Graduate Theses, Dissertations, and Problem Reports

2010

Proximity-based sentiment analysis

S. M. Shamimul Hasan
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Hasan, S. M. Shamimul, "Proximity-based sentiment analysis" (2010). *Graduate Theses, Dissertations, and Problem Reports*. 4603.
<https://researchrepository.wvu.edu/etd/4603>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

PROXIMITY-BASED SENTIMENT ANALYSIS

by

S.M.Shamimul Hasan

**Thesis submitted to the College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of**

**Master of Science
in
Computer Science**

Approved by

**Dr. Donald A. Adjeroh, Committee Chairperson
Dr. Arun Abraham Ross
Dr. James D. Mooney**

Lane Department of Computer Science and Electrical Engineering

**Morgantown, West Virginia
2010**

**Keywords: Sentiment Analysis, Text Mining, Proximity Pattern, Machine Learning,
Natural Language Processing.**

Copyright 2010 S.M.Shamimul Hasan

Abstract

PROXIMITY-BASED SENTIMENT ANALYSIS

by S.M.Shamimul Hasan

Sentiment analysis is an emerging field, concerned with the detection of human emotions from textual data. Sentiment analysis seeks to characterize opinionated or evaluative aspects of natural language text thus helping people to discover valuable information from large amount of unstructured data. Sentiment analysis can be used for grouping search engine results, analyzing news content, reviews for books, movie, sports, blogs, web forums, etc. Sentiment (i.e., bad or good opinion) described in texts has been studied widely, and at three different levels: word, sentence, and document level. Several methods have been proposed for sentiment analysis, mostly based on common machine learning techniques such as Support Vector Machine (SVM), Naive Bayes (NB), Maximum Entropy (ME).

In this thesis we explore a new methodology for sentiment analysis called proximity-based sentiment analysis. We take a different approach, by considering a new set of features based on word proximities in a written text. We focused on three different word proximity based features, namely, proximity distribution, mutual information between proximity types and proximity patterns. We applied this approach to the analysis of movie reviews domain. We perform empirical research to demonstrate the performance of the proposed approach. The experimental results show that proximity-based sentiment analysis is able to extract sentiments from a specific domain, with performance comparable to the state-of-the-art. To the best of our knowledge, this is the first attempt at focusing on proximity based features as the primary features in sentiment analysis.

DEDICATION

First of all I want to dedicate this thesis to Almighty “Allah” for giving me opportunity to prove myself in this world. Then I want to dedicate this thesis to my lovely and caring parents- “Hasina Sardar” and “Shahjahan Sardar”.

ACKNOWLEDGMENTS

I would like to mention my gratitude and to express my acknowledgements to all the people who have supported me. I would like to put this in words without missing a chance to mention how fortunate I am to have all these people around. I would like to thank all my professors, friends and family members without whom this difficult task wouldn't have been completed.

My deepest gratitude and appreciation goes to Dr. Donald A. Adjero for his guidance, patience, support and encouragement throughout my study at West Virginia University, which led to this thesis. I would like to thank my committee members, Dr. Arun Abraham Ross and Dr. James D. Mooney for reviewing my thesis and providing knowledgeable comments and suggestions.

I especially thank the Lane Department of Computer Science and Electrical Engineering at West Virginia University for providing me the opportunity to study here. My thanks to all faculty members from whom I have taken courses. Their interesting classes have inspired me and kept me loving computer science.

I would like to thank my parents and my brothers for their undying prayers, love, encouragement and moral support. 'Thank you', Mom and Dad for standing behind me and encouraging me always to take a step forward, you are the greatest people in the world. Last but not least, I want to thank all my friends and colleagues both in Bangladesh and in West Virginia University who stayed by me throughout this period of time constantly encouraging me to work hard and at the same time who made my stay and work at the West Virginia University a very pleasurable one.

TABLE OF CONTENTS

Abstract.....	ii
Dedication.....	iii
Acknowledgements.....	iv
Table of contents.....	v
List of Figures.....	vi
List of Tables.....	vii
Chapter 1: Introduction.....	1
1.1 Introduction.....	1
1.2 Problem and Motivation.....	2
1.3 Prior Work.....	3
1.4 General Approach and Thesis Contribution.....	3
1.5 Thesis Organization.....	4
Chapter 2: Background and Literature Review.....	5
Chapter 3: Proximity-Based Analysis.....	10
3.1 Data Cleaning.....	11
3.2 Measuring Distances.....	11
3.2.1 Polarity Words.....	12
3.2.2 Polarity & Non-Polarity Words.....	13
3.3 Proximity Models.....	14
3.4 Proximity-Based Features.....	15
Chapter 4: Sentiment Classification Techniques.....	19
4.1 Unsupervised Approach.....	19
4.1.1 Feature Weighting.....	20
4.2 Mean & Median Approach.....	20
4.3 Machine Learning Approach.....	21
4.3.1 Naïve Bayes (NB) classification.....	21
4.3.2 Support Vector Machines (SVM).....	22
4.3.3 K-Nearest Neighbours (K-NN).....	23
4.3.4 J-48 Algorithm.....	23
4.3.5 Multilayer Perceptron (MLP).....	24
Chapter 5: Results & Discussion.....	25
5.1 Dataset.....	25
5.2 Packages Used.....	25
5.3 Results.....	25
5.3.1 Polarity Word Count.....	26
5.3.2 Unsupervised Approach.....	26
5.3.3 Mean-Median Approach.....	29
5.3.4 Machine Learning Approach.....	33
5.3.5 Different Dataset.....	40
Chapter 6: Conclusion & Future Work.....	41
References.....	42

LIST OF FIGURES

Figure 1: Assumption on human writing style.....	10
Figure 2: General schematic diagram of the approach.....	11
Figure 3: Segmentation of Input Text.....	12
Figure 4: Two Proximity Models.....	15
Figure 5 : Illustration of an SVM classifier.....	22
Figure 6: Three layer multilayer perceptron.....	24
Figure 7: Correctly classification of positive review.....	27
Figure 8: Correctly classification of negative review.....	27
Figure 9: An undefined situation.....	27
Figure 10: Impact of feature weighting on the overall performance on classification accuracy.	28
Figure 11: Mean of the training set.....	29
Figure 12: Median of the training set.....	29
Figure 13: Classification Accuracy mean (Positive).....	31
Figure 14: Classification Accuracy mean (Negative).....	31
Figure 15: Classification Accuracy median (Positive).....	32
Figure 16: Classification Accuracy median (Negative).....	32
Figure 17: Mutual Information between Proximity Types.....	36

LIST OF TABLES

Table 1: A confusion table.....	6
Table 2: Existing work on sentiment analysis.....	8
Table 3: Distance Notation with polarity words.....	13
Table 4: Distance Notation with polarity and non-polarity words.....	14
Table 5: Polarity Pattern Example.....	18
Table 6: Polarity and Non-polarity Pattern Example.....	18
Table 7: Performance of polarity word count.....	26
Table 8: Performance of Unsupervised Approach (both positive and negative review).....	26
Table 9: Classification accuracy by considering mean of the training set (Scheme 1).....	30
Table 10: Classification accuracy by considering median of the training set (Scheme 1).....	30
Table 11: Overall classification accuracy by considering mean (Scheme 2).....	32
Table 12: Overall classification accuracy by considering median (Scheme 2).....	32
Table 13: Overall classification accuracy by considering mean (Scheme 3).....	33
Table 14: Proximity Distribution 1 Bin.....	34
Table 15: Proximity Distribution 9 Bins.....	34
Table 16: Proximity Distribution 10 Bins.....	35
Table 17: Proximity Distribution 100 Bins.....	35
Table 18: Best Performance.....	35
Table 19: Mutual Information between Proximity Types.....	36
Table 20: Proximity Patterns (N=2).....	37
Table 21: Proximity Patterns (N=3).....	37
Table 22: Proximity Patterns (N=5).....	37
Table 23: Best accuracy rate of the above three.....	38
Table 24: Best 3 fusion result after combining 3 algorithms.....	38
Table 25: Best 3 fusion result after combining 5 algorithms.....	38
Table 26: Comparative performance with state-of-the-art algorithms.....	39
Table 27: Drug Review.....	40
Table 28: Music Review.....	40

Chapter 1: Introduction

1.1 Introduction

Enormous volume of textual data is available in online. The amount of data accumulated each day by various business, scientific, and governmental organizations around the world is daunting. It becomes impossible for human analysts to cope with such vast amounts of data. Human analysts with no special tools can no longer make sense of enormous volumes of data that require processing in order to make decisions.

