



Comparing Feature Sets and Classifiers for Sentiment Analysis of Opinionated Free Text

Master Thesis by: Halleh Khoshnevis
Student Number: 4116402
Research Group: Man-Machine Interaction
Group Leader: Prof. dr. C. M. Jonker
Supervisor: Dr. ir. P. Wiggers
Period: September 2011 -July 2012

Contents

Abstract	4
Introduction	5
Chapter 1: Methods and Approaches	9
1.1 What is Sentiment Analysis?	9
1.2 Document Level Sentiment Analysis vs. Sentence Level Sentiment Analysis ..	11
1.3 Why Sentiment Analysis is Arduous	12
1.4 Opinion Words Generation	13
1.5 Document Level Sentiment Classification	14
1.6 Knowledge Transfer and domain Adaptation	19
1.7 Sentence Level Sentiment Classification	21
1.8 Conclusion	26
Chapter 2: Goal and Survey Hypothesis	28
2.1 Main Hypothesis	28
2.2 Subordinate Hypothesis	29
Chapter 3: Data and tools	30
3.1 Dataset	30
3.2 Classification	32
3.3 Density Estimation Classification	32
3.3.1 Parametric Approach	32
3.3.2 Non-Parametric Approach	34
3.4 Structural Risk Minimization Approach	35
3.4.1 Support Vector Classifier	35
3.5 PRTools	37
Chapter 4: Model and Features	39
4.1 Common Features	41
4.1.1 Bag of Words	41
4.1.2 Dictionary	42
4.1.3 Bag of words with Part of Speech Tags	44
4.2 Selected Features	46
4.2.1 Pointwise Mutual Information Based on Positive and Negative Labels	47
4.2.2 Pointwise Mutual Information Based on Positive and Negative Labels &	
“good” and “bad” terms	49
4.2.3 Pointwise Mutual Information Based on Positive and Negative Labels &	
Adjective and Negation Part of Speech Tags	50
4.3 Feature Extraction by PCA	51
4.4 Cross Validation	53
Chapter 5: Experiments and Evaluations	54

5.1 Bag of Words	54
5.2 Bag of words with Part of Speech Tags	58
5.3 Selected Features	60
Chapter 6: Conclusion and Discussion	68
6.1 Conclusion	68
6.2 Future Work	67
Acknowledgement	71
Appendix	72
References	81

Comparing Feature Sets and Classifiers for Sentiment Analysis of Opinionated Free Text

Author: Halleh Khoshnevis

Student Number: 4116402

Email: halleh.khoshnevis@gmail.com

Abstract

This master thesis is about the sentiment analysis of the societal theme documents and categorizing them in positive or negative groups. The application of this thesis can be widely used in review blogs, public polls and etc. In this study, we have compared different feature sets as well as different classifiers on datasets of opinionated texts with societal themes. These datasets consist of one large and 6 small sets in terms of number of documents. By considering the often used “Bag of Words” feature set as the base line we have tested 4 other models and came to this conclusion that selecting features with their part of speech tags can always improve the results of sentiment classification while adjective and negation tags can describe the opinionated documents more informatively in much smaller matrices which saves a lot of memory and processing time. Moreover, by selecting these tags according to their PMI ranks in positive and negative labeled documents, we obtained the most informative sentimental words. On the other hand, based on the obtained results, in contrast with the predominant attitude in the sentiment analysis field that support vector machine (SVC) is the best classifier for binary classification of opinionated documents, we found the linear discriminate classifier (LDC) can perform as well as support vector machine but 10 times faster. The consumed time is convincing enough to substitute SVC with LDC in sentiment analysis when we have a large number of features as is the case in a Bag of Words model due to the fact that the time that SVC needs for predicting labels is quadratic in terms of number of documents. The feature vectors obtained through PMI analysis are relatively small, we found that as a consequence, the k-nearest Neighbor Classifier (KNNC) could train well and gave the most accurate results in comparison with LDC and SVC in both large and small datasets. It should be stressed that Principal Component Analysis (PCA) is used in this study in order to extract mathematically the most common features in all the models.

Graduation Committee:

Prof. dr. C. M. Jonker, Faculty EEMCS (WIS), TU Delft

Dr. ir. P. Wiggers, Faculty EEMCS, TU Delft

Ir. H.J.A.M. Geers, Faculty EEMCS, TU Delft

Introduction

Consider the situation that President Obama has taken into account the necessity of a Health Insurance Reform and called on Congress to hold a final up or down vote on it.

As of 2011, the legislation remains controversial in the US due to the fact that some states are of the belief that the reform eventually does not benefit everyone. Therefore the opposition decides to ask people about their preference. In order to achieve this goal, the opposition provides people with an official review blog and asks them to take part in a public poll and write about their attitudes toward the health insurance reform.

The number of comments in the blog is excessively growing which brings about a huge amount of documents to be discussed. Now it is the time to extract the opinion of people towards the health insurance reform.

Usually in review blogs people do not write directly about their point of view, rather they try to give examples or write about the pros and cons of the subject and reflect their way of thinking in natural language. For instance one of the reviews about the health insurance reform is:

“This put the government in the role of arbiter of what practice is healthy and unhealthy. But I believe that U.S health care costs more because Americans are less healthy.”

According to this review, it can be concluded that the author is not satisfied by the current health insurance and therefore he/she is the supporter of the reform.

Thus, the opposition turns to the question what is the best possible way of recognizing the sentiment behind documents. That is to say how the opposition can find out if individuals are against the reform or not.

It would be a safe assumption to say that the most accurate results are obtained when the sentiment analysis is done manually, but this will undoubtedly consume too much time because the number of documents to be discussed is high. Therefore this process should be done automatically.

Generally speaking, these kinds of processes are settled in the Natural Language Processing and more precisely in the Sentiment Analysis field. Realizing the dominant opinion of a document and granting labels such as Positive or Negative to it, is done in the Sentiment Analysis area, with the aid of Classifiers.

The term Classifier refers to a mathematical function that maps input data to a category. In other words, the documents (input) are explained in the terms of variables which are called Features and the category (label) of each document is predicted based of the classes.

In Document Level Sentiment Analysis, it is widely belief that the Linear Support Vector machine is of the most accurate classifier especially in the 2-Class classification and therefore it is broadly used in this area. It would not be too far off to say that SVM is of the significant importance in the 2-Class classification, but total reliance on that seems not always correct. The amount of time which SVM needs to predict the label of each document is much more than the time other classifiers spend while the accuracy does not differ a lot.

Therefore, the opposition is left with no other choice but to think that what the best features are to describe opinionated documents, in addition which classifier is sufficiently good and fast for this purpose.

Main Goal:

The example above indicates one of the applications of this thesis research in the real world. Existing research on the analysis of sentiment has been done mostly on product or movie reviews and there are very few studies concentrated on societal themes. It is merely because of the difference between the nature of these documents. In product or movie review, the author mentions his/her opinions toward the whole product or movie in very simple sentences or gives his/hers on divided attention to eminent properties of them in short sentences. Sarcasm and complicated grammatical sentences are not usually seen in these kinds of documents. In contrast, in societal theme documents, the topic is very general and the author expresses his/her attitudes in long sentences with complicated grammar. Due to the generality of the discussed topic, authors usually do not concentrate only on the prominent points but also talk about the pros and cons of it or their personal experience related to subject. Below examples have been selected from Product reviews (Alarm Clock) and a social blog (God):

1) Alarm Clock:

The i-Home is a great product for so many reasons.

1.) It is very small and fairly light.

2.) Day Light Saving setting

3.) Great sound quality for the price!

4.) Easy to set up

The Sound:

The sound is great for what you are paying for. The speakers can be heard from my room on the third floor in the kitchen on the first. The i-Home has the great 3D sound setting and Reson-8 speaker chambers with a bass export on the back.

Overall:

I would recommend this product to anyone looking for one of the best alarm clocks on the market (especially for the money)!

2) God:

I wonder what you folks think about Angels and the Devil. Do you believe in them as well as God? Also if you believe God created us out of Love... was there something lacking in the Angels as in God did not see them sufficient enough to Love him freely? I mean why create

Humans if you have Celestial Beings (Angels / Demons) right there in Heaven who you created first who can choose to Love you or Diss you? Heck God could have made some Angels in his own Image if he wanted could he have not?

As it is obvious from the examples above, there is a wide difference between product or movie reviews and societal theme documents. The difficult nature of the latter makes that sentiment analysis of societal themes has not been given much attention by researches so far while in this thesis we attempt to take a much closer look at this issue.

Another point which is noteworthy to mention is that different studies on sentiment analysis use different feature sets that are not compared on the same data. While there are several kinds of classifiers in pattern recognition which can perform 2-class classification task (Positive/Negative Classification), most of the Document Level Sentiment Analyses has been done by Support Vector Machine (SVM). Although in general SVM achieves good accuracy as well as copes well with high dimensional data and is less prone to overtraining than other classifiers, it is time consuming and therefore not appropriate for a huge amount of data like the situation described above. On the other hand, the time can be easily influenced by the size of documents and therefore the “Time Issue” takes the precedence and the priority goes to the finding a fast and accurate classifier. In pursuance of fulfilling this objective, one is left with no other choice but to think that what the best substitute for “Linear Support Vector Machine” is. Given the fact that natural language documents have special characteristics and that subtle choice of words or grammar may completely change the sentiment of words, what is the best mathematical function that describes the behavior of such documents.

To wrap up, there are several important issues in sentiment analysis which should be investigated more to overcome them and try to overcome the lack of information in these areas. Therefore, this thesis research has been conducted and we hope that at the end of this study, we can come up with the answers of the research questions below:

- What is the best set of features to be extracted from a dataset with a large number of documents?
- What is the best set of features to be extracted from a dataset with a small number of documents?
- Which classifier performs better on a large opinion dataset in terms of time and accuracy?
- Which classifier performs better on a small opinion dataset in terms of time and accuracy?
- What is the overall performance of sentiment analysis on documents that contain opinions on societal themes?

This study is organized as follow:

In Chapter 1, Methods and Approaches, the state of the art in Sentiment Analysis and the necessity of it in the communication world is studied. Moreover, Document Level Sentiment Analysis along with its methods and approaches is discussed comprehensively.

Chapter 2 gives a more detailed discussion on the Goal and Hypotheses in this thesis. Main hypotheses, subordinate hypotheses and the relation between independent and dependent variables will be discussed in details.

In Chapter 3, Data and Tools are given as well as Classifiers which are surveyed in detail.

In Chapter 4, Model and Features, not only the model is discussed but also the possible feature sets are explored. Feature Reduction and Evaluation methods are also investigated in this chapter.

The Evaluation is described in Chapter 5 which concentrates on evaluation of each classifier on different types of document representations and finally Conclusion and Discussion based on the current conducted research are given in the Chapter 6.

Chapter 1: Methods and Approaches

1.1 What is Sentiment Analysis?

Sentiment Analysis aims to determine the attitude of a writer with respect to some topics or the overall sentiment polarity of a document, such as positive or negative. This attitude can be his or her judgments or evaluations or any other emotional perceptions.

With the growing availability of opinion resources such as blogs or review sites, the challenge for seeking out the opinions of others has increased as well. Exploiting the traditional information retrieval methods has not been accurate enough to calm the sudden eruption of activity in the area of opinion mining and sentiment analysis. Researchers have come up with some interesting approaches so far [4, 5, 6, 7, 10], but work is still in progress.

Recently, people can post their reviews of products on merchant sites such as “www.amazon.com” or express their opinions toward everything in the internet discussion groups or blogs. Now, if one wants to buy a product, (s)he is no longer limited to ask his or her friends; but can find different opinions existing in product reviews in terms of a special good. Companies can find consumer opinions about their products without conducting surveys. Travelers can find their suitable hotels according to the rates or opinions corresponded to the different aspects of hotels in tourism sites.

Most of the time, the opinions and sentiments on a topic are hidden in long reviews or blogs. So, it is difficult for a reader to go through a huge amount of documents to mine attitudes to a topic or different angles to a topic. Therefore, the task should be automated and computers should be involved in this procedure. According to the nature of the sentiment analysis, traditional text mining approaches (based on topic) are not enough to satisfy our needs (for more details refer to [16 section 3.2]). The computational study of opinions, sentiments and emotions expressed in texts are the challenges of natural language processing. It has been shown that a combination of Information Retrieval and Natural Language Processing approaches can be helpful in terms of opinion mining and sentiment analysis [6].

Before going further, it is better to warm up with some definitions:

Subjective Sentence:

Is a sentence in which the writer expresses his or her feelings or sentiments toward entities, events and their properties.

For example: “I like swimming.”

Objective Sentence:

Is a factual sentence about entities, events and their properties.

For example: “The schedule contains swimming, diving and ... “

Opinion:

Is a belief or judgment based on special knowledge towards a topic. Opinions are sometimes expressed explicitly like: “The voice quality of this phone is amazing.” But sometimes they are hidden in the sentiment of a sentence, for instance: “The earphone broke in two days”. Since the concept of opinion is very wide, sentiment classification mostly concentrates on the general feeling expressed by opinions (Positive / Negative). In fact, positivity or negativity is determining the *Polarity* of an opinion. In other words, one of the main subtasks of sentiment analysis is determining the polarity of documents or in more details, determining the polarity of each subjective sentence in a document.

Opinion words:

Are words that are commonly used to express positive or negative sentiment.

For example:

{beautiful, pretty, love} → positive sentiment

{ugly, awful, hate} → negative sentiment

Opinion Orientation (Polarity):

Indicates whether the expressed opinion by opinion words is positive, negative or neutral.

For example: “The camera takes wonderful pictures “ → Positive

Opinion Sentence:

Is a sentence which contains one or more opinion words.

For example: “The story was amazing as was the playing of actors. “

Object / Features:

So far, we have used “topic” to refer the main subject in reviews which is going to be discussed. Hence forth, we call it “Object”. In opinionated documents, objects and their components or attributes are going to be reviewed and sentiments toward them are expressed in terms of “opinion words”; these components or attributes are called “object-features”.

For Example:

The voice quality of the phone is good.

Object : phone

(Explicit)Object – Feature : voice quality
Opinion Word : good

In this example the Explicit Feature is voice quality, but sometimes object-feature should be inferred from the sentence. This kind of feature is called “Implicit Feature”.

For Example:

The phone is too large.
Object : phone
(Implicit)Feature : size
Opinion Word : large

Classifier:

Is a function to classify different objects and label them as an output. In sentiment analysis, classifiers are used to determine the polarity of a subjective sentence with respect to the topic.

There are two types of classification:

Supervised Classification, such that a classifier is inferred from training set. The classifier should predict the correct label (positive or negative) for any valid input object. In contrast *unsupervised classification* infers the hidden structure of raw data. In sentiment analysis both classification types are widely used [5, 8].

The main task of Sentiment Analysis is extracting suitable features and constructing an engineered feature vector as an input for classifiers^[*1].

1.2 Document-Level Sentiment Classification vs. Sentence-Level Sentiment Classification:

In the first step of opinion mining, the analyzer gets a huge amount of documents on a special topic which exists in different review blogs. These documents should go through a classification task. From an analytical point of view, the classification task is divided into two groups:

Document Level Sentiment Classification: classifying an opinionated document as expressing a positive or negative opinion [4,5,6] and *Sentence Level Sentiment Classification*: classifying a sentence as subjective or objective and for a subjective sentence, classifying it as expressing positive, negative or natural opinion [10,11,12]. Each of these classifications is going to be discussed in Chapter 3 and Chapter 4 in detail.

Document level sentiment classification is often used to give a dominant sentiment towards a topic as a whole. It does not take into account the different angles of the object. For example in the

[*1]: Notice that here term “feature” refers to a data attribute in *Machine Learning*, and should not be confused by feature of an object in subjective documents. In case of confusion between these two terms, we have used object-feature verses feature.

context of movie reviews, when the document is positive it means that the writer likes the movie in general, but if he or she concentrates on different aspects such as the story or the playing of actors, maybe the opinions are negative. In this case, if each feature and related opinion is extracted from each subjective sentence, the result is going to be a Sentence-Level sentiment classification. Sentence-level sentiment classification is required in order to obtain an accurate analysis of an object; for example in terms of products, such a precise analysis is necessary in order to make product improvements by distinguishing between what features (components or attributes) of a product are liked and disliked by consumers. Such information is not obtained by Document-Level sentiment classification.

1.3 Why Sentiment Analysis is Arduous...

Sentiment analysis is a difficult task for computers to perform. Identifying some patterns is hard for machines or even impossible while it is easy for human beings. Below you can find some intractable situations for computers:

Consider *ironies* or *sarcasm*, it is difficult to understand that the opposite meaning of a sentence is required. Sometimes ironies can be identified by special punctuation marks such as (!!!) but it is not that common to be a rule or sign for these types of expressions.

Pronoun resolution is another effortful job. Although there are some algorithms that can solve it [18], it is still demanding task in sentiment analysis. For example there are opinion words in a sentence but because the corresponding feature is a pro-noun, it is not easy to find which feature is expressed by those sentiment words.

For example:

This aspect is really good for this MP4 player in comparison with other players.

object: MP4 player

feature: this aspect →

It is a pronoun and should be resolute. For instance in this case should be "sound quality".

opinion word: good

Determining the *Strength* of an opinion also should be recognized as a demanding task in this area. Opinions have different strengths [19]. Some of them are very strong: "This phone is a piece of junk" and some of them are weak: "I think this phone is fine". Although it is possible to create a seed list of weak or strong opinion words depending on the application need [20], it is still not doable for computers when the strength of opinion mixes with the position of that opinion and changes the polarity of the document completely.

For example:

"The story was superficial, actors played awful, sound quality was terrific, but I liked it"

- ➔ Has positive polarity in the context of movie review although it has lots of negative opinion words.

Another difficulty is finding *implicit features*, in order to recognize the expressed sentiment. Although there are some studies which have investigated on recognizing Implicit Sentiment of texts [21], most of the time, explicit features are taken into account to perform opinion mining tasks [10, 11, 12].

So far, we have discussed about sentiment analysis in abstract. Now we want to focus on the classification task from a practical point of view. First, we are going to look at document level sentiment analysis: approaches, classifiers, features and other related sub-tasks and then focus on Sentence-Level sentiment classification. But before going further we have to know about opinion words and their generation in detail as these is an important component in sentiment analysis.

1.4 Opinion Words Generation:

In this subchapter we will discuss one of the main sub-tasks of sentiment analysis which is Opinion Words Generation.

As mentioned before, opinion words are used to express the sentiment of a document or a sentence. Therefore, they should have sentiment orientation as well in order to demonstrate whether a document or a sentence is positive or negative.

Opinion words, themselves, are divided into two types: *Base-Type*, such as beautiful, comfortable, quiet, and *Comparative-Type* which consists of comparative or superlative adjectives such as better or best. Base-Type opinion words express their polarities as positivity and negativity while Comparative-Type is just used to compare two or more objects together and emits no sentiment polarity.

There are three approaches which collect opinion words and their sentiment polarity:

- 1 Manual Method
- 2 Dictionary-Base Method
- 3 Corpus-Base Method

We are going to discuss each of them in detail.

Manual Method:

In this approach each opinion word such as nice (Adjective), fast (Adverb) love (Verb) is selected manually and the corresponding polarity is assigned. This approach is very time consuming [22, 23] and therefore it hardly ever used alone.

Dictionary-Base Method:

This approach has three steps. In the first step, a seed list of opinion words with their sentiment orientations is constructed manually. Then, in the second step, the seed list is grown by searching for synonyms and antonyms of seed words in an online dictionary such as WordNet[24]. The search results are added to the seed list with the same polarity as their synonyms in the list or the opposite polarity of their existing antonyms, and the seeking process is started again until no new word is found in the dictionary. In the third step, a correction process is done manually to remove any existent errors [25]. By using Machine Learning techniques and using additional information in

WordNet such as “hyponym”^[*1], it is possible to generate better and richer opinion words lists [26, 27, 28].

The most important drawback of this simple approach is that it is unable to distinguish between opinion words with respect to their domains. For example “quiet” is expressing positive sentiment in the context of a car but a negative sentiment for a speakerphone.

Corpus-Base Method:

This method is intended to solve the problem of the dictionary-Base approach. In [29], the proposed method consists of 2 steps. The first step is constructing a seed list of opinion words which have Adjective part of speech tags, and their polarities. In the second step, a set of linguistic constraints is introduced to find additional opinion words from the existing corpus as well as their sentiment orientations.

These linguistic constraints are based on the idea of “Sentiment consistency”. According to sentiment consistency, people usually express the same opinion on both sides of conjunctions (for instance AND) and the opposite opinion around disjunctions (for instance BUT). This idea helps to find new opinion words in a corpus. For example in the sentence “This car is beautiful and spacious “if we do not have “spacious” in our seed list, we can conclude from “beautiful” and conjunction (AND) that “spacious” has the same polarity as “beautiful”. Therefore, we can extend our list.

In practice, sentiment consistency is not observed, so in order to gain better results, a “Log-Linear Model” is applied to a large corpus to determine if two conjoined adjectives are of the same or different orientation. Same and different orientation links between adjectives form a graph. Finally, clustering is performed on the graph to produce two sets of words: positive or negative.

However, the corpus-based approach alone is not sufficient to cover all the opinion words and is not as effective as the dictionary-based approach.

In conclusion, we might expect that using the combination of dictionary-base and corpus-based approaches can give acceptable results for performing better sentiment analysis.

1.5 Document Level Sentiment Classification:

In this chapter we are going to present different approaches of categorizing reviews into 2 groups: Positive and Negative, with respect to their strong points and drawbacks, for Document-Level Sentiment Analysis. Existing supervised learning methods can be applied such as Naïve Bayesian or Support Vector Machine (SVM). One of the bottlenecks of using supervised learning is the manual effort involved in annotating a large number of training examples. To save the manual labeling effort, a bootstrapping approach is reported in [30, 31] to label training data automatically. The algorithm works by first using two classifiers to automatically identify objective and subjective sentences. These classifiers use a list of lexicon items that are good for expressing subjectivity. These classifiers classify a sentence as subjective if it contains two or more strong subjective clues and as an objective sentence if there is no subjective indicator. The extracted sentences are then added to the training data to learn patterns. A set of syntactic templates are provided to restrict the kinds of patterns to be

[*1]: **hyponym** denotes a subcategory of a more general class: “Chair” and “table” are hyponyms of “furniture”.

learned.

Some examples of these syntactic patterns are shown below:

Table 1.1 Syntactic patterns to be extracted from documents.

Syntactic Template	Example Pattern
<subject>passive-verb	<subject> was satisfied
<subject>active verb	<subject> complained
Active-verb<object>	Endorsed <object>
Noun auxiliary-verb<object>	Fact is <object>
Passive-verb preposition<noun>	Was worried about <noun>

Review sites have a huge amount of documents. Mining these documents in order to extract expressed sentiments toward a subject is not possible unless categorizing them according to the topics and objects, due to the fact that not all parts of reviews are informative in terms of expressing attitudes of a writer, extracting subjective statements is necessary and helpful to reduce redundant efforts to analyze a review. [2] tries to find opinion statements related to the topic in blog comments and reviews. The proposed detection system introduced Indri Queries which are used to find related passages among documents. The documents are parsed with the Minipar parser^[*1] and part of speech tagging is applied to the titles, descriptions and narrative terms. Meanwhile, the dictionary is checked to find positive and negative verbs, adjectives and etc.

Indri Queries are:

- 1) Title
- 2) Title, nouns and adjectives from the description
- 3) Title, sentiment words
- 4) Title, nouns, adjectives from the description and sentiment words from the dictionary

After removal of duplicated passages from the retrieved documents and, the authors come up with several opinionated passages related to the topic.

Also in [3], the authors have employed an automatic method to recognize subjective collocations. First they identified subjective elements which are responsible for changing the sentence to a subjective sentence and then in order to find collocations, feature extraction was applied such that 1-gram, 2-gram, 3-gram, 4-gram features were extracted and the precision of each group was calculated. The precision of an n-gram is the number of subjective instances of that n-gram divided by the total number of instances of that n-gram.

Finally subjective collocations were selected based on 2 criteria of their precisions.

For example for 3-gram collocation:

$$Pc(w1, w2, w3) > 1$$

$$Pc(w1, w2, w3) > \max(Pc((w1, w2), Pc(w3)) \text{ or } \max(Pc(w1, w3)), (Pc(w2))$$

Where $w1$, $w2$ and $w3$ are words and Pc is the precision.

