University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Dissertations and Theses from the College of Business Administration

Business Administration, College of

4-1-2012

SENTIMENT ANALYSIS: A STUDY ON PRODUCT FEATURES

Yanyan Meng
University of Nebraska – Lincoln, mastmyy@yahoo.com.cn

Follow this and additional works at: http://digitalcommons.unl.edu/businessdiss

Part of the Management Information Systems Commons

Meng, Yanyan, "SENTIMENT ANALYSIS: A STUDY ON PRODUCT FEATURES" (2012). *Dissertations and Theses from the College of Business Administration.* Paper 28.
http://digitalcommons.unl.edu/businessdiss/28

This Article is brought to you for free and open access by the Business Administration, College of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Dissertations and Theses from the College of Business Administration by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

SENTIMENT ANALYSIS:

A STUDY ON PRODUCT FEATURES

by

Yanyan Meng

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Arts

Major: Business

Under the Supervision of Professor Keng L. Siau

Lincoln, Nebraska

April, 2012

SENTIMENT ANALYSIS—A STUDY ON PRODUCT FEATURES

Yanyan Meng, M.A.

University of Nebraska, 2012

Advisor: Keng L. Siau

Sentiment analysis is a technique to classify people's opinions in product reviews, blogs or social networks. It has different usages and has received much attention from researchers and practitioners lately. In this study, we are interested in product feature based sentiment analysis. In other words, we are more interested in identifying the opinion polarities (positive, neutral or negative) expressed on product features than in identifying the opinion polarities of reviews or sentences. This is termed as the product feature based sentiment analysis. Several studies have applied unsupervised learning to calculate sentiment scores of product features. Although many studies used supervised learning in document-level or sentence-level sentiment analysis, we did not come across any study that employed supervised learning to product feature based sentiment analysis. In this research, we investigated unsupervised and supervised learning by incorporating linguistic rules and constraints that could improve the performance of calculations and classifications. In the unsupervised learning, sentiment scores of product features were calculated by aggregating opinion polarities of opinion words that were around the product features. In the supervised learning, feature spaces that contained right features for product feature based sentiment analysis were constructed. To reduce the dimensions of feature spaces, feature selection methods, Information Gain (IG) and Mutual Information (MI), were applied and compared. The results show that (i) product features were good indicators in determining the polarity classifications of document or sentences; (ii) rule based features could perform well in supervised learning e; and (iii) IG performed better in document analysis, while MI performed better in sentence-level analysis.

ACKNOWLEDGEMENTS

I have been very lucky to study Management Information System at University of Nebraska-Lincoln, where I have the opportunity to gain much knowledge from a number of outstanding professors.

Firsthand, I would like to express my sincere gratitude to my academic advisor Professor Keng L. Siau for guiding me this thesis. His insights and guidance over these two years have been invaluable to me. Two years ago, I had no idea what I would do after I graduate. But now, under the guidance of Professor Siau, I have acquired a lot of skills on Business Intelligence and Text mining, and I am very clear and confident about what I will do in the future.

I would also like to thank Professor Fiona Nah and Professor Sidney Davis for serving as my committee members, reading my thesis, attending my defense and giving me valuable advises. In addition, I owe my friend Lingling Yuan a debt of gratitude for providing me accommodation and help during my visit to Lincoln to prepare for my thesis defense.

Last but not the least; I would like to express my great thanks to my husband Peng Yang, my parents, and my parents-in-law, who support me both emotionally and financially. Without their endless love, supports, and encouragements, I could not imagine that I can complete my study.

Table of Contents

SENTIMENT ANALYSIS: 0
A STUDY ON PRODUCT FEATURES 0
SENTIMENT ANALYSIS—A STUDY ON PRODUCT FEATURES i
ACKNOWLEDGEMENTS ii
Chapter 1: INTRODUCTION 1
Chapter 2: LITERATURE REVIEW 9
2.1 General Review 9
2.2 Sentiment analysis on product features 16
2.3 Fundamentals of supervised learning 19
2.3.1 Finding appropriate features 19
2.3.2 Using appropriate numerical feature values 25
2.3.3 Feature selection methods 31
2.3.4 Machine learning classification methods 34
Chapter 3: THEORETICAL FOUNDATIONS 38
3.1 Fundamental Theories 38
3.2 Rules and constraints 43
Chapter 4: RESEARCH METHODS 47
4.1 Product feature extraction 48
4.2 Extract opinion words around product features 50
4.3 Unsupervised learning—Calculate Sentiment Score of Product Features 51
4.4 Supervised machine learning methods 56
Chapter 5: EXPERIMENTAL SETUP 62
5.1 Experimental Data 62
5.2 Experimental steps 63
5.3 Classification tool—LIBSVM 65
5.4 Classification performance evaluation 65
Chapter 6: RESULTS AND DISCUSSIONS 67
6.1 Unsupervised learning – Sentiment score calculation 67
6.2 Supervised learning 72
Chapter 7: CONTRIBUTIONS AND CONCLUSIONS 84
REFERENCES 89

Table 2.1Part of extraction patterns from Rillo(1996;2003)…….……18
Table 2.2 F eatures for identifying contextual polarities (Wilson et al., 2005) 30
Table 2.3 Features for polarity classification from (Wilson et al., 2005)……. .31
Table 2.4 Stop words list…… 33

Table 3.1 Intensifier list (Brooke, 2009)..... 46
Table 4.1 Rule features in document vector …….. 58
Table 4.2 Rule features in sentence vector …….……..…...58
Table 5.1 Negation word list (Pott 2011)…….……65

Table 5.2 Example for classifier evaluations…….……..…..66

Table 6.1 Product features with their polarities……67

Table6.2 Averaged polarities of some product features..... 69
Table 6.3 Comparison of our method with that of Ding et al. (2008)..... 70
Table6.4 Validation results for different classifications 71
Table 6.5 Evaluation of used rules and constraints……..….72

Table 6.6 Comparison between features……..…….……. ….73
Table 6.7 Weighting values of part of features ……..76
Table 6.8 Comparison among features ……..81

Figure 3.1 Main attribute of Appraisal (Whitelaw et al., 2005) ……..40

Figure 4.1 Unsupervised sentiment score calculation of product features……..55

Figure 6.1 Performance of features in document-level and sentence-level analyses……74

Figure 6.2 Document level feature selection based on two methods……..…..77

Figure 6.3 Sentence level feature selection based on two methods……77

Chapter 1: INTRODUCTION

Large datasets are available on-line today, they can be numerical or text file and they can be structured, semi-structured or non-structured. Approaches and technique to apply and extract useful information from these data have been the major focuses of many researchers and practitioners lately. Many different information retrieval techniques and tools have been proposed according to different data types. In addition to data and text mining, there has seen a growing interest in non-topical text analysis in recent years. Sentiment analysis is one of them. Sentiment analysis, also known as opinion mining, is to identify and extract subjective information in source materials, which can be positive, neutral, or negative. Using appropriate mechanisms and techniques, this vast amount of data can be processed into information to support operational, managerial, and strategic decision making.

Researchers in sentiment analysis have focused mainly on two problems– detecting whether the text is subjective or objective, and determining whether the subjective text is positive or negative. The techniques relied on two main approaches: unsupervised sentiment orientation calculation, and supervised and unsupervised classifications based on machine learning.

The Sentiment Orientation (SO)/ opinion polarity calculations are lexicon based calculations, which calculate the polarity (positive or negative) scores of words using bootstrapping methods based on a small list of seed words with prior-polarities and WordNet lexicon. The SO calculations are based on the assumptions proposed by Osgood (1957) that one word can have its prior-polarity and its semantic can be represented by numbers. Many researchers (e.g. Turney, 2002; Kamps et al., 2004;

Wilson et al., 2005; Hu et al., 2004; Ding et al., 2008) have conducted sentiment analysis by proposing different sentiment orientation calculation algorithms. The supervised machine learning classifications mainly reside in document-level classification and sentence-level classification. Document-level classifications (Pang et al., 2002; Tan et al., 2009; Nakagawa et al., 2010) attempt to learn the polarity (positive, negative or neutral) of documents based on the frequencies of the various words in the document. This method usually uses Bag-of-Word (BoW) features. BoW does not consider the order of words or phrases in the documents (bags). The purpose of sentence-level classifications (Ding et al., 2008; Qu et al., 2008; Socher et al., 2011) is to discover the sentiments of texts in more detail. There are also many unsupervised machine learning methods (Lin et al., 2009; Guo et al., 2009; Zhai et al., 2010; Zhai et al., 2011) that have been proposed to bring more convenient usages to users.

In a document-level classification, a document can be classified based on the frequency of different words in the Bag-of-Word. Before supervised machine learning classification methods can be applied, document-level analysis needs the pre-labeled training data such as Thumb-up or Thumb-down labels (Pang et al., 2002; Turney, 2002). The problem with document-level analysis is that we cannot get more detailed information such as positive/negative sentiments regarding certain product features from the reviews. For example, a product such as a car consists several product features like engines, tires, and batteries. Most of the time, one person can express his/her opinion on more than one product features in the same review or even in the same sentence. For example, “I like the color, but its battery life is short.” So, it is meaningful to conduct sentiment analysis at a

more detailed level. In this research, we are interested in studying product feature based sentiment classification.

Several researchers (e.g. Hu et al., 2004; Ding et al., 2008; Liu, 2010; Guo et al., 2009; Zhai et al., 2010) have been trying to solve this problem. The works of Hu et al. (2004) and Ding et al. (2008) are related to our research interest, which is to identify the sentiment/polarity scores of product features in product reviews. *Sentiment score* refers to the numerated opinions that expressed on product features, while *sentiment orientation* refers to opinion polarity of an opinion word. Sentiment analysis that is to identify sentiment score of product features is usually based on unsupervised learning, which involves creating an opinion word lexicon that contains opinion words with opinion polarities annotated (e.g. positive or negative), and then calculating the sentiment/polarity scores of the product features by aggregating the opinion polarities of the opinion words. Unsupervised learning does not need labeled data for training.

Linguistic rules and constraints are usually used to improve performance of unsupervised sentiment score calculations in sentiment analysis (Ding et al., 2008; Zhai et al., 2010). In sentiment analysis, the most commonly used rules are (i) negative rules (see Pang et al., 2002; Hu et al., 2004; Wilson et al., 2005; Ding et al., 2008; Socher et al., 2011; Nakagawa et al., 2010), which refers to the usage of negation words (i.e., no, neither, never, etc.) that could change the polarity of the opinion words, and (ii) syntactic rules such as the usage of POS tags. Conjunction rules (Ding et al., 2008; Liu, 2010), refers to the rules using AND, BUT, OR, NOR, etc., are widely used as well. More advanced semantic rules have also been developed and used by many researchers such as Wilson et al. (2005) who developed 28 constraint features based on the semantic analyses to

discover the contextual polarity of opinion words. Researchers such as Wilson et al. (2005), Socher et al.(2011), and Nakagawa et al. (2010) applied semantic dependency trees for deeper analysis. A detailed semantic analysis of attitude expressions based on the appraisal theory was discussed in Whitelaw et al. (2005), and their approach received improved performance. Attitude expressions sometimes are not individual words, but rather appraisal groups such as “extremely boring”, “very good”, etc. In other words, it involves the application of intensifiers (i.e., extremely, very, etc.) to opinion words. Hence, intensification rules and the usage of other constraints could improve the classification performance. Zhai et al. (2010) improved the input accuracy by adding two constraints– must-links and cannot-links. A must-link constraint specifies that two data instances must be in the same cluster. A cannot-link constraint specifies that two data instances cannot be in the same cluster. Therefore, by using these rules and constraints, the classification performance can be enhanced.

In supervised learning, one way to improve the classification performance is to construct a feature space that contains the right features. Features that contain syntactic and semantic information, such as dependency tree patterns, usually contribute to higher performance in sentiment classifications (Nakagawa et al., 2010; Socher et al., 2011). In supervised sentiment analysis area, different tokens or annotations have been used as features to construct feature spaces. To the best of our knowledge, Pang et al. (2002) is the first one to use unigrams (words), bigrams, unigrams with POS tags, adjectives, and their combinations as features in the document-level sentiment analyses. Also, many tokens or patterns that contain syntactic and semantic information have been used as features in feature spaces for machine learning models in sentiment analyses. Information

Extraction (IE) patterns (Rillof 1996; Rillof, 2003; Rillof, 2006), and dependency tree patterns (Wilson et al., 2005; Socher et al., 2011; Nakagawa et al., 2010), which contain syntactic structures and dependency relations, could perform better than tokens that contain less syntactic and semantic relations, such as unigrams. Unigram models are usually based on conditional probabilities, in which the occurrence of next word is based on the occurrence of the former word. So, in a machine learning based sentiment analysis, selection of features that is based on the deep and detailed syntactic and semantic analysis that is specific to sentiment classification, rather than selecting features with no *specific purposes*, can result in high performance in supervised sentiment classification. *Specific purpose* means that some features, such as n-grams, part-of-speech tags, etc. can also be used in other Natural Language Processing (NLP) tasks. So, the usage of these features is not specific to sentiment analysis. In this way, it usually leads to large vector size in a feature space because the vector size usually depends on vocabulary size of dataset. Large sized feature space can reduce time and space efficiency. Therefore, selection of features with specific purpose for sentiment classifications could reduce the size of feature spaces.

For product review based sentiment classifications, product features should be good indicators in determining sentiment classification types of product reviews (one review is usually treated as one document) because product reviews are about product features. Hence, the right features can be selected based on product features. To construct a high performance feature space for product feature based sentiment classification, product features can be included and treated as features in the feature space. Further, based on the fact that high performance can be obtained by using linguistic rules and constraints, rule

based features may be possible to be developed for supervised learning and improve the classification performance.

In our research, both supervised and unsupervised learning were conducted. Objective in unsupervised learning was to increase calculation performance on sentiment score of product features by using different linguistic rules and constraints. Objective in supervised learning was to construct feature spaces that contained right features for machine learning models. In this research, a feature space was constructed with three feature sets – the first set was composed of product features, the second set consisted adjectives, and the third set was developed based on linguistic rules and constraints. In many unsupervised learning, studies preferred extracting adjectives as opinion words. In many supervised sentiment classifications, adjectives were also treated as features for machine learning models. In this research, feature spaces were constructed specifically for sentiment analysis on product features. The feature spaces contained information related to product features, opinion words (adjectives), and linguistic rules and constraints. Feature spaces for both document-level and sentence-level analysis were constructed. The reason for conducting document-level and sentence-level analysis is that one document can contain more product features than one sentence. If product features are good indicators in determining classifications of text, then document-level analysis can get higher classification performance than sentence-level analysis.

In machine learning processes, feature weighting and selection techniques are important in assigning feature values, selecting features, and improving the classification performance. In sentiment analysis, many weighting values have been used for feature values such as term frequency (TF), term presence, term frequency-inverse document

frequency (tf-idf), and many other derived values like Δ tf-idf (Martineau et al., 2009). Some studies also assigned polarities of dependency tree patterns (Nakagawa et al., 2010) and polarities of sentences (Maas et al., 2011) as feature values for feature spaces. As for feature selection methods for sentiment analysis, proposed methods include χ^2, Information Gain (IG), Term Strength(TS) (Yang et al., 1997), and Mutual Information (MI) (Turney, 2002). The major reason for applying these mathematical processes is to reduce feature space dimensions and enhance classification performance. Dimensions can also be reduced by giving threshold values (Yang et al.,1997), or applying Singular Value Decomposition (SVD) methods (Maas et al., 2011). After feature spaces were constructed, MI and IG feature selection methods were applied to feature spaces. Results of these two methods were compared using open source Support Vector Machine (SVM) tool LIBSVM provided by Chang et al. (2011).

To summarize, there are several approaches to improve the sentiment classification accuracy: (i) the selection of tokens and annotations – N-grams, POS tags, negation tags, dependency tree parsing patterns, or information extraction (IE) patterns; (ii) the selection of different rules and constraints; (iii) the selection of feature weighting methods—TF, tf-idf, or presence; (iv) the selection of feature selection methods –MI, IG, or CHI; and (v) the selection of different machine learning classifiers—Support Vector Machine (SVM), Naïve Bayes (NB), Maximum Entropy, or Conditional Random Field (CRF) (Nakagawa et al., 2010). The major objective of this research is not to compare the efficiencies of different machine learning classifiers, and we used the SVM tool that is generally considered to be the most efficient in text classification tasks.

