document language model, while the other is to accommodate generation of non-informative common words. The following experiment results further suggest that this smoothing measure is useful in the estimation procedure.

6. Experiment Results and Discussions

This study is interested in the subject of "digital product review", and all documents are obtained from digital product review web sites. In terms of evaluating the results of sentiment classification, the researchers employ average accuracy based on 3-fold cross validation over the polarity corpus in the following several experiments.

6.1 Document Set and Evaluating Measure

The datasets select digital product reviews where the author rating is expressed either with thumbs "up" or thumbs "down". For the works described in this study, the dataset only concentrates on discriminating between positive and negative sentiment.

To avoid domination of the corpus by a small number of prolific reviewers, the corpus imposes a limit of fewer than 25 reviews per author per sentiment category, yielding a corpus of 900 negative and 900 positive reviews, with a total of more than a hundred reviewers represented. Some statistics about the corpus are shown in Table 1.

Table 1. The two collections from the same domain (digital product reivew).

Collections	# of Documents	Average # of Subjective Sentences	Sizes (KB)
Positive	900	28.3	462.99
Negative	900	25.9	453.82

Note that these 1800 documents in the corpus have obvious semantic orientations to their products: favorable or unfavorable. Furthermore, in terms of positive documents, they contain an average of 28.3 subjective sentences, while negative document collections contain an average of 25.9. All these digital product reviews downloaded from several web sites are about electronic products, such as DV, mobile phones, and cameras. On the other hand, all of these Chinese documents have been pre-processed in a standard manner: they are segmented into words and Chinese stop words are removed. All of these labeled documents are to be naturally divided into three collections in every process of 3-fold cross validation, which are used either for training or for testing.

In evaluating processes, a document may be grouped into positive or negative. That is to say, there exist two kinds of classification errors called "false negative" and "false positive".

Thus, the authors could build the following Contingency Table.

Table 2. Contingency Table.

	Tagged Positive	Tagged Negative
True Positive	A	B
True Negative	C	D

In the table A, B, C and D respectively indicate the number of every case. When the system classifies a true positive document into "positive" or classifies a true negative document into "negative", these two are correct, yet the other two cases are wrong. Therefore, the accuracy is defined as a global evaluation mechanism:

$$Accuracy = (A + D)/(A + B + C + D)\ . \tag{20}$$

Obviously, the larger the accuracy value is, the better the system performance is. In the following experiments, the 3-fold cross validation based average accuracy is the major evaluating measure in the following experiments.

6.2 Term Extraction

The researchers extract term candidates using a term extractor from the previous work of the authors [Chen *et al*. 2005]. Following this study, the hybrid method for automatic extraction of terms from domain-specific un-annotated Chinese corpus is used through means of linguistic knowledge and statistical techniques. Then, hundreds of terms applied in the sentiment analysis are extracted from the digital product review documents. They are ranked by their topic-relativity scores.

The main idea in [Chen *et al*. 2005] lies in finding the two neighboring Chinese characters with high co-occurrence, called "bi-character seeds". These seeds can only be terms or the components of terms. For instance, the seed "分辨" is the left part of the real term "分辨率 (Resolution)". So the system has to determine the two boundaries by adding characters one by one to these seeds in both directions to acquire multi-character term candidates. Apparently, there exist many non-terms in these candidates, so one must take a dual filtering strategy and introduce a weighting formula to filter these term candidates via a large background corpus.

Although the authors have adopted the dual filtering strategy in this system to improve performance, it cannot separate the terms and non-terms completely. Therefore, it also needs manual selection of the suitable terms that strictly belong to the digital product domain. The terms were chosen from the candidate list one by one via their topic-relativity scores.

It is worth noting that all the selected terms are nouns/noun phrases that represent concepts that are usually evaluated in real-life contexts. For example, "数码相机 (digital

camera, one of the digital products)", "处理器 (processor, a key part of some digital products)".

6.3 Experiments and Discussions

Three experiments were designed to investigate the proposed method as compared to SVM. The first was to select the most suitable number of terms given their topic-relativity to the domain. The second was to select a suitable kernel from linear, polynomial, RBF and sigmoid kernels for sentiment classification. The last was to compare the performance between the language modeling approach and SVM.

With respect to these three experiments, the 1800 digital product reviews were split into three parts: 1000 training samples (500 positive and 500 negative); 600 test samples (300 positive and 300 negative); and the remaining 200 samples (100 positive and 100 negative) that were prepared for choosing a suitable number of terms.

Table 3 shows a series of contrastive results by testing on the 200 samples after training models of terms ranging from 20 to 200 given their topic-relativity ranks. This is a method for selecting a suitable term set. In this experiment, unigram models are employed by MLE. Here, all of the Chinese words occurring are used as unigrams to learn the language models, and this is different from selecting a portion of them in the following experiments (see Section 6.4).

Table 3. Average accuracy based on the number of terms from 20 to 200 according to their topic-relativity ranking scores. In this experiment, we employ the unigram model by MLE.