[*1]: MINIPAR is a broad-coverage parser for the English language. An evaluation with the SUSANNE corpus shows that MINIPAR achieves about 88% precision and 80% recall with respect to dependency relationships. MINIPAR is very efficient, on a Pentium II 300 with 128MB memory; it parses about 300 words per second. An executable version (Linux, Solaris or Windows 95/98) of MINIPAR is available free of charge for non-commercial use on the [Minipar Homepage](#). For more information, refer to [18].

According to the fact that there are hundreds of review sites which reflect attitudes of people in terms of movies or products, most of the sentiment analysis studies are done on datasets given by these sites. Since these datasets have large amounts of reviews on a product or a movie, the obtained output of analyzing them can be counted as accepted results. For instance in [1], product reviews are categorized into positive (recommended) and negative (not recommended) groups. A Support Vector Machine (SVM) is employed as the classifier to extract various text features such as “unigrams”(bag of individual words) “selected words” (only words with Verb, Adjective and Adverb part of speech tags), “words labeled with their part of speech” (all individual words with their part of speech tags) and combinations of them where unigram was the baseline. The accuracy of each combination is reported as well. Meanwhile, it has taken syntactic and semantic processing into consideration and since the accuracy was not good in comparison with other results given by SVM classifier (the accuracy of SVM is usually more than 85% on topical text classification), the authors have analyzed the errors and found 5 types of problems which caused errors in sentiment analysis, below 3 of them are mentioned:

1) Negation phrase

For example: “I would **never regret** purchasing it”

$\begin{cases} \text{never} \\ \text{regret} \end{cases} \rightarrow \text{are two negative words while the overall sentiment of the sentence is positive}$

2) Comments on parts:

Comments negatively on parts of a product which he or she has satisfied with the product as a whole. For example: “The best phone I’ve had. The **ONLY** bad point is that.... “

3) Need for inference:

Sometimes the comment has no apparent positive or negative opinion words and inference is needed in order to identify the sentiment polarity. For Example: “If the price dropped, the company would be surprised how it would sell”

Negation phrases have a major impact on the effectiveness of unigram features extracted by the SVM classifier. For example in the sentence: “I would never regret purchasing it”, the overall sentiment is positive while “never” and “regret”, two negative words, guide to negative sentiment in unigram-approach classifiers. These kinds of phrases are a common error in subjectivity classification. In order to solve this error, the authors have inferred patterns which show the negation in the passage and if a pattern exists in the sentence, it is applied in the feature vector as “negation phrase”. By exploiting these patterns, the resulting accuracy improved from 74% to 79%. Each negation and its adjacent words are combined to generate a new composite term (i.e. negation phrase) according to the patterns below:

< Verb > - < Negative Particle > - <Verb>

< Verb > - < Negative Particle > - <Adverb> - <Adjective >

Other studies consider that dealing with negation phrases improves the sentiment analysis’ accuracy. For instance in [4], the authors not only took advantage of negation phrases but also used SVM classifier. In addition, they have used probabilistic classifiers (Max Entropy & Naïve Bayesian) to

perform the same task. The features which have been extracted were unigrams (with negation tagging), bigrams, adjectives, part of speech tags (POS) + unigrams, position^[*1]+ Unigrams. First they have experimented with term frequencies^[*2] which led to low performance while Max-entropy could not use term frequency and needed term presence. So, they applied the term presence technique and found out that when features are independent, Naïve Bayesian performs well and when features are dependent, Max-entropy performs better than Naïve Bayesian. In general, SVM with unigram features performs better than other classifiers, while Naïve Bayesian performs worst in the same case.

According to position features, the results do not differ very much in comparison with only unigrams. Position could be a useful feature if it has been investigated more while only adjective features are not useful and part of speech tagging features (POS) can improve accuracy a little in combination with unigram feature.

It is possible not to be completely black and white but gray. In other words, it is possible to express how much a document is negative and how much it is positive, as it is shown in [5]. The authors have tried to label reviews based on 4 rates: 0, 1, 2, 3, such that 0 shows the least positive and 3 shows the most positive reviews, instead of Thumbs up or thumbs down, like [4]. They have applied two algorithms (one vs. all & regression) based on SVM classifiers.

After labeling the *items*^[*3] according to the above methods, *Metric Labeling* is used as similarity measure. In *Metric Labeling* all the items with different labels are demonstrated in 2 dimensional coordination and k similar neighbors to the target item are recognized and tried to minimize the sum of distances among labels. In other words, it tries to normalize the labels of the similar items. In [5], instead of using each item(x) as vector (x), they used:

$$\overrightarrow{\text{psp}(x)} = (\text{psp}(x), 1 - \text{psp}(x))$$

Where “psp(x) = number of positive sentences / all the subjective sentences” and the cosine similarity function between two items x, y are the cosine similarity between $\overrightarrow{\text{psp}(x)}$ & $\overrightarrow{\text{psp}(y)}$.

It is shown in [5] that the proposed meta-algorithm improves multi-classes and regression versions of SVMs when the novel mentioned similarity measure is exploited.

As was mentioned before, using topic-based text classification, which classifies documents into predefined topic classes, is not useful for sentiment classification since in this kind of classification, topic-related words are important and the majority of them in each document make the document selectable while in sentiment classification, topic-related words are unimportant and instead the opinion words which express the sentiment are more important. However, it is shown in [6] that

[*1]: Each word is tagged according to whether it appeared in the first Quarter, last quarter or middle half of the document. There is an intuition that the position of the word can express the overall sentiment in the document especially in Movie reviews which might begin with an overall sentiment statement, proceed with a plot discussion and conclude by summarizing the author's views.

[*2]: Term Presence vs. Term Frequency:

In the feature extraction procedure, a document is expressed as a feature vector such that each feature corresponds to an individual term in the document. In topic-base information retrieval, the number of word's occurrences is inserted as a corresponding entry in the feature vector while in sentiment analysis the entries indicate whether a term occurs (value 1) or not (value 0). In [4] it is shown that term presence obtains better performance in comparison with term frequency, in the context of sentiment analysis. That is obvious due to the fact that in topic-base classification occurrences of certain keywords can emphasize on special topic while the presence of opinion words expresses the sentiment of the document.

[*3]: Here, “items” refer to documents.

using information retrieval techniques in constructing features vector can lead to accurate classifiers. The authors have presented a new approach to classify product reviews to 2 groups: positive, negative. In this approach they have exploited information retrieval techniques and some other methods to extract features. Some of the extracted features are useful and improved the accuracy while some of them did not worked well although they imposed lots of cost to achieve. The extracted features can be classified as follows:

1) Metadata and statistical substitution:

Replacing numerical tokens with numbers or with –productName and low frequency words with –Unique. Also in terms of names which seem to accrue only in certain product categories, they used “–productTypeName”.

This feature was mostly effective.

2) Linguistic substitution:

At this step the authors have parsed the documents and obtained part of speech tag of each word and the relationships between parts of sentences (i.e. triples of: words (POS), Relation, word(POS))

This feature did not work well and improved the performance.

3) Language base modification:

- Stemming
- Negating phrase

These did not work well either.

4) N-grams and proximity:

- Trigrams
- Bigrams

Trigrams worked better that bigrams.

5) Substrings (variable-length feature):

Identifying arbitrary length substrings that provide optimal classification by building a tree of substrings up to a cutoff length from the words in the subjective sentences. It is somehow similar to the n-gram feature while n is arbitrary^[*1].

Performed the best.

After feature extraction, in order to normalize the count of features, the authors set the upper bound and lower bound which improves the relevance of the remaining features and reduced the amount of computations.

Smoothing is also applied to the count of extracted features in order to smooth the probability of features. In this part *Laplace smoothing* [46] performed the best with average accuracy of 83.5% while *Good-Turing* [47] performed the least with average 78.4%.

After smoothing, each feature has been given score and the summation of the scores placed the review in positive or negative category.

[*1]: The paper does not give details.

These scores have a sign (+, -) and are formulated as follow:

$$\text{score}(f_i) = [p(f_i | C) - p(f_i | C')] / [p(f_i | C) + p(f_i | C')]$$

Such that f_i : each feature, C : Set of positive reviews and C' : Set of negative reviews

The described approach outperformed traditional Machine learning techniques such as SVM. Especially “variable-length features” and “metadata substitutions” were helpful.

1.6 Knowledge Transfer & Domain Adaptation:

One of the objections in the field of Supervised Classification, is that sentiment classification depends highly on the domain in which classifiers are trained and the knowledge learned from one domain or features extracted from one training set in a specific domain, cannot be used in other domains and most of the time, they perform poor in those occasions, although both are classification tasks. The domain sensitivity can be worse when for example opinion words in one domain express completely opposite sentiment in another domain such as “unpredictable in movie reviews vs. car reviews”. In order to deal with this problem *Knowledge Transfer* techniques and *Domain Adaptation* have been proposed [34, 35].

One of the simplest approaches of knowledge transfer is using different corpora such as movie reviews, product reviews and social blogs and extracting unigram features which have been highly ranked in all domains in terms of occurrence. These features are considered as domain-independent features and the rest of the features in the training data are then removed. For example in the movie review, the term “film” has high occurrence so it is not domain-independent, neither useful for classifying the opinions. In other words, domain-dependent high frequent features should be removed from the vocabulary in knowledge transfer.

Also, it is possible to generate general opinion words [32, 33]. One of the existing approaches is using labeled data from one corpus and unlabeled data from other corpora and tries to extract those opinion words which have the highest occurrence in all corpora of different domains.

As we have mentioned before, unsupervised classification is also used in the sentiment analysis field. [7] has proposed a simple approach to classify documents as positive or negative based on unsupervised classification.

The authors took into account the point that adjectives and adverbs are opinion indicators and phrases which contain these two part of speech tags (POS) are good candidates to express the subjectivity and sentiment of a sentence or document. The proposed approach has 3 steps:

- 1) Extract all adjectives and adverbs with correspond words which satisfy the patterns bellow:

Table 1.2 Patterns of tags for extracting two-word phrases from reviews.

First Word	Second Word	Third Word (Not Extracted)
Adjective	Noun	Anything
Adverb	Adjective	Not Noun
Adjective	Adjective	Not Noun
Noun	Adjective	Not Noun
Adverb	Verb	Anything

- 2) Determine the orientation of the extracted phrase in step 1 using “Pointwise Mutual Information” (PMI) , with the association of positive reference “Excellent” and Negative reference “Poor”.

The Pointwise Mutual Information (PMI) between two words, w1 and w2, is defined as:

$$PMI (w1, w2) = \log [p (w1 \& w2) / p(w1) * P(w2)]$$

The Semantic Orientation (SO) of a phrase is calculated as:

$$SO (phrase) = PMI (phrase, “Excellent”) - PMI (phrase, “Poor”) \text{ [*1]}$$

If these results have positive sign then the phrase is positive otherwise it is negative.

In order to calculate SO (phrase) the authors used the PMI_IR (Pointwise Mutual Information, Information retrieval) which retrieves the co-occurrence of the phrase & “Excellent” or phrase & “Poor” queries by the help of search engines.

It is rewritten:

$$SO (phrase) = \log \frac{\text{hits(phrase NEAR[*2] “Excellent”) hits(“Poor”)}}{\text{hits(phrase NEAR “Poor”) hits(“Excellent”)}}$$

Where hits(query) is the number of hits returned.

- 3) The average of Semantic Orientation of all phrases shows the semantic orientation of the document.

The average performance of this approach is 74%. The exact performance depends on the context of analyzing. For example in the context of movie reviews where the sentiment polarity of a document is not the sum of the sentiment polarity of its parts, the performance is 66% while it can improve to 84% in the field of banking and automobiles where the whole is the sum of the parts [7].

It is worth mentioning that although this approach works well (comparable to SVM in topic-based text classification) especially in the case of banking and automobiles, the required time for mining the queries is costly and can be counted as the bottleneck of this approach as well as its memory consumption. The authors are still trying to improve the efficiency of the algorithm and make it more reasonable.

Although classifying opinionated texts at the document level is useful in many cases, it does not provide the necessary details needed for some other applications such as determining liked or disliked features or components of special goods in terms of product reviews which is done for product improvement in companies. They need much more focus as we will discuss in the next section.

[*1]: The references words “Excellent” and “Poor” were chosen because, in the five star review rating system, it is common to define one star as “Poor” and five star as “Excellent”. SO is positive when phrase is more strongly associated with “Excellent” and negative when phrase is more strongly associated with “Poor”.

[*2]: The Alta Vista search engine is used. NEAR operator searches the documents that contain the words within 10 words of one another, in either order. It has observed that operator NEAR outperforms operator AND during measuring the strength of semantic association between words.

1.7 Sentence Level Sentiment Analysis:

As it is mentioned at the end of the previous section, document level sentiment analysis does not give any details about the objects and related opinions inside each review. In other words, a positive opinionated document on a particular object does not mean that the writer has positive opinions on all aspects or features of the object. Likewise for negative opinionated documents. In order to obtain such detailed information we need to find features with the help of feature indicators and determine whether the opinions on the features are positive, negative or neutral.

However, there are several studies [10, 13, 12] which have been conducted in this field and attempted to find the optimal unsupervised approaches of extracting explicit features and corresponding opinions in the subjective sentences.

For instance in [13], 4 steps have been proposed in order to perform object and opinion extraction:

- 1) Frequent Features Identification
- 2) Opinion Orientation Identification
- 3) Infrequent Features Identification
- 4) Opinion Sentence Orientation Identification

Frequent Features Identification:

In order to perform this task, first part of speech (POS) tagging is applied and then Association Miner [8] based on the Apriori algorithm [9] is run on the reviews and those nouns^[*1] or noun-phrases which pass the threshold are selected^[*2]. (POS tagging is done by NLProcessor Linguistic Parser^[*3]). Association Miner is applied on the files of nouns and noun-phrases to identify frequent features, i.e nouns and noun-phrases which have appeared the most (threshold 1% of all reviews). Two types of pruning (compactness pruning and redundancy pruning [11]) are applied on the file to obtain more accurate features.

Opinion Orientation Identification:

In this step all adjectives^[*4] in a sentence which contains frequent features are selected as opinion words and the closest adjective to the feature (noun POS) is considered as the “effective opinion”. The sentence itself is called opinion sentence. All features and corresponding opinions (nouns and adjectives POS) are saved in a file.

Combining a hand-made set of seed adjectives and their sentiment orientation (positive / negative) with WordNet[24] for finding synonyms and antonyms of seed adjectives, lead the authors to gain a dataset of adjectives as opinion words and their positive or negative sentiment orientations.

Finally each feature is labeled positive or negative according to itself and corresponding opinion words and mentioned dataset.

Infrequent Features Identification:

In order to perform this task, after considering the appearance of negation words in each sentence,

[*1]: features which are talked about in reviews are nouns

[*2]: Notice that one of the drawbacks of this paper is that the authors have not investigated on pronoun resolution task which can be the reason for lower precision in feature extraction in comparison with the [10].

[*3]: Refer to “<http://www.infogistics.com/nlprocessor.pdf>”

[*4]: It is widely believed that most of opinions are expressed as adjectives toward features but they can also be verbs such as “Love”.

if there are no frequent features but one or more opinion words, the nearest noun or noun-phrase to the opinion words (adjective POS) is identified as infrequent feature. Infrequent features are also labeled.

Opinion Sentence Orientation Identification:

The Sentiment Orientation of each sentence is calculated as the average of the orientations for all opinion words that appeared in the sentence (positive = +1, Neutral = 0, Negative = -1). In the case of Neutral the average of effective opinions is calculated for each sentence and the orientation of the previous sentence is considered (according to the fact that most of the time reviews have continuous sentiment orientation).

Figure 1: Prediction the orientation of opinion sentences

```

Procedure SentenceOrientation()
begin
for each opinion sentence  $s_i$ 
begin
orientation = 0;
for each opinion word  $op$  in  $s_i$ 
orientation += wordOrientation( $op$ ,  $s_i$ );
/*Positive = 1, Negative = -1, Neutral = 0*/
if (orientation > 0)  $s_i$ 's orientation = Positive;
else if (orientation < 0)  $s_i$ 's orientation = Negative;
else {
for each feature  $f$  in  $s_i$ 
orientation += wordOrientation( $f$ 's effective opinion,  $s_i$ );
if (orientation > 0)
 $s_i$ 's orientation = Positive;
else if (orientation < 0)
 $s_i$ 's orientation = Negative;
else  $s_i$ 's orientation =  $s_{i-1}$ 's orientation;
}
end for;
end

Procedure wordOrientation(word, sentence)
begin
orientation = orientation of word in seed_list;
If (there is NEGATION_WORD appears closely around word in sentence)
orientation = Opposite(orientation);
end

```

From a technical point of view, it has been observed that frequent features alone contain many errors and need to be pruned. This improved the precision from 56% to 79%. Identifying infrequent features was useful as well and improved recall from 67% to 80% but degraded precision from 79 % to 72% in feature extraction. Finally, the accuracy of the Sentence Sentiment Orientation Prediction was 84%, which is a good result in terms of Sentiment orientation.

A similar approach has been proposed in [10] with some improvements in comparison with [13], which are going to be discussed:

The proposed approach has 2 main steps:

- 1) Frequent Features Identification
- 2) Opinion Orientation Identification

Frequent features identification:

In contrast with [13], the authors have used pronoun resolution which is helpful in terms of feature extraction precision. After part of speech (POS) tagging is done, nouns and noun-phrases which have higher frequencies are selected and Feature Assessor which is an instantiation of KnowItAll's Assessor, a web-based domain-independent information extraction system [14], evaluates them according to their PMI (Pointwise Mutual Information) scores with a **Meronymy discriminator** associated with the product class, in order to choose accurate features. Meronymy is a semantic relation used in linguistics. A meronym denotes a constituent part of, or a member of something. That is,

X is a meronym of Y if X s are parts of Y (s), or
 X is a meronym of Y if X s are members of Y (s).

For example, 'finger' is a meronym of 'hand' because a finger is part of a hand.

So, for:

f : Candidate Feature

d : Discriminator

$$PMI(f, d) = \frac{hits(f \wedge d)}{hits(f) * hits(d)}$$

For instance: product class "Scanner" has meronymy **discriminators** such as: "of scanner", "scanner has", "scanner comes with".

Also, in order to obtain better results, the authors have used Web PMI statistics in addition to reviews in the assessment of candidate features which improved precision 22% in comparison with [13] but recall was dropped 3%.

Opinion Orientation Identification:

In this step the authors have exploited the advantage of syntactic dependencies (Table 1.3) computed by the Minipar parser in order to identify opinion phrases. If the explicit feature is found in the sentence, 10 extraction rules, listed in Table1, are applied to find the heads of potential opinion phrases and each headword together with its modifier is returned as opinion phrase.

For example:

"I'm not happy with this sluggish driver"

feature: driver

Modifier: sluggish

potential opinion: sluggish

potential opinion phrase: sluggish driver

Table 1.3: 4 rules of 10 Domain-independent rules for opinion phrase extraction

Extraction rules	Examples
If $\exists (M, NP = f) \rightarrow po = M$	(expensive) scanner
If $\exists (S = f, P, O) \rightarrow po = O$	Lamp has (problems)
If $\exists (S, P, O = f) \rightarrow po = P$	I (hate) this scanner
If $\exists (S = f, P, O) \rightarrow po = P$	Program (crashed)

PO = potential opinion

M = Modifier

NP = noun phrase

S = subject

P = predict

O = object

At this level, 79% precision and 76% recall is reported for opinion phrase detection. Meanwhile, in order to perform semantic orientation identification, “Relaxation Labeling” is applied which is an un-supervised classification to assign a label to each feature. Relaxation labeling leads to high precision (86%) and high recall (89%).

Relaxation Labeling:

Relaxation labeling is an un-supervised classification technique which assigns a label to each feature in each sentence, according to the feature’s neighbor’s label.

The neighbors for feature ‘w’ are those words connected to ‘w’ through a relationship of type T, which should be defined first. For example one of the common relationships are conjunctions “AND” or disjunctions “BUT”. In other words, the label which is assigned to word ‘w’ is more like the label which had been assigned to its neighbor which has “AND” relationship.

This technique is helpful for those situations that the opinion word for the feature ‘w’ itself doesn’t have a dominant label. For example “hate” always has negative label but “hot” sometimes is positive and sometimes is negative. It depends on its (feature’s) neighbor.

The example bellow makes this clear:

Hot $\rightarrow$ Positive / Negative

In the Hotel domain:

Hot Water: $\rightarrow$ Positive

Hot Room: $\rightarrow ?$ {In the vicinity of Broken Fan} $\rightarrow$ Negative

The double propagation method can also be used to extract features based on extracting features and sentiment words together and labeling them simultaneously [12]. It considers the dependency between each feature and corresponded sentiment and vice versa. So, it tries to find features based on known sentiment, features based on known features, sentiment based on known sentiment and sentiment based on known features.

In order to figure out these relations, the authors have defined a “Grammar Dependency” [15] which describes the grammar of these relations.

Given a set of hand-made seed adjectives as initial opinion words, the process begins in each sentence. According to Table 1.4, the features and sentiments are extracted iteratively until no more features or sentiment words are identified. Meanwhile the labeling process is running and assigns a label to each pair according to heterogeneous and homogenous rules:

Heterogeneous Rule:

For sentiment words extracted by known features and features extracted by known sentiment words, the same polarity as the known one is assigned. According to the fact that features convey no polarities so the polarity of the features inherits those of associated sentiment words. Notice that “Negation” is considered

Homogenous Rules:

For sentiment words extracted by known sentiment words and features extracted by known features, the same polarity as the known one is considered unless there are contrary words between them such as “but”, “however”...

Table 1.4: rules for sentiment word and feature extraction

	Observations	constraints	Outputs
$R1_1$	$S_{i(j)} \rightarrow S_{i(j)} - Dep \rightarrow S_{j(i)}$	$S_{j(i)} \in \{S\},$ $S_{i(j)} - Dep \in \{CONJ\},$ $POS(S_{i(j)}) \in \{JJ\}$	$s = S_{i(j)}$
$R1_2$	$S_i \rightarrow S_i - Dep \rightarrow H \leftarrow S_j - Dep \leftarrow S_j$	$S_i \in \{S\},$ $S_i - Dep == S_j - Dep$ $POS(S_j) \in \{JJ\}$	$s = S_j$
$R2_1$	$S \rightarrow S - Dep \rightarrow F$	$F \in \{F\},$ $S - Dep \in \{MR\}$ $POS(S) \in \{JJ\}$	$s = S$
$R2_2$	$S \rightarrow S - Dep \rightarrow H \leftarrow F - Dep \leftarrow F$	$F \in \{F\},$ $S/F - Dep \in \{MR\}$ $POS(S) \in \{JJ\}$	$s = S$
$R3_1$	$S \rightarrow S - Dep \rightarrow F$	$S \in \{S\},$ $S - Dep \in \{MR\}$ $POS(F) \in \{NN\}$	$f = F$
$R3_2$	$S \rightarrow S - Dep \rightarrow H \leftarrow F - Dep \leftarrow F$	$S \in \{S\},$ $S/F - Dep \in \{MR\}$ $POS(F) \in \{NN\}$	$f = F$
$R4_1$	$F_{i(j)} \rightarrow F_{i(j)} - Dep \rightarrow F_{j(i)}$	$F_{j(i)} \in \{F\},$ $F_{i(j)} - Dep \in \{CONJ\},$ $POS(F_{i(j)}) \in \{NN\}$	$f = F_{i(j)}$
$R4_2$	$F_i \rightarrow F_i - Dep \rightarrow H \leftarrow F_j - Dep \leftarrow F_j$	$F_i \in \{F\},$ $F_i - Dep == F_j - Dep$ $POS(F_j) \in \{NN\}$	$f = F_j$

- The arrows mean dependency, for example $S \rightarrow S - Dep \rightarrow F$ means S depends on F through a relation of S-Dep.
- S(or f) : extracted sentiment word (or feature)
- {s} (or {F}) :Known sentiment words (or features)
- $-Dep$: dependency relation of S(or F)
- H: any words
- POS(S(or F)) : POS information of S(or F)
- JJ: Adjectives (Sentiment Words)
- NN: Nouns , noun-phrases (Features)
- JJR: Adjectives with comparative ending
- JJS: Adjectives with superlative ending
- Conj: conjunctions
- {JJ}: set of potential sentiment words (JJ , JJR , JJS)
- {NN}: set of potential features
- {MR}: dependency relations describing relations between sentiment words and features such as **mod** which means that one word modifies the other word
- {CONJ}: relation of conjunctions

In conclusion, we can say that the double propagation method seems to be effective while in terms of sentiment word extraction recall and precision is reported between 60% - 70%, and the accuracy of polarity assignment was between 35% to 68%^[*1].

1.8 Conclusion:

This chapter gave an introduction to sentiment analysis. Applications which depend on subjectivity were discussed briefly as well as our need for them. The obstacles in performing opinion mining were mentioned and supervised and unsupervised approaches were discussed to deal with these.