For textual data analysis statistical Natural Language Processing (NLP) and machine learning based NLP have gained great popularity since the late 1980s. This trend is caused by the rapid growth of the Internet and the progress made in computer technology including the huge increase in CPU speed and the improved capacity-cost of storage devices. These developments have allowed us to obtain a large number of natural language corpora, and to apply complex but powerful machine learning methods to large scale documents. This recent success in the development of statistical NLP has led to improvements in fundamental text analysis; such as part-of-speech (POS) tagging, phrase chunking, dependency analysis and parsing. Using these components as fundamental building blocks, many NLP researchers have become interested in analyzing text “semantically” or “contextually”. For example, named entity tagging, semantic role tagging and discourse parsing are being investigated in the NLP fields. This move towards taking contextual or semantic information into account has occurred in application areas of NLP such as text classification, text summarization, information retrieval and question answering. That is, the research interest in application areas of NLP has moved on to more challenging tasks. Even with text classification tasks, one of the traditional NLP tasks, the target classes have recently been diversified from topics such as ‘sports’ and ‘economics’ to the contents of texts such as ‘polarity’ or ‘subjectivity’, called sentiment classification or sentiment analysis. Sentiment classification is defined as a task of classifying texts into the sentiment matter contained in the texts, such as like vs. dislike, recommend vs. not recommend, and subjective vs. objective (Suzuki Jun, 2005).

Sentiment analysis methodology can be used for better understanding of users' emotion and preferences, concerned with the determination of human emotions from textual data. This is an emerging field. Positive sentiment indicates praise and negative sentiment indicates criticism. Sentiment analysis can be used for grouping search engine results, analyzing news content, reviews for books, movies, sports, etc and different types of blogs, social networks and web forums. Today, it is usual for business organizations to receive large volumes of opinion data from their customers. Human analysis of these types of data is, however, time consuming, expensive and could be highly subjective. Sentiment analysis technology can be used by business organizations to better understand their customers. Sentiment analysis does not eliminate human participation in solving the task completely; it significantly simplifies the job and allows an analyst to manage the process of extracting sentiment from text. It automates the process of finding relationships and patterns in textual data and delivers results that can be either utilized in an automated decision support system or assessed by a human analyst.

1.2 Problem and Motivation

It is really hard for the computer to pull out the tone and meaning of a document automatically because people express things in many different ways. In this thesis, we propose a novel approach to sentiment analysis. We call it *proximity-based sentiment analysis*. Our basic motivation for this approach starts with answering the question "how do people write?". We believe that most of the people follow a basic structure in writing. People follow this basic structure, even without thinking about it and the sentiment of a document can be identified by the structural characteristics and the distribution of distances between the words in the document.

We applied our proximity based sentiment analysis technique to the problem of movie review analysis. Accurate automatic prediction of how much an audience is going to enjoy a movie is still a challenging research question. Different classification techniques can be used for better classification but the choice of a classifier plays a big role for this type of dataset. In this thesis we have systematically studied the accuracy of different classifiers in a movie review domain, where the input to the classifiers are proximity based features.

1.3 Prior Work

An early approach to sentiment analysis was scanning a document to count the number of occurrence of certain keywords, based on which the document is categorized as positive or negative. But that method was error prone and fails to capture the subtleties that bring human language to life. Different approaches to sentiment analysis have been recently proposed. Most of these are based on common machine learning techniques such as Support Vector Machine (SVM), Naïve Bayes (NB), Maximum Entropy (ME), etc. Researchers doing sentiment analysis focus on specific tasks, such as finding the sentiments of words (Hatzivassiloglou & McKeown, 1997), subjective expressions (Wilson, Wiebe, & Hoffmann, 2005), subjective sentences (Pang & Lee, 2004) and topics (Hiroshi, Tetsuya, & Hideo, 2004). (Pang and Lee, 2004) proposed a machine-learning method that can be used to improve the effectiveness of sentiment classification. (Dasgupta S., and Ng V., 2009) proposed a text clustering algorithm based on the author's mood. (Prabowo Rudy and Thelwal Mike, 2009) proposed a combined approach for sentiment analysis.

1.4 General Approach and Thesis Contribution

In this thesis, our contribution is proximity-based sentiment analysis. Our basic motivation for this research starts with a psychological question “how do people write?”. We believe that when a person starts writing positively about a topic or subject they continue with this positive trend for a period of time. Later he/she will use inflexion words like “however” and then start writing negatively about the topic. We assumed that in a paragraph people don't repeatedly write one positive and one negative word together. Therefore we used proximity as a feature of our sentiment analysis work.

Although previous machine learning methods have shown improved performance in sentiment analysis, perhaps, a more fundamental issue is how we populate the feature space, upon which these machine learning methods rely. In this work, we take a different approach, by considering a new set of features based on word proximities in a written text. To our knowledge, this is the first attempt at focusing on proximity based features as the primary features in sentiment analysis. Combining this new feature for sentiment analysis with other standard approaches, such as part-of-speech (POS) tagging, bag-of-words (BOW)s, phrases or higher-order n-grams will no doubt lead to further improvements.

1.5 Thesis Organization

Chapter 2 provides general background and existing work in the area of sentiment analysis. A survey of the previous work on sentiment analysis is presented.

Chapter 3 introduces the concept of proximity based sentiment analysis which is the novel idea that we applied in our research. Proximity based analysis is discussed in detail including our proximity model. A description of three different feature sets that we applied in our research is presented.

Chapter 4 explains the detail of our research design and the classification techniques that we applied. We describe three different approaches and the machine learning algorithms that we used for our research in detail.

In Chapter 5, we discuss the results that we obtained. Experiments are undertaken on sentiment analysis task to evaluate the effect of the proposed method. Here we investigate the results of our three approaches: - unsupervised approach, mean-median approach, and machine learning approach. Results for various variations of the suggested scheme with varying parameters are also been included.

Finally, in Chapter 6 we conclude by summarizing the thesis and providing an outlook for future research.

Chapter 2: Background and Literature Review

As mentioned in Chapter 1, the topic of this thesis is an evaluation of the effect of proximity based sentiment analysis. Therefore, before explaining the proposed methods, this chapter provides a survey of previous work on sentiment analysis.

Sentiment analysis can help business organizations like a virtual currency by providing a better understanding of customer opinions. Researchers are using sentiment analysis techniques to analyze data on blogs, social networks, bulletin boards, web forums, product reviews, movie reviews, hate groups, recommender systems, etc. Early research on sentiment analysis (Hearst, M., 1992; Sack, W., 1994) used models inspired by cognitive linguistics or manual or semi-manual construction of discriminant-word lexicons (Huettnner and Subasic, 2000). Kessler et al. (1997) did research into classifying text based on their sentiment or genre. As discussed in (Prabowo and Thelwal, 2009), researchers doing sentiment analysis focus on specific tasks, such as finding the sentiments of words (Hatzivassiloglou & McKeown, 1997), subjective expressions (Wilson, Wiebe, & Hoffmann, 2005), subjective sentences (Pang & Lee, 2004) and topics (Hiroshi, Tetsuya, & Hideo, 2004), or extracting the sources of opinions (Choi, Cardie, Riloff, and Patwardhan, 2005). Pang and Lee 2004 proposed a machine-learning method that can be used to improve the effectiveness of sentiment classification. Dasgupta S. and Ng V. (2009) proposed a text clustering algorithm based on the author's mood.

A number of researchers investigated hate groups using sentiment analysis techniques. For instances Glaser, Dixit & Green (2002) studied hate crime on the internet; Crilley (2001) and Schafer (2002) studied the use of the Internet by terrorists, extremists and activists. Zhou et al (2005) and Abbasi & Chen (2005) did a thorough investigation about US hate group web sites. Burris Smith Strahm (2000) discussed the importance of analyzing web forums and chat-room content and they use social network analysis to examine the inter-organizational structure of the white supremacist movement. Gamon, M., (2004) used customer feedback dataset for assigning sentiment to documents using a 4-point scale. Similar work was proposed by Pang and Lee (2005) for movie review analysis. Earlier Pang et al (2002) presented sentiment analysis on movie, while Dave

Lawrence Pennock (2003) used a number of machine learning techniques to analyze product review data.

For automatic sentiment analysis, the following approaches have been proposed: Natural Language Processing (NLP) and pattern-based methods (Hiroshi et al., 2004), machine learning algorithms, such as Naive Bayes (NB), Maximum Entropy (ME), Support Vector Machine (SVM) (Joachims, 1998), and unsupervised learning (Turney, 2002). (Rudi & Paul, 2007) proposed method for sentiment orientation. More recently, (Prabowo Rudy & Thelwal Mike, 2009) discussed a combined approach (rule-based classification, supervised learning and machine learning) for sentiment analysis.

In machine learning based classification, two sets of documents are required: a training and a test set. A training set (Tr) is used by an automatic classifier to learn the differentiating characteristics of documents, and a test set (Te) is used to validate the performance of the automatic classifier. To measure performance confusion table is commonly used because it is easy to see if the system is confusing between classes.

Table 1: A confusion table.

	Machine says yes	Machine says no
Human says yes	tp	fn
Human says no	fp	tn

The existing work of sentiment analysis is listed in Table 2 and shows different types of objectives along with the associated models used and the experimental results produced. In an ideal scenario, all the experimental results are measured based on the micro-averaged and macro-averaged precision, recall, and F_1 as explained below:

$$Precision(P) = \frac{tp}{tp + fp}; Recall(R) = \frac{tp}{tp + fn}; Accuracy(A) = \frac{tp + tn}{tp + tn + fp + fn};$$

$$F_1 = \frac{2.P.R}{P + R}$$

1. Micro-averaging: Given a set of confusion tables, a new two-by-two contingency table is generated. Each cell in the new table represents the sum of the number of documents from within the set of tables. Given the new table, the average performance of an automatic classifier, in terms of its precision and recall, is measured.