Based on the analysis above, we conducted both unsupervised and supervised learning on product feature based sentiment analysis. The first experiment was unsupervised learning that calculated sentiment score for each product feature by applying different linguistic rules and constraints. The datasets used were product reviews provided by Ding et al. (2008). The calculation performance was then compared to that of Ding et al. (2008). In supervised learning, we conducted three experiments. The first experiment was to construct feature spaces for both document-level and sentence-level analysis. Three feature sets were developed with specific purposes to product feature based sentiment analysis. Product features used in this experiment were extracted from unsupervised learning experiment. After feature spaces were constructed, the next step in the experiment was to compare the performance among three feature sets. Datasets used in this experiment was the same datasets used in unsupervised learning experiment. The second experiment was to apply two feature selection methods to the proposed feature spaces. The third experiment was to compare the performance of proposed document-level feature spaces to performance of features used in Pang et al. (2002). Datasets used in the second and third experiments were provided by Pang et al. (2002).

Chapter 2: LITERATURE REVIEW

2.1 General Review

Much research has been focusing on sentiment classifications at different levels (document-level, sentence-level, and phrase-level) using supervised learning and unsupervised learning techniques. For supervised learning, selection of tokens/features (different from "feature selection"), assignment of feature values, and selection of feature selection methods are important to classification performance.

Prior to the popularity of polarity classification studies, which is to identify positive or negative polarities of the document or sentences, several research studies were on subjectivity classification, which is used to classify whether documents or sentences are subjective or objective. As mentioned in Riloff et al. (2003), subjective expressions include opinions, rants, allegations, accusations, suspicions, and speculations. Riloff et al. (2003) presented a bootstrapping process that learned linguistically rich extraction patterns for subjectivity expressions. The learned patterns were then used to automatically identify whether a sentence was subjective or objective. The results showed that their extraction patterns performed better then n-grams.

Rillof et al. (2003) introduced several steps to extract subjectivity patterns from subjectivity clauses and to label subjectivities of sentences.

First, subjectivity clues were divided into strongly subjective and weakly subjective by the rule that "a strong subjective clue is one that is seldom used without a subjective meaning, whereas a weak subjective clue is one that commonly has both subjective and objective meanings (p3)". Second, sentences were classified as subjective if they contain two or more strong subjective clues, and classified as objective if they contain no strong

subjective clue and at most one weak subjective clue in the current, previous, and next sentences. Third, a learning algorithm that was similar to AutoSlog-TS (Rillof, 1996) was applied to learn subjective extraction patterns using the annotated subjective and objective sentences as training corpus (dataset).

The learning process contained two steps. First, instantiate the extraction patterns in the training corpus according to the syntactic templates. For example, the pattern "<subj> passive-verb" can be used to extract phrases such as "<subj>was satisfied". Second, gather the statistics on how often each pattern occurs in subjective training corpus or objective corpus, and then ranked the extraction pattern using the conditional probability measure:

$\Pr(\text{subjective}|\text{pattern}_i) = \frac{\text{subjfreq}(\text{pattern}_i)}{\text{freq}(\text{pattern}_i)}$, where $\text{subjfreq}(\text{pattern}_i)$ and

$\text{freq}(\text{pattern}_i)$ were frequencies of subjective pattern i in subjective training corpus and the whole training corpus. The thresholds to select extraction patterns that are strongly associated with subjectivity in the training data set are $\text{freq}(\text{pattern}_i) \geq \theta_1$ and $\Pr(\text{subjective}|\text{pattern}_i) \geq \theta_2$. Finally, they used a bootstrapping method to apply learned extraction patterns to classify unlabeled sentences from un-annotated text collections. The Pattern Based Subjective Sentence Classifier classifies a sentence as subjective if it contains at least one extraction pattern with $\theta_1 \geq 5$ and $\theta_2 \geq 1.0$ in the training data.

Pang et al. (2002) conducted a study on sentiment analyses using movie review data. It was a document-level supervised learning and they applied SVM, Naïve Bayesian, and Maximum Entropy to the feature spaces they constructed. They chose several tokens such

as n-grams, POS tags, and adjectives as features to feature spaces. They found that the three machine learning methods outperformed the human conducted classifications (two students were asked to classify the corpus), and SVM outperformed other machine learning methods. They also found that bigrams did not perform better than unigrams with all three classification methods. To investigate performance of different weighting methods, they assigned binary feature values that denoted presences/ absences and frequencies as feature values. The results showed that presence could perform better than frequencies.

Gamon (2004) conducted a supervised learning for automatic sentiment classification using a very noisy domain customer feedback data. The motivation for their research was based on the fact that large volume of customer reviews is coming in every day, so it was necessary to propose a system that could deal with these large volume and noisy data automatically. Before applying machine learning classifiers, right features have to be selected for sentiment analyses. Gamon (2004) experimented with a range of different feature sets, from deep linguistic analyses based features to surface-based features. The surface-based features contain unigrams, bigrams, and trigrams. The linguistic features contain part-of-speech (POS) trigrams, constituent specific length measures (e.g., length of sentences), structure patterns (e.g., DECL::NP VERB NP denotes a declarative sentence consisting of a noun phrase, a verbal head, and a second noun phrase), and POS tags coupled with semantic relations (e.g., “Verb-Subject-Noun” indicates a nominal subject to a verbal predicate). Binary feature weighting values were assigned to the features. The results showed that the usage of linguistic analysis based features consistently contributed to higher classification accuracy in sentiment classifications.

Other than using right features (tokens or patterns) and assigning right feature weighting values, the application of feature selection methods is also important. Yang et al. (1997) pointed out that a major characteristic of text categorization problem is the high dimensionality of feature spaces. Features used in text categorizations are usually bag-of-word (BoW) features such as unigrams or n-grams in the corpus, the size of which are usually decided by the size of vocabularies contained in the corpus. A big corpus usually contains tens of thousands vocabularies. The high dimensionalities in a machine learning process could result in the curse of dimensionality, which refers to various phenomena that arise when analyzing and organizing high dimensional spaces (Wikipedia). High dimensions could cause a feature space to contain many sparse values. Yang et al. (1997) focused on evaluating and comparing several feature selection methods that can reduce dimensions of feature spaces in text categorizations. Feature selection methods that were compared in their studies included DF, IG, x^2, Mutual Information (MI), and term strength (TS). They used classification methods k-nearest-neighborhood (kNN) and Linear Least Squares Fit (LLSF) mapping. The reason that they chose these two classifiers was that both kNN and LLSF are n-nary (typical instance is binary) classifiers that provide a global ranking of categories given an input vector—the category ranking in kNN is based on similarities of the two neighbors measured by cosine value while the ranking in LLSF is determined by least square fit of the mapping. Using both of them could reduce the possibility of classifier bias. Each of the feature selection method was evaluated using a number of different term-removal thresholds. The results showed that IG, DF and x^2 could eliminate up to 90% or more unique features with either an improved or no loss in categorization accuracy under kNN and LLSF.

Rogati and Yang (2002) examined major feature selection methods (DF, IG, x^2 and IG2 (the binary version of IG)) with four classification algorithms—Naive Bayesian (NB) approach, Rocchio-style classifier, k-nearest-neighbors (kNN), and Support Vector Machine. They found that feature selection method that is based on x^2 statistics outperformed the other four selection methods.

Forman (2003) presented an empirical method to compare twelve feature selection methods to investigate which feature selection method or combination of methods was most likely to produce the best performance. They found that Information Gain (IG) could get highest precision among the twelve selection methods.

Except supervised learning, unsupervised learning is also used often for sentiment analysis. Unsupervised learning involves the calculation of the opinion polarities of opinion words, and classifies the documents or sentences by aggregating the orientation of opinion words.

Turney (2002) presented a simple unsupervised learning algorithm to classify the reviews based on recommended (thumbs up) or not recommended (thumbs down) reviews online. The sentiment classification of a review is predicted by the average semantic orientation (SO) of adjective or adverb phrases in the review. Opinions are usually expressed by adjectives and adverbs. They used Point-wise Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of pairs of words or phrases, which is to calculate semantic orientation (SO) of a word or phrase by subtracting mutual information between the word or phrase and the reference word “excellent” from the mutual information between the word or phrase and the reference word “poor”. The mutual information is the co-occurrence of the two words or phrase among millions of online documents. Using

410 reviews in 4 domain areas, they obtained 84% accuracy for the bank and automobile datasets, and 66% accuracy for the movie review datasets. They argued that movie reviews were difficult to classify, since movie reviews usually contain description words such as “bad scene” or “good scene” which are not sentiment words. Although they received a decent result, the way they calculated the semantic orientation (SO) of phrases was not efficient enough as it involved retrieving millions of online documents to get the co-occurrence of two words.

In sentiment analysis, especially in an unsupervised learning process, opinion word lexicons are usually created first. An opinion word lexicon is a list of opinion words with annotated opinion polarities. Then opinion word lexicons could be used to infer the polarities of other words in the context, or be treated as features in feature spaces for supervised learning. Based on Martin and White’s Appraisal Theory, Whitelaw et al. (2005) presented a method to extract *appraisal groups* to formulate a lexicon. “An appraisal group is a set of attribute values in several task-independent semantic taxonomies.” The authors focused on extraction and analysis of adjectival appraisal groups that were headed by an appraising adjective (such as ‘beautiful’ or ‘boring’) and optionally modified by a sequence of modifiers (such as ‘very’, ‘sort of’, or ‘not’). 1329 appraisal groups were extracted by using a seed list that contained a small number of appraisal groups and the corresponding opinion polarities, and bootstrapping methods. The extracted appraisal groups achieved high sentiment classification performance when treated as features.

Different domains or contexts usually need different opinion lexicons because opinion words are context dependent. One positive opinion word in one domain may be neutral in

another domain or context. So the already annotated polarity of an opinion word in one lexicon is usually called prior polarity. Wilson et al. (2005) proposed a method to automatically distinguish prior polarity from contextual polarity of a phrase. Beginning with a subjective clue (word or phrase) list provided by Rillof et al. (2003), Wilson et al. (2005) first expanded the list by combining subjective words provided by former studies and dictionaries, and annotated the polarities manually. A list of 8000 subjective clues, containing 33.1% positive, 59.7% negative, and 6.9% neutral subjective clues, was created. Because classifications that are based on prior polarities of opinion words are not accurate enough as discussed earlier, Wilson et al. (2005) conducted classification experiments by developing features such as word features, modification features, and structure features to identify contextual polarities of phrases. The authors finally developed 28 features for the subjectivity (neutral or subjective) classification and 10 features for polarity (positive or negative) classification. The developed features that took into account the contextual polarities produced high classification performance.

Several works (Kim et al., 2004; Eguchi et al., 2006) conducted topic-based sentiment analysis to find some relations between topics and sentiment expressions. Eguchi et al. (2006) proposed a method based on the assumption that sentiment expressions are related to topics. For example, negative reviews for some voting events may contain kinds of indicator word “flaw”. They combined topic relevance models and sentiment relevance models with parameters that were estimated from training data using retrieval models. Sentence-level analysis was conducted, and one sentence was treated as one statement. Each statement consisted topic bearing and sentiment bearing words. They trained the model by annotating S (sentiment) and T (topic) to sentiment words and topic words.

Then, S, T, and polarities of the sentiment words formed a triangular relationship, which was trained by a generative model. The classification obtained high performance using the trained models.

2.2 Sentiment analysis on product features

As depicted in the introduction, it is necessary and important to recognize product features and their related sentiments. Several studies (Hu et al., 2004; Mei et al., 2007; Ding et al., 2008; Titov et al.,2008; Lin et al., 2009) have proposed methods such as lexicon-based unsupervised learning to identify product features and their corresponding opinion polarities.

Based on the consideration that frequent nouns are usually the product features in product reviews, Hu et al. (2004) proposed a system to use association rule mining to extract frequent noun phrases as potential product features. In the first step, the explicit product features on which many people had expressed their opinions were extracted using association mining. After extracting the frequent nouns, two pruning methods were used to remove nouns that were unlikely to be the product features. In the second step, all adjectives that were treated as potential opinion words in sentences that contained product features were extracted. Then, for each product feature in the sentence, the nearby adjective was treated as its effective opinion. In the third step, the polarities (positive or negative) of opinion words were decided using WordNet and bootstrapping methods. A small list of seed words with prior-polarities was used to create opinion word lexicons. In the final step, the polarity of each sentence was decided by aggregating the opinion polarities of opinion words expressed on the product features in that sentence.

Ding et al. (2008) optimized the methods of Hu et al. (2004) by conducting a holistic rule-based analysis.

Popescu et al. (2005) proposed another method to improve the product feature extraction methods proposed by Hu et al. (2004). They evaluated each noun phrase by computing the PMI score between each phrase and its related phrases. The phrases are syntactic dependency tree patterns that were parsed from open source parser MINPAR. The related phrases (they called candidate phrases) were obtained by searching websites online. Then, the number of hits that denotes the co-occurrences between the phrases and their candidate phrases was calculated. The idea of this approach was very similar to the method proposed by Turney et al. (2002). If the PMI score was too low, then the phrase and its candidate phrases did not co-occur frequently. Therefore, they should not be grouped under the same product feature class. In this way, candidate phrases with low scores should be eliminated from the list of noun phrases. After extracting none phrases, Popescu et al. (2005) applied relaxation labeling methods to find out the semantic orientations of opinion words. Relaxation labeling is an iterative procedure whose output is an assignment of labels to objects.(Popescu et al., 2005, p4). They obtained an improved performance compared to Hu et al. (2004).

Instead of extracting product features, Choi et al. (2005) extracted opinion sources by using an extraction pattern learner called AutoSlog (Riloff, 1996). The opinion sources referred to the people or subjects that could express their opinions. AutoSlog relies on shallow parsers and can be applied exhaustively to a text corpus to generate information extraction (IE) patterns. AutoSlog can generate 17 types of extraction patterns such as passive-voice verb phrases (PassVP), active-voice verb phrases (ActVP), and infinitive

verb phrases (InfVP). Then, subjects (subj) or direct objects (dobj) can be extracted using these patterns. Based on the product feature properties, nouns or noun phrases that appear before the passive verbs or object words that appear after several verbs or preps are probably product features. So, using AutoSlog, nouns or noun phrases can be extracted accurately as product features. In the extraction patterns, we can find patterns that reflect passive voice and active voice. Table 2.1 shows some of extraction patterns.

Table 2.1 Part of extraction patterns from Rillof (1996;2003)

EXTRACTION PATTERNS	EXAMPLES
<subj> passive-verb	<VW Passat> was preferred
Passive-verb<dobj>	Preferred <cars>
Active-verb<dobj>	Lily likes <Passat>
Verb infin.<dobj>	Probably to buy <Passat>
Gerund <dobj>	Criticizing <Passat>
Noun prep<np>	
Active-verb prep<np>	

One product feature can have many expressions. For example, expressions such as “picture”, “image”, “photo” and “picture quality” could all be grouped as “picture quality”. Zhai et al. (2010) proposed a constrained semi-supervised learning method to group similar expressions of product features. This method used the Expectation-Maximum (EM) classification model. Based on their system, the users just need to provide a small list of labeled seeds for each feature group. The system then assigns other

similar feature expressions to suitable groups. However, the disadvantage of the EM algorithm is that it can only achieve local optimization, which is based on initial seed list. The authors proposed two soft constraints (prior knowledge) to provide a better initialization. The "soft" means that classification can be relaxed (modified) during the learning process. These two "soft" constraints are:

1. Feature expressions sharing some common words are likely to belong to the same group (e.g., 'battery life' and 'battery power').
2. Feature expressions that are synonyms in a dictionary are likely to belong to the same group (e.g., 'movie' and 'picture').

There are several researchers who have been working on grouping similar expressions of product features using topic modeling methods— Latent Dirichlet Allocation (LDA) (Blei et al., 2003; Lin et al., 2009; Guo et al.,2009; Zhai et al., 2011). LDA was first proposed by Blei et al. (2003) for topic classification. LDA is a generative model that allows sets of observations to be explained by unobserved groups. In LDA, each document may be viewed as a mixture of various topics. Although it is used for topic classification, LDA can be used to group the product features in sentiment analysis.

2.3 Fundamentals of supervised learning

2.3.1 Finding appropriate features

Sentiment analysis is a kind of text classification task. In supervised learning, a number of machine learning algorithms can be used in text classification to classify text. When using machine learning models, the major focuses of supervised learning have two aspects: constructing appropriate feature spaces and choosing appropriate classification

algorithms. In the Nature Language Processing (NLP) tasks, features are also called terms or tokens.