Document level sentiment analysis, which aims to determine the sentiment orientation of a whole document, was compared with the much smaller scale, sentence level sentiment analysis. Also, the ability of sentiment analysis in identifying each feature of the related topic as well as the expressed opinion was discussed at the sentence level.

The results of document level sentiment analysis are hidden in the outcomes of sentence level sentiment analysis; therefore the sentence level is more informative than its larger scale counterpart and most of evaluations are performed on that.

Generally speaking, the most often used features to identify the object's features at the sentence level, are unigram, bigram, adjectives and Part Of Speech tags (POS) where unigram is almost always used as the baseline due to the accurate results it generates. Regression and Support Vector Machine (SVM) classifiers are usually used while the most accurate results are generated by the SVM classifier.

A number of interesting approaches are mentioned in this survey. The substring Feature (arbitrary length feature) is one of them which seem to be very informative to determine the polarity of a sentence as well as the object's features if investigated more. Another eminent feature is the "Negation Phrase" which causes more accurate results in the polarity reporting. And finally the novel approach for measuring the similarity among the labels of features and normalize them (metric labeling) as well as the approach for assigning the best label to each object's feature according to the feature's neighbor's label (relaxation labeling) are the milestones of this study.

Having considered the recent studies carried out in this area, one is left with the question that among all the different feature sets discussed, which performs the best on the particular dataset or which classifier is of significant importance by considering that time and accuracy are both important for performing a good classification. Given the fact that on the whole experts have a tendency to think that SVM is the best classifier for sentiment analysis and it is widely used in the binary classification of opinionated documents, the question is how much this predominant attitude should be dignified or how much it leaves to be desired.

[*1]: Two drawbacks for this paper:

- 1) According to the obtained accuracies, the authors have just plotted graphs without explicit grid size; therefore above mentioned accuracy is not accurate.
- 2) In Observation 1: it is written that "It is usually the case that the same sentiment word has the same polarity" but it is not correct.

Counter Example:

In the Hotel Domain $\begin{cases} \text{hot water} \rightarrow \text{Positive} \\ \text{hot room} \rightarrow \text{Negative} \end{cases}$

In order to come up with the answers for the questions mentioned, the research in this master thesis has been conducted which discusses about the different angles of binary classification of sentiment analysis in details in the following chapters.

Chapter 2: Goal and Survey Hypothesis

Sentiment analysis is based on 3 main components: a dataset, feature and a classifier. Each of these components is of an importance because the results of sentiment analysis can be easily influenced by them. It should be stressed that the Dataset itself is not considered as a variable, but different representations of it are applied in the classification procedure. To be more precise, each dataset consists of a number of “documents” that can be presented with a set of features. Then, the classifier predicts the label of each of the documents in the dataset based on the represented feature set. Therefore, selecting good features as documents representations will help the classifier to predict the label of each document with high accuracy. On the other hand, not only a good feature set should be given proper attention but also selecting the classifier should be done precisely and neglecting it will undoubtedly lead to misclassification.

From the technical point of view, a classifier is a mathematical function $F(x)$ which is given an object (x) in the feature set as input and calculates the conditional probability of the object, given label (w).

X: An object in the feature set

W: Given Label

$$F(x) = p(w / x)$$

Therefore, selecting the best model which fits the data and consequently calculates the conditional probability with higher estimates, plays a major role in sentiment classification.

Another important topic on classification is the “Curse of Dimensionality” which is often used as a blanket for not dealing with high dimensional data. In order to obtain a statistically reliable result, the amount of data one needs to support the result often grows exponentially with the dimensionality. That is to say, in order to describe each document precisely, one needs more features that give more information about the outcome to the classifier to predict. On the other hand by increasing the number of features, the number of parameters that should be estimated by the classifier, is increased consequently. Thus, to obtain reliable results more training data is required. Considering all the factors mentioned above, in order to improve the results of sentiment classification, it is necessary to reduce the features, to the best features that are sufficient to identify the documents. It is often called “Dimensionality Reduction”. It should not be neglected that having too few features leads to misclassification as well, so, as a result finding the optimal number of features takes the precedence.

What it all comes down to is that sentiment analysis can be done accurately if and only if the optimal features, informative and quantity wise, as well as proper classifier are selected.

2.1 Main Hypotheses:

In document level sentiment analysis binary classification or positive/negative classification is widely used according to the importance of inferring the dominant opinion of a document. As it is explained in the previous chapters, it is of a significant importance to deal with the overall sentiment of a document fast and accurately.

During conducting the survey in the previous chapter, one of the issues we have encountered, was having many different feature sets and classification methods proposed by authors on different datasets, which makes the comparison among them difficult. In order to figure out which feature sets and which classifiers perform well, we have decided to apply different feature sets and classification approaches on a large single dataset and compare the results together.

The following Research Questions will be answered in the next chapters by considering the Support Vector Machine (SVM) as the base classifier since almost all studies use it:

- 1) What is the best set of features to be extracted from a dataset with a large number of documents?
- 2) What is the best set of features to be extracted from a dataset with a small number of documents?
- 3) Which classifier performs best on a large opinion dataset in terms of time and accuracy?
- 4) Which classifier performs best on a small opinion dataset in terms of time and accuracy?

2.2 Subordinate Hypotheses:

1) As discussed in the detail in the previous chapter, there are 2 kinds of dictionaries that can be used in sentiment analysis: "Dictionary Based" and "Corpus Based". We will investigate which of these 2 approaches performs best for opinion classification.

2) Most previous studies in sentiment analysis are based on movie or product reviews and there are very few studies carried out on societal themes, therefore another subordinate question can be: What is the overall performance on this field, provided that due to the nature of the societal documents predicting the dominant attitude of them is much more difficult in comparison with movie and product reviews.

Chapter 3: Data and Tools

As it discussed in the previous chapters, the main objective of conducting the current thesis research is comparing the functionality of different classifiers based on different feature sets to identify the best classifier in terms of accuracy and speed of classification as well as the best feature set, informative and quantitative wise, in document level sentiment analysis.

In pursuance of fulfilling this objective, the first thing we need is a unique dataset that all the experiments can be done on in order to make comparison of the results. This is important since as it is mentioned before; former studies in this area did not investigate on applying all the important feature sets as well as different classifiers on the same dataset which made the comparison impossible. But in this study, different feature sets with different number of features will be tested different classifiers, thus it can be used as a good source to select appropriate classifiers and feature sets for further studies.

3.1 Dataset

From a technical point of view, a dataset which is used in document level sentiment analysis consists of a number of documents such that each document expresses the opinion of the author in few sentences. The author expresses his/her opinion toward a special topic, therefore, the document has a dominant direction which is positive or negative in general classification. This positivity or negativity of the document is called the label of that document. To sum up, each dataset which is used in opinion analysis is made up by a number of documents and labels. These labels then play a major role in classification procedure which is discussed later in details.

The dataset^[*1] which is used in this research consists of 6 different smaller dataset. Each smaller set is about a special topic such as "Abortion", "Creation", "Guns", "God", "Gay Rights" and "Health Care". To be more clear, one of the documents in each dataset is given below, following with its label:

Abortion: "The fetus causes physical pain; the woman has a right to self-defense. The fetus causes sickness, discomfort, and extreme pain to a woman during her pregnancy and labor. It is, therefore, justifiable for a woman to pursue an abortion in self-defense." → **positive**

Creation: "Teaching evolution exclusively is dangerous to science and reason. To exclude a set of ideas a priori is to potentially exclude the truth. Thus, as a matter of principle, science should not be constrained to a single set of ideas. Creationists argue that both sides should be taught. Evolutionists, on the other hand, limit the scope of science to a narrow acceptance of just one theory. Thus, to teach evolution exclusively is to promote limitations on the power of scientific inquiry. To teach creationism alongside evolutionism is to promote open scientific inquiry and critical thought." → **positive**

Guns: "It's better to arm pilots and risk abuse than risk another 9/11. Dave Kopel& David Petteys. "Air Neglect". National Review Online. July 2, 2003 - "the airline executives would rather risk another 9/11 than take the (tiny) chance that an armed pilot might use his gun illegally, and the airline might be sued". " → **Negative**

[*1]: the dataset is downloading from: <http://www.cs.pitt.edu/mpqa/>

God: "Consider how much of our DNA is shared by chimpanzees (about 97%). Consider cross-species transfer of viruses, bacteria, etc. whereby humans get malaria, Hantavirus, dengue fever and other diseases. That by itself is sufficient to show that "Intelligent Design" is oxymoronic. As one of my Philosophy instructors often stated: "If God created the universe, he was an incompetent engineer.""→**Negative**

Gay Rights: "Some babies are born with a predisposition to homosexuality (both human and in other races), and their upbringing will not be affect their sexuality. Attempting to suppress this genetic predisposition has resulted in great misery for many people. Rather, we should accept this and look to embrace all gay people fully or we stay in this denial situation where having a different sexual orientation means not the right to adopt a child."→**Positive**

Health Care: Public health insurance offers citizens more choices Jacob Hacker. "The case for public plan". The Institute for America's Future: "public plan choice gives Americans the opportunity to choose for themselves how they value the strengths and weaknesses of a public, Medicare-like plan and competing private health plans."→**Positive**

Providing that in this research, the classification procedure is applied on small and big datasets in terms of number of documents, and since there are few documents in the 6 datasets, we were left with no other choice but to think that the best possible way of obtaining a big opinion based dataset is to aggregate the 6 datasets and make a big one out of them. I should be stressed that, as it is discussed in the next chapter, the size of datasets is of a significant importance in the supervised classification approach since if the number of documents in a dataset is high, then the number of training documents is consequently high and the classifier can be trained well while in the small datasets where the amount of training data is small, the classifiers cannot be trained well and therefore it is necessary to investigate more on the features in order to be much informative to obtain a good classification.

Table 3.1: The number of documents in each datasets

Dataset	Number of documents
Abortion	1151
Creation	1230
Guns	1069
God	953
Gay Rights	2064
Health Care	667
Aggregated	7134

It is noteworthy to know that most of the available datasets are movie or product reviews and finding a good labeled dataset with high number of documents, if it is not impossible, it is surely difficult. This, is the main reason of aggregating small datasets in order to obtain a big labeled opinion based dataset one in this research.

3.2 Classification

Generally speaking, supervised classification consists of 3 main components, the “Training Set”, the “Classifier” and the “Test Set”. Training set is used for training a system by the examples which are all labeled. In the training procedure, the classifier is learned how to classify each object based on its features and its label. At the end, the system is evaluated by the test set. Test set is a set of examples which neither been labeled nor been used in training procedure while classifier tries to predict their labels. When labels are available, the functionality of system can be evaluated.

In supervised classification, objects are defined by “Feature Vectors”. A feature vector is an n-dimensional vector of numerical features that represent some objects. In textual sentiment analysis, term occurrence is considered as a feature and therefore, each document is represented by an n-dimensional feature vector, such that n is the number of words in the dictionary which is used for classification. In the case of occurrence of a dictionary word in the document, the feature is 1 otherwise, it is 0.

The vector space associated with these vectors is often called the “Feature Space”. In order to reduce the dimensionality of the feature space, a number of dimensionality reduction techniques can be employed such as Principal Component Analysis (PCA).

3.3 Density Estimation Classification

There are several approaches to classify a dataset. Density Estimation and Structural Risk Minimization are studied in this research in order to classify the opinionated documents to positive or negative groups based on their dominant sentiment.

3.3.1 Parametric Approach

The density estimation approach can be summarized in two Parametric and Non-parametric techniques. In the parametric technique, a simple global model, mostly Gaussian, is assumed for each class and its parameters are estimated, while in the non-parametric technique, a simple local model is considered.

According to the “Central Limit Theorem”, the mean of large numbers of independent identically distributed random variables will have a Gaussian distribution irrespective of the form of the original distribution. That is the chief reason of using Gaussian distribution as a global model in parametric approach.

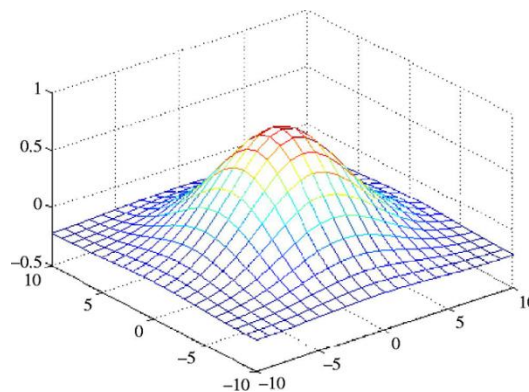


Figure 3.1: Two dimensional Zero Mean Gaussian Distribution

From a mathematical point of view, the Gaussian distribution has a continuous probability density function which in p-dimensional data space, with mean vector μ and covariance matrix Σ , it will be:

$$p(x) = \frac{1}{\sqrt{(2\pi)^p * \det(\Sigma)}} \exp\left[-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right]$$

In the parametric approach, Bayes rule plays a major role in positive/negative classification. In a dataset that all training objects have label w_1 as positive or label w_2 as negative, the classifier for text object x is defined based on Bayes rule. Assigned

$$p(w_1|x) = \frac{p(x|w_1) * p(w_1)}{p(x)}$$

$$p(w_2|x) = \frac{p(x|w_2) * p(w_2)}{p(x)}$$

And,

$$\begin{cases} \text{if } p(w_1|x) > p(w_2|x) \\ \text{if } p(w_1|x) < p(w_2|x) \end{cases} \quad \begin{array}{l} \text{positive label will be assigned to } x \\ \text{negative label will be assigned to } x \end{array}$$

Therefore the classifier will be:

$$F(x) = p(w_1|x) - p(w_2|x) \rightarrow \begin{cases} \text{if } F(x) > 0 & \text{Positive} \\ \text{if } F(x) < 0 & \text{Negative} \end{cases}$$

As it can be derived from the Bayes rule:

$$p(w_1|x) \propto p(x|w_1) * p(w_1)$$

$$p(w_2|x) \propto p(x|w_2) * p(w_2)$$

According to the fact that $p(w_1)$ and $p(w_2)$ are constant, therefore, the calculation of $p(x|w_1)$ and $p(x|w_2)$ take the precedence which based on parametric approach by plugging in Gaussian density function, we have:

$$p(x | w_1) = \frac{1}{\sqrt{2\pi^p * \det(\hat{\Sigma}_{w_1})}} \exp\left[-\frac{1}{2}(x - \hat{\mu}_{w_1})^T \hat{\Sigma}_{w_1}^{-1}(x - \hat{\mu}_{w_1})\right]$$

$$p(x | w_2) = \frac{1}{\sqrt{2\pi^p * \det(\hat{\Sigma}_{w_2})}} \exp\left[-\frac{1}{2}(x - \hat{\mu}_{w_2})^T \hat{\Sigma}_{w_2}^{-1}(x - \hat{\mu}_{w_2})\right]$$

And we have to estimate the parameters:

$$\hat{\mu}_{w_1} = \frac{1}{n} \sum_{i=1}^n x_i \quad \& \quad \hat{\Sigma}_{w_1} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_{w_1})(x_i - \hat{\mu}_{w_1})^T$$

$$\hat{\mu}_{w_2} = \frac{1}{m} \sum_{i=1}^m x_i \quad \& \quad \hat{\Sigma}_{w_2} = \frac{1}{m} \sum_{i=1}^m (x_i - \hat{\mu}_{w_2})(x_i - \hat{\mu}_{w_2})^T$$

The resulting classifier based on a separate mean and covariance matrix per class, is called Quadratic Classifier [37], because the design boundary is a quadratic function of x .

In order to make the classifier simpler and consequently linear, it is possible to average over all classes covariance matrices. Thus, the classifier is built based on separate mean and equal covariance matrix per class, which is called Linear Classifier [39].

By considering the identical covariance matrix per class and using the distance to the mean of the classes, another linear classifier is obtained which is called Nearest Mean Classifier [38].

3.3.2 Non-Parametric Approach

As far as non-parametric technique in Density Estimation is concerned, the nearest neighbor method is perhaps the simplest of all algorithms for predicting the class of a test object. The training phase is trivial; simply store every training object with its label. To make a prediction for a test object first compute its distance to every training objects and then keeps the k closest training objects where $k \geq 1$ a fixed integer is. Look for the label that is most common among these objects. This label is the prediction for this test object.

There are two main choices to make: the value of k and the distance function to use. In positive/negative classification, in order to avoid ties, the most common choice for k is a small odd integer for instance $k = 3$ and the most common distance function is Euclidean distance.

$$d(x, y) = ||x - y|| = \sqrt{(x - y)(x - y)^T} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Where x and y are points in R^n .

Therefore, for the nearest neighbor density, we define:

$$\hat{p}(x) = \frac{k}{n * V_k}$$

Where V_k is the volume of the sphere centered at x with radius r , the distance to the k -th nearest neighbor.

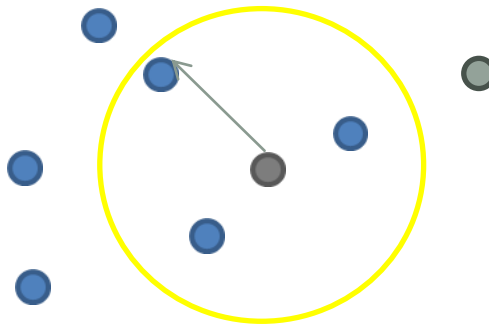


Figure 3.2: K-nearest Neighbor algorithm while $k = 3$

The classifier is:

$$F(x) = p(x|w_1) * p(w_1) - p(x|w_2) * p(w_2) \rightarrow \begin{cases} \text{if } F(x) > 0 & \text{Positive} \\ \text{if } F(x) < 0 & \text{Negative} \end{cases}$$

This is called the k-Nearest Neighbor Classifier [37].

3.4 Structural Risk Minimization Classification

Structural risk minimization in “Machine Learning” which refers to inferring a generalized model from dataset such that the resulted classifier from that model fits the data and therefore makes zero error on the training set. Due to the over fitting of classifier to the training data, it performs poorly on test set while according to the SRM principal, there are several tricks to benefit of this model to reduce the error on test set[43].

When a classifier fits to data, the correct complexity of the model, how well it can separate objects, should be chosen. The optimal complexity can be measured then by the performance on an independent test set. The Support Vector classifier minimizes its complexity while keeping a zero error on the training data.

3.4.1 Support Vector Classifier

The support vector classifier is one of the supervised techniques that construct a hyper plane which represents the largest separation or margin between the two classes in order to minimize the generalization error. The hyper lane is constructed based on support vectors.

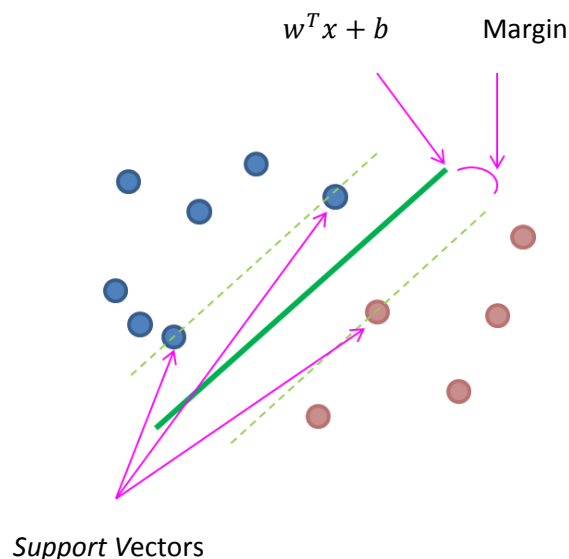


Figure 3.3: Support Vector Classifier

By considering:

$$D = \{(x_i, y_i) | x_i \in R^n, y \in \{Positive = +1, Negative = -1\}\}_{i=1:m}$$

$$\text{Positive (red): } w^T x + b \geq +1$$

$$\text{Negative (blue): } w^T x + b \leq -1$$

$$\text{Support vector: } w^T x + b = \pm 1$$

$$\text{Distance from the hyper plane to points: } \frac{|w^T x + b|}{||w||}$$

$$\text{Margin: } \frac{2}{||w||}$$

Then the optimization problem is represented as fallow:

Objective function:

$$\max_{w,b} \frac{2}{||w||} \text{ subjective to } \begin{cases} w^T x_i + b \geq +1 \\ w^T x_i + b \leq -1 \end{cases}$$

Equivalent quadratic optimization problem:

$$\min_{w,b} \frac{1}{2} w^T w \text{ subjective to } \{ y_i (w^T x_i + b) \geq +1 \}$$

Therefore the solution will be:

$$w = \sum_i \alpha_i y_i x_i$$

$$b = y_i - w^T x_i$$

So, the classifier will be:

$$w^T x + b = \sum_i \alpha_i y_i x_i^T x + b$$

where α_i are weights and x_i are support vectors.

When the training data is not linearly separable, it should be mapped to the higher dimensional feature space using kernel methods. Therefore, a linear function in the high dimensional kernel space is obtained while a non-linear function in the original feature space is remained.

In order to apply kernel method, the inner product should be replaced by another similarity measure.

$$\Phi: x \rightarrow \varphi(x)$$

$$\sum_i \alpha_i y_i x_i^T x + b \rightarrow \sum_i \alpha_i y_i K(x_i, x) + b$$

Where

$$K(x_i, x) = \varphi(x_i)^T \varphi(x)$$

There are several kernels which can be used in support vector classifiers but in this study the polynomial kernels have been evaluated.

$$K(x, y) = (x^T y + 1)^d$$

Where d represents the degree of polynomial kernel. For instance If $d = 1$ it is linear kernel while $d = 2$ represents the quadratic kernel [44].

3.5 PRTools

In order to perform the classification procedure in this research, we have used “*PRTools*^[*1]” which is a MATLAB based toolbox for Pattern Recognition and can be freely be used for academic research.

PRTools presented us with a lot of advantages. It can cover the main steps: representation, generalizing and evaluation. Training classifiers, linear feature extraction, feature reduction and error estimation have been done with the aid of this toolbox in this study. In this part we briefly introduce the modules that we used from PRTools.

Classifiers:

- **ldc:** Linear Bayes Normal Classifier

Computation of the linear classifier between the classes of the dataset by assuming normal densities with separate mean and equal covariance matrices per class. The joint covariance matrix is the weighted (by a priori probabilities) average of the class covariance matrices.

- **qdc:** Quadratic Bayes Normal Classifier

Computation of the quadratic classifier between the classes of the dataset assuming normal densities with separate mean per class [37].

- **nmc:** Nearest Mean Classifier

Computation of the nearest mean classifier between the classes in the dataset with separate mean and identity covariance matrix per class. Prior probabilities are not used because NMC is a plain classifier which is feature scaling sensitive and insensitive to class priors.

- **Knnc:** K-Nearest Neighbor Classifier

Computation of the K-nearest neighbor classifier for the dataset such that K is optimized with respect to the leave-one-out error on the dataset and class prior probabilities in it are neglected.

- **Svm:** Support Vector Classifier

Optimizes a support vector classifier for the dataset by quadratic programming. The non-linearity is determined by the kernel. There are several ways to define KERNEL, e.g. linear kernel is computed by (PROXM ([], 'p', 1)) or (PROXM ([], 'p', 2)) is for computing quadratic kernel.

[*1]: PRTools has been developed in Pattern Recognition Group of Delft University of Technology. For more information refer to: www.prtools.org

Linear Feature Extraction:

- **PCA:** Principal Component Analysis

It performs a PCA on the overall covariance matrix. In order to obtain the best variance in the data, the data is projected to a subspace (of the data space) which is built by the eigenvectors from the data. In that sense, the eigenvalue corresponding to an eigenvector represents the amount of variance that eigenvector handles. The mathematical formulation of PCA is discussed in the next chapter.

Cross Validation:

- **Crossval:** Cross Validation

Cross validation is the estimation of the error or performance of the untrained classifier using the dataset. The set is randomly permuted and divided in n_folds (almost) equally sized parts, using a stratified procedure. The classifier is trained on $(n_folds)-1$ parts and the remaining part is used for testing. This is rotated over all parts. Error is their weighted average over the class priors. [41, 42]

Chapter 4: Model and Features

As mentioned before, textual sentiment analysis or more precisely positive/negative classification is based on the use of a dataset and a classifier. The classifier is applied to the dataset in order to classify documents to two groups: positives and negatives. The more documents will be presented informatively in test phase, the more accurate the result of classification will be. Therefore, finding the best document representatives that can describe it (document) better is of a significant importance in sentiment analysis. From a technical point of view, a document representative is called a “feature”. A feature is an individual measurable heuristic property of document being observed. Due to the fact that choosing discriminative and independent features is key to successful sentiment classification, extracting features that are measurable by a computer is an art and hand selection of these, forms the base of opinion analysis. In this research we have investigated 2 different groups of features,” Common Features” that have been used in all similar previous studies and “Selected Features” which are expected to express the authors’ attitude in more details. Also, the WordNet dictionary [24] was of significant help in order to extract common features. Figure 1 shows the first phase of training procedure in this study.