2. Macro-averaging: Given a set of confusion tables, a set of values are generated. Each value represents the precision or recall of an automatic classifier for each category. Given these values, the average performance of an automatic classifier, in terms of its precision and recall, is measured (Prabowo Rudy & Thelwal Mike, 2009).

In our research, we applied our scheme for movie review classification. Kimitaka et al. proposed a method to classify movie review documents into positive or negative opinions. They describe multiple classifiers for the review document classification task. The method consists of three classifiers based on SVMs, ME and score calculation. They applied two voting methods and SVMs to the integration process of single classifiers. The integrated methods improved the accuracy as compared with the three single classifiers. The method identified the class (positive or negative) of a document on the basis of the distances that were measured from the hyperplane of each classifier. It obtained the better accuracy as compared with the single classifiers. This method, however, contained a problem for the determination of the final output, namely positive or negative. To normalize the classifier's outputs manually because the scale of the scoring method was different from that of SVMs they applied the 3rd machine learning method (Maximum Entropy) into the method based on the scoring and SVMs. Authors also compared two voting process, naive voting and weighted voting, and the integration with SVMs for the method. There methods consisting of three classifiers outperformed single classifiers. Even the naive voting produced higher accuracy than these classifiers. In experiment, SVMs in the integration process produced the best performance. If the weights for the voting method optimally tuned up, it obtained the best accuracy.

Although machine learning methods have shown improved performance in sentiment analysis, perhaps, a more fundamental issue is how we populate the feature space, upon which these machine learning methods rely. In this work, we take a different approach, by considering a new set of features based on word proximities in a written text. To the best of our knowledge, this is the first attempt at focusing on proximity based features as the primary features in sentiment analysis. Combining this new feature for sentiment analysis with other standard approaches, such as part-of-speech (POS) tagging, bag-of-words (BOW)s, phrases or higher-order n-grams will no doubt lead to further improvements.

Table 2: Existing work on sentiment analysis adapted from (Prabowo Rudy & Thelwal Mike, 2009).

Author	Polarity Objectives (No Classes)	N-gram	Model	Data source	Eval. method	Data set	Tr	Te	Accuracy	Precision	Recall	F1
Gamon (2004)	Documents		SVM	Customer feedback	Ten-fold cross validation	N/A	36,796	4084	77.5	N/A	N/A	N/A
Pang and Lee (2005)	Documents		SVM, regression, metric labeling	Movie reviews	Ten-fold cross validation	5006	N/A	N/A	66.3	N/A	N/A	N/A
Choi et al. (2005)	Sources sentiments		CRF and AutoSlog	MPQA corpus	Ten-fold cross validation	N/A	135	400	N/A	70.2–82.4	41.9–60.6	59.2–69.4
Wilson et al. (2005)	Expressions		BoosTexter	MPQA corpus	Ten-fold cross validation	13,183 expressions	N/A	N/A	73.6–75.9	68.6–72.2/ 74.0–77.7	45.3–56.8/ 85.7–89.9	55.7–63.4/ 80.7–82.1
König and Brill (2006)	Documents		Pattern-based, SVM, hybrid	Movie Reviews Customer Feedback	Five-fold cross validation	1000(+)	N/A	N/A	>91	N/A	N/A	N/A
Hatzivassiloglou and McKeown (1997)	Adjectives	N/A	Nonhierarchical clustering	WSJ corpus	N/A	657adj(+) 679adj(-)	N/A	N/A	78.1–92.4	N/A	N/A	N/A
Pang, Lee, and Vaithyanathan (2002)	Documents	Unigram and Bigrams	NB, ME, SVM	Movie reviews	Three-fold cross validation	700(+)	N/A	N/A	77–82.9	N/A	N/A	N/A
Turney (2002)	Documents	N/A	PMI-IR	Automobile, bank, movie, travel reviews	N/A	700(-) 240(+)	N/A	N/A	65.8–84	N/A	N/A	N/A
Yi et al. (2003)	Topics sentiments	-	NLP, pattern-based	Digital camera, music reviews Petroleum, Pharmaceutical Web pages	N/A	170(-) 735(+)	N/A	N/A	85.6	87	56 N/A	N/A
					N/A	4227(-) N/A	N/A	N/A	90–93	86–91		N/A
Nasukawa and Yi (2003)	Topics sentiments	–	NLP, pattern-based	Web pages	N/A	118(+)	N/A	N/A	94.3	N/A	28.6	N/A
				Camera reviews	N/A	58(-) 255	N/A	N/A	94.5	N/A	24	N/A

Dave, Lawrence, and Pennock (2003)	Documents	Unigram, bigram and trigrams	Scoring, smoothing, NB, ME, SVM	Product reviews	Macro-averaged	N/A	13,832(+)	25,910(+)	88.9	N/A	N/A	N/A
							4389(-) 2016(+) 2016(-)	5664(-) 224(+) 224(-)	85.8	N/A	N/A	N/A
Hiroshi et al. (2004)	Topics sentiments	-	NLP, pattern- based	Camera reviews	N/A	200	N/A	N/A	89–100	N/A	43	N/A
Pang and Lee (2004)	Documents	Unigrams	NB, SVM	Movie reviews	Ten-fold cross validation	1000(+) 1000(-)	N/A	N/A	86.4– 87.2	N/A	N/A	N/A
Kim and Hovy (2004)	Expressions		Probabilistic based	DUC corpus	Ten-fold cross validation	N/A	231 Adjectives	N/A	79.1–	N/A	93.2	N/A
							251 verbs	N/A	81.2	N/A	N/A	N/A
							N/A	100 sentence s	81			

Chapter 3: Proximity-Based Analysis

Sentiment analysis methods can be categorized into two broad groups:- i) symbolic techniques and ii) machine learning techniques. The symbolic approach uses manually crafted rules and lexicons, where the machine learning approach uses unsupervised, weakly supervised or fully supervised learning to construct a model from a large training corpus. The symbolic techniques using lexicon based such as bag-of-words approach, using a web search-to calculate the semantic orientation for a tuple, using WordNet-to determine the orientation and emotional content of a word and combination of lexical and grammatical approach. Machine learning approaches use unigrams, n-grams, lemmas, negation, opinion words and adjectives for feature selection and classification methods such as support vector machines, naïve Bayes multinomial, maximum entropy.

In this thesis, our contribution is proximity based sentiment analysis. Our basic motivation for this research starts with a psychological question “how do people write?”. We believe that when a person starts writing positively about a topic or subject they continue with this positive trend for a period of time. Later he/she will use inflexion words like “however” and then start writing negatively about the topic. We assumed that in a paragraph people don’t repeatedly write one positive and one negative word together. This model of writing is captured schematically in Figure 1.

POSITIVE POSITIVE POSITIVE NEGATIVE NEGATIVE POSITIVE
NEGATIVE NEGATIVE NEGATIVE POSITIVE POSITIVE NEGATIVE
~~**POSITIVE NEGATIVE POSITIVE NEGATIVE POSITIVE NEGATIVE**~~

Figure 1: Assumption on human writing style.

Typically segments of a written text (e.g. paragraphs or sentences) capture a concept or trend of thought over a short period of time. Such trends could fluctuate as we move along the written document. Within a segment, words used can give ideas on whether there is a positive, negative, or no sentiment being expressed in the segment. From the foregoing,

we have the following hypothesis: “The average distance between positive-oriented (negative-oriented) words is expected to be small for segments bearing positive (negative) sentiments”. Consequently “The average distance between positive-oriented (negative-oriented) words is expected to be relatively large for segments bearing negative (positive) sentiments”.

Figure 2 shows a schematic diagram of the proposed approach. Below we provide further details on the proposed proximity-based sentiment analysis.

Figure 2: General schematic diagram of the approach.

3.1 Data Cleaning

In order to capture the proximity between words, first we automatically remove punctuation and stop words from input text. After that we separated all the unique words of a given text. In this phase we also removed stop words in the text, but we did not perform any stemming.

3.2 Measuring Distances

By using the Stone Dictionary (Stone et. al 1966) we identify the polarity of the word. Here polarity means positive or negative. For distance measurement first of all we considered only polarity words that mean only positive and negative words. After that we considered polarity and non-polarity words. Here non-polarity means those words that are not available in the Stone Dictionary.

3.2.1 Polarity Words

The clean text and unique word list that we build up in previous phase allow us to perform proximity based analysis. We divide the input text into a number of segments or windows, each segment contains 100 words. Figure 3 is a visualization of the segmentation process for an input text.

Figure 3: Segmentation of Input Text.

Now we are interested in computing the distances between word pairs. We measure the distances for the following three types of pairs.

- POSITIVE-POSITIVE words.
- NEGATIVE-NEGATIVE words.
- POSITIVE-NEGATIVE words.