It is important to find out right features when using machine learning models for text mining. In sentiment analysis, many efforts have focused on finding right features to improve classification performance. If a particular feature tends to be highly consistent in the texts of a certain class (positive class or negative class), then the algorithm will generalize that this feature is a good indicator of that class (Brooke, 2009). For example, *beautiful* may be a good indicator to generalize a text as positive. To date, many features have been applied in sentiment analysis, such as unigrams, bigrams, trigrams, even higher level n-grams, POS tagged unigrams that reflect syntactic relations, dependency tree patterns that reflect semantic relations, negation-tagged tokens that reflect the effects of negation words, subjective extraction patterns, and adjectives. The objective of finding out these features is to find out good indicators to generalize text classifications. In the following sections, several feature types are discussed.

i. N-grams

An n-gram model is a type of probabilistic language model for predicting the next word conditioned on a sequence of previous words using Markov models. The probabilistic expression is $P(x_i|x_{i-(n-1)} \dots, x_{i-1})$. N-gram of size 1 is referred to as unigram, size 2 as bigram, and size 3 as trigram. Since n-grams are used for capturing dependencies between single words that stay in a text sequentially, the combination of words does not necessarily have syntactical or semantic relations. Unigrams performed much better than bigrams when used as features for feature spaces in Pang et al. (2002), while bigrams and trigrams contributed higher performance than unigrams in (Dave et al., 2003; Ng et al.,

2006). In Pang et al.(2002), unigrams also outperformed adjectives when treated as features.

ii. Negations

In sentiment analysis, both unsupervised learning and supervised learning deal with negation effects. In unsupervised learning, negation rules are usually applied to find out the contextual polarities of opinion words. In supervised learning, negation tags are usually used to tokens (features) that are behind a negation word. Negation tags for supervised learning will be discussed in this section, and negation rules for unsupervised learning will be discussed in Chapter 3.

Negation effect is one of the major effects to influence the contextual polarity of the opinion words and texts. Negation words or phrases, such as not, no, neither, and pattern-based negations such as “stop” + “vb-ing”, “quit” + “vbing” and “cease” + “to vb” usually reverse the polarities of the opinion words that adhere to them or follow closely behind them. In a sentence, words or phrases between a negation word and the first punctuation mark are usually tagged with negation tag _NOT to model the potentially important contextual effects of negations. But in supervised learning, it was pointed out by some research that negation tagged words that appeared after a negation word with special tags had a slightly helpful but mostly negligible effect on performance.

However, Pott (2011) applied the negation tagging methods proposed by Pang et al. (2002) and improved the classification accuracy from 0.886 to 0.895. Instead of tagging words between a negation word and the first punctuation, Wilson et al. (2005) tagged words within four words distance from the negation word to consider the negation effects.

Ikeda et al. (2008) proposed a polarity-shifting model to capture whether the polarity of a word is shifted (changed) or not. The model was a kind of binary classification model that determines whether the polarity is shifted by its context. The model assigns a score $s_{shift}(x, S)$ to the opinion word x in the sentence S. If the polarity of $x$is shifted in S, then $s_{shift}(x, S) > 0$, else $s_{shift}(x, S) < 0$. Compared to other features such as Bag-of-Word features, their model obtained higher performance.

Nakagawa et al. (2010) also pointed out that the consideration of interaction between words in sentiment analysis is necessary, and negation effects epically need to be considered. But, the simple Bag-of-Word features could not capture these interactions very well. Syntactic dependency tree patterns were used to capture the interactions between words. In their method, the sentiment polarity of each dependency sub-tree in the sentence is represented by a hidden variable, and the polarity of the whole sentence is calculated in consideration of interactions between hidden variables. They trained the model with Conditional Random Field (CRF) with hidden variables, and obtained higher performance with their model that was based on syntactic dependency features than Bag-of-Word (BoW) features with or without polarity reversal.

iii. Part-of-Speech (POS) Tagging

POS Tagging has been used for a long time in Nature Language Processing (NLP) and text classifications. Simple understanding of POS tagging is that to use some specific tags to differentiate syntactic meaning of words in a sentence, such as adjective, adverb, verb, none, conjunction, etc. Many English corpuses have been developed for POS tagging since the first major corpus called Brown Corpus was developed at Brown University.

Most often used POS tags are JJ to denote adjectives, RB to denote adverbs, VB to denote verbs, and NN to denote nouns.

In sentiment analysis, POS tagged words are usually used as features for supervised learning. Mejova et al. (2011) tested the effectiveness of different POS tagged features separately and with combination for supervised learning. The selected features contained adjectives, verbs, and nouns. The combination of adjectives, adverbs, and nouns performed better than individuals when treated as features in feature spaces. Adjectives performed the best among the three individual POS tagged features.

In our analysis, we applied Hidden Markov Model (HMM) based tagging provided by NLTK (www.nltk.org). HMM is a finite state automaton that has a set of states with probabilities attached to transitions between states.

iv. Syntactic dependency tree patterns

A syntax dependency tree is a syntax tree structure that is constructed by the syntax relation between a word (a head) and its dependents. Dependency structures identify useful semantic relationships. Dependency parsing transforms a sentence into quasi-semantic structures that can be useful for extracting sentiment information from texts (Pott, 2011). In syntactic dependency trees structures, each word or phrase is one leaf node, and two nodes are connected by one edge. The relations among nodes are based on dependency grammars. The parent word is known as the head in the structure, and its children are known as modifiers. Dependency parsing is for syntax analysis, which identifies the part-of-speech (POS), and syntactic relations, and then to determine the grammatical structure of sentences or phrases. Many researchers have focused on this

field to get efficient and accurate parsing tree patterns for sentiment analysis. Works such as (Collins 1997; Lin 1998; sha et al., 2003; Sang et al., 2002; Blache et al., 2001; Nakagawa et al., 2010) have applied the syntactic dependency trees to sentiment analysis and obtained higher performance than using Bag-of-Word features.

Words, phrases or patterns are usually given certain thresholds to be treated as features for machine learning models, the thresholds that measure effective frequency of occurrence. Syntactic dependency tree patterns are structured patterns, so they could occur very few times in a corpus, especially the longer syntactic patterns. Wilson et al. (2005) assigned thresholds for considering syntactic dependency trees as features. Those tree patterns that occur more than 70% in subjective expressions could be treated as potential features for machine learning models in sentiment analysis.

v. Extraction patterns

Riollf et al. (2003) used two different bootstrapping algorithms and a set of seed words to extract patterns from un-annotated data. Extraction pattern, which is a kind of features like N-grams, negations, or just word tokens, often represents role relationships surrounding noun and verb phrases.

When extraction patterns are treated as features, one feature is said to subsume another when the set of text spans that matched the first pattern (string) are supersets of the text spans that match the second. For instance, the unigram feature *good* would subsume the bigram feature *very good* or the information extraction (IE) pattern <*subject*> *is good*. In that way, complex features can be subsumed by simpler ones, and cut down the total number of features.

As an example mentioned in Rillof et al. (2003), ‘hijackings’ might subsume the pattern ‘hijacking of <x>’. The way it works is to look for the noun ‘hijacking’ and extract the object of the preposition ‘of’. The pattern ‘<x> was hijacked’ would extract the hijacked objects when it finds the verb ‘hijacked’ in a passive voice sentence, and the pattern ‘<x> hijacked’ would extract the hijacker when it finds the verb ‘hijacked’ in a active voice sentence.

2.3.2 Using appropriate numerical feature values

In sentiment analysis or the other NLP tasks, except that selection of appropriate features is important, the assignment of numerical feature values to selected features is also important. The feature value assigning methods are usually called feature weighting methods. The most widely used feature weighting methods are term frequency (TF) and presence.

When come across an input matrix (feature space) for a machine learning model, we may have a question that what the columns represent and what the rows represent. Turney et al. (2010) have discussed deeply on three vector space models, which are term-document matrix (space), word-context matrix, and pair-pattern matrix. Term-document matrix and word-context matrix are introduced in this section. The matrices are different based on different representations of columns and rows.

The term-document matrix is used to identify the similarity of documents. It is based on the hypothesis that if two documents have similar topics, then the two corresponding columns could have similar pattern of certain numbers such as frequencies. The row vectors of the matrix correspond to terms (features), and the column vectors correspond

to documents. Each column vector has the same size that depends on the size of vocabularies contained in the corpus. Each column is called one Bag-of-Word. The value in each cell represent the frequency of that word occurred in the corresponding document. Hence, most of cells should be weighted as 0, since each document only contains a small part of the vocabularies. If two documents have similar topics, then the two corresponding column vectors will tend to have similar pattern of frequencies (Turney, 2010).

The word-context matrix is used to identify the similarity of words (Tureney, 2010). It is based on the hypothesis that words occur in similar context could have similar meaning. Instead of looking at column vectors in a term-document matrix, we look at row vectors in a word-context matrix. The context is represented by words, phrases, sentences, paragraphs, chapters, documents, or more exotic contexts such as sequences of characters or patterns. In the matrix, each word is represented by a vector that contains different contexts of the word, which means that different contexts of a word can be developed and put into one row vector.

In sentiment analysis, the commonly used matrix (we call feature space in this study) is the reversed term-document matrix, which put documents or sentences in rows and features (terms) in columns. Pang et al. (2002) conducted document-level sentiment analysis using movie review dataset. They investigated the performance of several feature spaces with different features such as unigrams, bigrams or POS tagged features in the columns. Most often, especially in document-level sentiment analysis, the features are weighted by term frequencies (TF). For example, if a unigram *good* appears in the document *doc1* 3 times, then the feature *good* is weighted with number 3.

Another commonly used feature weighting values are binary numbers, which indicate the presence/ absence of tokens in the documents or sentences. For example, if unigram *good* appears (no matter how many times) in the document, then the feature value of *good* is 1, else the value is 0. Presences performed better than frequencies when it was used as feature weighting values in (Pang et al., 2002).

Term frequency- inverse document frequency (tf-idf) is another frequently used feature weighting method.

i. Term frequency- inverse document frequency (tf-idf)

Term frequency (TF) denotes the relative importance of a term to a document, which means that the more times a word appears in a document, the more important it is to that document, while tf-idf weight is a numerical statistic which reflects how important a word is to a document in a certain type (class) of collection or corpus (Wikipedia).

Based on TF weighting method, a word could get more weight when it appears frequently in one document. High weight values could contribute more information to the classification of the text. However, TF-based high weight values do not provide useful information to the classification all the time because a word that occurs frequently in one corpus may not be that important. For example, the term “the” may occur many times in almost all the documents in one corpus, but it cannot provide useful information to indicate the classification type of a document. So, simply assigning TF as weight values to a feature is not accurate enough. Based on the above considerations, $tf - idf$ weighting method was proposed, which is to reduce the weight of the word that occur most but have less contribution to the classification.

The mathematical representation of $\mathrm{tf-idf}$ is $\mathrm{tf-idf(t,d)} = \mathrm{tf(t,d)} \times \mathrm{idf(t)}$, in which $\mathrm{idf(t)} = \log_2 \frac{|D|}{|\{d:t \in d\}|}$, $|D|$is the total number of document in the corpus, and $|\{d: t \in d\}|$is the number of document where term t occurs. In this way, if the number of documents that contain the term is big, then $\mathrm{idf(t)}$ could get small value to reduce the weight obtained from the TF. For example, TF of word "the" is big in the corpus, and the corresponding idf is small, and then the weight from TF can be offset by the weight from idf. Vice versa, if a term has high TF values in some certain documents, and the number of documents that contain this term is small, which means that TF of the term in the whole corpus is small, and idf is big, then the term could be assigned a large weight to indicate the classification type of the certain documents. Many complex term frequency weighting methods also have been proposed by researchers such as Jones et al.(2000), Martineau et al. (2009), and Paltoglou et al. (2010), etc. Martineau et al. (2009) weighted the features by how biased the features are to one corpus by proposing $\Delta\mathrm{tf-idf}$ weighting method, which is calculated from the difference between $\mathrm{tf-idf}$ of features in the positive corpus and that in the negative corpus. The expression is as follows: $\Delta\mathrm{tf-idf(t,d)} = \mathrm{tf(t,d)} * \log_2 \frac{|P|}{|P_t|} - \mathrm{tf(t,d)} * \log_2 \frac{|N|}{|N_t|}$, in which P and N denotes the number of positive and negative documents in the training sets, P_t and N_t denotes the number of positive and negative documents that contain the term (feature) t. The object of this delta calculation is to boost the importance of words that are unevenly distributed between the positive and negative classes and to discount the evenly distributed words, since the value of an evenly distributed term (feature) could be zero under this calculation method. The more uneven the distribution, the more important the term should be to indicate the classification type of a document that contain the term. The proposed $\Delta\mathrm{tf-idf}$ weighting

method was evaluated using SVM, and obtained higher accuracy than the basic $\mathrm{tf} - \mathrm{idf}$ weighting method.

ii. Term presence (absence)

The presence (absence) is another frequently used feature weighting method. Presences are usually denoted by binary values – 1 denotes the presence, and 0 denotes the absence. It means that if a term (feature) appears in the document or sentence, then its weight value in that document or sentence is 1, else is 0. Pang et al. (2002) obtained higher accuracy using presences as features values than using frequencies as feature values. Paltoglou et al. (2010) also found that using binary features is better than raw term frequency (TF), although a scaled TF values performed as well as binary values.

iii. Other numerical features

Wilson et al. (2005) proposed a method to automatically classify the contextual polarity of expressions that contain subjectivity clues, which refer to words or phrases that have subjective usage (they may also have objective usage). In their experiment, they compiled 8000 subjectivity clues as an opinion word lexicon by expending a list of subjectivity clues from (Rillof et al., 2003) using dictionaries and thesauruses. Each word in the lexicon was tagged with reliability tag – strongsubj or weaksubj, and its prior polarity tag—positive, negative, both or neutral. The clues were divided into strong and weak subjective clues, where strong subjective (strongsubj) clues have subjective meanings with high probability, and weak subjective (weaksubj) clues have subjective meanings with low probability.

Wilson et al. (2005) found that words with non-neutral prior polarities frequently appear in neutral contexts. So it is necessary to consider interactions between words in a sentence when conducting sentiment analysis. The authors developed 28 features by analyzing linguistic rules that can capture contextual interactions among words for subjectivity classifications. 28 features that could capture contextual interactions were developed as shown in Table 2.2. For example, modification features are binary features that capture different types of relationships involving the subjectivity clue instances. The final results showed that 28 features performed better than Bag-of-Word features in subjectivity classifications.

Table 2.2 Features for identifying contextual polarities (Wilson et al., 2005)

Word Features word token word part-of-speech word context prior polarity: positive, negative, both, neutral reliability class: strongsubj or weaksubj	Sentence Features strongsubj clues in current sentence: count strongsubj clues in previous sentence: count strongsubj clues in next sentence: count weaksubj clues in current sentence: count weaksubj clues in previous sentence: count weaksubj clues in next sentence: count adjectives in sentence: count adverbs in sentence (other than not): count cardinal number in sentence: binary pronoun in sentence: binary modal in sentence (other than will): binary	Structure Features in subject: binary in copular: binary in passive: binary
Modification Features preceeded by adjective: binary preceeded by adverb (other than not): binary preceeded by intensifier: binary is intensifier: binary modifies strongsubj: binary modifies weaksubj: binary modified by strongsubj: binary modified by weaksubj: binary		Document Feature document topic

They also developed 10 features to capture interactions for opinion polarity classifications as shown in Table 2.3.

Table 2.3 Features for polarity classification from (Wilson et al., 2005)

Word Features
word token
word prior polarity: positive, negative, both, neutral
Polarity Features
negated: binary
negated subject: binary
modifies polarity: positive, negative, neutral, both, notmod
modified by polarity: positive, negative, neutral, both, notmod
conj polarity: positive, negative, neutral, both, notmod
general polarity shifter: binary
negative polarity shifter: binary
positive polarity shifter: binary

2.3.3 Feature selection methods

In text classification tasks, most techniques use Bag-of-Word features to represent documents, which can lead to big sized document vectors or sentence vectors in feature spaces. Different feature-selection methods are used to select most useful features to reduce the size of feature spaces and improve efficiencies. Feature selection methods are techniques that choose a small set of features out of a given set of features to capture the relevant properties or classifications of datasets.