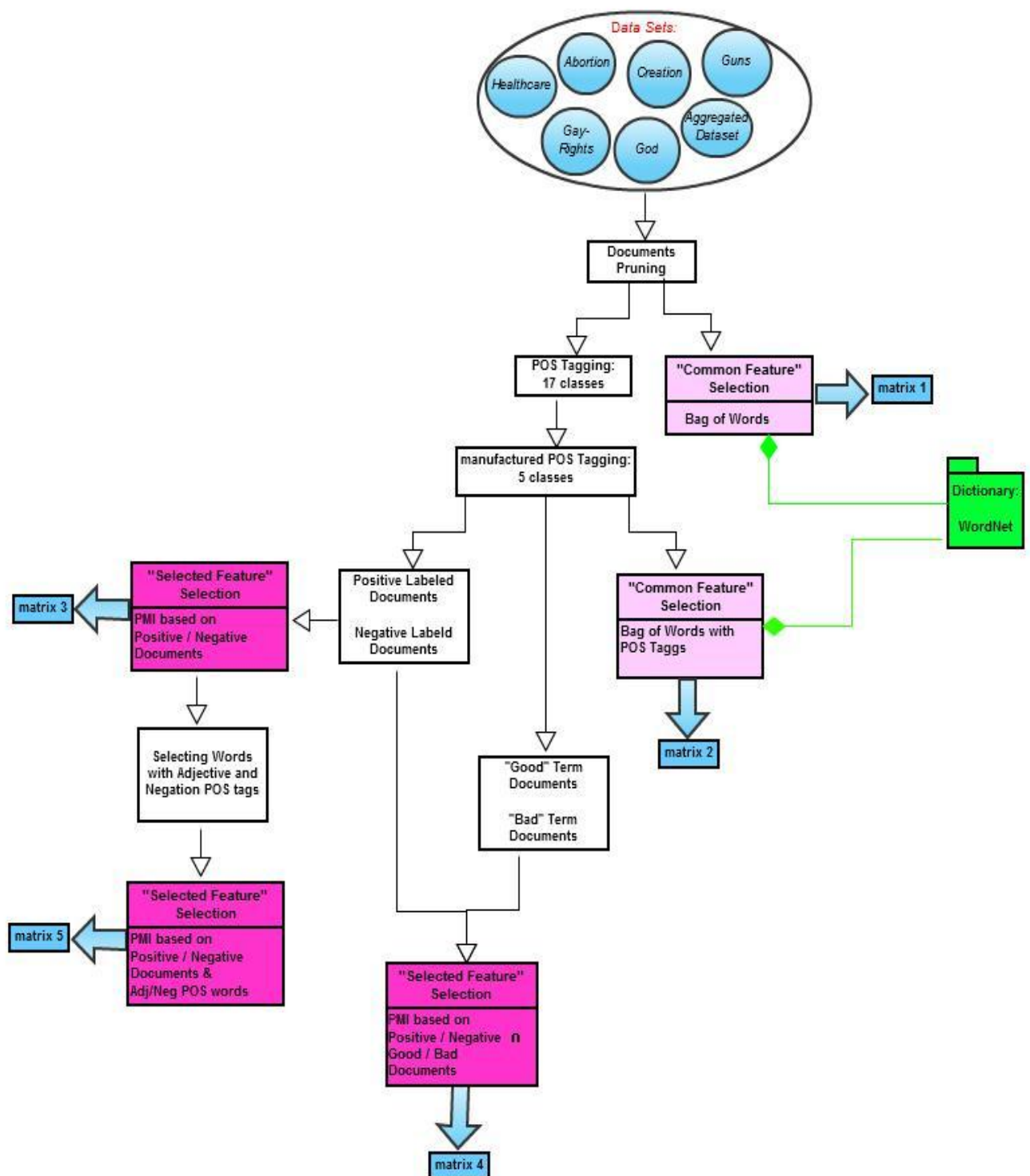


Figure 4.1: Phase 1 of Training Procedure. Note that in this model the purple blocks belong for common features while pink blocks show the selected features. The resulted matrix based on each feature set is shown by blue arrow.

Each dataset is represented as 5 different matrices based on 5 different feature sets. As it is shown in Figure 2, in the second phase of the training procedure, the datasets are transformed in order to be used much efficiently by classification procedure.

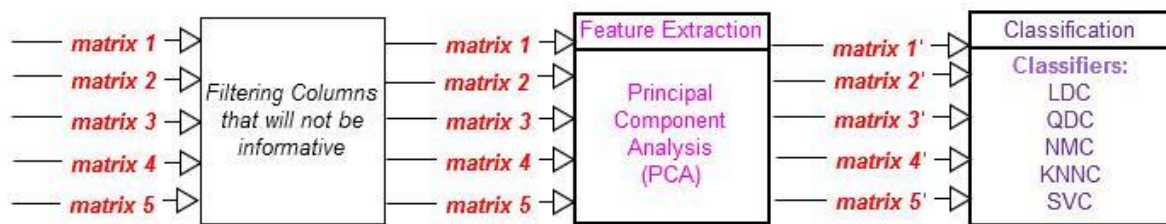


Figure 4.2: Phase 2 of Training Procedure. Note that the constructed matrices in the previous phase are given as input to this phase in order to be processed.

Finally in the testing phase, the efficiency of each feature set is evaluated by different classifiers. We have taken advantage of Cross Validation, which is more accurate and honest measure than using a single test set for estimating the performance of a predictive model, in this research.

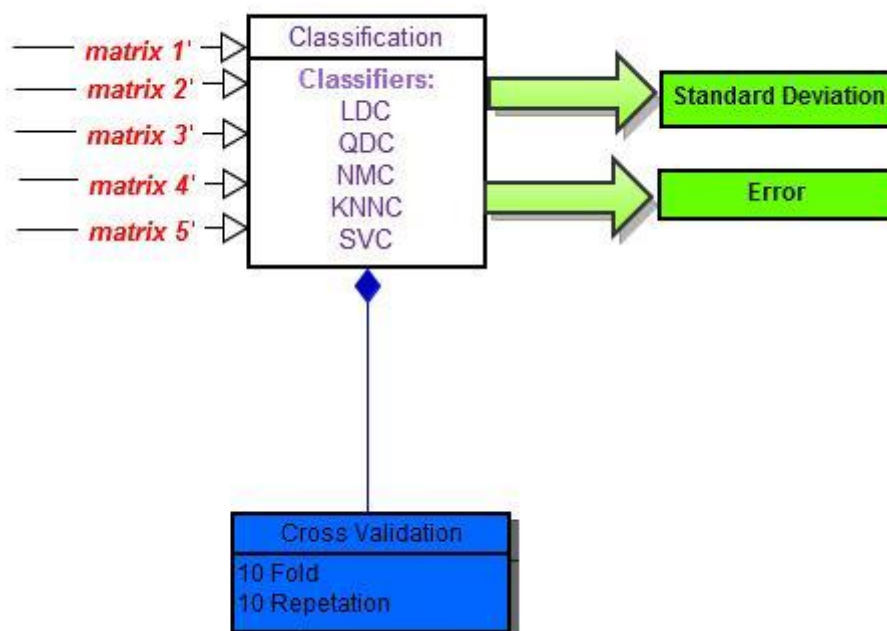


Figure 4.3: Testing Procedure. Note that the constructed matrices in the second phase from PCA procedure are given to classifiers to be tested based on 10 Fold-Cross Validation. The testing procedure is repeated 10 times to produce the most accurate results.

4.1 Common Features:

4.1.1 Bag of Words

In this model, which is widely used in supervised text classification, a document is represented as an un-ordered collection of words disregarding grammar and even word order. According to this model, during the training phase, a dictionary is constructed based on training data and is then used to discriminate between the positive and negative documents in the testing procedure.

For instance if we have the two documents below:

- 1) Healthcare is a very important issue.
- 2) Healthcare should be considered precisely.

The dictionary which is constructed based on BoW will be:

Dictionary= {1:"Healthcare", 2:"is", 3:"a", 4:"very", 5:"important", 6:"issue", 7:"should", 8:"be", 9:"considered", 10:"precisely"}

Therefore the feature vector of each document has "10 dimensionalities" based on the constructed dictionary.

As it is explained in chapter 1, in sentiment analysis, word appearance is very informative (in contrast with word frequency in information retrieval). According to the nature of natural language, one word can express the authors' attitude clearly while a sequence of words cannot. For instance, in the sentence below, only the words "like" and "not" show the polarity of the sentence while the whole sentence seems to have positive polarity.

"Swimming is one of the sports that most of people are fan of. Although swimming is good, I do not like it."

4.1.2 Dictionary

In this research, instead of constructing a dictionary based on documents, we have used the WordNet dictionary [24]. WordNet is a large lexical database of English with 58058 words and 4 part of speech tags. Nouns, verbs, adjectives and adverbs are grouped into set of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. There are 3 chief reasons for selecting WordNet.

First of all, the dictionary which was made based on "Aggregated Dataset" (aggregation of 6 small datasets) was too big, approximately 117660 words, and the constructed matrix based on it was practically unusable by MATLAB and PRTTools. Since the verbs have different tenses in documents and nouns have different forms of singular and plural, the size of the dictionary has increased uselessly while in the WordNet dictionary words are in their lemmatized form and therefore MATLAB and PRTTools could take the advantage of constructed matrix. I should press upon that the different tenses of verbs or singular or plural forms of nouns are not important in sentiment analysis while the appearance of them is informative. That is why it is a safe assumption to use lemmatized forms of words.

Second, since WordNet is a huge dictionary, it is more probable to have all the words as a source in comparison with the "Aggregated Dataset". Although in chapter 1, it was mentioned that corpus based dictionary will be useful in opinion classification, due to the small size of the available dataset in term of number of documents with societal theme, it would not be too far off if we say that WordNet dictionary will be of much more significant help.

Finally, WordNet is a popular and available dictionary. The experiments conducted based on it, are corpus independent and therefore the results can be generalized for other datasets as well.

In this research in order to use the WordNet^[*1] dictionary, it is necessary to prune and lemmatize the dataset. Thus, each word in dataset is lemmatized and all the plural nouns are changed to singular form and all different tenses of verbs are changed to the present first person tense with the help of MorphAdorn^[*2]. For example:

“I am going home” → “I be go home”

“There are 2 girls” → “There be 2 girl”

All the words are converted to their lower case form in terms of their letters, to be compatible with the WordNet words. Also all the numbers, punctuation marks and other elements except words with 2 or more letters are removed from dataset since they were not that much informative in comparison with other words. Therefore, terms such as 19, a, @ and words like that are removed from the datasets. For instance:

“For 4 years to come, corruption isn’t the extreme” →

“for year to come corruption be not the extreme”

After applying the pruning procedure on the dataset, documents are ready to be presented by the BoW model. Each dataset is described in a 2 dimensional matrix of documents and WordNet dictionary. The dimensionality of each dataset based on BoW model is shown in the table 4.1.

Table 4.1: Detailed information of the matrices constructed for each dataset based on BoW Model.

Note that in each matrix one element is considered for those words that do not exist in WordNet dictionary. Therefore, the number of columns is 58059 instead of 58058.

Dataset	Dictionary	Documents	Matrix	Label
Abortion	58058	1151	1151 * 58059	1151
Creation	58058	1230	1230 * 58059	1230
God	58058	953	953 * 58059	953
Guns	58058	1069	1069 * 58059	1069
Gay rights	58058	2064	2064 * 58059	2064
Health Care	58058	667	667 * 58059	667
Aggregated	58058	7134	7134 * 58059	7134

Each document is checked based on the words in the WordNet dictionary and existence of each word in each document is shown by 1, otherwise it is 0. Hence, a row in this matrix will be a feature vector corresponding to a document.

The resulting matrix has several zero columns which show that the head word of the column has not been used in any documents in the dataset at all. Thus, they are not informative and can be removed from the matrix.

Bag of Words is a supervised learning approach as the truth sense of word. This model trains a classifier by each word in each document. Consequently, it will give similar results for similar

[*1] All the words in WordNet dictionary are in their lemmatized form.

[*2] MorphAdorner is a Java command line program which acts as a pipeline manager for processes performing morphological adornment of words in a text. www.morphadorner.northwestern.edu

documents. For instance the two sentences below are close to each other in terms of Euclidean distance from the Bow point of view and receive the same labels in testing procedure.

- 1) I do likeswimming
- 2) I do likeapple.

While the two sentences below have totally opposite polarity, but likewise their feature vectors are close to each other according to BoW model and mistakenly receive the same labels which is surely wrong.

- 1) I do likeswimming
- 2) I do notlikeswimming

This sort of BoW mistakes convinced us to look for other feature models that are more accurate and take into account the details which BoW misses.

4.1.3 Bag of Words with Part of Speech Tags

Another common feature that has been studied in this research is the combination of the BoW model with part of speech tags.

In this model each word in a corpus is indicated with its corresponding part of speech which is determined based on both its definition as well as its context. Due to the fact that some words can represent more than one part of speech at different times, and since some part of speech are complex or unspoken in natural language, part of speech tagging is a tough job in order to be done accurately. For instance, “dogs” is usually considered as plural noun while it also can be a verb.

“The sailor dogs the barmaid”

Although part of speech tagging is very time consuming, it is broadly used in textual classification, merely because it describes documents with much more details. Grammatical information which it adds to feature vector of each document, is of enough value to be taken into consideration in sentiment analysis.

Providing that in this research supervised classification is conducted, this model can be very useful due to the fact that it extracts much more details from training.

In this research in order to use the BoW with PoS tags model, the same pruning procedure as BoW should be applied on dataset before the tagging phase.

In the tagging phase, the part of speech tagger^[*1], grants a PoS tag to each word based on the context as well as the definition of it. The tags are categorized to 17 major word classes which are shown in Table 4.2. Since all these classes are not informative for sentiment analysis, we have

[*1] MorphAdorner is a Java command line program which acts as a pipeline manager for processes performing morphological adornment of words in a text. www.morphadorner.northwestern.edu

reduced the part of speech tags to 5 classes of “verb”, “noun”, “adjective”, “adverb” and “negation” such that the “verb” tag refers to pure verbs which typically expressing action. Therefore, “to be verb” which expressing state or “auxiliary verb” is not considered as verb in this study. The chief reason is that in sentiment analysis, only subjective action verbs express feelings while in the sentences with “to be verb” or “auxiliary verb” the sentiment is expressing by adjectives or adverbs. The latter is shown in examples below:

- 1) I love my mother.
- 2) It is a nice car.

Also, all the negation indicators are summarized to “Not” in the negation tag.

In order to construct the matrix of each corpus, based on this model, each word with its PoS tag in a document is checked by the WordNet dictionary and the 5 tags mentioned, thus, in the case of equality the corresponding element in the feature vector turns to be 1, otherwise it is 0.

According to the fact that negation plays a major role in determining the polarity of a document, we decided to pay special attention to it and investigated on finding several negation patterns in documents such as:

<verb><negation particle><verb>
 <verb><negation particle><adjective>/<adverb>
 <verb><negation particle><conjunction><adjective>/<adverb><conjunction>

Table 4.2: 17 major POS tags

Major Word classes
adjective
adv/conj/pcl/prep
adverb
conjunction
determiner
foreign word
interjection
negative
noun
numeral
preposition
pronoun
punctuation
symbol
undetermined
verb
wh-word

But given the fact that there are many patterns to show negation in natural language and it is not possible to model all of them, we have come up with a new idea which was granting more weight to negation indicator “not” in the feature vectors of documents that contained it. This was done by adding another “not” to the WordNet dictionary^[*1], and therefore for those documents contained negation patterns, the weight of “not” was doubled in their feature vectors.

As a result, the difference between the feature vectors of the sentences below becomes more visible than before in BoW model.

[*1] WordNet dictionary has already adverb “not”.

- 1) I do like swimming.
- 2) I do not like swimming.

As in the BoW matrix, there are several zero columns in this matrix which show that special combinations of WordNet dictionary and PoS tags such as “beautiful _ verb” have not been used in any of the documents and therefore can be removed from the matrix which helps to reduce the matrix dimensionality.

Although the BoW with PoS tag model will be a significant help in sentiment analysis, it is not typically designed for opinion classification, hence, we decided to investigate on those features that can describe better the behavior of natural language.

4.2 Selected Features

The common features approaches show that the sentiment of a text can be found from the words in that text, but not all the words are equally important. For instance the word “lovely” can better express the sentiment while the word “God” cannot express any feelings. On the other hand, some times the combination of words matters. For example in the sentence “I guess you take me for granted”, the idiom “take for granted” shows the negative polarity while none of the consisting terms of this sentence is a sentimental word.

These indications motivated us to recognize the most common word patterns in opinionated documents as well as selecting popular subjective words in order to obtain more accurate classification. Moreover, these selected features lead to construct much smaller size matrices in comparison with huge matrices built by BoW and BoW with PoS tags which helps us to save a big portion of time in MATLAB based processing procedure.

In pursuance of fulfilling this objective the predominant assumption is Pointwise Mutual Information (PMI) which is explained in details in chapter 1. Based on this association measure, we can select those patterns that are of a high quality and more probable to express the dominant sentiment of a document.

From the statistical point of view, the PMI of a pair outcomes x and y belonging to discrete variables X and Y quantifies the discrepancy between the probability of their coincidence given their joint distribution and the probability of their coincidence given only their distributions, assuming independence. Mathematically:

$$pmi(x; y) = \log \frac{p(x, y)}{p(x)p(y)} = \log \frac{p(x | y)}{p(x)} = \log \frac{p(y | x)}{p(y)}$$

The measure can take positive or negative values, but is zero if X and Y are independent. PMI maximizes when X and Y are perfectly associated [45].

4.2.1 Pointwise Mutual Information Based on Positive and Negative Labels

One of the approaches that we can determine which words are mostly used in natural language to express positive or negative sentiment, is calculating the PMI of each word in a document with respect to the label of that document in the training phase. For instance the PMI of the word “Excellent” is positively higher in “Positive” labeled documents than in “Negative” labeled documents where the PMI is negatively lower.

Intuitively, by this approach, we can select those words that have higher PMI in positive and negative documents, which means they are mostly used to express positive and negative sentiments and consequently can lead to better classifiers. On the other hand, as we can select the best features (words) based on their PMI rank, we can manage the dimensionality of the feature vector for each document and therefore construct a more reasonable matrix in terms of dimensionality per each dataset, which implies faster classification.

In this research we have been taken the advantage of datasets with part of speech tags. Although this model can be applied on the BoW documents as well, we have decided to use PoS tags since it presents us with many advantages. The grammatical information it adds to a word, helps the PMI procedure to find more correct PMI rank for that word. For instance for word “fat” we can see the influence of PoS tag in determining the polarity of it:

fat_noun → especially profitable or advantageous work (positive polarity)

fat_adjective → having too much flabby tissue (negative polarity)

Technically speaking, the Pointwise Mutual Information for each word in a dataset based on existence in positive and negative documents and is calculated as follows:

$pmi(word; positive\ documents) =$

$$\log \frac{\left(\frac{\text{number of documents labeled positive and contain the word}}{\text{total number of documents in dataset}} \right)}{\left(\frac{\text{number of document labeled "positive"}}{\text{total number of documents in dataset}} \right) \left(\frac{\text{number of documents contain the word}}{\text{total number of documents in dataset}} \right)}$$

Likewise for $pmi(word; negative\ documents)$.

Therefore, for each word in a dataset we have 2 PMI ranks which relate to documents labeled “positive” (positive documents) and documents labeled “negative” (negative documents). Since the range of PMI for each word can differ from $-\infty \leq pmi(x; y) \leq \min[-\log p(x), -\log(y)]$, the results can be positive or negative. Thus, in order to compare words (features) with each other, it is wise to normalize the PMI ranks.

Pointwise Mutual Information can be normalized between $[-1, +1]$ resulting in -1 for never occurring together, 0 for independency and +1 for complete co-occurrence.

$$\text{normalized pointwise mutual information}(x; y) = \frac{pmi(x; y)}{-\log[p(x, y)]}$$

Although NPMI could be helpful, we have not used it in our study merely because the number of available documents with societal theme was comparatively low and the intersection of words with positive or negative documents to calculate $p(x, y)$ was not high enough to satisfy the Java accuracy for calculating $\log[p(x, y)]$, therefore, the results of NPMI were 0 for most of the words. This made us to use another normalization method namely Sigmoid.

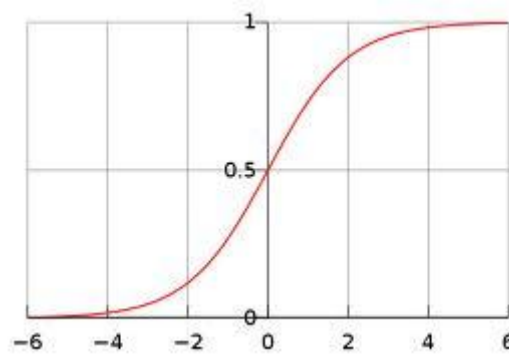


Figure 4.4: Sigmoid Curve

The sigmoid curve is produced by a mathematical function, having an “S” shape and changes between $[0, 1]$. The normalized PMI will be 0 if two words never occur together, 0.5 for independency and 1 for complete co-occurrence.

The sigmoid function is:

$$f(x) = \frac{1}{1 + e^{(-x)}}$$

Having 2 sigmoid normalized PMI ranks for each word, with regards to positive and negative documents, makes it possible to extract informative words in terms of expressing positive opinion or negative sentiment. Those words with sigmoid normalized PMI rank greater than 0.5 tend to express sentiment more than other words. Therefore we have decided to choose the top first 20% of words for each group (positive and negative). As result, we had 2 groups of words with their Part of Speech tags which have been used in most of the documents and could be considered as informative features.

Table 4.3: Sigmoid Normalized PMI Rank

Word	PMI in Negative documents	PMI in Positive documents
Happiness_noun	0.3841	0.7521
Rape_verb	0.6871	0.2908

I should press upon that in the list of sigmoid normalized PMI ranks for both groups of documents, there were some words with the highest PMI rank that have been removed from the top first 20% due to the fact that they were of low quality since their high PMI rank were spurious. To be much precise, total number of documents was equal to the number of co-occurrence documents which was at most 5. Therefore by removing those trinkets, we obtained 2 lists of informative words.

The combination of these two lists gives the feature vector of the documents in the dataset.

4.2.2 Pointwise Mutual Information Based on Positive and Negative Labels & “good” and “bad” terms

Generally speaking, natural language arises in an premeditated fashion as the result of the innate facility for language possessed by the human intellect. Although natural language is typically created for communication the range of words that are being used in it, is not wide. According to the Oxford English Dictionary there are 171476 current English words (obsolete words are not included) while less than half of them are used in normal speaking. It is noteworthy to know that the top 25 commonest words with adjective part of speech contain “good” and “bad” terms which are popular subjective words to express sentiments^[*1].

The latter convinced us to look for those documents which contained the “good” and “bad” terms and extract their structural pattern of words . As it was expected, it was observed that documents contained a positive indicator, had positive polarity and documents with negative subjective terms that had been labeled with negative polarity. In addition, it was observed that in most of documents which contained the “good” term (positive subjective indicator) but had negative polarity, the adverb “not” existed. For instance:

- 1) “The **good** point of democracy is that non-elected citizen, who has greatest to gain and most to lose in a political situation, has the right to determine by popular vote, their elected leaders”. (**positive** polarity)
- 2) “The gay marriage is a very **bad** idea because it has nothing to do with liberty and everything to do with providing the elite with a new moral mission”. (**negative** polarity)
- 3) “The right to defend yourself is a **good** part of human right. As extension of that, as it is in law, is equality of arms. If the police and criminals have guns, so should the rest of us, but on the whole I’m **not** in favor of arming children”. (**negative** polarity)

These indications encouraged us to extract the words with their PoS tags from the documents containing “good” and “bad” terms and calculate their PMI ranks with regard to the existence of them in document containing the mentioned subjective terms.

[*1] The OEC: Facts about the language... www.oxforddictionaries.com/words.

Mathematically speaking:

$$pmi(word; good\ documents^{[*1]}) =$$

$$\log \frac{\left(\frac{\text{number of "good" documents which contain the word}}{\text{total number of documents in dataset}} \right)}{\left(\frac{\text{number of "good" documents}}{\text{total number of documents in dataset}} \right) \left(\frac{\text{number of documents contain the word}}{\text{total number of documents in dataset}} \right)}$$

The same holds for $pmi(word; bad\ documents^{[*2]})$.