# of terms	20	40	60	80	100	120	**140**	160	180	200
Avg. Accuracy	48.31	50.50	57.11	58.78	70.83	74.27	**79.31**	77.04	76.78	73.50

The experiment proves that it is not clear whether or not one ought to use a large term set for achieving better system performance, because redundant terms may bring "noise" to semantic polarity decision. As seen in Table 3, experimental results achieve the greatest accuracy when keeping 140 terms by topic-relativity ranking scores in the term set. According to this result, the authors use the 140 terms next for smoothing of sentiment language models and comparison with SVM.

6.4 Comparison with SVM

Unigrams are extracted as input feature sets for SVM. The following experiments compare the performance of SVM using linear, polynomial, RBF and sigmoid kernels, the four conventional learning methods commonly used for text categorization. The SVM^{light} package [Joachims 1999] was used for training and testing on the document-level, and other

parameters of different kernel functions were set to their default values in this package. This experiment aims at exploring which method is more suitable for the sentiment detection problem (See Table 4).

To make sure that the results for the four kernels are not biased by an inappropriate choice of features, all four methods are run after selecting unigrams (Chinese words) appearing at least three times in the whole 1800 document collection. Finally, the total number of features in this study is 5783 for SVM, including those "terms" used in the language modeling approach.

Table 4. Comparison of four kernel functions on the digital product review training and test corpus and average performance over two categories. Linear kernel achieves highest performance on unigram feature set.

Features	# of features	Linear	Polynomial	Radial Basis Function	Sigmoid
unigrams	5783	**80.17**	61.25	53.09	51.26

The result with the best performance in the test set is the linear kernel. Thus, the language model based method is compared with the SVM using linear kernel. The next table gives the results achieved by the language modeling approach and the control group. In this experiment, the 5783 single word forms (*i.e.* vocabulary) are also used as the features for language models.

Table 5. Comparison between language model based method and SVM using linear kernel.

	# of features	AvgAccuracy	% change over SVM
SVM (Linear Kernel)	5783	80.17	—
Uni-MLE	5783	83.10	+3.65
Uni-Smooth (μ=1100)	5783	**85.33**	+6.44

Seen from table 5, Uni-MLE performs better on the unigrams features set than SVM, which achieved an average significant improvement of 3.65% compared with the best SVM result. As to the model smoothing, Dirichlet Prior smoothes unigram language model with parameter μ set to 1100 (In this experiment, the best result appears when $\mu = 1100$ in Dirichlet Prior smoothing). It makes a contribution to estimating a better unigram language model leading to a significantly better result than SVM (+6.44%). The effect of the smoothing method in sentiment analysis is just like its effect on most language model based applications in NLP. In practice, the unigram model built up from the two limited collections by simple MLE has not enough reasonability in terms of the unseen words. The smoothing method gives the unobserved ordinary words of every term a suitable non-zero probability and improves the system performance.

The better results obtained by this generative model may be due to the sentiment

description within sentences, which proves that the two assumptions in Section 4.1 may be reasonable. The authors use the triggered unigram models to describe the classifying contribution of features of every term, and then construct sentiment language models. Accordingly, the motivation to further explore the refinement of sentiment language models based on learning higher order models and introduce more powerful smoothing methods in future is acquired.

7. Conclusions

In this paper, the authors have presented a new language modeling approach for sentiment classification. To this generative model, the terms of a domain are introduced as counting terms, and their contexts are learnt to create sentiment language models. It was assumed that sentences have complete semantic orientation when they contain at least one term. This assumption allows one to design models to learn positive and negative language models from the subjective sentence set with polarity. The approach is then used to test a real document in steps: first to generate all the subjective sentences in the document, and then to generate each ordinary word in turn depending on the terms by positive and negative sentiment models. The difference between the generation probabilities by the two models is used as the determining rule for sentiment classification.

The authors have also discussed how the proposed model resolves the sentiment classification problem by refining the basic unigram model through smoothing. When the language model based method is compared with a popular discriminative model, *i.e.*, SVM, the experiment shows the potential power of language modeling. It was demonstrated that the proposed method is applicable for learning the positive and negative contextual knowledge effectively in a supervised manner.

The difficulty of sentiment classification is apparent: negative reviews may contain many apparently positive unigrams even while maintaining a strongly negative tone and vice-versa. In terms of the Chinese language, it is a language of concept combination, allowing the usage of words to be more flexible than in Indo-European languages, which makes it more difficult to acquire statistic information than other languages. All classifiers will face this difficulty. Therefore, the authors plan to improve the language model based method in the following three possibilities:

Future works may focus on finding a good way to estimate better language models, especially the higher order n-gram models and more powerful smoothing methods.

The authors have assumed an independent condition among sentences so far. It is also possible to introduce a suitable mathematic model to group the close sentences. Constructing an enlarged sentiment analyzing area may utilize more linking information between words.