Feature selection may be viewed as a form of weighting, in which some terms may get a weight of zero and hence can be removed from feature spaces (Turney, 2010). The idea

of selection is to give more weight to surprising features and less weight to expected features (Turney 2010). The hypothesis is that surprising features, if shared by two vectors, are more discriminative of similarities between the vectors than less surprising features. Based on information theory, a surprising feature has higher information content than an expected feature (Shannon, 1948).

In text domains, an effective feature selection method is essential to make the learning tasks efficient and accurate. Many feature selection methods have been proposed, such as Information Gain (IG), Mutual Information (MI), x^2-test (CHI), term strength (TS) and term presence (absence).

In feature spaces of machine learning models for text classifications, the size of document vectors or sentence vectors that are composed of Bag-of-Word features is usually big because it depends on the size of vocabularies in the whole corpus (dataset). For example, a corpus contains 5000 sentences, and the average number of vocabularies in one sentence is 5, then the size of document vectors or sentence vectors will be 25000 when unigrams are treated as features. Large sized vectors can slow the system down and they are inefficient. A common way to get rid of less effective features are applying feature selection methods. Before applying feature selection methods such as IG and MI, there are several simple ways to preprocess the feature spaces and reduce the size. One way is to use stop words. Stop words are usually domain specific, so it is important to find out the domain dependent stop word list. Table 2.3 shows a stop words list that can be applicable to most of the domains, which was provided by http://karpathy.ca.

2.4 stop words list

i	de	on	who
a	en	or	will
about	for	that	with
an	from	the	und
are	how	this	the
as	in	to	and
at	is	was	but
be	it	what	its
by	la	when	it's
com	of	where	

Another preprocessing way to eliminate the features is to use frequency thresholds. For example, Pang et al. (2002) selected unigrams that occurred at least 4 times in the corpus to delete unigrams that occurred less than 4 times. Forman (2003) pointed out that half of the total number of distinct words (vocabularies) may occur only one time, so eliminating words under a given low rate of occurrence could yield great savings. But this statement was not totally correct. From $tf - idf$ weighting method, rare terms could have high idf score, which means that rare terms may be good indicators for text classifications, depending on how it could balance well with TF scores.

The more advanced methods for feature selection can then be applied to select the most efficient features, such as IG, MI, or CHI. Theoretically, features selected by these selection methods can have the same or improved performance than the full feature set without selections (Yang et al., 1997). Two feature selection methods, IG and MI, will be applied to reduce the size of the proposed feature spaces in this study.

2.3.4 Machine learning classification methods

Text classifications using machine learning methods usually focus on finding right features, appropriate feature weighting values, feature selection methods, and right machine-learning algorithms.

The supervised machine learning algorithms are usually corpus-based classification methods, which are to find out co-occurrence patterns (e.g., frequency) of words in the corpus to determine the sentiments of words or phrases. Bayes Theorem is the basic theorem of many classification algorithms in text classifications. The theorem provides a way to calculate the probability of hypothesis based on its prior probability (Mitchell 2003, p156).

Bayes theorem is expressed as $\mathrm{p(h|D)} = \frac{\mathrm{p(D|h)p(h)}}{\mathrm{p(D)}}$, which is the cornerstone of Bayesian learning methods, because it provides a way to calculate the posterior probability $\mathrm{p(h|D)}$ from prior probabilities $\mathrm{p(h)}$, $\mathrm{p(D)}$, and $\mathrm{p(D|h)}$, in which, h is the hypothesis. More intuitively, h is the target classification in space H, and D is the training dataset. We are often interested in determining the best hypothesis h from the space H. In our problem, the target classification (the best probable hypothesis) is positive or negative. So, the Bayes theorem provides a direct method for calculating such probabilities. Any such maximally probable hypothesis is called maximum a posterior (MAP) hypothesis.

$\mathrm{h_{MAP}} = \mathrm{argmax_{h\in H}}\ \mathrm{p(h|d)} = \mathrm{argmax_{h\in H}} \frac{\mathrm{p(D|h)p(h)}}{\mathrm{p(D)}} \propto \mathrm{argmax_{h\in H}}\ \mathrm{p(D|h)p(h)} \propto$ $\mathrm{argmax_{h\in H}}\ \mathrm{p(D|h)}$, in which the terms $\mathrm{p(D)}$ and $\mathrm{p(h)}$ are dropped, because $\mathrm{p(D)}$ is constant independent of h, and, most often, we assume each hypothesis in H is equally

probable, in which priori $p(h_i) = p(h_j)$. So, h_{MAP} , which is the Maximum of $p(D|h)$, is also called maximum likelihood hypothesis.

i. Naïve Bayes classifier

Naïve Bayes classifier is based on the above maximum likelihood hypothesis. The Bayesian approach classifies a new instance by assigning the most probable target values v_{MAP} to the instance. $v_{MAP} = \text{argmax}_{v_j \in V}\, p(v_j | a_1, a_2 \dots a_n) = \text{argmax}_{v_j \in V}\, p(v_j) * \prod_i^n p(a_i | v_j)$, in which, attribute (feature) values$<a_1, a_2 \dots a_n>$ describe the instances. The assumption of the classifier is that attributes are conditionally independent given the target values. In real word situations, the conditional independence assumption clearly does not hold (Pang et al., 2002). But it is pointed out that this classification algorithm had performed pretty well in text classification tasks.

ii. Conditional random field (CRF)

Conditional Random Fields (CRF) is a statistical modeling method that is often applied in pattern recognitions. Pattern recognition is a task that assigns some sort of output values such as Tags or labels to given input values such as tokens using some specific algorithms (Wikipedia). CRF is often used for labeling or parsing sequential data, such as natural language text or biological data.

Lafferty et al. (2001) defined CRF on observations X and random variables Y as follows:

Let $G = (V, E)$ be a graph such that $Y = \{Y_v,\ v \in V\}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case. When conditioned on X, the random variables Y_v obey the Markov probability with respect to the graph:

$P(Y_v|X, Y_w, w \neq v) = P(Y_v|X, Y_w, w{\sim}v)$, where $w{\sim}v$ means that w and v are neighbors in G.

Choi et al. (2005) proposed sequence tagging and pattern matching techniques to train a linear-chain Conditional Random Field (CRF) to identify opinion sources for sentiment analysis. Features used in the study contain syntactic, semantic, and orthographic lexical features such as dependency parse features and opinion recognition features. They presented two source recognition methods—sequence tagging with Conditional Random Field (CRF) and pattern extraction with AotoSlog proposed by Riloff (1996). The performance was improved by combining the two methods.

iii. Support Vector Machine (SVM)

Support Vector Machine (SVM) is popular machine learning method for classification, regression, and other learning tasks (Chang et al., 2011). It is claimed to be an appropriate tool for sentiment analyses because it can be resistant to noise, and can handle large feature sets. SVM performed better than Naïve Bayes and Maximum Entropy Pang et al. (2002) for sentiment classifications. It also performed better in (Rogati and Yang, 2002) than kNN used in (Yang et al., 1997).

LIBSVM is one of the most widely used SVM tools currently. LIBSVM support two classification types, C-support vector classification and V-support vector classification. A good classifier should have higher classification accuracy for points that are farther from the margin. SVM is a discriminative method, and it needs to find out a margin to classify the categories. For instance, in a two-class classification, it needs to find out a linear

hyper-plane that is represented by $\vec{w}$ to classify two classes. An SVM can provide the distance of a test point from the margin.

iv. Bootstrapping

Bootstrapping is a machine learning method that seeks to combine many weak learners into one highly accurate classifier. The weak learners are trained in iterations, by adding a new weak learner to the classifier in each iteration step. Many studies have used bootstrapping to increase the seed list of opinion words for sentiment analysis such as works of Wilson et al. (2005) and Hu et al.(2004).

Chapter 3: THEORETICAL FOUNDATIONS

3.1 Fundamental Theories

Osgood et al. (1957) made an assumption that the semantic orientation (opinion polarities) of words can be expressed as numerical values. Based on this assumption, many studies on sentiment analysis have proposed sentiment orientation (opinion polarity) calculation methods for opinion words. The calculation often involves creating opinion word lexicons using dictionaries such as WordNet, or using statistical methods such as searching the co-occurrence of the words online. Lexicon-based approaches create opinion word lexicons with a small list of opinion words with their polarities and their synonymous words in WordNet using boot strapping methods. Wilson et al. (2005) developed an opinion word lexicon for sentiment analysis. The lexicon contains 8000 subjective clues (opinion words) with tagged prior-polarities and other annotations such as annotating the strength of opinion words. This lexicon was used in both unsupervised learning and supervised learning experiments in this thesis.

3.1.1 Sentiment consistency and lexicon-based approach

Opinion words lexicons are usually applied to a sentiment orientation calculation process. Normally, there are two ways to generate lexicons—corpus-based lexicon and dictionary-based lexicon. The methods in the corpus-based approach rely on syntactic or co-occurrence patterns of words in the corpus, and a seed list of opinion words. Dictionary-based approaches use synonyms and antonyms of words that can be searched in dictionaries such as WordNet to increase the opinion word seed list.

Liu (2010) pointed out that the major shortcoming of using dictionary-based approach to generate lexicons is that the approach is unable to find opinion words with domain specific opinion polarities. For example, “quiet” in “the speakerphone is quiet” is negative, while in “the engine of the car is quiet” is positive. Corpus-based approach can resolve the major limitation of dictionary-based approach, but it also has limitations that a large corpus is difficult to be prepared to cover all English vocabularies. Further, the same word can have different opinion polarities even in the same domain. Thus, finding domain dependent opinion words is tricky and may not be sufficient, and has a large space to be improved.

One way to improve the domain based or context based accuracy is to apply the linguistic rules and conjunction rules. For example, in one sentence, if the polarity of one opinion word is known, then polarity of another opinion word in that sentence can be inferred if these two opinion words are connected by AND, or other conjunction or negation words. This is called *sentiment consistency* by Liu (2010).

Ding et al. (2008) proposed a holistic lexicon-based approach to identify the polarities of context dependent opinion words based on linguistic rules – conjunction rules and negation rules. Conjunction rules basically state that when two opinion words are linked by AND or other conjunction words in a sentence, their opinion polarities are the same.

3.1.2 Appraisal theory for sentiment intensity

In a detailed semantic analysis, attitude expressions in the form of a well-designed taxonomy of attitude types and other semantic properties are needed. Whitelaw et al. (2005) presented a new method for sentiment classification based on extracting and

analyzing appraisal groups such as “very good”. Appraisal groups refer to the attitude expressions in the form of a well-designed taxonomy of attitude types and other semantic properties. Based on Martin and White’s Appraisal Theory, Whitelaw et al. (2005) assigned four main types of attributes to appraisal groups: Attitude, Orientation, Graduation, and Polarity.

$$Appraisal \begin{cases} Attitude \begin{cases} affect \\ appreciation \\ judgement \end{cases} \\ Graduation \begin{cases} forced \\ focused \end{cases} \\ Orientation \begin{cases} positive \\ negative \end{cases} \\ Polarity \begin{cases} marked \\ unmarked \end{cases} \end{cases}$$

Figure 3.1 Main attribute of Appraisal (Whitelaw et al., 2005)

Attitude provides a type of appraisal phrase being expressed as affect, appreciation, or judgment. Orientation indicates whether the appraisal phrase is positive or negative. Graduation describes the intensity of appraisal phrase such as using “very”, “slightly”, or “truly” to modify an adjective (or verb). Graduation consists of two dimensions – force (or ‘intensity’) and focus (‘prototypical’). Polarity of an appraisal phrase is marked if it is scoped in a polarity marker (such as ‘not’). Otherwise, it is unmarked. Brooke (2009) pointed out that there was a distinction between force graduation and focus graduation. Focus graduation involves sharpening or softening of attitude assessment (modifiers such as “really” or “truly”), whereas force graduation involves the scaling up or down of sentiments (modifiers such as very or extremely). However, it is pointed out that they do not differ much in their overall effects on the intensity to a word they modify. In sentiment analysis, these modifiers could be used by no differentiations. These modifiers

could be referred jointly as intensifications. An intensification list provided by Brooke (2009) was used in the unsupervised learning experiment in this thesis.

Appraisal groups are phrases that compose modifiers and being modified words, such as *very beautiful*. Whitelaw et al. (2005) focused on extracting and analyzing adjectival appraisal groups headed by appraising adjectives (such as beautiful) and optionally modified by modifiers to build a lexicon using semi-automatic techniques. They created a lexicon contains 1329 adjectival appraisal groups classified to the above appraisal taxonomies. Finally, they treat their appraisal groups as features, and compared them to the Bag-of-Words (BoW) classification methods. The approach received high performance when the appraisal groups are treated as features alone or when they combined with BoW features.

From the above discussions about appraisal groups, we could figure out that the modifiers in appraisal groups usually play the roles of intensifications, and these intensifiers could improve the performance of classifications.

3.1.3 Semantic differential theory

The semantic differential measures people's reactions to stimulus words and concepts in terms of ratings on bipolar scales which are defined with contrasting adjectives at each end – bad and good (Heise, 1970).

Osgood et al. (1957) tried to quantify the words in their famous book: The Measurement of Meaning. Osgood's semantic differential was designed to measure the connotative meaning of concepts through classifying adjectives. They found through factor analysis three recurring attitude factors that people use to evaluate words and phrases: evaluation,

potency, and activity. Evaluation loads highest on the adjective pair 'good-bad'; potency loads on 'strong-weak'; and activity loads on 'active-passive'. Based on Osgood's theory, we can imagine that one adjective can be mapped to a multidimensional semantic space as a point, with the attribute factors as the axes. So, the distance of the point to every axis could demonstrate which attribute factor it belongs. The lesser the distance from that axis, the higher the possibility for the word belonging to that attribute factor. Regardless of its long distance from the other axis, it could still be affected by the other attribute factors. This means that if a word is evaluative, it could still express "strong-weak" or "active-passive" meanings. In other words, Osgood pointed out that the other attribute factors could have unpredictable effects on the evaluative attribute of a word.

Sentiment analysis, which is mining peoples' attitudes towards products or other objects, most often deals with evaluative words, especially adjectives. So, it is appropriate for Turney et al. (2002) to choose "poor" and "excellent" as two reference words to calculate the opinion polarities of opinion words by measuring the distances between the opinion words and the reference words.

Kamps et al. (2004) used path length distance in WordNet to derive semantic differential values. Basically, they counted the minimum number of synonym relation links intervening between a word and the prototypical examples of each of the three factors (i.e., good/bad for Evaluation, strong/weak for Potency, and active/passive for Activity). For example, the expression of sentiment orientation of evaluative adjectives is $\mathrm{EVA}\ (\mathrm{w}) = \frac{\mathrm{d(w,bad)-d(w,good)}}{\mathrm{d(bad,good)}}$, where $d(w, bad)$ is a distance that is a straightforward generalization of synonym relation between w and bad. A synonym relation connects words with similar meaning. The range of the expression is between [-1, 1]. The negative

word w is in the range of [-1, 0], and the positive word w is in the range of [0, 1]. With the same idea, the polarity of potency adjectives and activity adjective can be expressed as: $\mathrm{POT(w)} = \frac{\mathrm{d(w,weak)-d(w,strong)}}{\mathrm{d(weak,strong)}}$, and $\mathrm{ACT(w)} = \frac{\mathrm{d(w,passive)-d(w,active)}}{\mathrm{d(passive,active)}}$.

3.2 Rules and constraints

The identification of opinion words expressed on the product features (Hu et al., 2004; Kim et al., 2004; Popescu, et al., 2005; Ding et al., 2008) always involves the usage or creation of domain based opinion lexicons, which contain opinion words with their polarities. Usually the positive opinion words are assigned value +1, and negative opinion words are assigned value -1. The polarities of opinion words around the product features are usually aggregated to the product features by considering linguistic rules and constraints, the linguistic rules such as negation rules, conjunction rules and intensification rules. For example, “beautiful” has polarity +1, and if there is a “not” between a product feature and the “beautiful”, then the contextual polarity of the “beautiful” is -1. Hence, the consideration of these linguistic rules could capture the contextual polarities, and improve the final classification accuracies. Moreover, we could infer the polarities of the unknown words by using linguistic rules. For example, in “very beautiful and long”, we know the polarity of “beautiful” is positive, then we can infer from the conjunction rule that the polarity of “long” is also positive. Further, the polarities of these two opinion words could be intensified by the intensifier “very”.