The same sigmoid normalization has been applied on PMI ranks. Also likewise in positive/negative PMI ranks, each word had 2 PMI ranks according to “good” and “bad” documents. The first top 20% words with PMI greater than 0.5 have been selected.

But we found that these words are not informative enough to discriminate between positive and negative documents due to the examples below. In these examples, although we have a positive indicator, the polarity of the document is negative and when we have a negative indicator, the polarity of document is positive:

- 1) “Whether universal social healthcare is good inconsequential? this question have been answered that have no bear on whether universal system can provide level of service proponent claim” (negative polarity)
- 2) “Public healthcare just receives more scrutiny criticism by public because they are public accountable, this often gives impression that these systems are bad But this impression is simply bi-product of high level of scrutiny, these system receive” (positive polarity)

Therefore we made our mind to use the equal words between selected normalized PMI words based on positive/negative labeled documents and selected normalized PMI words based on “good” / “bad” documents, in order to construct the feature vector. The motivating idea was that the words which have been used frequently in positive labeled documents as well as “good” documents can be the best representations of positive attitude while popular words in negative labeled and “bad” documents can quid us to negative sentiment.

Feature vector of each document constructed based on these intersected words, has 335 dimensionalities.

4.2.3 Pointwise Mutual Information based on Positive and Negative labels & Adjective and Negation Part Of Speech Tags

As discussed before, the words which have been selected from positive and negative documents in

[*1] Documents contained “good” term.

[*2] Documents contained “bad” terms.

order calculate their PMI ranks; have accompanying part of speech tags. These PoS tags consist of 5 major classes: verb, noun, adjective, adverb and negation, but not all of them are informative in determining the polarity of a document. For instance nouns hardly can express feelings while adjectives and adverbs always can. Or negation can easily change the polarity of document while some verbs cannot. Examples below show the influence of some PoS tags in comparison with others.

- 1) "Many Americans who pay taxes are opposed to abortion; therefore it is morally **wrong/adjective** to use tax dollars to fund abortion".
- 2) "Abortion is **not/negation** a **viable/adjective** alternative to the means of contraception".
- 3) "Basically people **dislike/verb** abortion because you are killing a beautiful human being before it even has a chance to come into this world".

These three examples express negative sentiment because of their adjective, verb and negation part of speech tags while in the example below, since there is no adjective, adverb or negation, the author's opinion is not indicated.

- 4) "A majority, perhaps as many as 75 percent, of abortion clinics are in areas with high minority populations. Abortion apologists will say this is because they want to serve the poor. While they take their money to terminate their children".

Although some verbs such as love, hate, like and dislike show the direct polarity of a sentence, due to the small number of these sorts of words, we have overlooked them and just investigated on adjective, adverb and negation part of speech tags.

In other words, the first 20% words which have been selected based on their normalized PMI from positive and negative labeled documents, were sifted again in terms of their part of speech tags and those words with adjective, adverb and negation PoS tags have been chosen in order to construct feature vectors of documents in a dataset.

It should be stressed that this model could also been applied on words that have been selected based on their normalized PMI from positive, negative and good, bad documents which leads even to more informative and discriminative features, but due to the comparatively small number of documents and lack of enough training data, we did not investigate on that.

4.3 Feature Extraction by PCA

After constructing 5 different matrices for each dataset based on the 5 discussed feature sets, these matrices should be automatically transformed, such as filtering zero columns due to non-informativity and extracting eminent eigenvectors of covariance matrices of datasets in order to find important directions of data which are interpreted to much common features. Based on these common features, new matrices are constructed for each dataset which are given to classifiers for training. In this research, we have used Principal Component Analysis (PCA) which is one of the popular feature extraction methods.

Generally speaking, the basic idea of the curse of dimensionality is that high dimensional data is difficult to work due to the fact that adding more features can increase the noise, and hence the error and there are not enough observations to get good estimates; therefore, dimensionality reduction which is the process of reducing the number of random variables under consideration, takes the precedence. With fewer parameters accuracy estimation can be done faster and easier. Since we want to improve the results by throwing away information therefore **extracting** “explaining” variables (**features**) is of a significant importance. Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions.

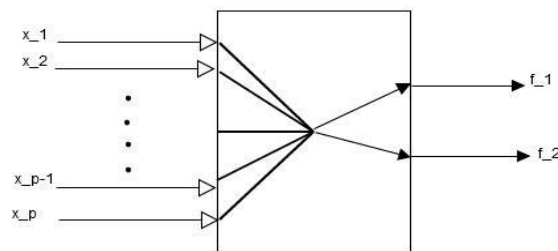


Figure 4.5: Feature Extraction maps p measurements to d measurements

Principal Component Analysis (PCA) is the main linear technique for dimensionality reduction which performs a linear mapping of the data to a lower dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. That is to say, it finds directions in data which retain as much variation as possible and projects data on those directions.

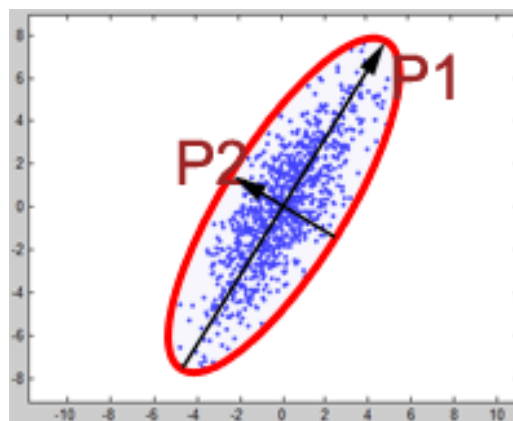


Figure 4.6: PCA finds the most variant directions

The first principal component is the eigenvector corresponding to the first largest eigenvalue of the covariance matrix of data. The second component is the eigenvector corresponding to the second largest eigenvalue of the covariance matrix of data, etc. These orthogonal principal components make projected data uncorrelated which minimizes the squared reconstruction error.

In this research, PCA is used in order to find the best number of features which can describe the opinionated documents. We have started our experiments with a large number of features and reduced them with PCA. Based on the classification accuracy of values obtained by the matrices constructed by these extracted features, we have reached to the best number of features in terms of the “confidence intervals of the number of features” which can express a sentimental document which is given in the next chapter.

4.4 Cross Validation

Classification is applied on the matrices derived from PCA. In this procedure, classifiers are trained based on given examples which have been labeled before, and tested on unlabeled data. In order to achieve a good classification, we need a large training set to train classifiers well and large test set to obtain reliable and unbiased error estimates. But in this research, we have just a single design set which should be used for training as well as testing. Therefore, use cross validation which is especially designed for situations that further samples are hazardous, costly or impossible to collect. In this study, we have used 10-fold cross validation which randomly partitions the given dataset into 10 parts. Of the 10 parts, a single part is retained as the validation data for testing and the remaining 9 parts are used as training data. The cross validation process is then repeated for 10 times (the folds), with each of the 10 parts used exactly once as the testing data. The 10 results from the folds are then averaged to produce a single estimation.

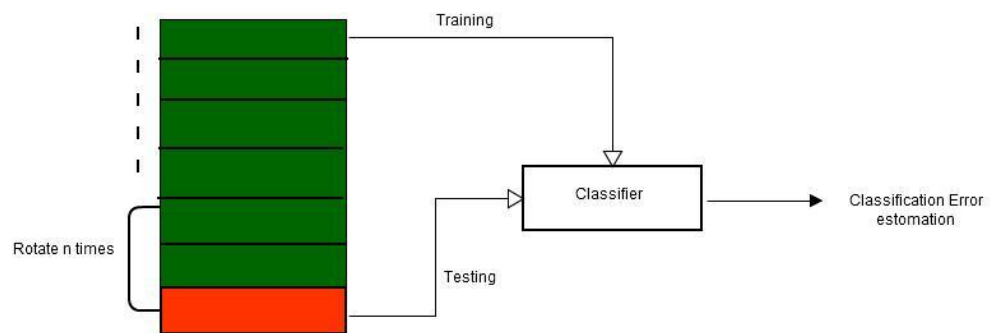


Figure 2: Cross Validation

In cross validation all documents are used for both training and testing, and each document is used for validation exactly once. The results obtained by cross validation show the worst case accuracies and in practice they are lower than other means of validations. Also, in order to obtain the most reliable results, we have done cross validation with 10 times repetition and averaged all over the results.

Chapter 5: Experiments and Evaluations

In the previous chapters, we have explained different feature sets which have been investigated as well as constructed matrices based on them. Mathematical optimizations which have been imposed on constructed matrices were discussed as well as the procedure. Now, in this chapter, it is time to evaluate the results obtained by the experiments in order to answer to the research questions of this thesis, mentioned in chapter 2.

5.1 Bag of Words

As far as the BoW feature set is concerned, each of the 6 small datasets as well as the large dataset (Aggregated Dataset) is represented in terms of matrices. The detailed information of these matrices is given in Table 5.1.

Table 5.1: Matrices constructed based on datasets in Bag of Words Model

Datasets	WordNet Dictionary	Documents	Matrix	Labels	Positive Labels	Negative Labels
Aggregated	58058	7134	$7134 * 58059$	7134	3964	3170
Abortion	58058	1151	$1151 * 58059$	1151	630	521
Creation	58058	1230	$1230 * 58059$	1230	451	779

After removing zero columns from these matrices due to non-informativity, the label of each document in each matrix was attached to it to make a new set which had to be given to PCA. Then, PCA was applied on these new sets in order to reduce the number of features. We set PCA to extract 50 to 500 features with intervals of 50 from two small sets (Abortion and Creation) and one large set (Aggregated). Each group of three mapped sets was tested with each of the classifiers listed in the Table 5.2 using 10 fold cross validation. In order to assure the results, this procedure was repeated 10 times and the given mean error as well as standard deviation is the average of the previous results.

Table 5.2: Classifiers

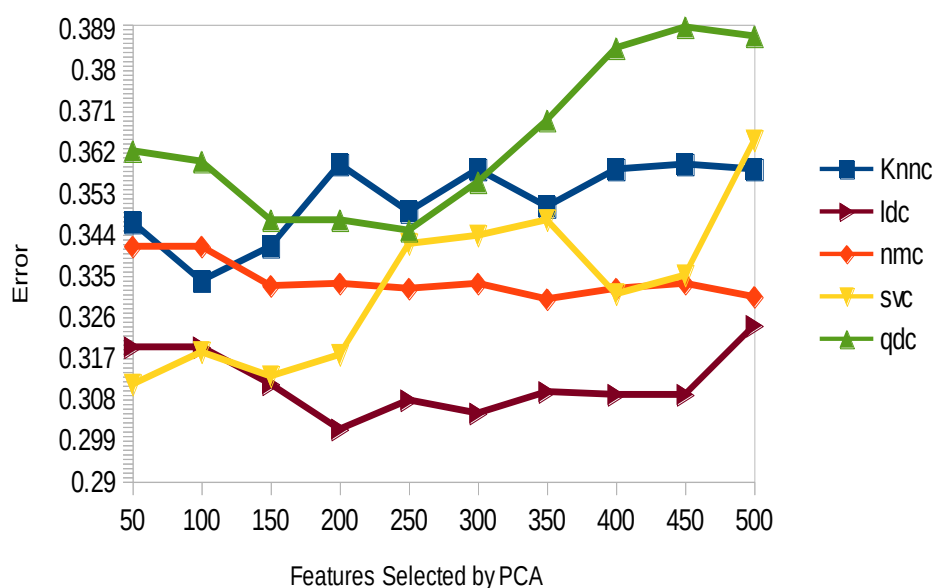
<i>Classifiers</i>
LDC
QDC
NMC
KNNC
SVC

It was observed in the results of these two sorts of datasets (Figure 5.1), the confidence interval of the number of features^[*1] were determined as 100 to 250 which gave higher accuracies in terms of different classifiers in contrast with other number of features extracted by PCA. Also, LDC and SVC

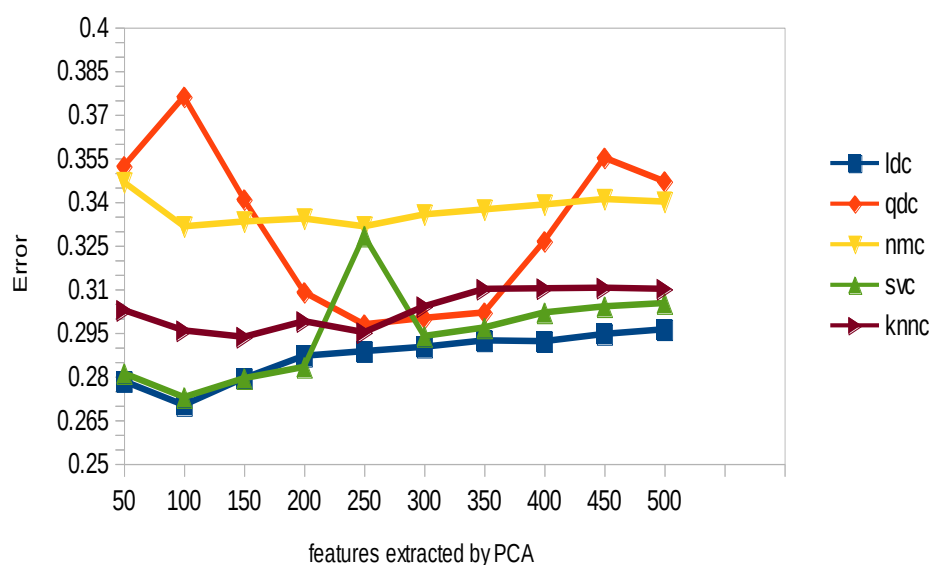
[*1] "Confidence interval of the number of features" refers to the number of features that produce the least error and the highest accuracy in the binary sentiment classification. In this study, we try to reach to the interval of number of features which are optimized in terms of accuracy.

respectively, predicted the labels of test data better in comparison with other classifiers. According to the fact that these classifiers are linear classifiers and outperformed non-linear ones, we can conclude that linear combination of words such as (x and y) is better. That is to say, these classifiers can discriminate between “good” and “not good” better than KNNC and QDC classifiers.

Bag of Words Model on Abortion



Bag of words Model on Creation



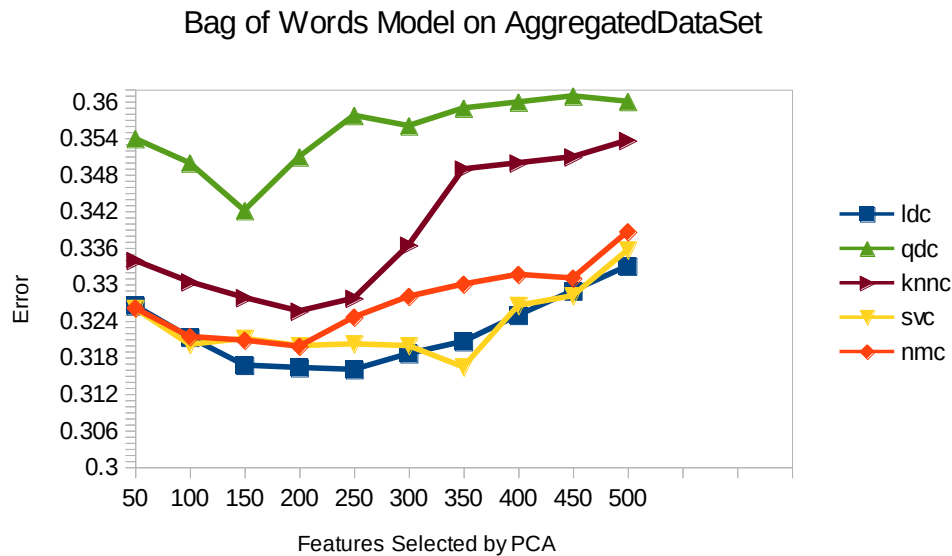
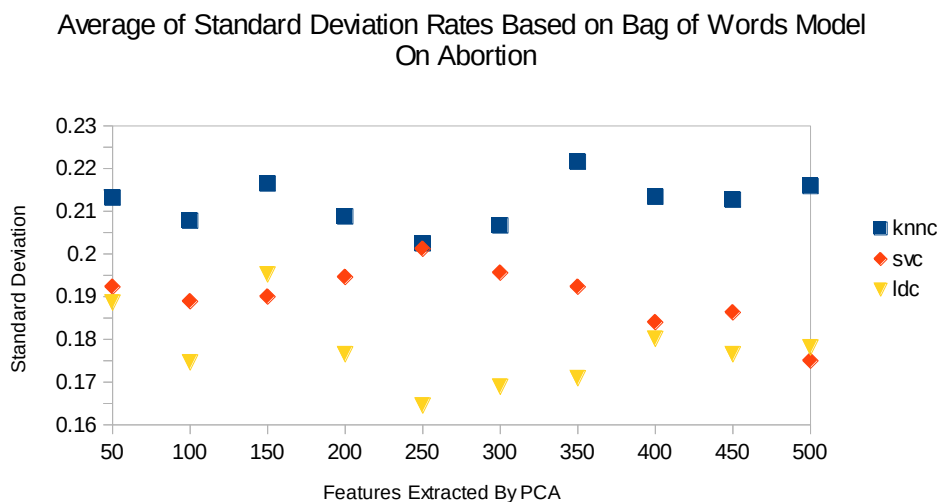


Figure 5.1: comparison between classifiers based on large and small size documents

Since non-linear classifiers are sensitive to features, in this case (words) in comparison with linear ones, and according to the nature of natural language documents in which for instance sarcasm or even one punctuation mark such as “?” can change the polarity of a sentence, non-linear classifiers are not good for sentiment analysis and as you will see in next experiments, they never outperform linear classifiers in terms of common features. That is to say in the case of strange behavior of natural language, linear classifiers treat them as misclassification, while non-linear classifiers try to adapt themselves to those examples, therefore they are overtrained due to redundant features and produce higher error rate.

It would be fair to say that SVC in this feature set is of a low quality in comparison with LDC, in contrast with the predominate attitude in sentiment analysis, merely because of higher mean error obtained by cross validation. In addition the time which has been consumed by SVC was 10 times more than LDC while it gave a much higher error rate. It should be stressed that this extra time is increased quadratic by the size of dataset.



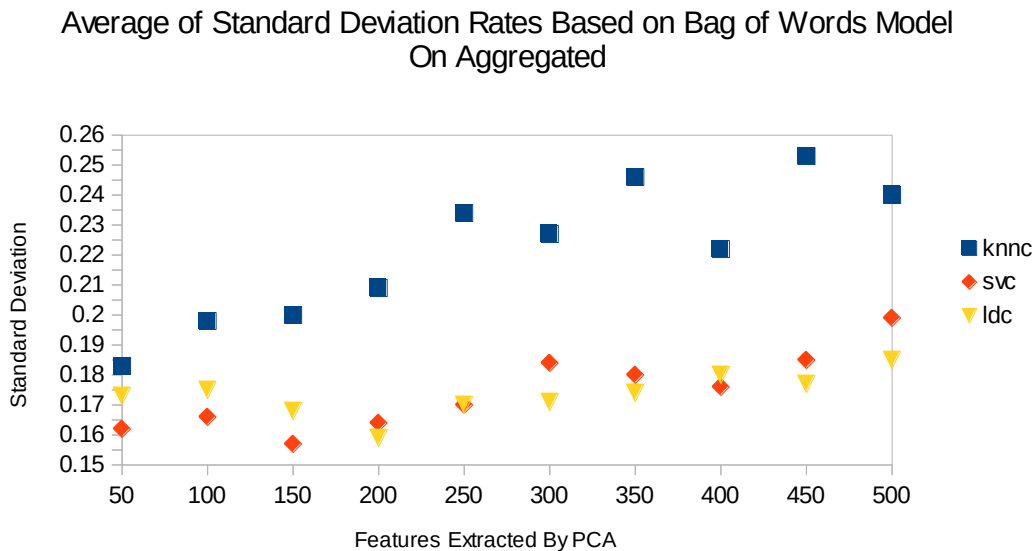
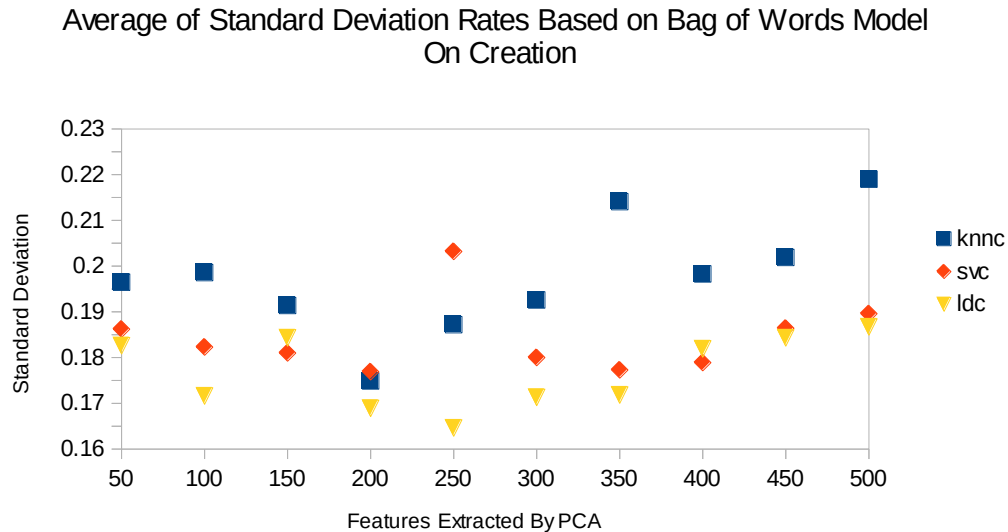


Figure 5.2: comparison between the averages of Standard Deviation of classifiers based on large and small size documents

According to the average of the mean standard deviation of classifiers (Figure 5.2), the graph of the non-linear classifiers, in this case KNNC, are fluctuating due to the fact that they are sensitive to the data and their results variant by different features (due to overtraining) while linear classifiers (LDC and SVC) are more stable and fluctuate less.

To wrap up, we concluded that as far as the BoW model is concerned, LDC outperforms in both small and large datasets within the interval (100,250) of number of features extracted by PCA. The best results achieved in this part are shown in Table 5.3.

Table 5.3: The best results obtained based on Bag of Words model

Small Dataset	Large Dataset	Classifier	
0.2868	0.3222	LDC	Min (Mean Error Rate)
0.1645		LDC	Min (Mean Standard Deviation)
	0.1570	SVC	Min (Mean Standard Deviation)

I should press upon that the results for other 4 datasets are given in Appendix.

5.2 Bag of Words with Part of Speech Tags

As for the previous model, we have constructed 3 matrices based on 2 small and 1 large datasets. After filtering zero columns, motivated by the confidence interval of the number of features were obtained in the Bow model, we set PCA to extract 50 to 250 features with intervals of 50.

Table 5.4: Matrices constructed based on datasets in Bag of Words with Part of speech Tags Model

Note that PoS tags are: verb, noun, adjective, adverb and negation.

Datasets	WordNet Dictionary	PoS Tags	Documents	Matrix	Labels	Positive Labels	Negative Labels
Aggregated	58058	5	7134	7134 * 348354	7134	3964	3170
Abortion	58058	5	1151	1151 * 348354	1151	630	521
Creation	58058	5	1230	1230 * 348354	1230	451	779

In this model, the confidence interval of the number of features (words) extracted by PCA remained the same as in the previous model, [100,250] while these are those words that co-occurred in every or most of documents. According to the PoS model, it is possible that these words contain some terms that are not sentimentally informative at all such as “go_verb”, but because they are popular words in normal talking, they produce one of the main directions of the data. The latter is the motivating idea for “PMI based on Positive and Negative & Adjective and Negation PoS tag” model. It is noteworthy to know that the first 50 features extracted by PCA cannot find considerably discrimination between classes due to the same reasoning, therefore, produces higher error rates and are not improving the results.

Similar to BoW, in this model, LDC and SVC outperform the other classifiers. This could be predicted since the nature of these two models is similar, just part of speech tags add extra information to the previous model which influences the results. In other words, the mean errors rates of BoW with PoS tags are comparatively lower than those of the BoW model due to the grammatical information added by PoS tags.