From the unique word list we select a pair of words. Then, using the Stone Dictionary (Stone et. al 1966) we identify the polarity of the words in a pair. For polarity we only consider two classes positive and negative. Then we measure the different distances between the pair of words and count the number of occurrences of that pair for each particular distance in a segment of a given text. Our distance range is 1-100 since we used different segments of 100 words each. We repeated the above procedure in each

segment for all possible combination of words available in the unique word list. Consider the following example text segment.

Example: the *funniest* **horror** movie ever made and wile coyote evil **dead** dawn is the result of an **unstable** fusion.

The word “funniest” is a positive word and “horror” is a negative word. There is no word between funniest and horror so we consider it as positive–negative distance measurement and in this case distance is 1. In a similar manner, the distance between bold negative-negative words “horror” and “dead” will be 7, while that between “horror” and “unstable” will be 3 (ignoring words, such as “of”, “and”, “the”, etc.)

Thus, for each positive and negative review we calculated all the above mentioned distances. We used following notation (Table 3) for the reviews and for their distances.

Table 3: Distance Notation with polarity words.

Review True Class	Distance	Notation
POSITIVE	POSITIVE-POSITIVE WORDS (++)	P++
	NEGATIVE-NEGATIVE WORDS(--)	P--
	POSITIVE-NEGATIVE WORDS(+--)	P+-
NEGATIVE	POSITIVE-POSITIVE WORDS(+++)	N++
	NEGATIVE-NEGATIVE WORDS(---)	N--
	POSITIVE-NEGATIVE WORDS(++-)	N+-

3.2.2 Polarity & Non-polarity Words

Here we perform similar distance measurement mentioned in section 3.2.1 but we consider non-polarity words also. We measure the distances for the following three types of pairs.

- POSITIVE-POSITIVE words.
- NEGATIVE-NEGATIVE words.
- POSITIVE-NEGATIVE words.

- POSITIVE-NON-POLARITY words
- NEGATIVE- NON-POLARITY words
- NON-POLARITY- NON-POLARITY words

For each positive and negative review we calculated all the above mentioned distances. We used following notation (Table 4) for the reviews and for their distances.

Table 4: Distance Notation with polarity and non-polarity words.

Review True Class	Distance	Notation
POSITIVE	POSITIVE-POSITIVE WORDS (++)	P++
	NEGATIVE-NEGATIVE WORDS(--)	P--
	POSITIVE-NEGATIVE WORDS(+/-)	P+/-
	POSITIVE-NON-POLARITY WORDS (+0)	P+0
	NEGATIVE- NON-POLARITY WORDS (-0)	P-0
	NON-POLARITY- NON-POLARITY WORDS (00)	P00
NEGATIVE	POSITIVE-POSITIVE WORDS(++)	N++
	NEGATIVE-NEGATIVE WORDS(--)	N--
	POSITIVE-NEGATIVE WORDS(+/-)	N+/-
	POSITIVE-NON-POLARITY WORDS (+0)	N+0
	NEGATIVE- NON-POLARITY WORDS (-0)	N-0
	NON-POLARITY- NON-POLARITY WORDS (00)	N00

3.3 Proximity Models

As we mentioned in Chapter 1, the main topic of this thesis is an evaluation of the effect of handling proximity information within text for sentiment analysis. In this section we have formalized the proposed methods of this thesis.

We consider two models of proximity between words in a text. One is an all-against-all model of proximity and the other is the sequential model. The two models are shown in Figure 4. Thus, for a given document, we can compute the proximity types, and their distances, based on the chosen proximity model. We can observe that the all-against-all

model contains much more information than the sequential model. However, it will require much more time, typically time in $O(n^2)$, where n is the number of words. The sequential model is more succinct, contains less redundant information, and requires less time to compute, typically in $O(n)$ time complexity. Our experiments are based on all-against-all model.

Figure 4: Two Proximity Models; (a) all-against-all model (b) sequential model

3.4 Proximity-Based Features

Using the proximity models, we determine the word proximities for given pairs of words in the text being analyzed. The question then becomes how we can extract features from the pair-wise distances for the three proximity types for the required problem of sentiment analysis. We consider three proximity-based features:

Proximity Distributions:

Distribution of pair wise distances as returned by the proximity models, for distances from 1 to 100. We considered different number of bins, from 1 bin (cumulative distance over all

the bins), to 10 bins (eqi-spaced), 9 bins (non-uniform spacing), and 100 bins (one distance/bin). Followings are three different distance ranges that we have used in our research. Range 1 is just 10 equal bins. For each review with polarity words we take the sum of first 10 and divide it by sum of ++, -- & +- of that review for normalization purpose and we did it second 10, third 10 and so on. We repeat the previous mentioned procedure for polarity and non-polarity words with six distance pairs. Range 2 has 9 bins. Here bins are not equal but for normalization we follow same procedure as for 10 bins. First three dimensions of range 3 are different bins, next three dimensions are the distance between the bins and last three are correlations. For first 3 dimensions of range 3 we used 10 bins (Range 1), 9 bins (Range 2) and 1 bin (cumulative of distance1-100).

Range 1: 1-10, 11-20, 21-30, 31-40, 41-50, 51-60, 61-70, 71-80, 81-90, 91-100.

Range 2: 1-3, 4-8, 9-15, 16-25, 26-40, 41-55, 56-70, 71-85, 86-100

Range 3: [P(++,-,+) P(F1(++,+), F2(++,-), F3(+,-,-)) P(C1(++,+),C2(++,-),C3(+,-,-))]

[N(++,-,-) N(F1(++,+), F2(++,-), F3(+,-,-)) N(C1(++,+),C2(++,-),C3(+,-,-))]

F(++,+)= Distance(++,+)

C(++,+)=correlation (++,+)

P=Positive Review.

N=Negative review.

Here, Distance means Manhattan Distance. Formula for Manhattan distance is

$$F(X,Y) = Dis\ tan\ ce(X,Y) = \sum_{i=1}^n |x_i - y_i|$$

For correlation calculation we used following formula:-

$$C(X,Y) = Corr(X,Y) \frac{n \sum XY - (\sum X)(\sum Y)}{\sqrt{[n \sum X^2 - (\sum X)^2][n \sum Y^2 - (\sum Y)^2]}}$$

Here, a series of n measurements of X and Y written as x_i and y_i where $i = 1, 2, \dots, n$, the sample correlation coefficient can be used to estimate the population Pearson correlation r between X and Y.

Mutual Information between Proximity Types:

Given the three proximity types (++ , +- , --), we expect that, for a given document, we could use the relationship between the types to have some idea on the document polarity. For example, for a document with positive review, we expect that the sequence of proximity values of type (++) will contain more information about the (+-) proximity type, when compared with the sequence with the (--) proximity type. Thus, for such positive documents, we should expect the mutual information between the (++) and (+-) sequences to be larger than those between the (--) and (+-) sequences. To capture this intuition, we use information theoretic quantities of entropy for each sequence, and mutual information between pairs of proximity types. The entropy of a sequence S is defined as:

$$H(S) = -\sum_{i=1}^{|\Sigma|} p(s_i) \log p(s_i) \dots\dots\dots(1)$$

Here $p(s_i)$ is the probability of the i -th symbol in the alphabet, Σ . Let $P_{ij}(k)$ be the joint probability of finding symbols i and j separated (in that order) by k positions in the sequence. Let p_i and p_j be the respectively probability of symbol i and j in the sequence. The mutual information is defined as follows (Cover & Thomas, 1991):

$$I(k) = \sum_{i \in \Sigma} \sum_{j \in \Sigma} P_{ij}(k) \log \frac{P_{ij}(k)}{p_i p_j} \dots\dots\dots(2)$$

Weiss, O., and Herzel, H., (1998) approximated the mutual information as :

$$I(k) = \frac{1}{2 \ln 2} \sum_{i \in \Sigma} \sum_{j \in \Sigma} \frac{D_{ij}^2(k)}{p_i p_j} + O(D_{ij}^3(k)) \dots\dots\dots(3)$$

$$\text{Here } D_{ij}(k) = P_{ij}(k) - p_i p_j .$$

Based on this, they defined correlation functions $C_{\vec{a}\vec{b}}(k)$ as a linear combination of the distances above:

$$C_{\vec{a}\vec{b}}(k) = \sum_{i \in \Sigma} \sum_{j \in \Sigma} a_i D_{ij}^2(k) b_j \dots\dots\dots(4)$$

The weights a_i and b_j are binary digits (0 or 1). When $\vec{a} = \vec{b}$, we have the autocorrelation function, $C_{\vec{a}\vec{a}}(k)$. For a given separation parameter, k , we computed the mutual information for all proximity values from 1, 2, ..., k .

For polarity and non-polarity words we applied mutual information formula to (++) and (+0) pairs and (--) and (-0) pairs.