The conceptual analysis of Chinese words may be helpful to sentiment analysis because this theory pays more attention to counting the real sense of concepts. In future works, the authors may integrate more conceptual features into the models.

Acknowledgement

This work is supported by NSFC Major Research Program 60496326: Basic Theory and Core Techniques of Non Canonical Knowledge.

References

Bai, X., R. Padman, and E. Airoldi, "Sentiment extraction from unstructured text using tabu search-enhanced markov blanket," In *Proceedings of the International Workshop on Mining for and from the Semantic Web*, 2004, Seattle, WA, USA.

Basili, R., "Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms by Thorsten Joachims," *Computational Linguistics*, 29(4), 2003, pp. 655-661.

Beineke, P., T. Hastie, C. Manning, and S. Vaithyanathan, "Exploring sentiment summarization," In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications* (AAAI tech report SS-04-07), 2004.

Biber, D., *Variation across Speech and Writing*, The Cambridge University Press, 1988.

Burges, C., "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, 2(2), 1998, pp. 121-167.

Chen, X., X. Li, Y. Hu, and R. Lu, "Dual Filtering Strategy for Chinese Term Extraction," In *Proceedings of FSKD(2)*, Changsha, China, 2005, pp. 778-786.

Church, K. W., and P. Hanks, "Word association norms, mutual information and lexicography," In *Proceedings of the 27th Annual Conference of the ACL*, 1989, Vancouver, BC, Canada.

Cristianini, N., and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*, The Cambridge University Press, 2000.

Das, S., and M. Chen, "Yahoo! for Amazon: Extracting market sentiment from stock message boards," In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*, 2001, Bangkok, Thailand.

Eguchi, K., and V. Lavrenko, "Sentiment Retrieval using Generative Models," In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2006, Sydney, Australia, pp. 345-354.

Gamon, M., "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis," In *Proceedings the 20th International Conference on Computational Linguistics*, 2004, Switzerland.

Hatzivassiloglou, V., and K. McKeown, "Predicting the semantic orientation of adjectives," In *Proceedings of the 35th ACL/8th EACL*, 1997, Madrid, Spain, pp. 174-181.

Hatzivassiloglou, V., and J. Wiebe, "Effects of Adjective Orientation and Gradability on Sentence Subjectivity," In *Proceedings the 18th International Conference on Computational Linguistics*, 2000, Germany, pp. 299-305.

Hearst, M., "Direction-based text interpretation as an information access refinement," *Text-based intelligent systems: current research and practice in information extraction and retrieval*, ed. by Paul Jacobs, Lawrence Erlbaum Associates, 1992, pp. 257-274.

Joachims, T., "Text categorization with support vector machines: Learning with many relevant features," In *Proceedings of the European Conference on Machine Learning*, 1998, Chemnitz, pp. 137-142.

Joachims, T., "Making large-scale SVM learning practical", *Advances in Kernel Methods-Support Vector Learning*, ed. by Bernhard Scholkopf and Alexander Smola, The MIT Press, 1999, pp. 44-56.

Pang, B., and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," In *Proceedings of the 42nd ACL*, 2004, Barcelona, Spain, pp. 271-278.

Pang, B., L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," In *Proceedings of The Conference on Empirical Methods in Natural Language Processing,* 2002, Philadelphia, USA.

Pone, J., and W. B. Croft, "A language modeling approach to information retrieval," In *Proceedings of the 21st Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, Melbourne, Australia.

Riloff, E., and J. Wiebe, "Learning extraction patterns for subjective expressions," In *Proceedings of the 41st Conference on Empirical Methods in Natural Language Processing*, 2003, Sapporo, Japan, pp. 105-112.

Rosenfeld, R., "Two decades of statistical language modeling: where do we go from here?" In *Proceedings of the IEEE*, 88(8), 2000.

Sack, W., "On the computation of point of view," In *Proceedings of the Twelfth AAAI, Student abstract*, 1994, Seattle, WA, USA, pp. 1488.

Song, F., and W. B. Croft, "A general language model information retrieval," In *Proceedings of the 22nd Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, Berkeley, CA, USA.

Tong, R.M., "An operational system for detecting and tracking opinions in on-line discussion," Workshop Notes, *SIGIR Workshop on Operational Text Classification*, 2001, New Orleans.

Turney, P.D., "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," In *Proceedings of the ACL*, 2002, Philadelphia, Pennsylvania, USA, pp. 417-424.

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

Turney, P.D., and M. L. Littman, "Unsupervised learning of semantic orientation from a hundred-billion-word corpus," Technical Report EGB-1094, National Research Council, Canada, 2002.

Yu, H., and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, Sapporo, Japan.

Zhai, C. and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," In *Proceedings of SIGIR*, 2001, New Orleans, USA.

Zhai, C. and J. Lafferty, "Two Stage Language Models for Information Retrieval," In *Proceedings of S IGIR*, 2002, Tampere, Finland.