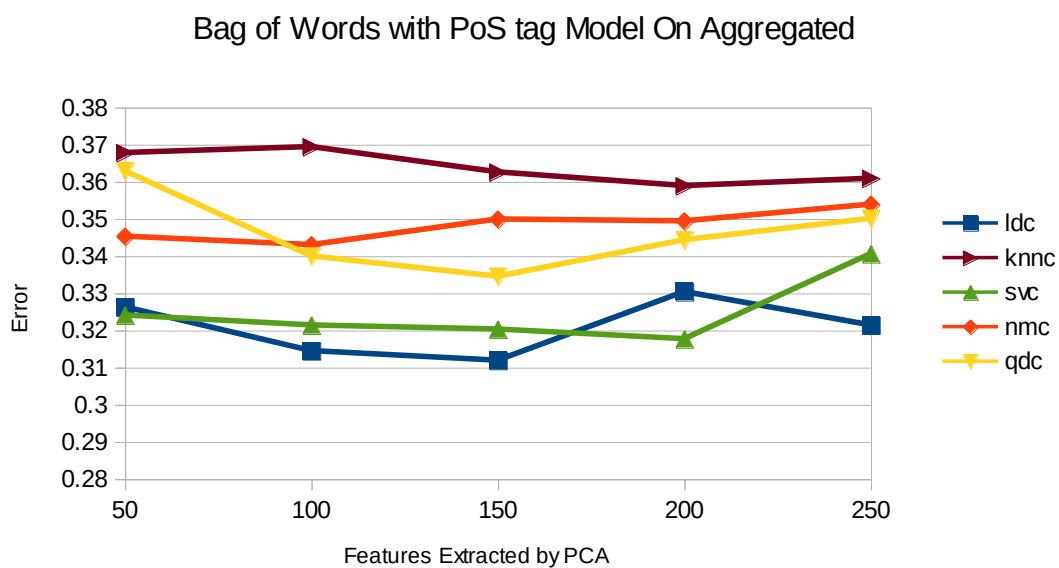
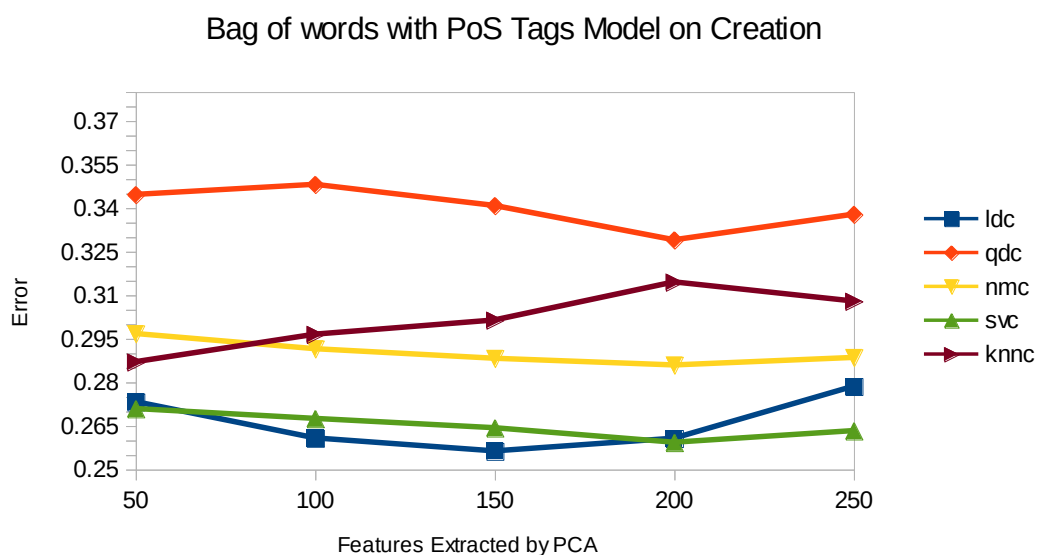
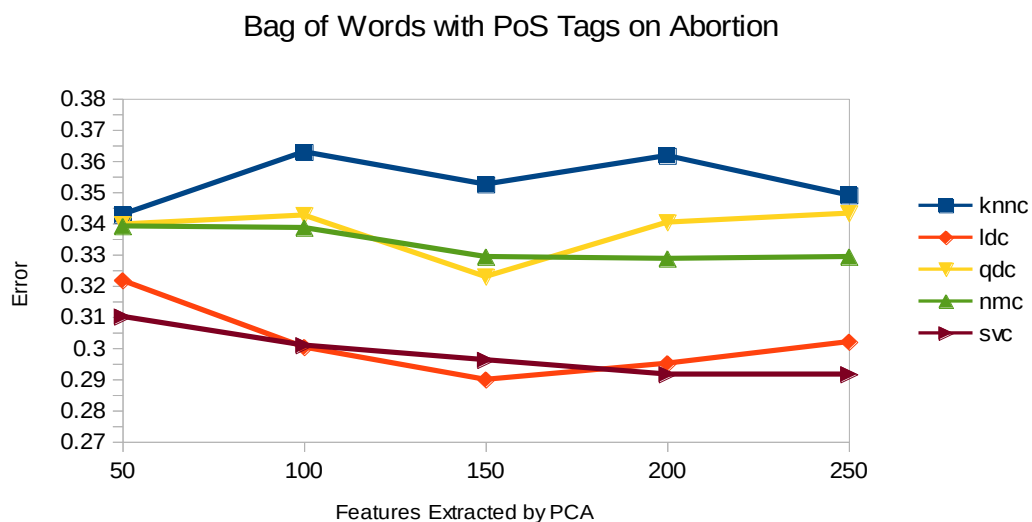


Figure 5.3: comparison between classifiers based on large and small size documents

It should be stressed that since the accuracy of classifiers changes based on features, and there is no large difference between results of the classifiers in this model, it shows that the features represent the properties of data in such a way that the documents are discriminative enough for different classifiers to set their decision boundary close to each other, that is to say there is a visible distance between positive and negative documents.

The best results achieved by this model are shown in Table 5.5.

Table 5.5: The best results obtained based on Bag of Words with Part of Speech Tags model

Small Dataset	Large Dataset	Classifier	
0.2659	0.3202	LDC	Mean Error Rate
0.1683	0.172	LDC	Mean standard Deviation

5.3 Selected Features

As mentioned before, selected features refer to those words that have been selected from positive and negative labeled documents such that their normalized PMI ranks in positive or negative documents are higher than 0.5 which means that they have tendency to express dominant positive or negative sentiment. The main model based on these features is called PMI rank based on positive and negative documents and the detailed information about the constructed matrices based on it is given in the table below.

Table 5.6: Matrices for “PMI Rank Based on Positive/Negative Documents” model

Note that the dictionary which is used in this model is constructed based on the 20% top words selected by this model from Aggregated dataset

Dataset	Dictionary	Documents	Matrix	Labels	Positive Label	Negative Label
Aggregated	669	7134	7134 * 669	7134	3964	3170
Abortion	669	1151	1151 * 669	1151	630	521
Creation	669	1230	1230 * 669	1230	451	779

Motivated by previous models and the results obtained for the confidence interval of the number of features to be extracted by PCA in order to achieve the best accuracy of classification, we have selected 50 to 250 features with interval 50 and observed that this interval reduced to [100,200] but still likewise the common features mode, the first 50 features are usually not informative due to the fact that they are words that are mostly used in every sentence in ordinary talking and not sentimentally informative to be useful for classification.

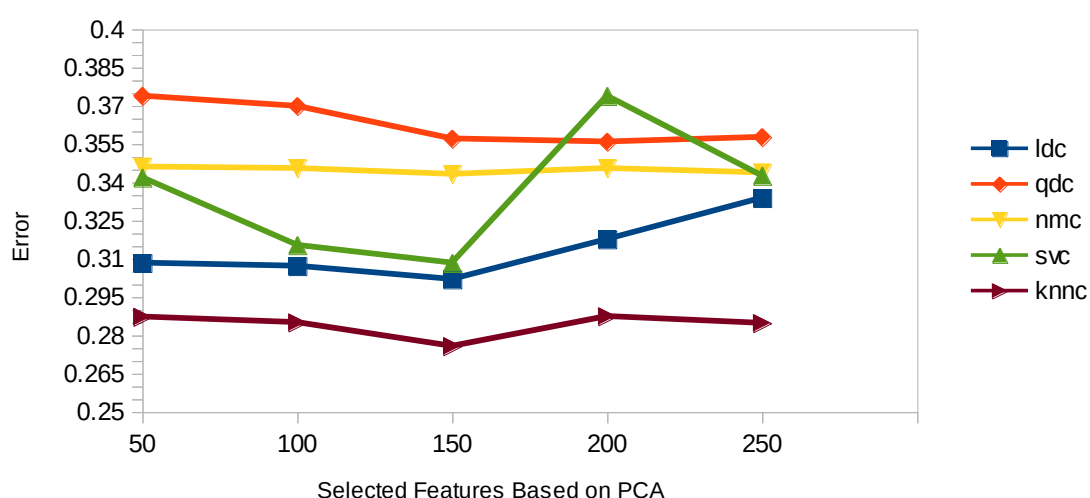
Table 5.7: Matrices for “PMI Rank Based on Positive/Negative Documents & Adjective and Negation Part of Speech Tags” model

Note that the dictionary which is used in this model is constructed based on the 20% top words which have adjective and negation part of speech tags selected by this model from Aggregated dataset

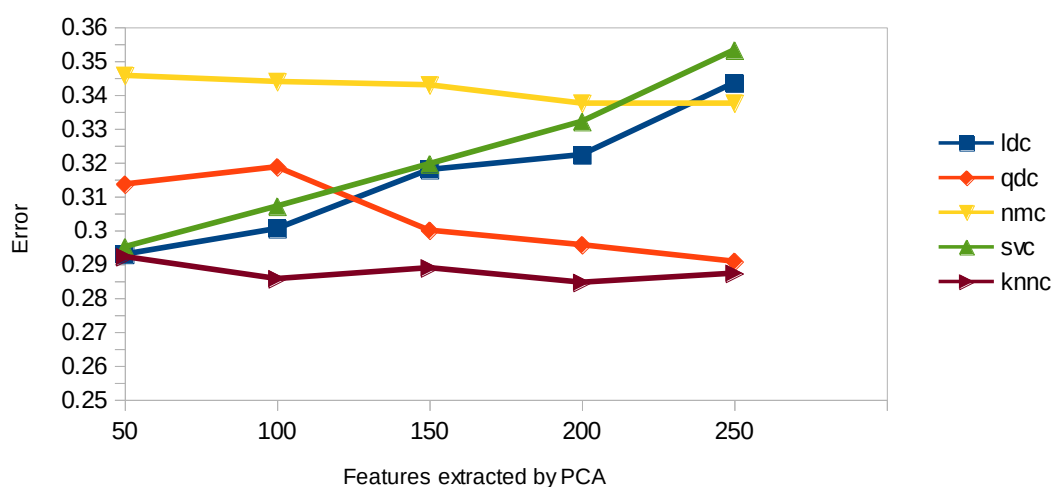
Dataset	Dictionary	Documents	Matrix	Labels	Positive Label	Negative Label
Aggregated	151	7134	7134 * 151	7134	3964	3170
Abortion	151	1151	1151 * 151	1151	630	521
Creation	151	1230	1230 * 151	1230	451	779

The amazing point about the results obtained by these experiments is that the nearest neighbor classifier (KNNC) outperforms the support vector classifier (SVC) as well as the linear discriminate classifier (LDC). The results gained by this classifier were considerably much accurate than other classifiers especially linear ones. This observation made us to check another model “PMI Rank Based on Positive/Negative Documents & Adjective and Negation Part of Speech Tags” which is the detailed form of the previous model, before making any conclusion about the observed fact. As it was anticipated, the same results was obtained based on this new model but this time with higher accuracies in comparison with previous model.

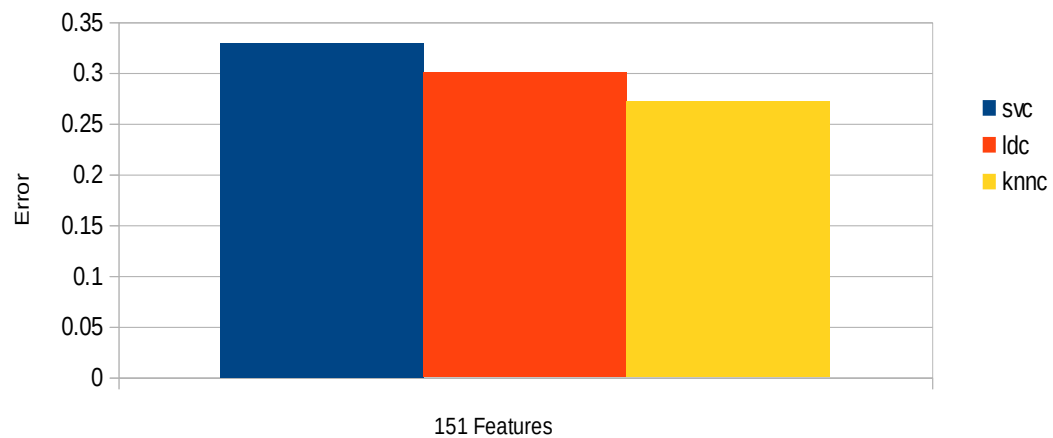
PMI Based on Posirive/Negative Documents Model on Abortion



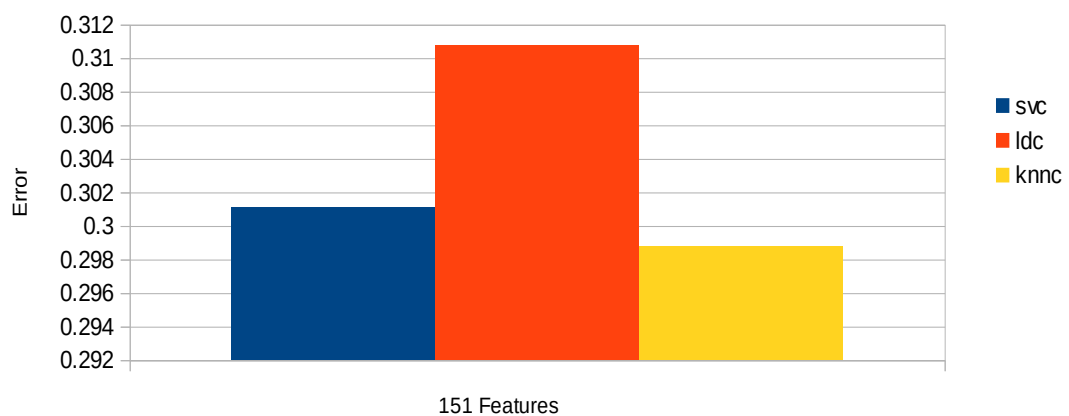
PMI Based on Positive/Negative Documents Model on Creation



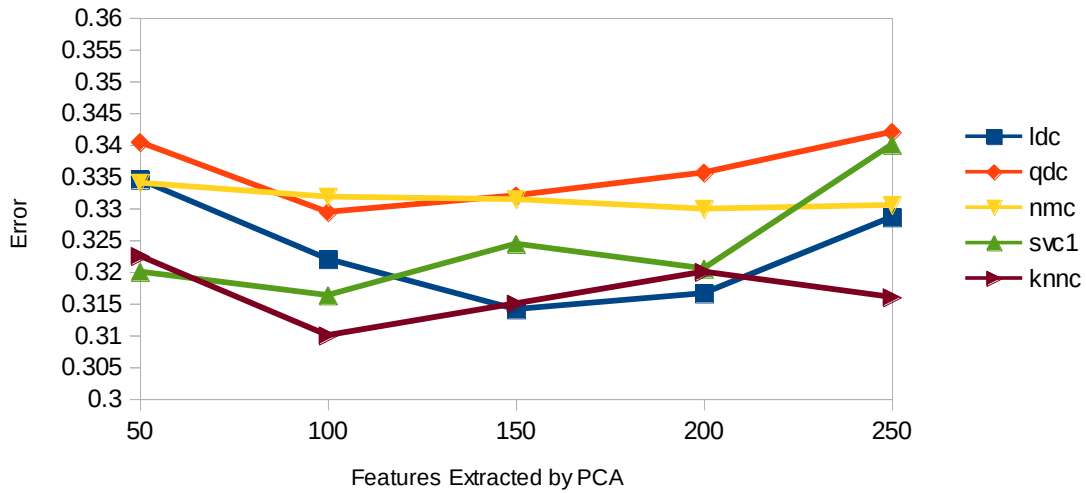
Mean Error Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Abortion



Mean Error Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Creation



PMI Based on Positive/Negative Documents on Aggregated



Mean Error Rates Obtained By Cross Validation for PMI Rank Based on Positive/Negative Documents & Adjective/Negation POS Tags On Aggregated

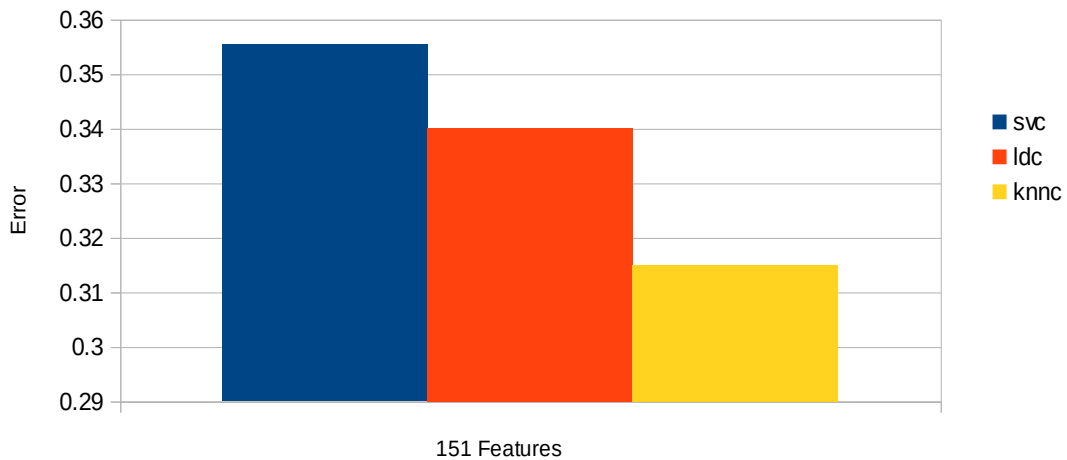


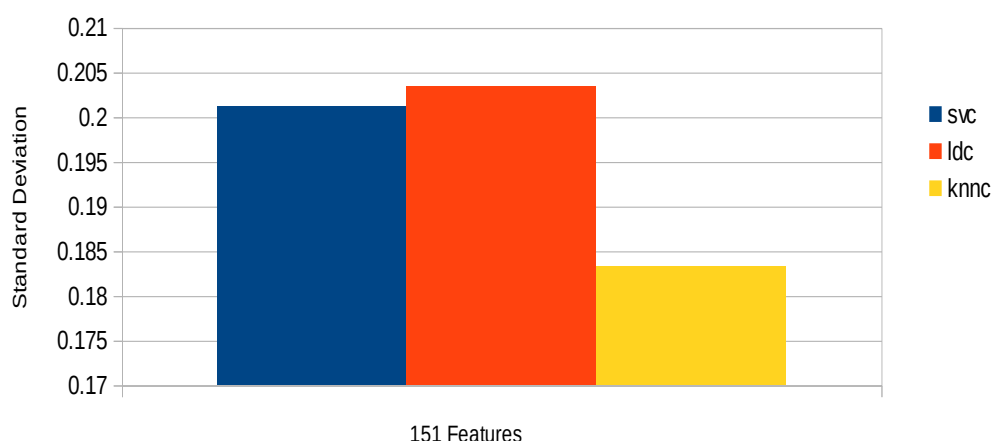
Figure 5.4: comparison between “PMI Rank Based on Positive/Negative Documents” and “PMI Rank Based on Positive/Negative Documents & Adjective and Negation Part of Speech Tags” models according to different classifiers on large and small size documents

It should be stressed that the dimensionality of the features vector constructed based on PMI rank Based on positive/negative documents& adjective and negation part of speech tags model is 151. That is to say the number of words with adjective and negation part of speech tags in the list of words with PMI rank higher than 0.5 which is used as dictionary for PMI rank based positive or negative documents model is 151. Therefore, we did not use PCA to extract features, simply because there are no or very few non-informative words among them. First of all, they are adjectives and adjectives are those part of speech tags that express the feeling the most. Secondly, they are selected from highly PMI rank words obtained by positive and negative documents and the number

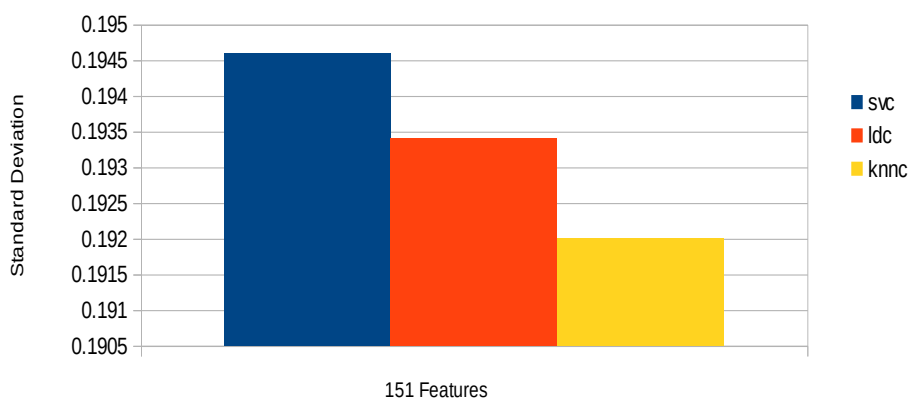
of them is 151 which according to the results obtained from conducted experiments in this study so far, is itself optimized number and stands in the confidence interval of number of features.

According to the standard deviation obtained by each classifier, the classification results on each number of features is less than 0.2, which shows that the variations of the error rates produced by the classifiers are not high and all the measurements are close to the average of the error rate.

Mean Standard Deviation Rates Obtained by Cross Validation for
PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Abortion



Mean Standard Deviation Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Creation



Mean Standard Deviation Rates Obtained by Cross Validation for
PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Aggregated

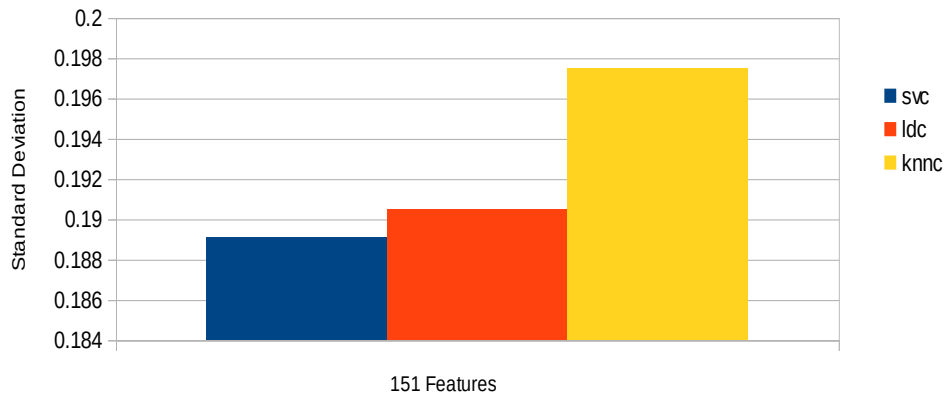


Figure 5.5: comparison Among Standard Deviation of different classifiers on large and small size documents based on “PMI Rank Based on Positive/Negative Documents & Adjective and Negation Part of Speech Tags” model

Providing that subjective sentences mostly express their sentiment with adjective part of speech tags and to show the opposite feeling use the negation tag, by having these key words we can discriminate between positive sentences and negative ones. On the other hand, those sentences that express positive feeling are similar to each other in terms of sentiment words used in them. This clarifies why the nearest neighbor classifier outperforms other classifiers. According to the principal of this classifier, the intuitive concept is that data instances of the same class should be closer in the feature space. As result for a given data of unknown class, the nearest neighbor simply computes the distance between the data and all other points in the training data and assigns the class determined by the k nearest examples to it. In this experiment k is optimized based on PRTTools. On the other hand, the nearest neighbor classifier can be easily overstrained by a large number of features which is the reason that it did not perform well in BoW and BoW with PoS tags. While if the number of features is comparatively small compared to the number of training examples (documents) and they are informative features as well. The nearest neighbor classifier can be trained well and predicts the label of the test data better. However, linear classifiers such as LDC and SVC used a linear function to map feature vectors into a hyper plane. If the hyper plane is higher than a threshold, then the corresponding feature vectors belong to one class, otherwise they belong to the other class. Therefore, these classifiers can also predict the label of test data accurately plus the fact that they can cope better with the high dimensionality of features and that is the reason that they performed well, in common features model.