Proximity Patterns:

In some sense, the above features considered summary or global information about the word proximities in a given document. They did not pay much attention on the context of these pair-wise proximities, and hence may not be able to capture some local variability in the sentiments being expressed by an author. Yet, the sentiment of the overall document could depend critically on how the word polarities are changing within a small segment of the text, for example, the concluding part. So, could be a powerful method on segmented text. To capture this observation, we define proximity patterns. First of all we consider words with polarity which essentially is a binary sequence describing the polarity of the words used in the document. After that we consider words with polarity and without polarity which is a ternary sequence. Polarity is defined by the Stone Dictionary used. We then consider n-length blocks of this proximity patterns, and use these for classification. In this work, we considered n=2, 3, and 5. The patterns that we get, it follows power of 2 for polarity words and power of 3 for polarity and non-polarity words. For polarity words, if n=2 we have 4 patterns, for n=3 we have 8 patterns and for n=5 we have 32 patterns. For polarity and non-polarity words, if n=2 we have 9 patterns, if n=3 we have 27 patterns and if n=5 we have 243 patterns. Following is the example (Table 5 for polarity & Table 6 for polarity and non-polarity words).

Table 5: Polarity Pattern Example.

N=2	N=3	N=5
Pattern 1: ++ Pattern 2: +- Pattern 3: -+ Pattern4: --	Pattern 1: + + + Pattern 2: + + - Pattern 3: + - + Pattern 4: + - - Pattern 8: - - -	Pattern 1: +++++ Pattern 2: +++++- Pattern 3: ++++++ Pattern 4: ++++-- Pattern 32:- - - - -

Table 6: Polarity and Non-polarity Pattern Example.

N=2	N=3	N=5
Pattern 1: + + Pattern 2: + - Pattern 3: + 0 Pattern4: - + Pattern 9: 0 0	Pattern 1: + + + Pattern 2: + + - Pattern 3: + + 0 Pattern 4: + - + Pattern 27: 0 0 0	Pattern 1: + + + + + Pattern 2: + + + + - Pattern 3: + + + + 0 Pattern 4: + + + - + Pattern 243:0 0 0 0 0

Chapter 4: Sentiment Classification Techniques

After extracting the features, the next phase is sentiment classification, using the generated features. In this chapter we present sentiment classification techniques that we employed. Therefore, this section provides the background knowledge needed to understand the thesis. We performed our empirical study using three approaches, namely unsupervised method, mean & median method, supervised method using machine learning approaches.

4.1 Unsupervised Approach

From the results obtained in the previous phase, for polarity words only we sum all the occurrence of Positive-Positive, Negative-Negative and Positive-Negative word pairs. As we mentioned earlier our distance range is 1-100. So, after that we automatically calculated the cumulative probability from range 1-100 for all three pairs (Positive-Positive, Negative-Negative and Positive-Negative) that we considered. After that we perform classification. We have three classes: - positive, negative and undefined. We calculated the difference of cumulative probability between Positive-Positive and Negative-Negative pair and after that based on the result we did classification. We used three classification thresholds 0, 0.025, 0.05. The classification scheme is described as follows:

Cumulative Probability=CP

Classification threshold $\tau=0, 0.025, 0.05$

$\Delta = \text{CP (Positive-Positive)} - \text{CP (Negative-Negative)}$

Positive, if $\Delta \geq \tau$

Negative, if $-\Delta \geq \tau$

Undefined, if $|\Delta| < \tau$

For classification we also used ratio instead of only subtraction. Following is the scheme

Cumulative Probability=CP

$\Delta = (\text{CP (Positive-Positive)}) : (\text{CP (Negative-Negative)})$

Positive, if $\Delta > 0.02$

Negative, if $\Delta < -0.02$

Undefined, if $|\Delta| = \text{range of } 0-0.02$

We also calculated precision and recall based on the following formula.

$$Precision = \frac{\text{Correct Classification by System}}{\text{Total Classified (as Positive or Negative)}} \dots\dots\dots (5)$$

$$Recall = \frac{\text{Correct Classification by System}}{\text{Total Classified (Total Number of Records)}} \dots\dots\dots (6)$$

4.1.1 Feature Weighting

An important issue in classification is how we need to weight the different features that we used. Our observation from the movie reviews¹ is that, using say proximity patterns with polarity, the predominantly positive patterns (example, +++++, ++++-, etc,) tended to dominate, even for documents with overall negative sentiment. What this means is that, even when the people have negative opinion about a subject, they still use a lot of words with positive polarity in expressing themselves. But a human reading the text will still be able to pickup the overall sentiment, in spite of the use of words with positive polarity. This problem is handled in part by the specific pattern of occurrence of the proximity words. However, this observation also implies that we may need to assign appropriate weights to the polarity patterns, before using them in decision making. Typically, this means we need to use a factor less than 1 on the predominantly positive patterns, in order to reduce the bias due to the general human writing preference, before we can use these features.

4.2 Mean & Median Approach

This is a supervised approach, with training and testing phases. For this approach from our 1400 (700 positive and 700 negative) movie review corpus we randomly selected 1000 review (500 positive and 500 negative) for system training and 400 reviews (200 positive and 200 negative) for system testing. We calculated the mean and median of the training set. Suppose T is a test review then D(T+-,P+-) means Euclidian distance between test review T and train set P. Here +- means Distance between positive-negative words. We perform classification as follows:.

¹ polarity dataset version 1.0, which is available at:
<http://www.cs.cornell.edu/people/pabo/movie-review-data/>

$$d+++ = \alpha.D(T++,P++) + \beta.D(T--,P--) + \gamma.D(T+-,P+-)$$

$$d--- = \alpha.D(T++,N++) + \beta.D(T--,N--) + \gamma.D(T+-,N+-)$$

if, $d+++ \leq d---$, classify as +ve

otherwise, classify as -ve

Here, α , β , γ are three different weights and $\alpha+\beta+\gamma=1$. We tested between the ranges of 0 to 1 within an interval of 0.02.

if $\alpha=0$, $\beta=0$ we obtain result 1a (Section 5.3.2)

if $\beta=0$, $\gamma=0$ we obtain result 1b (Section 5.3.2)

if $\alpha=0$, $\gamma=0$ we obtain result 1c (Section 5.3.2)

if $\gamma=0$ we obtain result 2 (Section 5.3.2)

else we obtain result 3 (Section 5.3.2)

4.3 Machine Learning Approach

Our objective of this work is to inspect whether it is possible to do sentiment analysis by using word proximity as a feature. We also experimented feature with different standard machine learning algorithms:- Naïve Bayes classification, support vector machines, K-Nearest Neighbours, J48 (also called C4.5) and multilayer perceptron. Although from algorithmic point of view these three are different but each has been shown to be effective for analyzing textual data. Here once again we have used 1000 review (500 positive and 500 negative) for system training and 400 reviews (200 positive and 200 negative) for system testing. Training and testing set are selected randomly.

4.3.1 Naïve Bayes (NB) classification:

This method is based on probabilistic knowledge. It uses the Bayes theorem to find the probability of all the classification in the database. Naïve Bayes is based on supervised learning. The probability model for a classifier is a conditional model $Pr(c|f_1, f_2, \dots, f_n)$ over a dependent class variable c with a small number of outcomes or classes, conditional on several feature variables f_1 through f_n . The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more

tractable. The Bayes' theorem relates the conditional and marginal probabilities of stochastic events c , and f .

$$\Pr(c | f) = \frac{\Pr(c) \Pr(f | c)}{\Pr(f)} \dots\dots\dots(7)$$

Where, $\Pr(c)$ is the prior probability of hypothesis c ; $\Pr(f)$ is the prior probability of training data f ; $\Pr(c|f)$ is the probability of given f and $\Pr(f|c)$ is the probability of f given c . Using Bayes' theorem for several feature variables f_n , we can rewrite this as:

$$\Pr(c | f_1, \dots, f_n) = \frac{\Pr(c) \Pr(f_1, \dots, f_n | c)}{\Pr(f_1, \dots, f_n)} \dots\dots\dots(8)$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on c and the values of the features f_i are given, so that the denominator is effectively constant (Pop, I. ,2006).

4.3.2 Support Vector Machines (SVM)

Support vector machine (SVM) classifiers are based on Vapnik's structural risk minimization principle. Given a set of N linearly separable training examples $S = \{x_i \in R^n | i = 1, 2, \dots, N\}$ where each sample belongs to one of the two classes, $y_i \in \{+1, -1\}$, the SVM approach seeks the optimal hyperplane $w \cdot x + b = 0$ that separates the positive and negative examples with the largest margin. The problem can be formulated as solving the following quadratic programming problem.

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \|w\|^2 \\ \text{Subject to} & y_i(w \cdot x_i + b) \geq 1 \end{array}$$

Figure 5: Illustration of an SVM classifier, showing examples of separating hyper-planes (solid lines), optimal hyper-plane (bold solid line) and support vectors (data on the dashed lines). The dashed line identifies the maximum margin.

The basic SVM formulation can be extended to the nonlinear case by using nonlinear kernels. Interestingly, the complexity of an SVM classifier representation does not depend on the number of features, but rather on the number of support vectors (the training examples closest to the hyperplane). This property makes SVMs suitable for large dimensional classification problems. SVMs are often considered to be the state of the art classification technique in many text classification problems. A shortcoming of SVM classifiers is that they do not deal with unobserved features well. They simply ignore all unobserved features by assigning them zero weight. Feature smoothing for SVMs is not as straightforward as for probabilistic models, such as naïve Bayes classifiers, and remains an ongoing research problem (Peng F., 2003).