We considered negation rules, conjunction rules, and intensifiers in our analyses.

3.2.1 Negation rules

In supervised machine learning processes, the negation effects are always considered by using negation tags. As mentioned by Das et al. (2002) and Pang et al. (2002), the words between negation words and the end of the sentences are all tagged with _NEG, and then these negation tagged words could be used as features and assigned Bag-of-Word feature values.

In unsupervised learning, which usually involves the sentiment orientation calculation, the negation rules can be considered in several ways. The most straightforward way of representing negation in a quantificational framework is using polarity switch: 1 -> -1. But most of the time whether a word can be negated by a negation word depends on the contextual situation. Brooke (2009) gave an example about *functional* and *not functional*, in which *functional* has the polarity of +1, but *not functional* seems somewhat worse than -1. These negation subtleties could be classified as contradictory versus contrary negations (Brooke, 2009).

Godbole et al. (2007) reversed the polarity of a sentiment word whenever it was preceded by a negation and increased/decreased the polarity strength when a word is preceded by a modifier. For example, not good = -1; good = +1; very good = +2.

The application of negation rules to the unsupervised learning usually involves the usage of negation list, such as “no”, “not”, “never”, “rarely”, and words in the patterns such as “stop vb-ing”(POS tag vb denotes verb), “cease to vb” , etc. For example, if there is a “stop” in the datasets followed by a verb that is in the vb-ing form, then the phrase “stop vb-ing” will be treated as negation word. We use the negation list provided by Pott et al. (2011) in our experiments.

Ding et al. (2008) considered the negation in three situations: negation-negative is positive, negation-positive is negative and negation-neutral is negative, but their rules only applied to the bigrams or consecutive phrases they extracted. Their first rule is that if a negation word exists with negatives, then the whole phrase is positive. The second rule is that if a negation word exists with positives, then the whole phrase is negative; and the third rule is that if a negation words exists with neutral, then the whole phrase is negative. In reality, we know that negation words could also have effects on the words that are far away from them. In our work, we also consider the situations in which negation words and opinion words are not consecutive.

Some works such as Godbole et al. (2007) consider the "far away" effect by dividing the polarities of opinion words by the distance between the two words. The significance of modification decreases as the distance increase.

3.2.2 Conjunction rules

Conjunction rules basically state that when two opinion words are linked by AND, BUT or other conjunction words in a sentence, their opinion polarities are the same or different. In this way, the polarity of one word can be inferred by the polarity of another one.

Hatzivassiloglou et al. (1997) hypothesized that adjectives separated by AND have the same polarities, while those separated by "BUT" have opposite polarities. Liu et al. (2010) also proposed a sentiment consistency concept based on the conjunction rules, which consider other constraints to the connectivity -- OR, EITHER OR, NEITHER-NOR, and BUT.

3.2.3 Intensification rules

Quirk et al. (1985) classified intensifiers into two major categories: amplifiers (e.g., very) that increase the semantic intensity of appraisal words that appear not far away, and down-toners (e.g., slightly) that decrease the intensity of appraisal words that appear around.

Brooke (2009) generalized a bunch of intensification effects by annotating a list of intensifiers numerated with their intensification percentages. Intensifiers are usually adverbs. The list could cover most of the common intensifiers. The list was used in the unsupervised learning experiment in this study. Table 3.1 shows part of the intensifiers with their numerated intensification percentages. As an example, if "sleazy" has a polarity value of -3, then "somewhat sleazy" would have a polarity value of -3 + (-3 *-30%) = -2.1. Intensifiers are additive. If "good" has a polarity value of 3, then "really very good" has a polarity value of 3 + (3 *15%) + (3 *25%) = 4.3.

Table 3.1 intensifier list (Brooke, 2009)

Intensifier	Modifier%
Slightly	-50%
somewhat	-30%
pretty	-10%
Really	15%
Very	25%
extraordinarily	50%
(the)most	100%

Chapter 4: RESEARCH METHODS

This research aims to study people's sentiments/ opinions expressed on product features. In this section, we introduced and proposed methods to conduct unsupervised and supervised learning on sentiment analysis that focused on product features in product reviews. In unsupervised learning, we utilized linguistic rules and constraints to calculate sentiment score of product features, and to investigate whether intensification rules could improve the performance of the method (Ding et al., 2008) that only used conjunction rules and negation rules. In supervised learning, we conducted document-level and sentence-level sentiment analyses to investigate whether product features were good indicators in determining classifications of documents or sentences, and to investigate whether the features that developed by considering linguistic rules could perform well in supervised learning either.

The unsupervised learning process is a sentiment score calculation process – to calculate sentiment scores of product features by aggregating polarities of opinion words expressed on product features. Opinion polarities of opinion words also called sentiment orientations. The calculation was based on the equation provided by Ding et al. (2008), and the rules and constraints we discussed in Chapter 3. One more rule (intensification rule) and sentence constraints were added to the method proposed by Ding et al. (2008). An improved calculation performance was expected by considering additional rules and constraints.

In the supervised learning, product features were included as features in document vectors or sentence vectors in feature spaces. To the best of our knowledge, prior studies that were related to sentiment analysis on product features are all unsupervised learning. The consideration of product features as features in feature spaces of machine learning methods in supervised learning is a pioneering effort. Further, we applied linguistic rules to the feature spaces by developing rule-based features.

The phases in our supervised learning research are: (i) choose the right features for product feature based sentiment analysis and construct document-level and sentence-level feature spaces; (ii) apply two feature selection methods to the proposed feature spaces and compare the results; and (iii) compare our proposed feature spaces to those of Pang et al. (2002).

Although the unsupervised sentiment calculations and supervised machine learning are two different methods, we applied the same linguistic rules to the two problems and expected improved performance in both of the tasks. Product features extracted from the unsupervised learning could also be applied to the supervised learning directly.

4.1 Product feature extraction

The goal of this step is to extract product features that have been commented on in the product reviews, and to determine whether their opinions on the product features are positive or negative.

Before extracting the product features, we considered three properties of product features based on the considerations in Hu et al. (2004). First, the product features are noun (POS:

N) or noun phrases (POS: NP). Second, the product features are usually objects in a sentence. Third, product features are directly related to opinion words or phrases.

One product can have many features. For example, a product such as computer can have features such as monitor, CPU, memory, hard drive, etc. Each feature can be expressed with a finite set of words or phrases. For example, monitors may be expressed as pictures, images or screens. So, it is difficult for computer to understand such fuzzy phrases and features. Hu et al. (2004) applied POS-Tagging to extract the product features after several preprocessing steps – removing stop-words, stemming, and fuzzy matching. Both Hu et al.(2004) and Popescu et al. (2005) used the association rule mining to extract the frequently occurred noun phrases as potential product features. We have discussed the detailed steps in Chapter 2. Popescu et al. (2005) obtained higher precision than Hu et al. (2004) when extracting the product features. The major difference was that Popescu et al. (2005) used the feature assessor that could evaluate each candidate noun phrase by computing the PMI scores between the noun phrases and the whole-part discriminators (they had already known the product class information and could figure out whether the properties, parts, or features of parts should belong to that product class).

To extract more accurate product features is beyond the scope of this research. Product features that were provided by Hu et al. (2004) were directly used in the unsupervised learning experiment. In the second and third experiments of the supervised learning, we extracted the product features using the extraction method proposed by Hu et al. (2004).

4.2 Extract opinion words around product features

Adjectives are found as effective terms for identifying opinion words in either subjectivity classifications or polarity classifications. Wiebe et al. (1999) used statistical methods to validate that adjectives had positive correlations with opinion words. Adverbs sometimes are also used to identify the opinion words. Verbs and nouns can also be used to express opinions. Based on the method of Hu et al. (2004), we extracted adjectives as potential opinion words.

Hu et al. (2004) proposed an interesting yet efficient method when extracting opinion words around product features. They first extracted frequently occurred noun phrases to treat them as potential product features. Then they extracted the potential opinion words (adjectives) from the sentences that contain the frequent noun phrases. They stated that if the sentence contains both product features and opinion words, then the sentence would be an opinion sentence. After extracting the potential opinion words, they indentified the polarities of the opinion words by utilizing synonymous set and antonymous set in the WordNet, and a small list of opinion words with opinion polarities.

The major shortcoming of the method proposed by Hu et al. (2004) was that, after identifying the polarities of opinion words, they assigned the same polarities to the product features that were adjacent to the opinion words. This does not work well in most situations. First, in most situations, the product features and adjectives do not appear adjacent to each other. So, using this polarity assigning method, many product features cannot obtain polarities. Second, if the opinion words are adjacent to negation words, then the polarity expressed on the product feature should be reversed. So, finding a more

accurate way to assign polarities to product features is the major objective of unsupervised learning in this research.

Ding et al. (2008) proposed a holistic rule-based method to calculate the sentiment score of product features based on linguistic rules and constraints. They also used the same methods proposed by Hu et al. (2004) to extract product features using part of the same dataset. To compare the result obtained in this learning with Ding et al.(2008), we used the product features and the opinion words lexicon provided by Hu et al (2004). Ding et al. (2008) also used the same opinion word lexicon. We used the opinion word lexicon provided by Wilson et al. (2005) to expend the opinion words coverage. Therefore, in this research, the identification of opinion polarities of opinion words was not conducted.

4.3 Unsupervised learning—Calculate Sentiment Score of Product Features

The ability to establish relatedness, similarity, or distance between words and concepts is at the heart of computational linguistics (Kamps et al., 2004). WordNet is a syntactic lexicon to group English word into sets of synonyms and antonymous. Research that is related to calculating the opinion polarities of opinion words usually use WordNet to create a dictionary that contains opinion words and their prior polarities. 8000 opinion words with their prior polarities (positive, negative, both or neutral) were annotated by Wilson et al. (2005), and they were called subjective clues. 6800 opinion words were tagged with positive or negative polarities by Hu et al. (2004). Instead of creating opinion word lexicon ourselves, we used the above two lexicons in both unsupervised learning and supervised learning directly. In a contextual environment, the opinion polarity and/ or the strength of an opinion word may be changed because of the existence of negation words, conjunction words, or intensifiers.

Ding et al. (2008) proposed a method to calculate sentiment scores of product features by aggregating polarities of opinion words expressed on product features. Using the opinion word lexicon that have already been created, sentiment score of each product feature that was obtained from one opinion word was calculated using the proposed equation: $\mathrm{Score(f)} = \frac{\mathrm{SO(w)}}{\mathrm{dis(w,f)}}$, where f is a product feature, w is an opinion word, $SO(\mathrm{w})$ is the opinion polarity (sentiment orientation) of w that was contained in the opinion lexicon (-1 was assigned to negative opinion words and +1 was assigned to positive opinion words), and $\mathrm{dis(w, f)}$ is the distance between the opinion word and the product feature in one sentence. The distance is represented by the number of words between the product feature and the opinion word.

The same equation was used to calculate sentiment score of each product feature that was obtained from one opinion word in this unsupervised learning experiment. The major difference between the method we proposed and the method by Ding et al. (2008) in this step is in $\mathrm{dis(w, f)}$. The distance within one sentence was considered in Ding et al. (2008), while the distance within two consecutive sentences was considered in the method we proposed. Based on the properties of product feature we have discussed, pronouns in the next sentence may be related to the product features in the first sentence. Therefore, if there are two consecutive sentences, the first sentence contains a product feature, and the second sentence does not contain product features but pronouns, then we assume that the pronouns in the second sentence may refer to the product feature in the first sentence (we call this sentence constraints). If it is the case, then the distance between the product feature that is in the first sentence and the opinion words that are in the second sentence was also considered in the calculation.

After the sentiment score of each product feature obtained from one opinion word was calculated, the aggregated sentiment score of each product feature obtained from all the opinion words that were contained in two consecutive sentences was aggregated by considering three linguistic rules. The three linguistic rules were negation rules, conjunction rules, and intensification rules. Two linguistic rules – negation rules and conjunction rules were applied in Ding et al. (2008). The way that the linguistic rules were used to aggregate the sentiment score of product features is as follows:

(i) Negation rule. We followed the negation rules that were used in Ding et al. (2008) — negation-positive is negative, negation-neutral is negative, and negation-negative is positive. If an opinion word was adjacent to a negation word, then its polarity was reversed. However, the way that we used the negation rules was a little bit different. First, if there was a negation word between the product feature and opinion word, then the polarity of the opinion word was reversed, so it was not restricted to "adjacent". Second, the sentence constraint was also applicable, which mean that if the two consecutive sentences satisfied the restrictions described above, then the negation words between the product feature and opinion words could also reverse the polarities of the opinion words.

(ii) Intensification rule. Intensifier list provided by Brooke (2009) was applied. The list contains the numerated intensification percentages of the intensifiers. If the opinion word was adjacent to an intensifier, then its polarity was multiplied by the intensification percentage of the intensifier.

(iii) Conjunction rule. Conjunction rules were majorly used to determine the contextual polarities of opinion words by Ding et al. (2008). Each opinion word has two polarity attributes: prior polarity and context polarity. Prior polarity is already given in opinion

word lexicon. Contextual polarity is the polarity of an opinion word in the context, and it may different from its prior polarity. So, if there was a conjunction word between two opinion words, then the polarity of one opinion word could be inferred from that of another opinion words. The sentence constraint was also applicable when using conjunction rules. If two opinion words were connected by the conjunction word “AND”, then polarities of two opinion words were the same. Figure 4.1 shows the pseudo codes used in this unsupervised leaning.

```
Assume positive polarity equals 1; negative polarity equals -1
negated = false
intensified = false
foreach word w in a sentence:
   if w is a negation word:
      negated = true
   if w is a intensifier:
      intensified = true
   if w is a conjunction word:
      negated = false
      intensified = false
   if w is a known opinion word       #i.e. with known prior polarity
      if negated == false and intensified == false:
         w's context polarity = w's prior polarity
      elif negated == false and intensified == true:
         w's.contex_polarity = intensifying_rule(w's prior_polarity, the intensifier before w)
      elif _winfo.negated == true and _winfo.intensified == false:
         w's.contex_polarity = reversed w's prior_polarity
      else:
         w's.contex_polarity = intensifying_rule(w's prior_polarity, the intensifier before w)
foreach word w in a sentence:
   if w's context polarity is not known, but it is a adjective, adverb, or verb:
      foreach word v in the same clause with w:
         if v's context polarity is known:
            w's context polarity = v's context polarity
            break
      if w's context polarity is not known:
         if w is in the first clause:
            foreach word v in the next clause:
               if v's context polarity is known:
                  w's context polarity = conjunction_rule(v's context polarity, the conjunction word
between v and w)
         else:
            foreach word v in the previous clause:
               if v's context polarity is known:
                  w's context polarity = conjunction_rule(v's context polarity, the conjunction word
between v and w)

def intensifying_rule(p, w):
   level = w.level          #level > 1, p is strengthened; otherwise, p is weakened
   return p * level

def conjunction_rule(p, w):
   if w is a negative conjunction word (e.g., but):
      return the reverse of p
   else:
      return p
```

Figure 4.1 Unsupervised sentiment score calculation of product features

4.4 Supervised machine learning methods

4.4.1 Constructing feature spaces for machine learning methods

Feature spaces of machine learning models usually contain numerical data. In the text mining processes, the texts are always preprocessed by tokenization, and annotations, and then go through mathematical processes to assign numerical weights to features.

Bag-of-word features were commonly used in sentiment analysis or other NLP tasks. Let $\{f_1, \dots, f_m\}$ be a predefined set of m features that appear in a document, such as n-grams, dependency tree patterns or other tokens that could be treated as features. Let $n_i(d)$ or $n_i(s)$ be the number representation of token f_i in the document d or sentence s, either frequency or binary value. Then, each document d is represented by the document vector $\vec{d} := \left(n_1(d),\ n_2(d), \dots, n_m(d)\right)$ or sentence vector $\vec{s} := \left(n_1(s),\ n_2(s), \dots, n_m(s)\right)$. Document vectors or sentence vectors formulate document level or sentence level feature spaces. Both document level and sentence level feature spaces were constructed for the supervised learning experiments in this study. The objective of conducting both document-level and sentence-level analysis was to see whether these two level analysis will have different performance under the selected features.