Table 5.8: The best results obtained based on PMI Rank Based on Positive/Negative Documents & Adjective and Negation Part of Speech Tags

Small Dataset	Large Dataset	Classifier	
0.2713	0.3149	KNNC	Mean Error Rate
0.1833		KNNC	Mean standard Deviation
	0.1891	SVC	Mean standard Deviation

As far as “PMI Rank Based on Positive/Negative & Good/Bad Documents” model is concerned, determining the positivity or negativity of documents based on the existence of “good” or “bad”

terms in those documents, is not an accurate approach due to the fact that these sorts of documents contain redundant words which are not sentimentally informative. Based on the obtained results, these redundant words increase the noise and decrease the accuracy of classification.

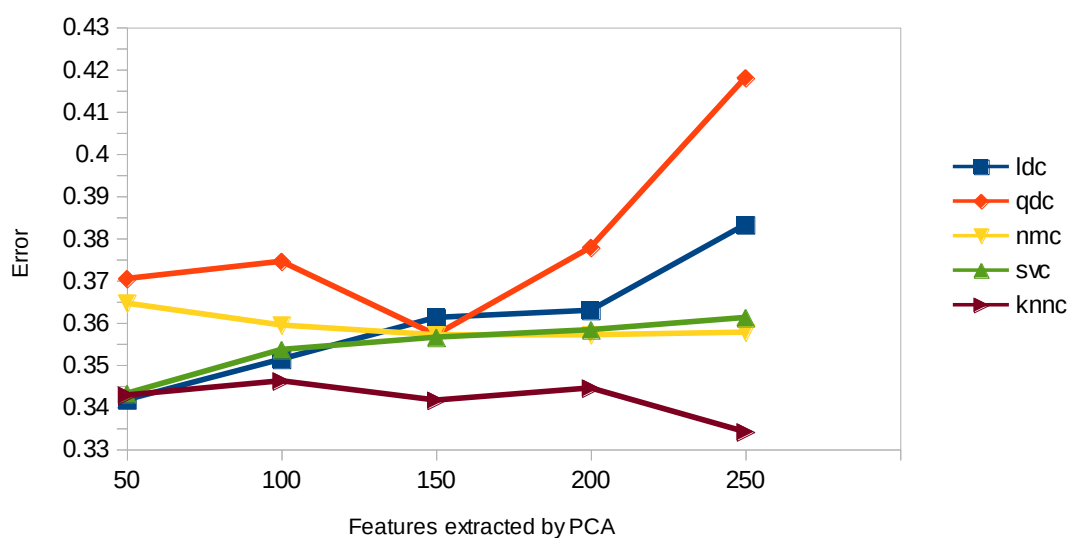
Table 5.9: Matrices for “PMI Rank Based on Positive/Negative & Good/Bad Documents” model

Note that the dictionary which is used in this model is constructed based on the 20% top words selected by this model from Aggregated dataset

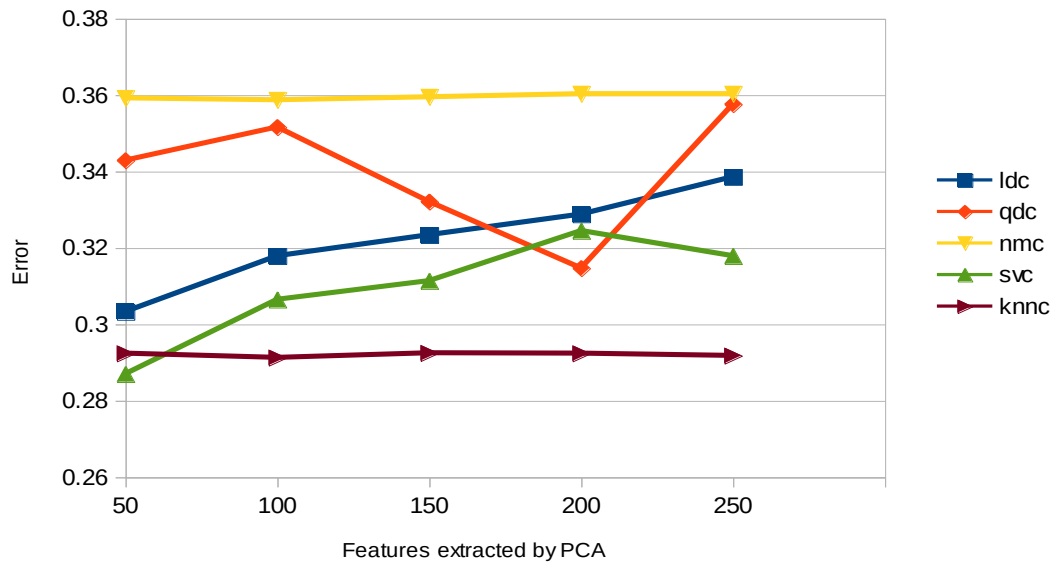
Dataset	Dictionary	Documents	Good Documents	Bad Documents	Positive Documents	Negative Documents	Matrix	Labels
Aggregated	335	7134	1376	449	3964	3170	7134 * 335	7134
Abortion	335	1151	1376	449	630	521	1151 * 335	1151
Creation	335	1230	1376	449	451	779	1230 * 335	1230

In conclusion, although the confidence interval of the number of features still is [100,200] and the nearest neighbor classifier outperforms in comparison with other studied classifiers, the error rate obtained by this model is higher than for the two other models and therefore not recommended until it has been investigated sufficiently in terms of removing the non-informative words.

PMI Based on Positive/Negative & Good/Bad Documents Model on Abortion



PMI Based on Positive/Negative & Good/Bad Documents Model on Creation



PMI Based on Positive/Negative & Good/Bad Documents on Aggregated

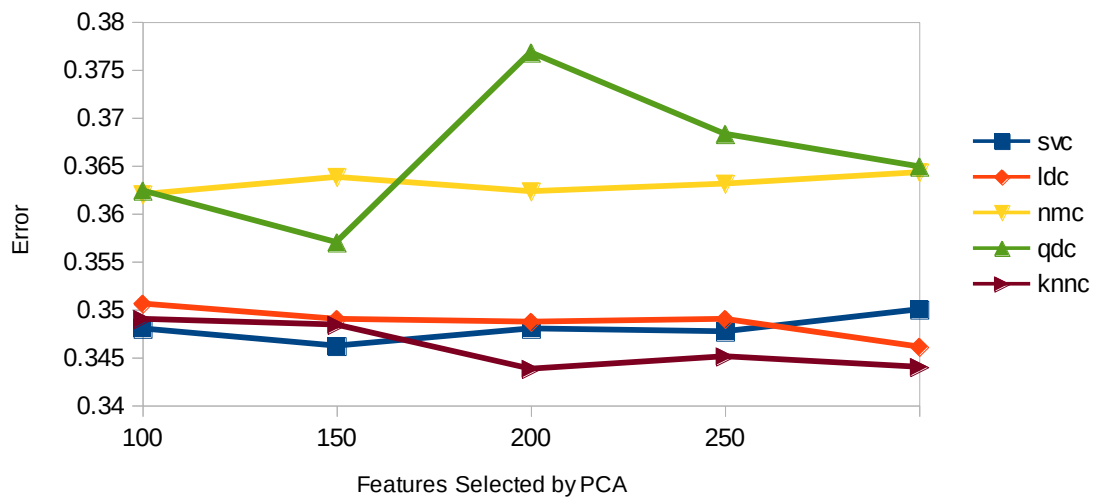


Figure 5.6: comparison between classifiers based on large and small size documents

The best results achieved by this model are shown in Table 4.

Table 5.10: The best results obtained based on PMI Rank Based on Positive/Negative & Good/Bad Documents

Small Dataset	Large Dataset	Classifier	
0.2920	0.3460	KNNC	Mean Error Rate
0.1983	0.2304	KNNC	Mean standard Deviation

Chapter 6: Conclusion and Discussion

6.1 Conclusion

In this thesis, we have investigated common feature sets which have been used in previous similar studies on sentiment analysis as well as the new feature sets that we introduced, that contain mostly the high PMI rank sentimental words based on positive and negative documents, on societal theme datasets. Also, we have intended to find the best classifier in terms of accuracy and time, on large and small datasets represented by the mentioned feature sets. Classifiers which have been evaluated in this study categorized to linear and non-linear ones. Determining the best number of features to be extracted is another challenging issue in this research.

So far, in the previous chapters we have explained different feature sets as well as classifiers and experiments which have been conducted based on them. Now, it is time to take a much closer look at the results and summarize them in brief but high-content conclusions.

Generally, based on the conducted experiments we are of the belief that classifiers especially LDC and SVC show similar behavior on small and large datasets based on the common features model. However, the results obtained by the small datasets, are more accurate. In common features model, each of the small datasets relates to the specific topic and thus, most of the documents use same structures or words which helps the classifier to be trained faster and better while if we choose randomly a few number of documents from all 6 datasets and try to train the classifiers with these documents, surely we will obtain the bad results. In the selected features model they have been taken the advantage of the large PMI based dictionary made by training examples of the large dataset. Therefore, we can conclude that if we have much training data for large heterogeneous datasets as well, the classifiers can be trained better and give much accurate results.

There is every indication that the general performance of SVC and LDC are similar (even in some cases LDC is better than SVC) but the time that SVC needs to complete its prediction is 10 times more than the time consumed by LDC and this amount increases quadratically with dataset size. Therefore, in sentiment analysis, in which usually there is a huge amount of data to be processed or in the case of public polls, the SVC is of a low quality and in contrast with the predominant attitude in this field, it is fair to say that LDC performs better and is a good substitute for SVC in opinion classification.

According to the fact that common features model such as BoW or BoW with PoS tags use the whole data in training examples to train the classifiers instead of using typical sentimental information, LDC and SVC or generally linear classifiers outperform; since they generally discriminate the positive documents from negative ones. In their feature vectors, there are many redundant words such as “get_verb” which are not sentimentally informative and therefore, KNNC and QDC or generally non-linear classifiers which are sensitive to the features cannot perform well due to overtraining. While we can see in the selected feature model, which constructs feature vectors based on sentimental words especially in PMI based on positive/negative documents & adjective/negation PoS, that KNNC outperforms all the linear classifiers. This is because these sentimental features have direct polarity (positive or negative) and each documents represented based on them, can better describe themselves in terms of positive or negative attitude. According to KNNC, in the sphere it builds for itself to examine training documents based on their features and labels, when each object

represented by key features, the classifier can be trained better draw its decision boundary with more accuracy and classify objects with less error.

Based on the evidence obtained, the use of terms such as “good” or “excellent” for determining positive labeled documents and “bad” or “poor” for negative ones are not recommended. According to the accuracy results of classifiers, PMI based on positive/negative documents model performed better in comparison with PMI based on positive/negative & good/bad documents which shows that the assumption that these words can express the sentiment toward a topic is not always correct, rather they act as noise instead of informative features. The popular non-informative words which co-occurred in both model also are the reason for the bad performance.

There is every indication that the best discriminative features for sentiment analysis when there are lots of features exist, stand in the interval [100,250] of the data’s main directions and in the case of a small number of features (less than 100), the interval [35,65] are the most informative directions in the data.

There is a trade-off between cost and accuracy in order to determine the best feature sets. Generally speaking, the results obtained based on common features are more accurate but the constructed matrices based on them are very large in terms of memory and sometimes non-usable by applications. On the other hand, to make these matrices more informative, we should increase the size of the dictionary or add more fine grained PoS tags where both of them make the matrices huge. While, using selected features can reduce the size of matrices and by pruning these features based on extracting terms with adjective or negation part of speech tags, most of the non-informative words in terms of expressing sentiment are removed from the feature vectors. To wrap up, by investigating more on the selected features model in terms of training examples, surely we can get the better results in much less time since despite of smaller size of constructed matrices, they are more informative.

As far as societal theme documents are concerned, they have a difficult nature to be processed. However, despite the fact that we have comparatively little training data in comparison with other studies in this field, we could obtain in the worst case a mean accuracy of 68.51% in the large dataset which surely can be improved by increasing the number of training data samples and an average 73.41% accuracy on the small datasets which have been trained using the PMI dictionary gained by large dataset.

6.2 Future Work

Due to the limited time to conduct this research, some of the ideas and experiments could not be evaluated while they can give us a deeper insight into sentiment classification based on text.

One of the important features of sentiment analysis is the “position feature” which means the position of sentiment words in a sentence. This feature is of a significant importance since it plays a major role in opinion classification. For instance in the sentence below, the dominant opinion is mentioned in the last word of the sentence and while the whole sentence seems negative, the polarity is positive.

“The weather in Holland is awful, the sky is always cloudy, it is always raining, but I like it”

In order to increase the accuracy of sentiment classification, it is possible to take the advantage of pattern recognition techniques such as combining classifiers. Based on this technique, we can combine several classifiers and apply them on a dataset and benefit from each classifier which outperforms in a part of the data. Ensemble can be another option as well.

Creating a large dictionary based on sentiment words and words that have a high PMI rank in sentiment classification is another trick to obtain good results for textual opinion analysis since in this model we take the advantage of dictionary based as well as corpus based constructed matrices for datasets. More informative features lead classifiers to more accurate results.

Acknowledgement:

Apart from the efforts of myself, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to show my greatest appreciation to Dr. Pascal Wiggers. I can't say thank you enough for your tremendous support and help. I feel motivated and encouraged every time I attend your meeting. Without your encouragement and guidance this project would not have materialized. I will never forget you as my best supervisor.

I am grateful to Dr. Sepideh Babaei for her able guidance and useful suggestions, which helped me in completing the project work, in time. Sepideh, you helped me a lot not only in providing me valuable information as the guidance of my project but also in supporting me when I was under a lot of pressure. Moreover, drinking tea every day, every hour with you in the coffee room was my best break during this project.

My thanks and appreciations also go to Ing. Ruud de Jong and Ing. Bart Vastenhout for giving me such attention and technical support. Dear Ruud, you lent me a large amount of memory, I will give it back to you in few days, thank you!

Special thanks to my dear friend Roy Straver for his understanding, supporting, and wishes for the successful completion of this project.

My thanks to Seyran Khademi and all the other nice people who I did not mention here, for their contribution and support.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessings, understanding & endless love, through the duration of my studies.

Appendix

As mentioned before, the results of experiments on the other 4 small datasets is given in this part. In graphs below, the Mean Error Rate as well as Mean Standard Deviation of 4 small datasets based on three classifiers LDC, SVC and KNNC, are given. Likewise the previous results, in the common features model (Bag of Words with Part of Speech Tags) which has many features, the linear classifiers (LDC and SVC) perform better while in Selected features model (PMI Rank Based on Positive/Negative Documents & Adjective/Negation Part of Speech Tags) the non-linear classifier (KNNC) outperforms since there is a few number of redundant features in this model and therefore the classifier can train well.

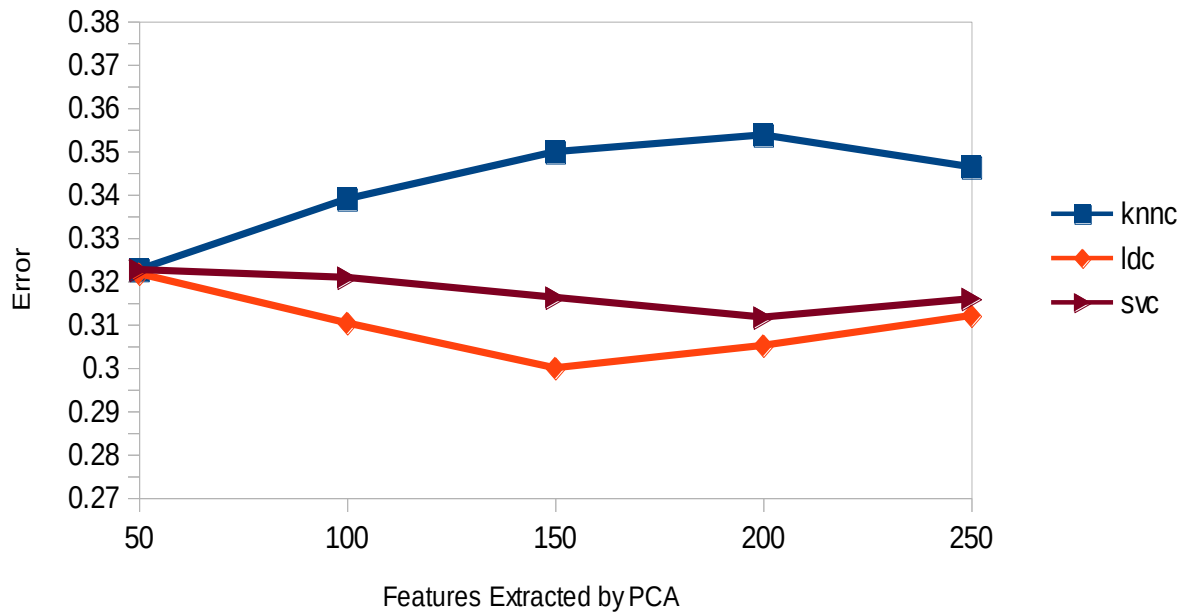
Table 1: Detailed information of the matrices constructed for each dataset based on “BoW with POS Tags” Model.

Dataset	Dictionary (WordNet)	Documents	Matrix	Label
God	58058	953	953 * 348354	953
Guns	58058	1069	1069 * 348354	1069
Gay rights	58058	2064	2064 * 348354	2064
Health Care	58058	667	667 * 348354	667

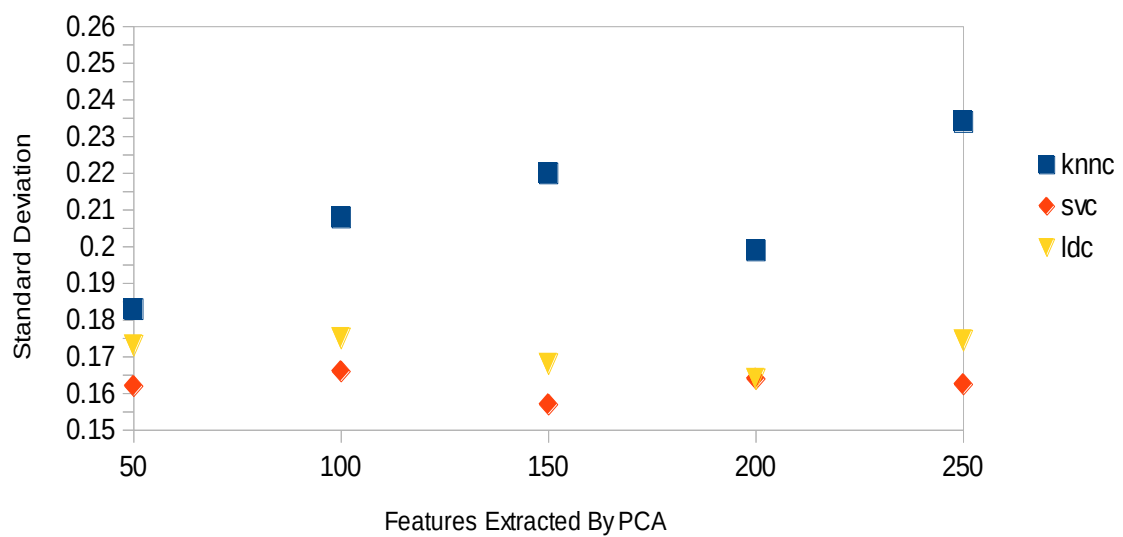
Table 2: Detailed information of the matrices constructed for each dataset based on “PMI Rank Based on Positive/Negative Documents & Adjective/Negation Part of Speech Tags” Model.

Dataset	Dictionary	Documents	Matrix	Label
God	151	953	953 * 151	953
Guns	151	1069	1069 * 151	1069
Gay rights	151	2064	2064 * 151	2064
Health Care	151	667	667 * 151	667

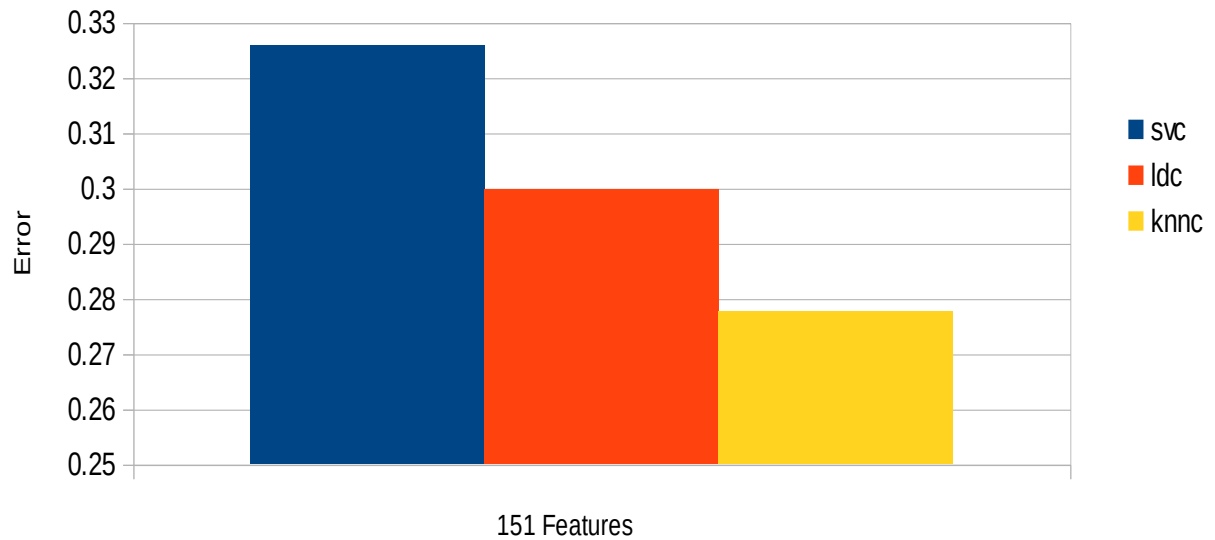
Bag of Words with PoS Tags on Gay-Rights



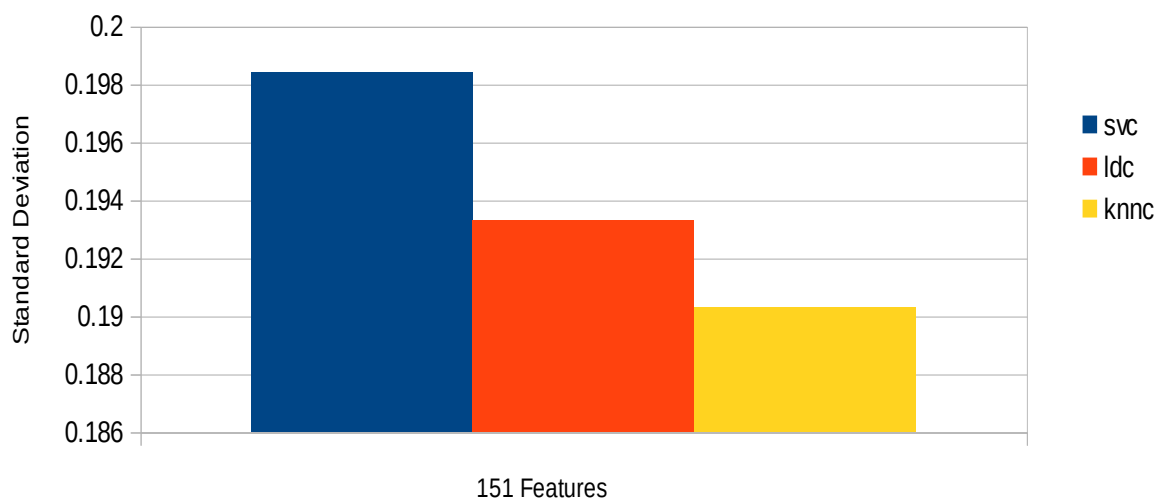
Average of Standard Deviation Rates Based on Bag of Words with POS Tags On Gay-Rights