4.3.3 K-Nearest Neighbours (K-NN)

The k-nearest neighbours algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. In k-nearest neighbor algorithm an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The k-nearest neighbor algorithm is sensitive to the local structure of the data. Nearest neighbor rules in effect compute the decision boundary in an implicit manner. It is also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity (Bremner et. al 2005).

4.3.4 J-48 Algorithm

The C4.5 algorithm is a part of the multiway split decision tree. C 4.5 yields a binary split if the selected variable is numerical, but if there are other variables representing the attributes it will result in a categorical split. That is, the node will be split into C nodes where C is the number of categories for that attribute. The J4.8 decision tree in WEKA is

based on the C4.5 decision tree algorithm. The C4.5 learning algorithm is described in the following (Eapen G. 2004):-

1. Build the decision tree from the training set (conventional ID3)
2. Convert the resulting tree into an equivalent set of rules. The number of rules is equivalent to the number of possible paths from the root to a leaf node.
3. Prune each rule by removing any preconditions that result in improving its accuracy, according to a validation set.
4. Sort the pruned rules in descending order according to their accuracy, and consider them in this sequence when classifying subsequent instances.

4.3.5 Multilayer Perceptron (MLP)

MLP neural networks consist of units arranged in layers. Each layer is composed of nodes and in the fully connected network considered here each node connects to every node in subsequent layers. Each MLP is composed of a minimum of three layers consisting of an input layer, one or more hidden layers and an output layer. The input layer distributes the inputs to subsequent layers. Input nodes have linear activation functions and no thresholds. Each hidden unit node and each output node have thresholds associated with them in addition to the weights. The hidden unit nodes have nonlinear activation functions and the outputs have linear activation functions. Hence, each signal feeding into a node in a subsequent layer has the original input multiplied by a weight with a threshold added and then is passed through an activation function that may be linear or nonlinear (hidden units). Figure 6 shows an example three layer perceptron. For the actual three layers MLP, all of the inputs are also connected directly to all of the outputs (Nazzal J., et. al 2008).

Figure 6: Three layer multilayer perceptron.

Chapter 5: Results & Discussion

5.1 Dataset

We used sentiment polarity dataset v1.0 which is a movie review dataset introduced by Pang and Lee (2002)². It contains 1,400 movie reviews in text files, among these 700 are positive and 700 are negative reviews.

5.2 Packages Used

Our programs are written in C, Python & MATLAB programming language and we used Visual Studio 6.0 for C, Python 2.7 for Python and MATLAB version 7.5.0.342 (R2007b) programming task. For data analysis and plotting we used Microsoft Excel 2007.

For machine learning algorithms we used Waikato Environment for Knowledge Analysis (WEKA) tool in our study. The WEKA software was developed in the University of Waikato, New Zealand. A number of data mining methods are implemented in the WEKA software. Some of them are based on decision trees like the J48 decision tree, some are rule-based like ZeroR and decision tables, and some of them are based on probability and regression, like the Naïve bayes algorithm. We used WEKA version of 3.6.2.

5.3 Results

In data cleaning phase, for each review we automatically remove all the punctuation and stop words and after that by using stone dictionary and unique word list we count distance between positive-positive words, negative-negative words and positive-negative words. Our distance range is 1-100. Results obtain from our approaches are presented in the following.

² polarity dataset version 1.0, which is available at:

<http://www.cs.cornell.edu/people/pabo/movie-review-data/>

5.3.1 Polarity Word Count

First of all for classification we applied a simple scheme called polarity word count. For each review, we calculated the sum of positive and negative words. If sum of positive words is greater than negative then we classify that review as positive else we classify that review as a negative review. We use 700 positive and 700 negative reviews for this experiment and got the following result presented in the Table 7.

Table 7: Performance of polarity word count.

	Accuracy Rate
Positive Reviews	65.29%
Negative Reviews	59.43%
Overall Performance (both positive & negative)	62.36%

5.3.2 Unsupervised Approach

For the unsupervised approach, we use some simple calculations on the features of polarity words and make decisions on the overall polarity of the document, without initial training. For this approach we used 100 positive reviews and 100 negative reviews from our movie review corpus which contains 1400 reviews. Reviews are randomly selected. The results of our experiment are presented in Table 8.

Table 8: Performance of Unsupervised Approach (both positive and negative review).

	$\tau=0$	$\tau=0.025$	$\tau=0.05$
Precision	67%	66%	69%
Recall	67%	59.5%	58%

From Table 8 we found that the highest overall performance is 69% at $\tau=0.05$ and recall is 67% at $\tau=0$. So from our above result we can conclude that this approach is a better prediction tool for negative documents.

To better understand our dataset we also plotted distance cumulative probability of all the 200 reviews that used for this approach. Figures 7 to 9 show results for three sample review: one correctly classified as positive (Figure 7), one is correctly classified as negative (Figure 8) and one that cannot be classified, which we call undefined (Figure 9).

For this approach subtraction and ratio scheme mentioned in section 4.1 gives same result.

Figure 7: Correctly classification of positive review.

Figure 8: Correctly classification of negative review.

Figure 9: An undefined situation.

Feature Weighting Approach:

As we mentioned previously we have 1000 (500 positive and 500 negative) reviews for training and 400 (200 positive and 200 negative) reviews for testing. We applied feature weighting technique in the 400 testing set of polarity words with proximity pattern only.

Feature Weighting for Proximity Pattern:

Following figure 10 shows the impact of feature weighting on the performance of proximity patterns with polarity words only, using $n=5$. From the 32 original patterns, we selected 12 patterns, as follows:

Predominantly positive:

P1=++++, P2=++++-, P3=+++++, P4=+++-, P5 =+-+- and P6 = -++-

Predominantly negative:

N1=+----, N2=-+++, N3=----+, N4= ---+-, N5 = ----+, and N6 = ----

Weights were then applied only to the predominantly positive patterns. The variation of the performance as a function of the applied weights is shown in Figure 10. Clearly, the weights have a significant impact on the overall performance. A similar weighting can be applied to the other features before classification. This feature weighting is important, whether we use machine learning methods, or direct unsupervised classification.

Figure 10: Impact of feature weighting on the overall performance on classification accuracy. The weight was varied from 0 to 2.5. The following notations are used in the figure: 2D: P1 vs N6; 6D: P1+P2+P3 vs N4+N5+N6; 12D: P1+P2+...+P6 vs N1+N2+...+N6. The performance peaked at around 0.33 for the weight.

5.3.3 Mean-Median Approach

Followings are results of the mean-median approach's three schemes only based on polarity words.

1. For this approach we used all the 1400 movie reviews. Figure 11 and 12 shows the mean and median view of the training dataset. Here for normalization purposes we divide P+- and N+- by 2.

Figure 11: Mean of the training set.

Figure 12: Median of the training set.

Table 9: Classification accuracy by considering mean of the training set.

Scheme	Train Set	Test Set
1a.	51.8%	50.75%
1b.	61.6%	63.25%
1c.	53.9%	58.75%

Table 10: Classification accuracy by considering median of the training set.

Scheme	Positive Train Set	Positive Test Set
1a.	50%	50.75%
1b.	58.1%	60.5%
1c.	54.6%	55.5%

Tables 9 and 10 show the results of our experiment. Our approach improves the classification accuracy than previous approach. By using mean of the training data for positive review test set best accuracy rate is 83.50% and for negative test set it is 69%. The median scheme gives 87.50% for positive test set and 71.50% for negative test set. In this scheme we also ran our classification program on the training set. For mean, best result for positive was 86.20% and 69.60% was the best result for negative review and for median 86.80% is best result for positive and 73.20% is for negative review.

2. Figures 13, 14, 15 and 16 present classification accuracy of mean and median test and train set. The best accuracy rate from this scheme are as follows:- for test set if we use mean then for positive review we get 70.83% and negative review 76% and for median 70.3% for positive review and 72% for negative review. For training set if we use mean we get 62.2% for positive review and 73.2% for negative review and for median 70.4% for positive and 68.4% accuracy rate for negative review. Overall performance of both positive and negative reviews is presented in Tables 11 and 12.

Figure 13: Classification Accuracy mean (Positive).

Figure 14: Classification Accuracy mean (Negative).

Figure 15: Classification Accuracy median (Positive).

Figure 16: Classification Accuracy median (Negative).

Table 11: Overall classification accuracy by considering mean.

Train Set	Test Set
59.2%	63.66%

Table 12: Overall classification accuracy by considering median.

Train Set	Test Set
63.1%	66%

3. This scheme improves our classification accuracy in a good margin. For positive train set highest accuracy is 95.6% and for negative train set accuracy is 76%. For positive test set we get highest 96% accuracy and for negative test set it is 76%. In this scheme we used only mean. These are very good results with significant improvement than previous research of this area mentioned in related work section. But from our research we found that the best accuracy rate does not occur at the same values for α , β , γ . For one set of values of α , β , γ positive reviews gives good result but for negative it performs very poorly and vice-versa. Overall performance of both positive and negative reviews is presented in Table 13.

Table 13: Overall classification accuracy by considering mean.