Features that were selected for sentiment analysis on product features:

The first feature set in feature spaces was product feature set. Product features were those extracted in the unsupervised learning. Feature values that were assigned to each product feature were the distances (the number of words between them) between the product feature and its nearest opinion word in that document or sentence. Based on the hypothesis that a product features has closer distance with positive words in a positive

review and has closer distances with negative words in a negative review, positive distance was assigned to the product feature when its nearest opinion word is positive, and negative distance was assigned to the product feature when its nearest opinion word was negative. (i) The sentence constraint used in the unsupervised learning could also be applicable here. If the two consecutive sentences satisfied the restrictions as described in unsupervised learning section, then the opinion words in the second sentence could be considered if the first sentence did not contain opinion words. (ii) If both of these consecutive sentences did not contain opinion words but adjectives, then the distance between the product feature and its nearest adjective was considered. (iii) If these two consecutive sentences did not contain opinion words or adjectives, then we assign the product feature a large number, which is 30, as feature value, since most of sentences cannot contain more than 30 words. To consider linguistic rules and constraints in supervised learning, rule-based features were developed based on the rules and constraints we discussed in chapter 3.

The second feature set of feature spaces was composed of adjectives. In many former studies such as Pang et al. (2002) and Turney (2002), researchers used adjectives as features for sentiment classifications. For adjective features, we assigned the frequency of these words in that document or sentence as feature values. As the corpus size and vocabulary size increase, the number of these features should increase. Top 30% the mostly occurred adjectives were retained in the feature set. Pang et al. (2002) retained features that occurred more than 4 times.

The third feature set took into account rules and constraints that were discussed in Chapter 3. The rule-based feature sets for both document-level and sentence-level

analysis were developed by following the methods proposed by Wilson et al. (2005), which was to identify phrase level contextual opinion polarities of opinion words. Table 4.1 shows the rule-based features for document-level analysis.

Table 4.1 Rule features in document vector

1. Count for positive words in the document
2. Count for negative words in the document
3. Count for negation words in the document
4. Count for sentences that have positive words and conjunction words
5. Count for sentences that have negative words and conjunction words

Table 4.2 shows the rule-based features for sentence-level analysis.

Table 4.2 Rule features in sentence vector

1. Count for positive words in current sentence
2. Count for negative words in current sentence
3. Count for positive words in the next sentence if the next sentence contain possessive pronouns
4. Count for negative words in the next sentence if the next sentence contain possessive pronouns
5. Count for negation words in the current sentence
6. Count for negation words in the next sentence
7. Count for conjunction words in the current sentence
8. Count for conjunction words in the next sentence

4.4.2 Comparison of feature selection methods: MI vs. IG

1. Information Gain (IG)

Information Gain (IG) is widely used as a goodness criterion in the field of machine learning. We need to understand "Entropy" before explaining the definition of IG. Entropy quantifies the expected value of information that contained in a specific message and measures the uncertainty of random variables. Maximum Entropy was used often as feature weighting method. Abbasi et al. (2008) began with a fairly wide range of syntactic (e.g., N-grams, POS) and stylistic features (e.g., appearance of function words, vocabulary richness, even appearance of individual letters) and showed how feature selections that were based on Maximum Entropy could be effective in significantly boosting performance.

Information Gain could be calculated from Maximum Entropy. For example, the estimated probability of female population is 0.5 in the world, but if it is the conditional probability based on countries or districts, then 0.5 will not hold. In China, it is said that the male population is bigger than female population. Therefore, if the input could be classified to its own class first, which is not uniformly distributed, then it should have reduced entropy. This is expressed as follows: given $X = \{x_1, x_2, \ldots x_n\}$, if $p(X = V_1) = p_1, \ldots p(X = V_n) = p_n$, then the entropy of X is denoted as $H(X) = -\sum_{i=1}^{i=2} p_i \times \log_2 p_i$ .With the assumption that input vectors are independent of each other, then IG can be explained with the expression: $IG(Y|X) = H(Y) - \sum_i P(X = v_i)\, H(Y|X = v_i)$. The larger the information gain, the less effort used to transmit from X toY. Yang et al. (1997) treated IG in another way. The number of bits of information obtained for category prediction is measured by knowing the presence or absence of a term (feature) in a

document. Let $\{c_i\}_{i=1}^{m}$ denote the set of categories in the target space. Then, IG of term t is defined to be:

$$G(t) = -\sum_{i=1}^{m} P(c_i) log P(c_i) + P(t) \sum_{i=1}^{m} P(c_i|t) log P(c_i|t) + P(\bar{t}) \sum_{i=1}^{m} P(c_i|\bar{t}) log P(c_i|\bar{t})$$

Given a training corpus, the information gain for each term is computed and the terms whose information gains is less than some predefined threshold will be removed from the feature space. Based on the IG expression, common features could get higher IG values than rare features (low frequencies).

2. Mutual Information (MI)

Mutual information (MI) is a commonly used criterion in statistical modeling of word associations and related applications (Yang et al., 1997). The MI of two random variables is a quantity that measures the mutual dependence of two random variables. In text classification, as Yang et al. (1997) introduced, there is a contingency table of term t and a category c, where A is the number of times t and c co-occur, B is the number of times the t occurs without c, C is the number of times c occurs without t, and N is the total number of documents, then the MI between t and c is defined to be $I(t,c) = log \frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)} = log P_r(t|c) - log P_r(t)$, and is estimated using

$I(t,c) \approx log \frac{A \times N}{(A+C) \times (A+B)}$. $I(t,c)$ is zero if the t and c are independent. Based on the equation, for terms with an equal conditional probability $log P_r(t|c)$, rare terms will have higher score than common terms (Yang et al., 1997).

So, the terms that will be deleted based on IG and MI selection are different—common terms may be deleted by MI, while rare terms may be deleted by IG.

Chapter 5: EXPERIMENTAL SETUP

5.1 Experimental Data

5.1.1 Corpuses

i. Movie review dataset (Pang et al., 2002)

The movie review dataset provided by Pang et al. (2002) is a corpus of customer reviews of movies in IMDB web set. The corpus contains 1000 positive reviews and 1000 negative reviews. The corpus also contains 5331 positive and 5331 negative processed sentences.

ii. Amazon.com product review dataset (Hu et al., 2004; Ding et al., 2008)

The annotated datasets provided by Hu et al. (2004) and Ding et al. (2008) contain reviews for 5 products and 9 products respectively. Each dataset contains hundreds of reviews (documents), and each document and most of sentences in every document are labeled polarities. They also annotated the product features that appeared in the sentences.

5.1.2 Lexicons

i. Subjective clues (Riloff, et al., 2003; Wilson et al., 2005)

Subjectivity clues are words and phrases that have subjectivity expressions, such as emotions or attitudes. The lexicon was developed by Wilson et al. (2005), and it contains 8000 subjective clues. The phrases in the lexicon are adjectives, adverbs, nouns, and verbs with their polarities annotated. Each line in the lexicon contains one subjectivity clue and its corresponding types. The types contain reliability type which refers to strong subjective or weak subjective; length type that refers to the length of the clues; and prior-

polarity type that refers to whether the prior polarity of a clue is positive, negative, both, or neutral.

ii. WordNet

WordNet is a large lexical database of English created by Princeton University. Nouns, verbs, adjectives, and adverbs are grouped into sets of conceptual groups. Adjectives are organized into synonym and antonym clusters with a given adjective. We can search the synonyms and antonyms of a word from the WordNet.

iii. Opinion words for product review (Hu et al., 2004)

We also added the 6800 positive and negative opinion words provided by Hu et al. (2004) and Ding et al. (2008) to the lexicon list. These opinion words were mainly developed using the product reviews from amazon.com.

5.2 Experimental steps

Unsupervised learning and supervised learning for sentiment analysis on product features were discussed in this study. In unsupervised learning, sentiment scores of product features are calculated by aggregating opinion polarities of opinion words around the product features. In supervised learning, construction of feature spaces that are specific to sentiment analysis can improve time and space efficiency. In this experiment, feature spaces that contained right features was constructed by considering the product features and linguistic rules. Two feature selection methods were then applied to the proposed feature spaces. In the final experiment of the supervised learning, the proposed feature spaces were compared to those of Pang et al. (2002).

In the Natural Language Processing (NLP) tasks, it always involves preprocessing texts by tokenizing, normalizing, and annotating with different annotations to discriminate syntactic and semantic differentiations.

5.2.1 Tokenization

Text needs to be preprocessed as tokens or strings before applying machine learning models. This process is called tokenizing. We used the tokenization tools provided by www.nltk.com.

5.2.2 Normalization

The purpose of Normalization is to reduce the size of feature space. The most common types of normalization are case folding (converting all words to lower case) and stemming (reducing inflected words to their stem or root form) (Terney et al., 2010). Some studies found that stemming could not improve the classification performance. Normalization could also decrease the precision. Case folding normalization was applied in the preprocessing steps in this study.

5.2.3 Annotation

After turning the text into a list of tokens, the next step is to identify the syntactic and semantic groupings, and relationships that are relevant to sentiment. Part-of-speech tagging and negation tagging were applied in this study.

Sentiment words usually have opposite meaning when they are correlated with negation words in their semantic scope. The negation tagging is to get the semantic influences. Das et al. (2001) and Pang et al. (2002) proposed a method for approximating the effects

of negations, which is to append a _NEG suffix to each word that appeared between a negation word and a punctuation mark, the punctuation mark such as : ^[.:;!?]$. We use the negation word list provided by Pott (2011) as shows in Table 5.1.

Table 5.1 Negation word list (Pott 2011)

never	no	nothing	nowhere	none	none	not
Haven't	hasn't	Hadn't	cannot	couldn't	shouldn't	
Won't	wouldn't	don't	doesn't	didn't	Isn't	Aren't

5.3 Classification tool—LIBSVM

Many research works in sentiment analysis achieved high performance using SVM tools. In our supervised learning, we used an open source SVM tool LIBSVM provided by Chang et al. (2011). LIBSVM is a widely used SVM tool in many areas.

5.4 Classification performance evaluation

Accuracy is one of the assessments, which is denoted as the correct guesses divided by all guesses. The equation is $\mathrm{Acc} = \frac{\sum_i \mathrm{cii}}{\sum_i \sum_j \mathrm{cij}}$. The website www.christopherpotts.net provided an example on the accuracy as shown in table 5.2. In the example, the accuracy of positive is $\frac{15}{15+10+100} = 0.12$, while that of the objective is $\frac{1000}{10+100+1000} = 0.9$. *Accuracy* is almost useless if the categories are highly imbalanced (15 vs. 1000), because one can often guess that the largest category will have the highest accuracy. In our study, we used about equal number of examples in the two class folders. The accuracy-based evaluation is, therefore, still effective in our tasks.

Table 5.2 Example for classifier evaluation

		Predicted		
		Positive	Negative	Objective
Observed	Positive	15	10	100
	Negative	10	15	10
	Objective	10	100	1000

The second performance assessment is *precision*, which is the percentage of items classified as positive that are actually positive. The equation is $\mathrm{Pre} = \frac{\mathrm{cii}}{\sum_{\mathrm{j}} \mathrm{cji}}$, which means the fraction of targets assigned to class i that are actually in class i. Precision is the correct guesses penalized by the number of incorrect guesses. In binary classification, precision is analogous to positive predictive value, which could be denoted as true positive divided by the sum of true positive and false positive.

The third performance assessment is *recall*, which is the fraction of documents that are relevant to the query and those have been successfully retrieved— the percentage of positives that are classified as positive. The equation is $\mathrm{Rec} = \frac{\mathrm{cii}}{\sum_{\mathrm{i}} \mathrm{cij}}$, which means the fraction of targets in class i that are classified correctly. In binary classification, recall is called sensitivity. It is denoted as the true positive divided by the sum of true positive and false negative.

F measure is also a commonly used performance assessment. It is one of the combinations of precision and recall, which is denoted as $2 * \frac{\mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$. The F measure is usually the most important performance evaluation tool in the text mining area.

Chapter 6: RESULTS AND DISCUSSIONS

We used the product review dataset provided by Hu et al. (2004) to conduct the unsupervised sentiment score calculations. The lexicons we used consist of 6800 opinion words tagged by Hu et al. (2004) and 8000 subject clues tagged by Wilson et al. (2005).

6.1 Unsupervised learning – Sentiment score calculation

The unsupervised sentiment score calculation method was implemented in Python using the Natural Language Tool Kit (NLTK). The objective of this experiment was to calculate sentiment scores of product features and evaluate that calculation accuracy. We carried out the experiment using customer reviews of 5 products from www.amazon.com provided by Hu et al. (2004). We used the product features that were provided by Hu et al. (2004) directly in the calculations. Instead of creating opinion word lexicon for the calculation, we used the opinion word lexicons provided by Hu et al. (2004) and Wilson et al. (2005). If the polarities of certain opinion words are unknown, which means that they are not contained in any of the opinion lexicons, then their polarities were inferred by the linguistic rules and constraints that were introduced.

Based on the polarities of the opinion words around a product feature, sentiment scores of product feature were calculated and aggregated by the equation provided by Ding et al. (2008), and the rules and constraints that were discussed.

Table 6.1 lists part of the product features with their calculated sentiment scores from one of the five datasets provided by Hu et al. (2004), which is the "Apex AD2600 Progressive-scan DVD player" dataset. This dataset contains 99 product reviews, and contains about 840 sentences. The number of product features provided by this dataset is

101. 585 product features with their sentiment scores annotated were contained in the final result list. Among the 585 product features, 62 product features were distinct, but the same product features could have different opinion polarities. This means that average number of times a product feature can occur in the dataset was about 10. Table 6.1 shows part of the product features and their polarities.

Table 6.1 Product features with their polarities

remote	-1.0
zoom	-1.0
quality	-1.0
apex	1.0
service	1.0
support	1.0
quality	1.0
player	1.0
apex	-1.0
player	1.0
price	1.0
quality	-1.0
quality	1.0
apex	0
amazon	-1.0
service	-1.0

Note that one product feature could appear more than one time and even with different polarities. This means that the product features received different opinions from different product reviews. For example, the “quality” in the table had two different polarities. Table 6.2 shows part of the averaged polarity of each distinct product features.

Table6.2 Averaged polarities of some product features

features	polarity
ad-1220	1
ad-1600	1
aff	1
amazon	0.5
apex	0.3
button	0.5
case	1
cd	1
color	0
design	0.8
dvd	0.21

We obtained the average polarities of 62 distinct product features. For the other 39 product features, we could not obtain their polarities. The major reason for reduced product feature coverage was that not all product features in the product feature list appeared in the reviews. It can be explained by the fact that some product features were implicit product features and they were not explicitly expressed in the reviews.

Intensification rule and a sentence constraint were added to the rules used in Ding et al. (2008) in this experiment. To investigate which consideration could improve the performance, we validated intensification rule and sentence constraint separately.

The final calculation list of this experiment contained product features with their calculated sentiment score and their corresponding sentences. The calculation performance was validated manually. First, the “true” polarities of product features were annotated manually by reading the corresponding sentences. Table 6.3 shows the calculation performance of the five datasets provided by Hu et al. (2004). The performance of Ding et al. (2008) is also shown in the table.

Table6.3 Comparison of our method with that of Ding et al. (2008)

Dataset	intensification			sentence relation			Ding et al. methods		
	pre	rec	F	pre	rec	F	pre	rec	F
Apex	0.66	0.63	0.64	0.63	0.65	0.64	0.89	0.88	0.89
GanG3	0.53	0.74	0.61	0.64	0.76	0.69	0.93	0.92	0.93
Nikcool	0.61	0.76	0.64	0.64	0.75	0.67	0.96	0.96	0.96
Nomp3	0.58	0.65	0.6	0.576	0.64	0.6	0.87	0.86	0.87
No6610	0.66	0.79	0.72	0.68	0.82	0.74	0.95	0.95	0.95

Compared to Ding et al. (2008), the method we proposed obtained lower performance because of the coverage of the product features in our algorithm. In Apex dataset, 62 product features with annotated polarities were obtained from a total of 101 product features whereas the other 39 product features in the testing data were not predicted. One reason is that some product features that were provided by Hu et al. (2004) were not explicitly expressed in reviews. They were implicit product features within the text. So product features annotated by Hu et al. (2004) might not appear in reviews explicitly. For example, in the sentence "When you put this phone in your pocket, you forget it is just there; it is unbelievably small and light". "Small" is the implicit feature of cell phone *size,* and "light" is the implicit feature of *weight*. In our computation, we only considered explicit product features.

The second reason is that we only considered the current sentence that contained product feature and the next sentence if that sentence did not contain other product features. If the two sentences had no corresponding opinion words or the polarities of opinion words that could not be obtained or inferred from the opinion lexicons, then the product features was not included in final result list. Therefore, the number of opinion words that were

included in the calculation of the sentiment scores of the product features was reduced. Hence, the final performance was affected. The third reason for lower performance is that all product features that appeared in one sentence were assigned polarities by calculation. But sometimes, the sentence was not talking about that product feature. For example, in the sentence "The Nokia 6610 excels as a cell phone, thank god", both "Nokia 6610" and "cell phone" were assigned positive scores by the calculation, but "cell phone" here should be neutral. Therefore, many product features were assigned polarities while they should be neutral. In this way, product features that had true neutral values received low performance. Table 6.4 shows the performance of three different sentiment classes using the Nikon dataset.

Table 6.4 validation results for different classifications

	intensification			sentence relation		
	positive	neutral	negative	positive	neutral	negative
precision	0.91	0.38	0.31	0.88	0.08	0.41
recall	0.82	0.13	0.71	0.77	0.30	0.75
F	0.86	0.19	0.43	0.82	0.13	0.53

The results show that product features with true neutral polarities performed the lowest compared to product features with positive and negative polarities. Therefore, one way we could improve the calculation performance is to find out the exact opinion words that were expressed on product features in one sentence. Many product features and opinion words were contained within one sentence, and product features that were not expressed opinions were also assigned polarities from the opinion words that did not modify the product features. The other way to improve the calculation performance is to develop an

algorithm to consider implicit product features, which were not expressed explicitly in the product reviews.

Because of the above reasons, the unsupervised calculation algorithm proposed in this thesis did not produce results that were comparable to those of Ding et al. (2008). To show that the usage of intensification rule and sentence constraints could improve calculation performance, we compared results that used intensification rule and sentences constraints to the result that used neither of them. Table 6.5 shows the comparison results.

Table 6.5 Evaluation of used rules and constraints

	Neither	Intensification	Sentence
precision	0.61	0.67	0.63
recall	0.62	0.63	0.65
F measure	0.61	0.64	0.64

"Neither" in table 6.5 means that the calculation algorithm used neither intensification rule nor sentence constraints. "Intensification" means that intensification rule was added to the algorithm, and "sentence" means that sentence constraints was added to the algorithm. As shown in table 6.5, calculation performance was improved by using intensification rule and sentence constraint.

6.2 Supervised learning

6.2.1 First result: evaluation of each part of features

Three possible feature sets were included in document-level and sentence-level feature spaces: (i) product features that were extracted from corpus; (ii) adjectives; and (iii) features that were developed based on the linguistic rules and constraints (referred to rule features in the following discussion). The rule features were developed based on the work

of Rillof (2005) and Wilson et al. (2005). The objective of this section is to compare the efficiency of the three sets of features using the Apex DVD player dataset provided by Hu et al. (2004). We used the open source SVM tool LIBSVM in this task.

For the second set of adjective features, not all the adjectives were considered. We applied a threshold value and filtered out adjectives with low occurrences, since rare words may have little contributions to the classification. Forman (2003) pointed out that usually half of the total number of distinct words may occur only one time, so eliminating words with a given low rate of occurrence can yield great savings. To filter out the adjectives that occur rarely, we only considered the adjectives that had occurrence rate that were in the top 30 percent. Pang et al. (2002) considered the frequency that is above four. This means that only 30 percent of adjectives were included in the feature spaces. Table 6.5 shows the performance for different feature combinations.

Table 6.6 Comparison between features

	Features	Accuracy
Document-level	product features	74.4
	adjectives	71.8
	rule features	89.74
	prftr+adj+rule	82.05
Sentence-level	product features	69.43
	adjectives	67.4
	rule features	80.05
	prftr+adj+rule	81.35

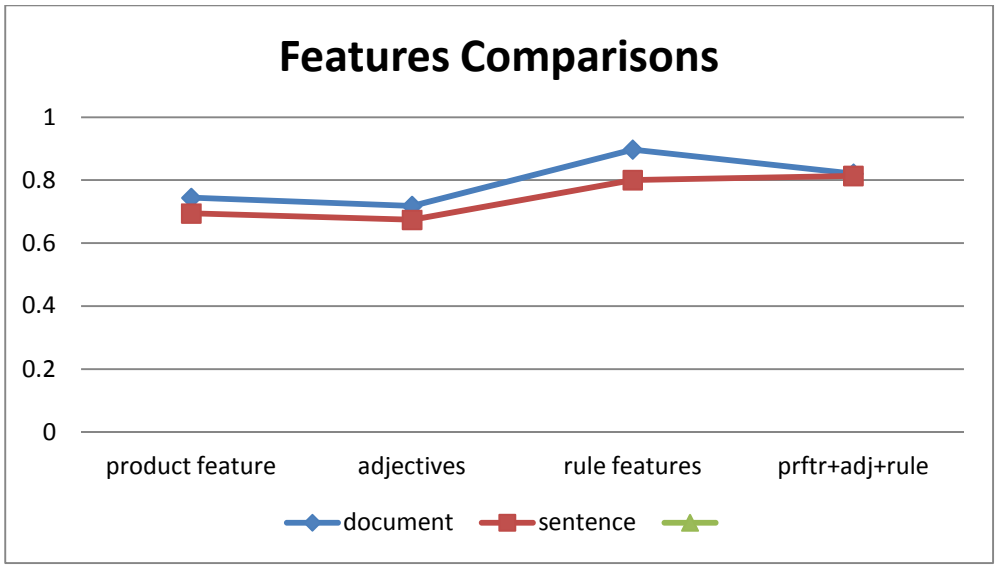

Figure 6.1 Performance of features in document-level and sentence-level analyses

Table 6.6 and Figure 6.1 show that the document-level accuracy (74.4%) is higher than the sentence-level accuracy (69.43%) if the feature space contains only product feature set. Feature spaces were used. It is not surprising that a document-level analysis has higher performance than sentence-level analysis because one document can contain more product features than one sentence. Many sentences even do not contain product features, which mean that coverage of product features in a document is much higher than that of a sentence. In other word, one document contains more information than a sentence to indicate their classification types (positive or negative). The result also proved our assumption that product features could be good indicators in determining the classification types of product reviews.

Results also show that rule based features performed very well in both document-level analysis (89.74%) and sentence-level analysis (80.05%). The results suggest that the usage of linguistic rules and constraints can improve the classification performance. The combination of all three sets of features in both document-level (82.05%) and sentence-

level (81.35%) analyses perform better than product features (document-level: 74.4%; sentence-level: 69.4%) and adjectives (document-level: 71.8%; sentence-level: 67.4%) separately. The improved performance of the combinations should have been benefited from the usage of rule based features, which means that the rule based features made the most contributions to the overall performance of the feature spaces. Therefore, the results support the hypothesis that the usage of linguistic rules and constraints can increase the performance of the classifications in sentiment analyses. Hence, the future research can consider to develop rule based features and to reduce the vector space most.

Moreover, the results show that document-level analyses perform better than the sentence-level analyses because of the sparseness of the sentence-level feature spaces, i.e., many features in sentence vectors have feature values zero. Hence, we expected an improved performance after applying feature selection methods and reducing dimensions of feature spaces, which can delete features with too many zeros as feature values and with little contributions to the performance.

6.2.2 Second result: compare feature selection methods

To further reduce dimensions of feature spaces, we need to rank each feature according to its contribution to the classification performance, and then take the best k features. Yang (1997) and Forman (2003) conducted comprehensive studies on feature selection methods (i.e., ranking the features) in text mining problems. In order to investigate whether the commonly used feature selection methods can improve the performance of sentiment analysis and whether our proposed features can get different performance under the usage of different feature selection methods, we conducted experiments using two feature selection methods, Mutual Information (MI) and Information Gain (IG). The

purpose is to study whether the usage of feature selection methods can lead to an improved performance.

In the experiment, we used product reviews (Hu et al., 2004 & Ding et al., 2008) of Apex AD2600 Progressive-scan DVD player to evaluate MI and IG. Based on the results of the previous experiments, the product features and rule features have the abilities to determine the classification of a sentence or a document, so it is meaningless to reduce any of these features from feature spaces. Hence, we only applied the two feature selection methods to the adjectives. Part of the results of the feature selection methods are shown in Table 6.7.

Table 6.7 Weighting values of part of features

	f65	f66	f67	f68	f69	f70	f71
Sen-MI	0.05403	0.038361	0.024747	0.028605	0.007595	0.00893	0.010633
Sen-IG	0.374213	0.677399	0.677399	0.741937	0.271934	0.04879	0.191891
doc-MI	-0.00969	-0.01089	0.009729	-0.02719	-0.03982	-0.0362	0.049198
doc-IG	0.182322	0.356675	0.405465	0.287682	0	0.133531	0.693147

Sen- means the sentence-level, and doc- means the document-level. The f in columns represents the features, and the numbers means the nth feature in the feature space.

Based on the ranking of each feature selection results, we retained $\frac{1}{3}$, $\frac{2}{3}$ and $\frac{3}{3}$ features in three experiments, and combined them with the product features and the rule based features. We obtained the performance of each combination by applying LIBSVM to the feature spaces. Figure 6.2 and 6.3 show the changing of cross validation accuracy for each of the feature selection methods as the number of features decreases in the document-level analysis and sentence-level analysis.

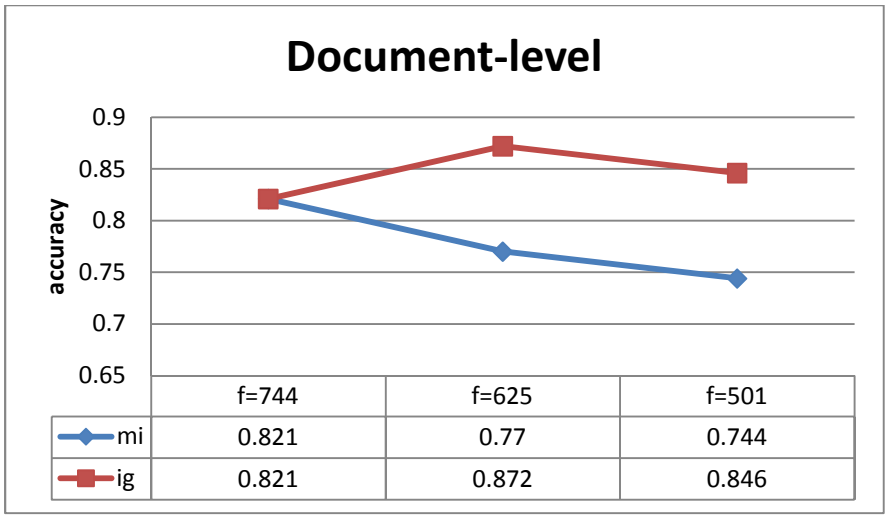

Figure 6.2 Document-level feature selection based on two methods

Figure 6.3 Sentence-level feature selection based on two methods

For document-level feature selection processes, the deletion of about 120 adjective features (at the point f=628, which is the number of combined features) based on the Information Gain (IG) improved the performance of the combined features from 0.821 to 0.872. For Mutual Information (MI), the deletion decreases the performance from 0.821 to 0.77. Further, the deletion of 240 adjectives (at the point f=501) for both of the

selection methods resulted in lower performance than when 120 adjectives were deleted. In our case, the optimal number of deleted adjective features when using IG was between 120 and 240. Yang et al. (1997) pointed out that IG can eliminate up to 90% or more of the unique terms (features) with either an improvement or no loss in classification accuracy. However, when using MI, the deletion of adjectives decreased the performance all the way.

For sentence-level comparisons, Figure 6.3 illustrates that the deletion of adjectives based on MI have minimal effects on the classification performance than based on IG. The performance of MI based deletion was about the same (0.814, 0.808 and 0.808 separately). This means that the adjectives deleted based on MI at the two points (f=628 and f=504) had little effects on the overall performance. For document-level analyses, IG based features have higher performance than MI based features. This is not surprising because of the properties of MI and IG. As discussed in Chapter 4, the common terms can result in higher IG scores, while rare terms could produce higher MI scores. In our prior processing of the data, we only selected the top 30% most frequently occurred adjectives as features in the feature spaces. So, most of the features in feature spaces are common features. In other words, IG based feature selections could have higher performance than MI based feature selections.

The weakness of IG is that, when using IG ratio, the classifiers are biased for attributes with a large number of distinct values. Attributes that have many distinct values could receive the most information gain and are likely to be selected as the relevant attribute to predict the classification. This may present a problem that because distinct values are not able to predict other values. For example, credit card number attribute has high

information gain value since each card number uniquely identify a customer, but we cannot determine it also has a problem that we cannot determine other customer's attributes based on the credit card number. This, fortunately, is not an issue in our study because none of the attributes in our input matrix has this property.

A weakness of Mutual Information (MI) is that the score is strongly influenced by the marginal probabilities of terms. For terms with an equal conditional probability($P(t|c)$), rare terms can have higher score than common terms (Yang, 1997). The scores, therefore, are not comparable across terms of widely differing frequencies.

6.2.3 Third results: Comparison of our feature spaces with Pang et al. (2002)

In many kinds of analyses, especially in empirical analyses, the selection of variables is a critical task as the variables have major impacts accuracy and effectiveness of the final results. Similarly, the selection of features for machine learning tasks is important. The objective this thesis is to identify the sentiment polarities of product features and to investigate whether product features can be effective features for sentiment analyses using machine learning tools. The principle of machine learning is to determine/ predict the unknown data by learning the "behavior" using training data. As such, the reason we propose the usage of product features as features in a feature spaces is that the presence of product features in one document or sentences can or may be a good indicator to determine the classification of unknown documents or sentences. For example, in the training data, if most of the documents that contain "button" and "Apex player" are labeled negative, then the co-occurrence of these two words may help to classify documents that have not been labeled. The feature value of each product feature we

assigned to document vectors is the distance between the product feature and its corresponding opinion words within two consecutive sentences as discussed in Chapter 4.

The second set of the feature space is adjectives. The reason for including adjectives as features is that opinion words are usually adjectives and they have direct relationships with product features. Many of prior research works in the area such as Pang et al. (2002) used adjectives as features in machine learning based sentiment analyses. The values assigned to these features are their frequency count in one document.

The third set of the feature space is proposed based on linguistic rules and constraints as discussed in the Chapter 3. The methods proposed by Wilson et al. (2005) can identify the contextual polarities of phrases based on the rule based features developed. We believe that developing the rule features for sentence-level and document-level analyses can also enhance the performance. To evaluate the performance of the proposed feature spaces, a comparison experiment was conducted, which to compare proposed feature spaces to those of Pang et al. (2002) using movie review dataset they provided. They conducted document-level analyses using Maximum Entropy, Naïve Bayes, and SVM, and tried several kinds of tokens or token combinations such as n-grams, adjectives, POS-tagged word tokens as features. In our unsupervised learning, we directly used the product features that were provided by Hu et al. (2004). In this supervised learning, we needed to extract product features by ourselves. We followed the methods proposed by Hu et al. (2004) to extract nouns or noun phrases that occurred frequently from the dataset and conducted pruning to delete impossible nouns or noun phrases. We extracted top 10% most frequently occurred nouns and noun phrases, and then conduct the "pruning" manually to delete nouns or noun phrases that were not product features. The

adjectives we retained were also the top 10% of the most frequently occurred adjectives. We used 100 positive documents and 100 negative documents from the dataset provided by them. After pruning manually, the number of product features retained is 377, and the number of adjectives retained is 362. After adding the proposed rule features based on the rules and constraints, the total number of features is 744. Table 6.8 shows the cross validation accuracy of our approach and Pang et al. (2002)'s method.