Train Set	Test Set
54%	54.5%

5.3.4 Machine Learning Approach

From our previous experiment we found that in unsupervised approach cumulative probability works better for negative review but for positive review it's performance is not good. Mean-median approach gives some very good result but problem is when one part of the result is good at same time other part is poor for example if a scheme's accuracy rate for positive review is good than this scheme's perform once is bad in negative review and vice-versa. We need an approach that performs well for both positive and negative reviews at the same time. Thus we used our word proximity features on some machine learning algorithms. We tested with a number of machine learning algorithms but following five are the primary algorithms used: Naïve Bayes, support vector machines, k-nearest neighbor, J-48 and multilayer perceptron. All the algorithms are implemented in WEKA package. Like previous methods in the literature, here we use all the 1400 movie reviews. We randomly selected 1000 for training and 400 for testing. From earlier discussion we know that each document (whether positive or negative) have three proximity distances for polarity words and six proximity distances by considering polarity and non-polarity words. We thus computed the proximity features as described earlier, and then pass the features to the machine learning algorithms.

Proximity Distributions:

Here we have considered different bins with our distance range of 100 and after that we pass it to WEKA for machine learning classification. Following table shows the performance of different machine learning algorithms for proximity distribution of 1 bin that means cumulative of distance 1-100, also combination of this in range 3 and polarity & non-polarity words. Here highest accuracy rate is 67.75% given by SVM algorithm.

Table 14: Proximity Distribution 1 Bin.

Algorithms	Proximity Distribution 1 Bins		
	Cumulative of 1-100 (Polarity Words only)	Cumulative of 1-100 & Range 3 (Polarity Words only)	Cumulative of 1-100 (Polarity & Non-Polarity Words)
SVM	67% (PUK Kernal)	67.75% (Normalize Poly Kernal)	53.75%
NB	66%	67%	53.5%
MLP	66.25%	63.25%	52.75%
KNN (K=100)	66.5%	67.25%	52
Best Result	67% (SVM)	67.75% (SVM)	53.75% (SVM)

Following table shows the performance of different machine learning algorithms for proximity distribution of 9 bins that is using range 2, also combination of range 2 and range 3 and polarity & non-polarity words. Here highest accuracy rate is 68.25% given by KNN algorithm.

Table 15: Proximity Distribution 9 Bins.

Algorithms	Proximity Distribution 9 Bins		
	Range 2 (Polarity Words only)	Range 2 & Range 3 (Polarity Words only)	Range 2 (Polarity & Non- Polarity Words)
SVM (PUK Kernal)	67%	65.5%	65.5%
NB	66.5%	63.25%	64.75%
MLP	64%	51.5%	62.25% (J48)
KNN (K=139)	68.25%	64%	63.5%
Best Result	68.25% (KNN)	65.5% (SVM)	65.5%(SVM)

Following table shows the performance of different machine learning algorithms for proximity distribution of 10 bins that is using range 1, also combination of range 1 and range 3 and polarity & non-polarity words. Here highest accuracy rate is 69.75% given by SVM algorithm.

Table 16: Proximity Distribution 10 Bins.

Algorithms	Proximity Distribution 10 Bin		
	Range 1 (Polarity Words only)	Range 1 & Range 3 (Polarity Words only)	Range 2 (Polarity and Non-Polarity Words)
SVM (PUK Kernal)	68.25%	65.75%	66%
NB	66.5%	60%	65.25%
MLP	61.5%	58.75%	60.5% (J48)
KNN (K=100)	69.5%	62.75%	63.25%
Best Result	69.5%(KNN)	65.75% (SVM)	66%(SVM)

Following table shows the performance of different machine learning algorithms for proximity distribution of 100 bins. That means here we considered all the distance point (1-100) as a bin, also combination this with range 3 and polarity & non-polarity words. Here highest accuracy rate is 63.75% given by NB algorithm. We did not perform any correlation experiment with 100 bins.

Table 17: Proximity Distribution 100 Bins.

Algorithms	Proximity Distribution 100 Bins	
	Polarity Words only	Polarity and Non-Polarity Words
SVM	53%	58.25%
NB	54.5%	63.75%
MLP	49.25%	54.5%
KNN (K=7)	50.5%	61.5%
Best Result	57% (J48)	63.75% (NB)

Table18 shows the best performance for Proximity Distributions feature.

Table 18: Best Performance.

No.	Bin	Accuracy Rate
1	1	67.75% (SVM)
2	9	68.25% (KNN)
3	10	69.5%(KNN)
4	100	63.75% (NB)

Mutual Information (MI) between Proximity Types:

Best Performance for mutual information between proximity types with polarity words only is given in the following. Detail description of this technique is available in chapter 3. Table 19 shows the result of mutual inform for different values of N from 2-100. Best performance of 68.25 was obtain at N=75.

Table 19: Mutual Information between Proximity Types.

No.	N	Best performance
1	2	59.5% (SVM/PUK)
2	5	61.25% (Bayesianlogistic Regression ,Logistic, SVM/PUK)
3	10	62.25% (Naivebayes with mutlinomial)
4	15	62.75% (Multilayerperceptron)
5	20	64.75% (SVM/PUK)
6	25	65.25% (BaysianLogisticRegression)
7	50	67.75% (Logistic)
8	75	68.25% (SVM/PUK)
9	100	66.5% (BaysianLogistic regression)

Figure 17: Mutual Information between Proximity Types

Using polarity and non-polarity words we got an accuracy rate of 68.5% with SVM, PUK kernel at N=75.

Proximity Patterns:

Performance of the proximity patterns is available in the Tables 20, 21, and 22. For N=2, best accuracy is 67.5%, N=3 it is 67% and for N=5 it is 64.75%. For proximity pattern of polarity and non-polarity words, if we found a number of consecutive non-polarity word sequence then we consider only one non-polarity word of that sequence and perform our experiments.

Table 20: Proximity Patterns (N=2).

Algorithms	Polarity Word Only	Polarity & Non-Polarity words	Polarity & Non-Polarity words (by consider one non-polarity word of a sequence)
SVM	64.75%	66.5%	67.5%
NB	63.5%	66%	66.5%
MLP	65.75%	62%	60%
KNN (K=89)	63.5%	67.25%	65.25%
Best Accuracy	64.75% (SVM)	67.25% (KNN)	67.5% SVM (PUK)

Table 21: Proximity Patterns (N=3).

Algorithms	Polarity Word Only	Polarity & Non-Polarity words	Polarity & Non-Polarity words (by consider one non-polarity word of a sequence)
SVM	66.75%	65.25%	66% (SVM/PUK)
NB	66.25%	66%	64.75%
MLP	64.25%	61%	62.25%
KNN (K=89)	64.5%	62%	65%
Best Accuracy	67% (J48)	66%(NB)	66%(SVM/PUK)

Table 22: Proximity Patterns (N=5).

Algorithms	Polarity Word Only	Polarity & Non-Polarity words	Polarity & Non-Polarity words (by consider one non-polarity word of a sequence)
SVM	64.75%	62%	61%
NB	64.25%	58.75%	54.75%
MLP	59.25%	51.75%	58.25%
KNN (K=7)	61.5%	54.25%	55.75%
Best Accuracy	64.75% (SVM)	62%(SVM)	61% SVM(PUK)

Table 23 shows best accuracy rate in proximity patterns with polarity words only.

Table 23: Best accuracy rate of the above three

No.	N	Accuracy Rate
1	2	67.5% (SVM/PUK) (Polarity & Non-Polarity words :-by consider one non-polarity word of a sequence)
2	3	67% (J48) (Polarity words)
3	5	64.75% (SVM) (Polarity words)

Fusion:

In previously mentioned result none of these go beyond 69.5%. So, to get better result we combined classification performance of our three features. Our experiment demonstrates that combining different classifier out performs the individual classifier's performance. We experimented with different classification algorithms and try to combine 3 algorithms and 5 algorithms to get good result. So, our best result using fusion is 72.5%. The results are shown in Tables 24 and 25.

Table 24: Best 3 fusion result after combining 3 algorithms

Algorithms	Accuracy Rate
10 bin (KNN) + Mutual Information (N=75,SVM)+ Proximity Pattern (N=3, J48)	72.5%
10 bin (KNN) + Mutual Information (N=50, Logistic) + Proximity Pattern (N=3, J48)	71.75%
10 bin (KNN) + Mutual Information (N=75,SVM)+ Mutual Information (N=50, Logistic)	70%

Table 25: Best 3 fusion result after combining 5 algorithms

Algorithms	Accuracy Rate
10 bin(KNN) + Mutual Information (N=75, SVM)+ Proximity Pattern (N=3, J48)+ 9 bin (logistic)+ Correlation Bin1 (NaiveBayes)	72.5%
10 bin (KNN) + Mutual Information (N=75,SVM)+ Proximity Pattern (N=3, J48)+ 9 bin (logistic)+ Mutual Information (N=50, Logistic)	72.5%
10 bin (KNN) + Mutual Information (N=75,SVM)+ 9 bin (logistic)+ Mutual Information (N=50, Logistic)	72.5%

This final result with proximity features can be compared with results from current state-of-the-art algorithms on the same or similar movie review data set; as shown in Table 26.