Table 6.8 Comparison among features

Features	# of features	Feature values	Cross-valid Acc	Predict Acc
Unigrams(Pang)	16165	presence	82.9	--
Bigrams(Pang)	16165	presence	77.1	--
Uni+POS(Pang)	16695	presence	81.9	--
Adjectives(Pang)	2633	presence	75.1	--
prftr+adj+rule(our)	744	dstns+fre+fre	76.2	57.6
Product feature(our)	377	distance	62.8	57.6
Adjective (our)	362	frequency	74.7	55.8
Rule features (our)	5	frequency	78.1	53.9

When all three feature sets in feature spaces (product features + adjective features + rule features) were used, we found that cross validation accuracy was 76.2%, which was lower than unigrams (82.9%), bigrams (77.1%) and unigrams with POS tags (81.9%) but a little higher than adjectives (75.1%) used in Pang et al. (2002). However, the feature spaces with only adjective features in our proposed feature space performed almost the same with adjectives used in the Pang's analyses (74.7% vs. 75.1%). Unigrams, bigrams or n-grams selection were based on the assumption that the occurrence of a word or token could affect the occurrence of the next word or token. Thus, tokens need to be extracted

one by one consecutively. This kind of features can also be used in many other Natural Language Processing (NLP) classification tasks. This means that Bag-of-Word (BoW) features such as unigrams are not specifically prepared for sentiment analysis, and they can be used to other NLP classifications either. The disadvantage of these kind of features is that they can lead to large document vector size or sentence vector size, because vector sizes are determined by vocabulary size of corpus when they include BoW features. The bigger the corpus is, the larger the number of vocabularies, and the larger the vector size.

Hence, the appropriate feature selection techniques according to specific problems can reduce the size of feature spaces, and improve the time and space efficiency. If we could select the features based on specific problems/ context, this would improve classification performance. For example, for sentiment analysis problems, the time efficiency and space efficiency or even the performance could be improved by analyzing the linguistic rules and constraints that are related to sentiment classification. Zhai et al. (2011) received a better result and efficiency when they added two constraints. Unigrams, bigrams or POS tags have a long history of usages in NLP classification tasks. Although sentiment analysis is a kind of NLP task and the usage of these features in sentiment analysis tasks could get a decent result, it is still too time and space consuming, and not efficient. The system has to spend a lot of time and memory space to store the less useful information for the classifications. Although the performance of proposed feature spaces in this research was not as good as expected, selection of features with a specific purpose to product feature based sentiment analysis and with more information show potentials.

Further, the proposed feature spaces in this research could result in smaller sized feature spaces, and be more time and space efficient.

We found that the product features with Pang's dataset (62.8%) did not perform as well as those with Hu et al. (2004)'s dataset (68.4%) in first supervised learning task. One possible reason is the differences in the dataset. Another possible reason is the accuracy of our product feature extraction process. We used the product features provided by Hu et al. (2004) directly in the first supervised learning experiment, while we extracted the product features from Pang et al. (2008)'s dataset ourselves by following Hu et al. (2004)'s extraction methods, and we conducted the pruning manually instead of using the pruning methods they provided. In this way, we may have retained some nouns that are not product features, and deleted some product features that should be retained. One of our future works is to improve our product feature selection methods.

Chapter 7: CONTRIBUTIONS AND CONCLUSIONS

In this study, we conducted supervised learning and unsupervised learning for sentiment analysis on product features.

In the unsupervised learning, the objective was to calculate the sentiment score of product features by aggregating opinion polarities of opinion words around the product features. The approach incorporated different linguistic rules and constraints. We followed the sentiment score calculation equation provided by Ding et al. (2008), and added the intensification rule and sentence constraints to the rules used in Ding et al. (2008). The method we proposed did not performed as well as that of Ding et al. (2008). There reasons could explain the low performance. The first one is that we did not considered implicit features, so not all the product features provided by Hu et al. (2004) were in the final result list. We only considered explicit product features that appeared in the datasets. The second reason may reside in the coverage of opinion words was not enough. The third reason was that many product features that were not commented on were assigned sentiment scores.

In the supervised learning, our major focus was on how to derive specific and appropriate feature spaces for sentiment analysis. We considered several techniques that could improve the classification performance. In Natural Language Processing (NLP) classification tasks, many Bag-of-Word (BoW) features, such as n-grams, are used as features. In this way, size of a feature space is big because it depends on the vocabulary size of the corpus. Therefore, choosing the features that are directly related to sentiment analysis is important, because it can improve performance and time and space efficiency.

To construct feature spaces that are specific to sentiment analysis, and to achieve high performance, several aspects were considered:

i. Choose appropriate features for sentiment analysis on product features.

Product features were treated as features in the proposed feature spaces. The product reviews are about the products, so the product features should be good indicators in determining the class types (positive or negative) of documents or sentences. The second feature set in the feature spaces was composed of adjectives, which are generally considered to be related to opinion words. The third feature set in the feature spaces is composed of rule based features, which were proposed based on linguistic rules. Linguistic rules are widely used in sentiment analyses to improve the classification performance.

The result shows that product features in document-level analysis performed better than in sentence-level analysis. The rule based features in sentence-level analysis performed better than in document-level analysis. The rule based features can improve the overall performance of the feature spaces.

ii. Choose appropriate feature selection methods.

MI and IG are two different feature selection methods. We applied two feature selection methods to the proposed feature spaces. The results show that IG performed better than MI in document-level analysis while MI performed better than IG in sentence-level analysis.

For evaluation, we compared the proposed feature spaces to those of Pang et al. (2002). Pang et al. (2002) analyzed several Bag-of-Word features using machine learning models.

So the comparison could evaluate whether the proposed feature spaces with specific purposes for sentiment analysis could perform better. The major advantage of proposed feature spaces in this research is to improve time and space efficiency of classification models. The feature spaces we proposed did not perform as well as those used in Pang et al. (2002). One possible reason is the difference in dataset size. We used 100 examples in each class while they used 1000 examples. We will use larger dataset in our future research. The second possible reason for lower performance is the coverage of the opinion words in the lexicon is low. Feature values assigned to product features were distances between product features and opinion words covered within two consecutive sentences. If two consecutive sentences that satisfied the restrictions of sentence constraints proposed in Chapter 4, but no opinion word that was annotated in the lexicon appeared in these two sentences, and even worse, the sentences contain opinion words with unknown polarities and the polarities cannot be inferred, then feature values assigned to product features are not correct.

Contributions to research:

i. This is a pioneering research that incorporates product features, linguistic rules and constraints as features in supervised machine learning. The combination of frequent adjectives with product features and rule based features could improve classification performance. Specially, the usage of rule based features could achieve high classification performance.

ii. For feature selection method, Information Gain (IG) appears to be performing better in document-level analyses than in sentence-level analyses. On the other hand, Mutual

Information (MI) appears to be performing better in sentence-level analyses than in document-level analyses. This finding needs to be further investigated and substantiated.

iii. Based on the approach, researchers can gather the final product feature list and the corresponding opinion polarities of the features of a product. The result can then be used to analyze and study (and may be predict) the performance of the product and the impact on company's performance.

Contributions to practice:

i. The sentiment analysis on product features is useful for customers and shoppers. A better and more accurate sentiment analysis can help buyers make better decisions and select the right products to fit their needs.

ii. The sentiment analysis on product features is helpful to companies and organizations. They can use the analysis to enhance their products and better their offerings.

iii. Bag-of-Word based document classification cannot give the information miners specific knowledge about the products or product features. Most users are interested in discovering these specific features of a product and the polarity of the product features. Our approach provides the specific polarity knowledge that can help the users make the right decisions.

iv. The approach has many potential applications in various industries and domains. Its application is not restricted to studying the sentiments of different product features. The approach can also be used to study the sentiments of different features of an academic program or a school or an institution. It is also possible to apply the approach to study

the factors impacting the popularity of politicians and such analysis may be helpful in predicting the outcomes of elections.

REFERENCES

Abbasi, A., Chen, H., Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums, *ACM Transactions on Information Systems (TOIS), Volume 26 Issue 3.*

Blache, P., Balfourier, J.-M. (2001). Property Grammars: a Flexible Constraint-Based Approach to Parsing, *in proceedings of IWPT.*

Blei,D., Ng,A., and M. Jordan (2003). Latent dirichlet allocation, *Journal of Machine Learning Research, 3(5):993–1022.*

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM : a library for support vector machines, *ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27.*

Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns, *Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005).*

Collins, M. (1997). Three generative, lexicalized models for statistical parsing, *In Proceedings of the 35th Annual Meeting of the ACL.*

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning, *In Proceedings of the 25th International Conference on Machine Learning (ICML-08), pp. 160–167.*

Daelemans,W., et al. (2003). Combined optimization of feature selection and algorithm parameter interaction in machine learning of language, *In Proceedings of the 14th European Conference on Machine Learning (ECML-2003), pp. 84–95.*

Das, S. and Chen, M. (2001). Yahoo! for Amazon: Extracting market sentiment from stock message boards, *In Proceedings of the Asia Pacific Finance Association Annual Conference (APFA), 2001.*

Dave,K., Lawrence, S., and Pennock, D.M.(2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews, *In Proceedings of WWW, pages 519–528.*

Ding, X., Liu, B. and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining, *In Proceedings of the Conference on Web Search and Web Data Mining (WSDM).*

Duric, A., Song,F. (2011). Feature selection for sentiment analysis Based on Content and Syntax Models, *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT 2011, pages 96–103, 24 June, 2011, Portland, Oregon, USA.*

Eguchi, K. and Lavrenko, V. (2006). Sentiment retrieval using generative models, *In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 345–354, Sydney, Australia, July 2006, Association for Computational Linguistics.*

Fan, R.E., Chen, P.-H., and Lin, C.J. (2005). Working set selection using second order information for training SVM, *Journal of Machine Learning Research 6, 1889-1918.*

Forman,G. (2003). An extensive empirical study of feature selection metrics for text classification*, The Journal of Machine Learning Research.*

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis, *Proceedings of the 20th international conference on Computational Linguistics, p.841-es, August 23-27, 2004, Geneva, Switzerland.*

Godbole, N., Srinivasaiah, M. and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs*, In CWSM '07.*

Guo, H., Zhu, H., Guo, Z., Zhang, X. and Su, Z. (2009). Product feature categorization with multilevel latent semantic association, *Proc. of CIKM.*

Harris, Z. (1954). Distributional structure. *Word, 10 (23), 146-162.*

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives, *In Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL, pages 174-181, Madrid, Spain, July. Association for Computational Linguistics.*

Hatzivassiloglou, V. and Wiebe, j. (2000). Effects of adjective orientation and gradability on sentence subjectivity, *In Proc. of COLING.*

Hu, M. and Liu, B. (2004). Mining Opinion Features in Customer Reviews, *To appear in AAAI'04.*

Ikeda, D., Takamura, H., Ratinov, L. and Okumura, M. (2008). Learning to Shift the Polarity of Words for Sentiment Classification, *In Proceedings of the 3rd International Joint Conference on Natural Language Processing, pages 296–303.*

Jones,K., Walker,S., and Robertson, S.(2000). A probabilistic model of information retrieval: development and comparative experiments, *Inf. Process. Manage., 36(6):779–808.*

Kamps, Jaap, Marx, Maarten, Mokken, Robert J., and de Rijke, Maarten (2004). Using WordNet to measure semantic orientation of adjectives, *In Proceedings of 4th International Conference on Language Resources and Evaluation (LREC 2004), 1115-1118. Lisbon, Portugal.*

Kim, S. and Hovy, E. (2004). Determining the Sentiment of Opinions, *COLING'04.*

Lafferty, J., McCallum, A., Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Proc. 18th International Conf. on Machine Learning. Morgan Kaufmann. pp. 282–289.*

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis, *In CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management, pages 375–384, New York, NY, USA. ACM.*

Lin, D. (1998). Dependency-based evaluation of MINIPAR, *In Workshop on Evaluation of Parsing Systems at ICLRE.*

Liu, B. (2010).Sentiment Anlaysis and Subjectivity, *Invited Chapter for the Handbook of Natural Language Processing, Second Edition.*

Maas, A. L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y. and Potts, C. (2011). Learning Word Vectors for Sentiment Analysis, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 142–150, Portland, Oregon.*

Martineau, J. and Finin, T.(2009). Delta TFIDF: An Improved Feature Space for Sentiment Analysis, *In Proceedings of the Third AAAI Internatonal Conference onWeblogs and SocialMedia, San Jose, CA.*

Mei,Q., Ling,X., Wondra, M., Su, H. and Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs, *In WWW '07: Proceedings of the 16th international conference on World Wide Web, pages 171–180, New York, NY, USA.*

Mejova, C., Srinivasan, P. (2011). Exploring Feature Definition and Selection for Sentiment Classifiers, *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media.*

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources, *In Dekang Lin and Dekai Wu, editors, Proceedings of EMNLP-2004, pages 412–418, Barcelona, Spain, July 2004. Association for Computational Linguistics.*

Nakagawa, T., Inui, K. and Kurohashi, S. (2010). Dependency tree-based sentiment classification using CRFs with hidden variables, *In NAACL, HLT.*

Ng, V., Dasgupta, S. M., and Arifin, N. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews, *Proceedings of the COLING/ACL on Main Conference poster sessions, p.611-618, Sydney, Australia*

Osgood, C.E., Suci, G., & Tannenbaum, P. (1957), The measurement of meaning. *Urbana, IL: University of Illinois Press.*

Paltoglou, G., Thelwall, M.(2010). A study of Information Retrieval weighting schemes for sentiment analysis, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1386–1395, Uppsala, Sweden. 2010 Association for Computational Linguistics.*

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques, *In Proc. of EMNLP.*

Pang, B. and Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, *In Proceedings of the ACL.*

Popescu, A. and Etzioni, O. (2005). Extracting product features and opinions from reviews, *In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP).*
Potts, C. (2007) The Expressive Dimension. Theoretical Linguistics 33:165-198.

Qu, L., Ifrim, G., Weikum, G. (2010). The bag-of-opinions method for review rating prediction from sparse text patterns, *Proceeding in COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics.*

Quirk, R., Greenbaum, Sidney, Leech, Geoffrey, and Svartvik (1985). A Comprehensive Grammar of the English Language, *London: Longman.*

Riloff, E (1996). Automatically Generating Extraction Patterns from Untagged Text, *In: Proceedings of the Thirteenth National Conference on Artificial Intelligence, The AAAI Press/MIT Press 1044–1049.*

Riloff, E., Wiebe, J. and Wilson, T. (2003). Learning subjective nouns using extraction pattern bootstrapping*, In Conf. on Natural Language Learning (CoNLL), pages 25–32.*

Riloff, E. and Patwardhan, S. and Wiebe, J. (2006). Feature Subsumption for Opinion Analysis, *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 440–448, Sydney. Association for Computational Linguistics.*

Rogati, M., and Yang, Y. (2002). High-performing feature selection for text classification, *Proceedings of the eleventh international conference on Information and knowledge management, November 04-09, 2002, McLean, Virginia, USA.*

Sang, E. F. T. K.(2002). Memory-based shallow parsing. Journal of Machine Learning Research, *2:559-594*

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields, *In Proceedings of HLT-NAACL.*

Socher, R., Pennington, J., Huang, E., Ng, A. and Manning, C.D. (2011). Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions, *Conference on Empirical Methods in Natural Language Processing (EMNLP 2011).*

Tan, S. X., Cheng, Y., Wang and Xu, H. (2009). Adapting Naïve Bayes to Domain Adaptation for Sentiment Analysis, *ECIR 2009.*

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text, *Journal of the American Society for Information Science and Technology 61(12):2544–2558.*

Titov,I. and McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization*, In Proceedings of ACL-08: HLT, pages 308–316, Columbus, Ohio. Association for Computational Linguistics.*

Toutanova,K., Klein, D., Manning, C. and Singer, Y. (2003). Feature-Rich Part-of-Speech tagging with a Cyclic Dependency Network, *In Proceedings of HLT-NAACL 2003, pp. 252-259.*

Turney, p. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*, In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania.*

Turney, P., and Pantel, P.(2010). From Frequency to Meaning:Vector Space Models of Semantics*, Journal of Articial Intelligence Research* 37 (2010) 141-188.

Whitelaw,C., Garg,N., and Argamon, s.(2005). Using appraisal groups for sentiment analysis, *In CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management, pages 625–631, NY, USA. ACM.*

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2005). Learning subjective language, *Computational Linguistics, 30(3):277–308.*

Wilson, T., Wiebe, J. and Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, *In Proceedings of the 2005 Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 347–354.*

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization, *In International Conference on Machine Learning, pages 412–420.*

Zhai, Z., Liu, B., Xu, H. and Jia, P. (2010). Grouping Product Features Using Semi-Supervised Learning with Soft-Constraints, *To appear in Proceedings of the 23rd International Conference on Computational Linguistics (COLING-2010),Beijing, China.*

Zhai, Z., Liu, B., Xu, H. and Jia, P. (2011). Clustering Product Features for Opinion Mining, *Proceedings of Fourth ACM International Conference on Web Search and Data Mining (WSDM-2011), Feb. 9-12, 2011, Hong Kong, China.*