Table 26: Comparative performance with state-of-the-art algorithms

Algorithm Ref.	AB 08	DN 09	WY 09	ITR 08	BA 10	Proposed
Accuracy (%)	68.5%	70.9%	81%	77%	62%	72.5%

[AB 08]: (Andreevskaya A. and Bergler S., 2008).

[DN 09]: (Dasgupta S. and Ng V, 2009).

[WY 09]: (Whitehead M. and Yaeger L., (2009),

[ITR 08]: (Ikeda et. al. 2008).

[BA 10]: (Bloom K. and Argamon S, 2010).

In [WY09] accuracy rate is 81% because they used cross-domain models with lexicons that are similar to the target domain's lexicon. They build their model using lexicon similarity. In [ITR08] the authors used polarity shifting technique. They train their system with word-wise and sentence-wise learning, they used hybrid model, also experimented with a number of features like baseline, bag-of-words, simple voting, negative voting, hybrid bag-of-words etc.

Above mentioned papers used a number of features for experiment but with one proximity-based feature we achieved 72.5% accuracy rate. Combining this new feature for sentiment analysis with other standard approaches, such as part-of-speech (POS) tagging, bag-of-words (BOW)s, phrases or higher-order n-grams will no doubt lead to further improvements.

5.3.5 Different Dataset.

We also applied our approach in a drug and music review dataset. These are available in https://www.cs.indiana.edu/~mewwhite/html/opinion_mining.html. These reviews are relatively shorter than movie review dataset. Here we used our proximity pattern (N=2) feature with polarity words only. Following is the performance in Table 26 & 27.

Table 27: Drug Review.

Algorithms	N=2
SVM (PUK)	62.24%
NB	50.62%
MLP	60.99%
KNN (K=89)	60.17%
Best Accuracy	62.24%(SVM/PUK)

Table 28: Music Review.

Algorithms	N=2
SVM (PUK)	64.09%
NB	51.93%
MLP	60.22%
KNN (K=89)	63.54%
Best Accuracy	64.09%(SVM/PUK)

Chapter 6: Conclusion & Future Work

The goal of this thesis is to study a new methodology for sentiment analysis. To achieve this goal three different proximity based features:- Proximity distributions, mutual information between proximity types & proximity patterns are proposed. These proposed methods were evaluated by using sentiment classifications techniques. In chapter 3 we describe about our methodology that is proximity based analysis, our system architecture and how we perform proximity measurement. Chapter 4 focused on sentiment classification techniques that we employed in our research. We used three approaches for sentiment classification namely unsupervised approach, mean & median approach and machine learning approach. Finally in chapter 5 we showed our experimental results on the three features with different classification techniques. The empirical results show that proximity-based sentiment analysis is able to extract sentiments from a specific domain, with performance comparable to the state-of-the-art.

Our results are heavily affected by the dictionary and the stop word list that we used. Thus, for future work we suggest to generate a dictionary from the corpus and to perform analysis without using any stop word list. The results could be further improved if instead of only words we consider phrase polarity, phrase proximity, and inflexion words, such as “not”, “but”, “however”, and quantifiers such as “very”. It will be interesting to study the impact of text segmentation with blocks and overlapping on the results. Feature weighting scheme can be tested with different bins and for mutual information. Also Hidden Markov Model (HMM) can be applied for learning proximity patterns.

References:

- Abbasi, A. and Chen, H., Identification and comparison of extremist-groupWeb forum messages using authorship analysis. *IEEE Intell. Syst.* 20, 5, 67–75, 2005.
- Alison H. and Pero Subasic., Fuzzy typing for document management. In *ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes*, pages 26-27, 2000.
- Andreevskaja A. and Bergler S., When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. *ACL* 2008.
- Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G. "Output-sensitive algorithms for computing nearest-neighbor decision boundaries". *Discrete and Computational Geometry* 33 (4): 593-604, 2005.
- Bloom K. and Argamon S., Automated learning of appraisal extraction patterns. in *Corpus-linguistic applications: Current studies, new directions*, 2010.
- Burris, V., Smith, E., and Strahm, A., White supremacist networks on the Internet. *Sociol. Focus* 33, 2, 215–235, 2000.
- Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP 2005)* Vancouver, BC, Canada, (pp. 355–362), October 6–8, 2005.
- Cover, T and Thomas ,J, *Elements of Information Theory*, Wiley-Interscience, 1991.
- Crilley, K., Information warfare: New battle fields, terrorists, propaganda, and the Internet. *Aslib Proc.* 53, 7, 250–264, 2001.
- Dasgupta S, and Ng V.,. Topic-wise, Sentiment-wise, or Otherwise? Identifying the Hidden Dimension for Unsupervised Text Classification. *EMNLP* 2009.
- Dave, K., Lawrence, S., & Pennock, D. M., Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international WWW conference Budapest, Hungary*, (pp. 519–528), May 20–24, 2003.
- Eapen G., *Application of Data mining in Medical Applications*, Master's Thesis, University of Waterloo, Ontario, Canada 2004.
- Glaser, J., Dixit, J., and Green, D. P., Studying hate crime with the Internet: What makes racists advocate racial violence? *J. Social Issues* 58, 1, 177–193, 2002.
- Gamon, M. ., Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland, 23–27 August 2004, pp. 611–617, 2004.

- Hatzivassiloglou, V., & McKeown, K. R., Predicting the semantic orientation of adjectives. In Proceedings of the 8th conference on European chapter of the association for computational linguistics (pp. 174–181), 1997.
- Hearst M, Direction-based text interpretation as an information access refinement. In Paul Jacobs, editor, Text-Based Intelligent Systems. Lawrence Erlbaum Associates, 1992.
- Hiroshi, K., Tetsuya, N., & Hideo, W. , Deeper sentiment analysis using machine translation technology. In Proceedings of the 20th international conference on computational linguistics (COLING 2004) Geneva, Switzerland, (pp. 494–500), August 23–27, 2004.
- Ikeda D., Takamura H., Ratinov L-A, Okumura M., Learning to Shift the Polarity of Words for Sentiment Classification. IJCNLP 2008.
- Joachims, T., Making large-scale SVMlearning practical. In B. Schölkopf, C. J. C. Burges, & A. J. Smola (Eds.), Advances in kernel methods: Support vector learning. The MIT Press, 1998.
- Kessler, B., Nunberg, G., & Schutze, H., Automatic detection of text genre. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (pp. 32–38). Somerset, NJ: Association for Computational Linguistics, 1997.
- Nazzal, J., El-Emary, I. & Najim, S., “Multilayer Perceptron Neural Network (MLPs) For Analyzing the Properties of Jordan Oil Shale” , World Applied Sciences Journal 5 (5): 546-552, ISSN 1818-4952, 2008.
- Pang, B., Lee, L., & Vaithyanathan, S., Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the conference on empirical methods in natural language processing (EMNLP 2002) Philadelphia, PA, USA, (pp. 79–86), July 6–7, 2002.
- Pang, B., & Lee, L., A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting of the Association for Computational Linguistics (ACL) Barcelona, Spain, (pp. 271–278), July 21–26, 2004.
- Pang, B., & Lee, L., Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL) (pp. 115–124). USA: University of Michigan, June 25–30, 2005.
- Peng F.. Language Independent Text Learning with Statistical n-gram Language Models. PhD thesis, University of Waterloo, 2003.
- Pop, I., An approach of the Naive Bayes classifier for document classification, General Mathematics, Vol. 14, No. 4, pp.135–138, 2006.

- Richard M. Tong., An operational system for detecting and tracking opinions in on-line discussion. Workshop note, SIGIR 2001 Workshop on Operational Text Classification, 2001.
- Rudy Prabowo and Mike Thelwal. Sentiment analysis: a combined approach. *Journal of Informetrics* 2009.
- Sack, W., On the computation of point of view. In *Proc. of the Twelfth AAAI*, page 1488. Student abstract, 1994.
- Schafer, J., Spinning the web of hate: Web-based hate propagation by extremist organizations. *J. Criminal Just. Popular Culture* 9, 2, 69–88, 2002.
- Stone P., Dunphy D., Smith M., Ogilvie D., and associates. "The General Inquirer: A Computer Approach to Content Analysis," The MIT Press, 1966.
- Suzuki Jun, Kernels for structured data in natural language processing. PhD Thesis, Nara Institute of Science and Technology, 2005.
- Turney, P. D. , 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 (ACL)* Philadelphia, PA, USA, (pp. 417–424), July 6–12, 2002.
- Weiss O., and Herzel H., Correlations in protein sequences and property codes, *Journal Theoretical Biology*, 190: 341—353, 1998.
- Whitehead M. and Yaeger L., Building a General Purpose Cross-Domain Sentiment Mining Model, *CSIE*, 2009
- Wilson, T., Wiebe, J., & Hoffmann, P., Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceeding of the conference on empirical methods in natural language processing (EMNLP 2005)* Vancouver, BC, Canada, (pp. 347–354), October 6–8, 2005.
- Zhou, Y., Reid, E., Qin, J., Chen, H., and Lai, G., U.S. extremist groups on theWeb: Link and content analysis. *IEEE Intell. Syst.* 20, 5, 44–51, 2005.