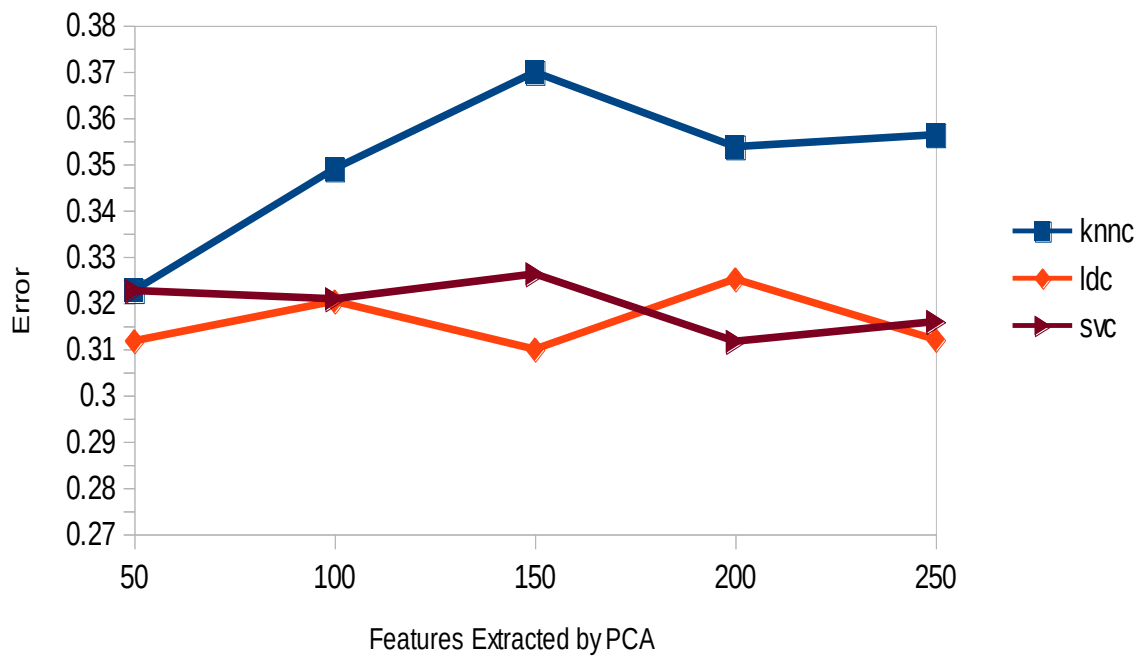
Mean Error Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Gay-Rights



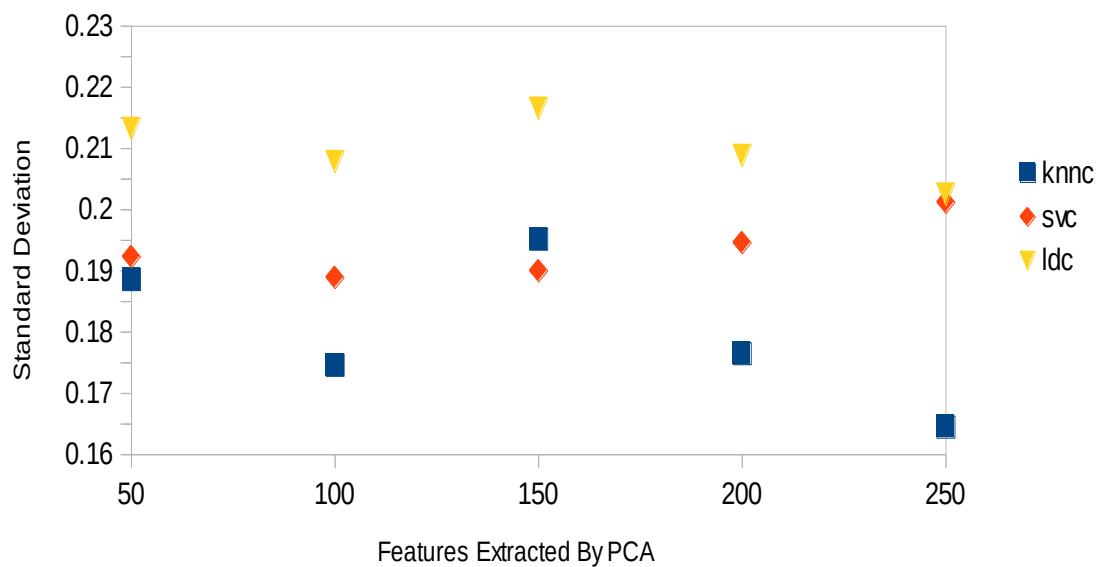
Mean Standard Deviation Rates Obtained by Cross Validation for
PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Gay-Rights



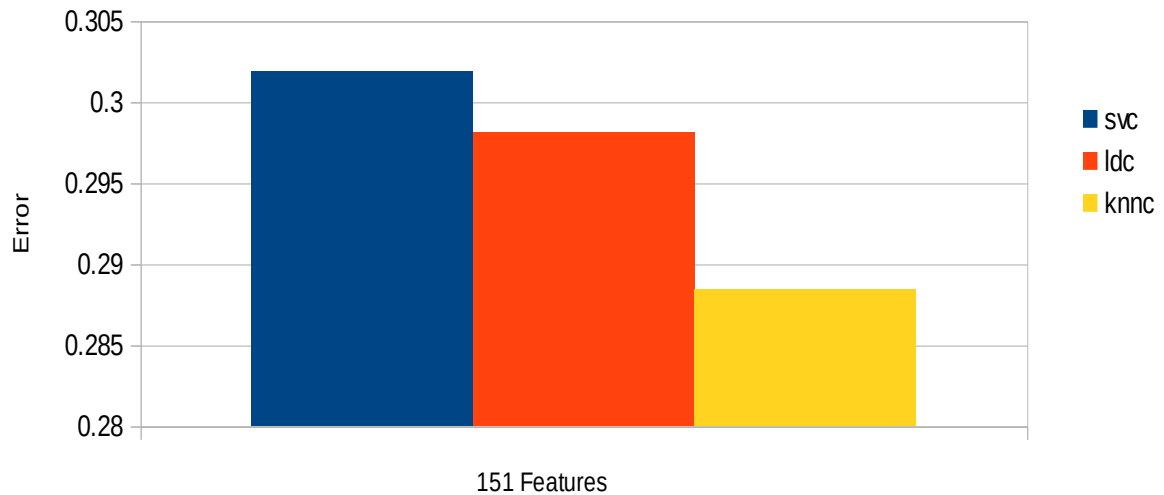
Bag of Words with PoS Tags on God



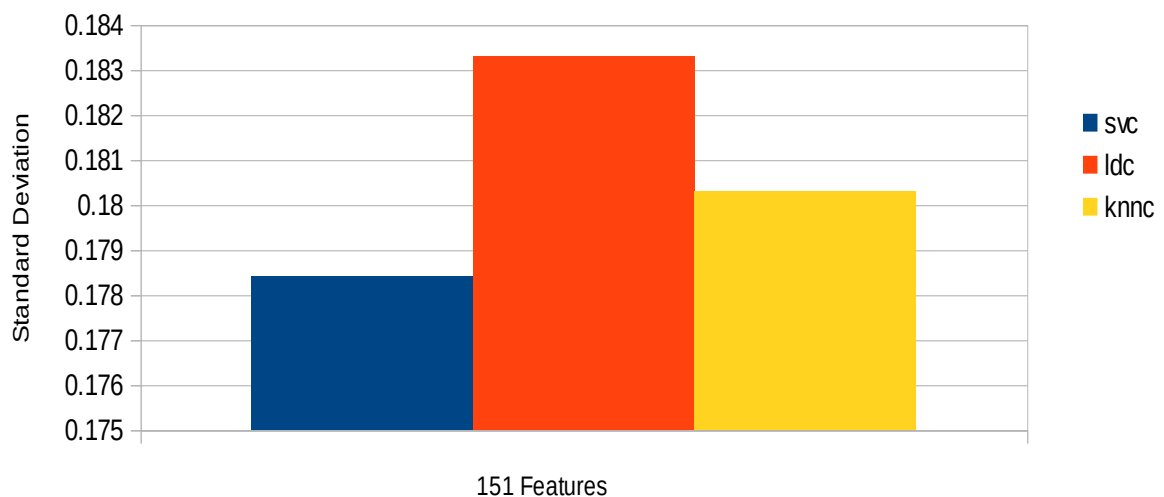
Average of Standard Deviation Rates Based on Bag of Words with POS Tags On God



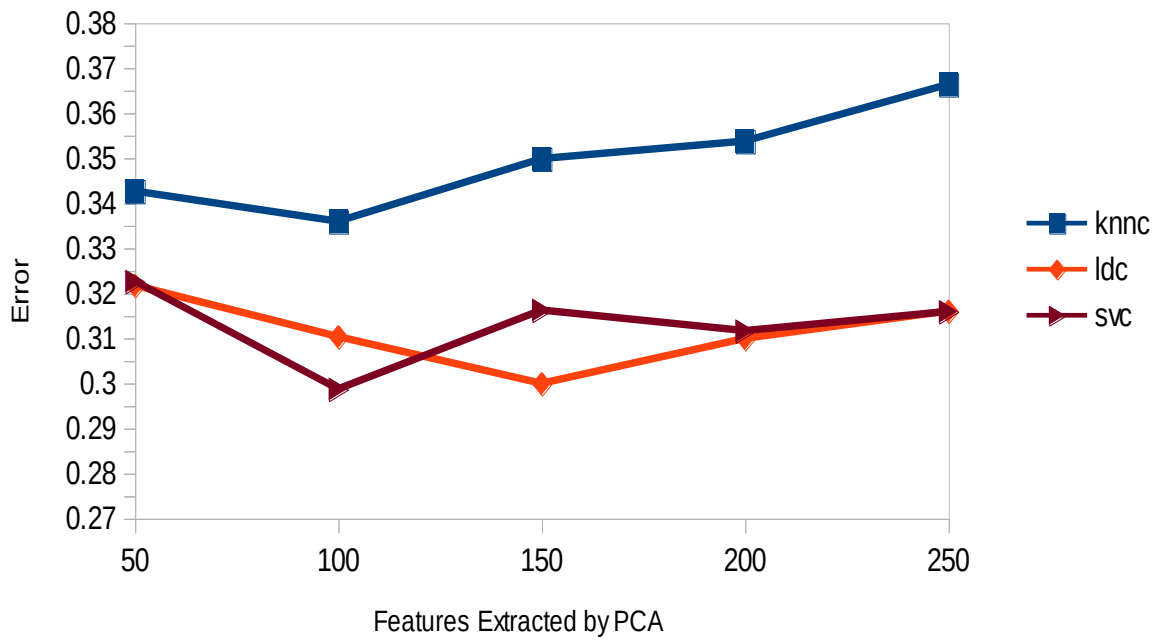
Mean Error Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On God



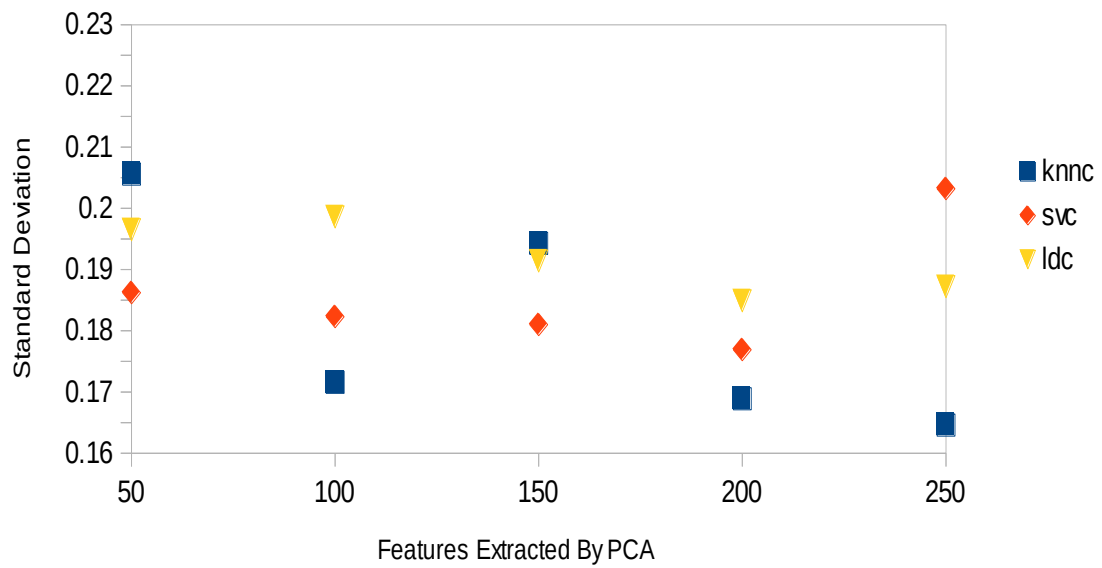
Mean Standard Deviation Rates Obtained by Cross Validation for
PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On God



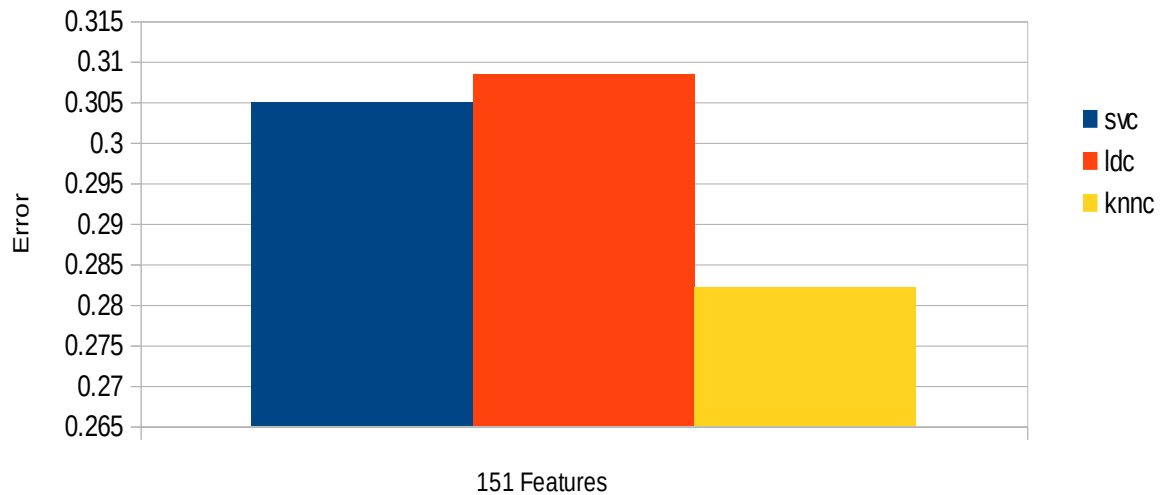
Bag of Words with PoS Tags on Guns



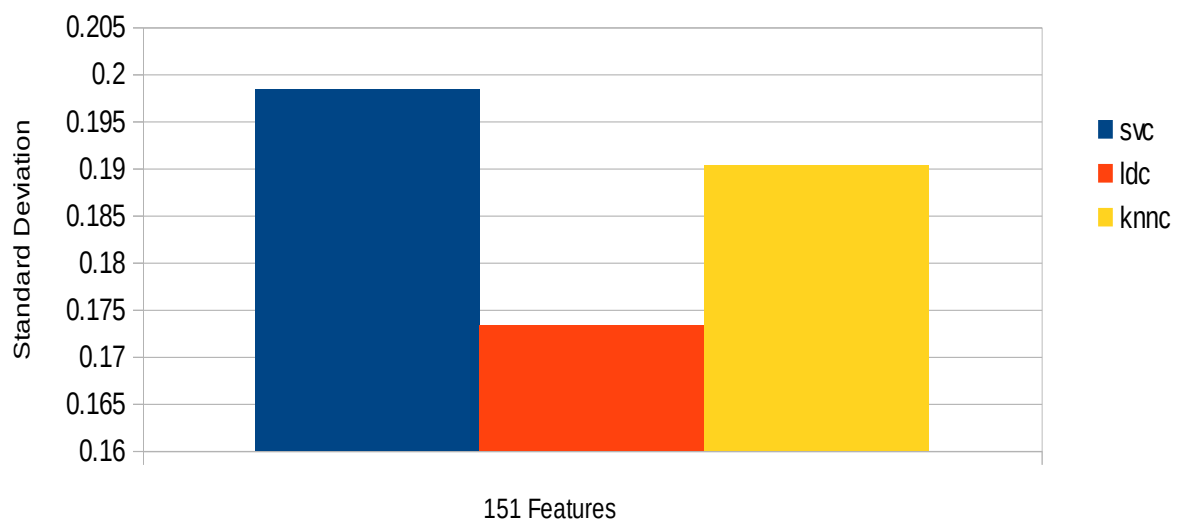
Average of Standard Deviation Rates Based on Bag of Words Model On Guns



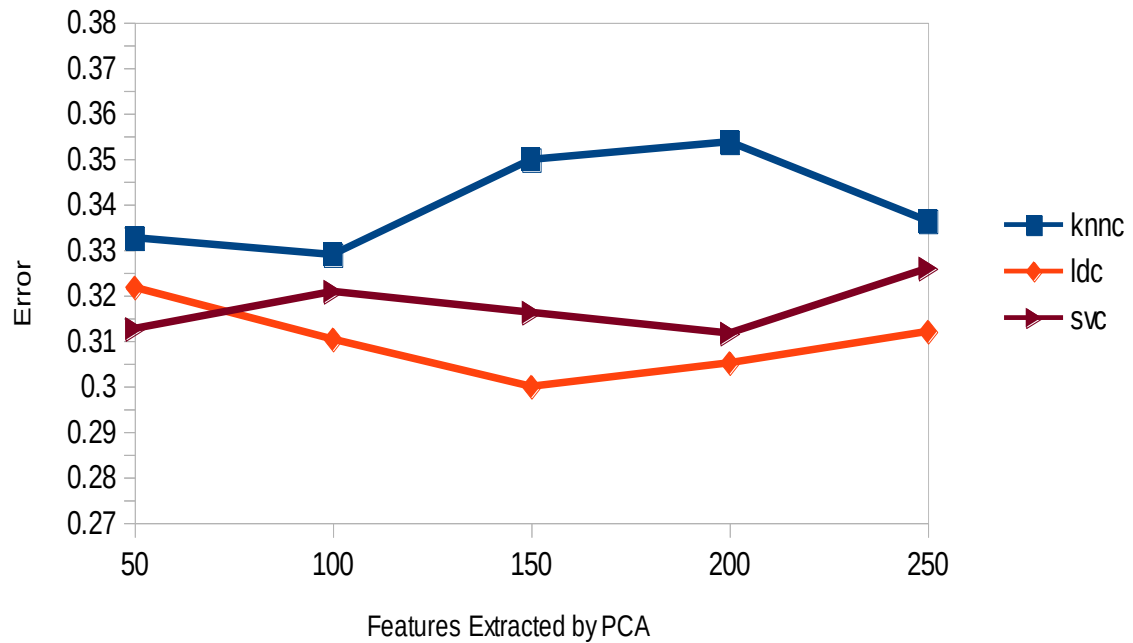
Mean Error Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Guns



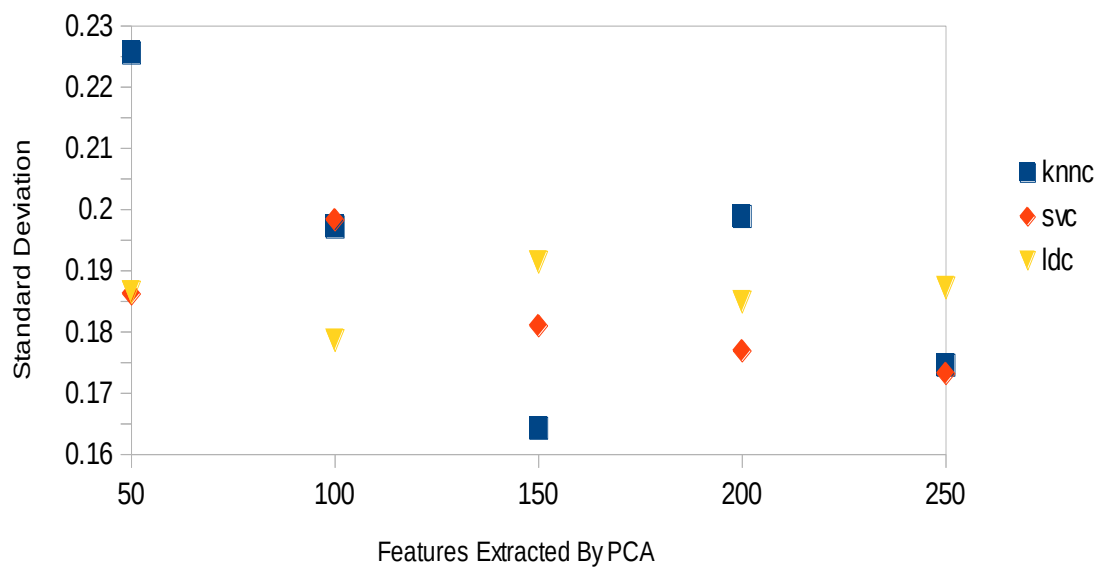
Mean Standard Deviation Rates Obtained by Cross Validation for
PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Guns



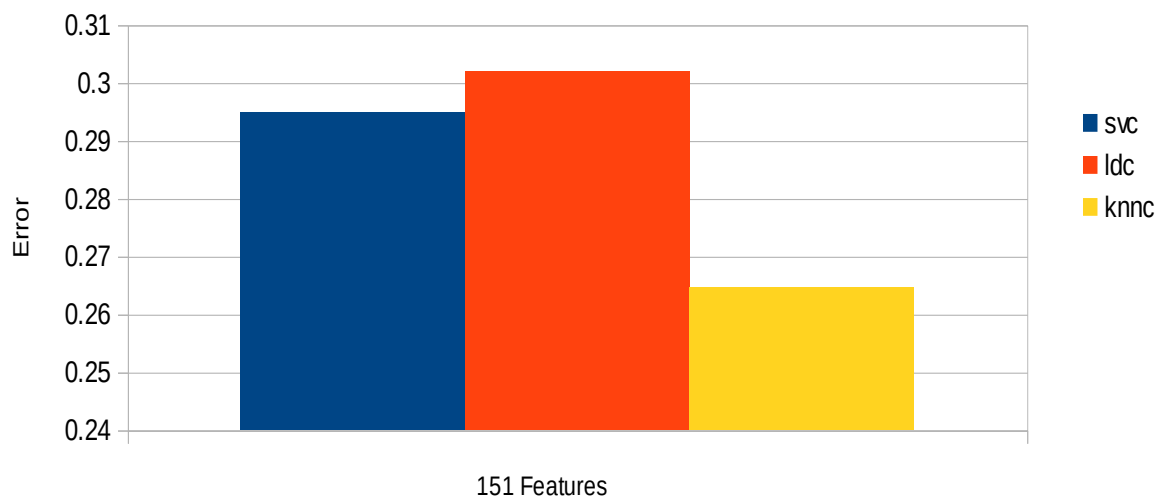
Bag of Words with PoS Tags on Health Care



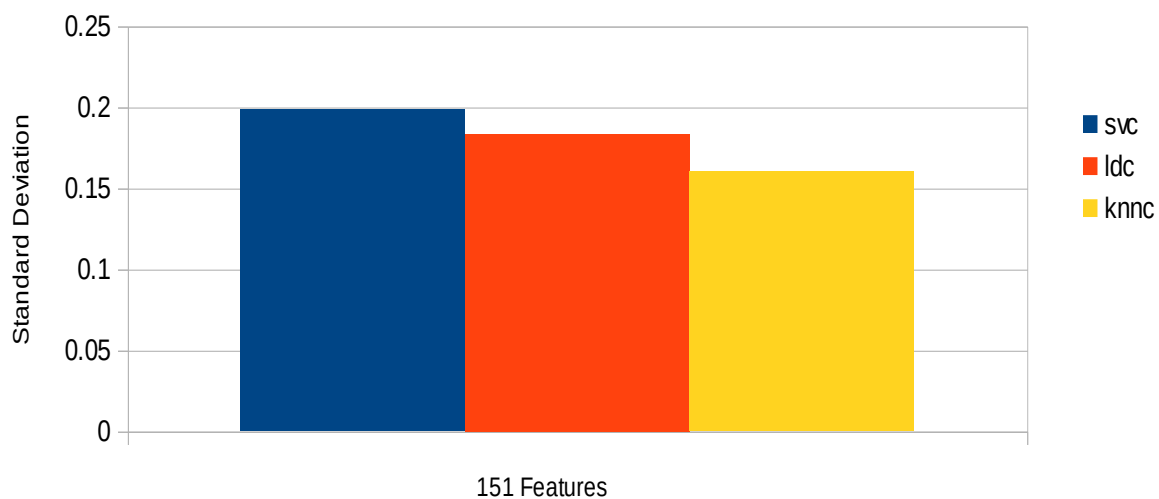
Average of Standard Deviation Rates Based on Bag of Words Model On Health Care



Mean Error Rates Obtained by Cross Validation
for PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Health Care



Mean Standard Deviation Rates Obtained by Cross Validation for
PMI Rank Based on Positive/Negative documents & Adjective/Negation POS Tags
On Health Care



References:

- [1]Na,J.C., Sui, H., Khoo, C., Chan, s., & Zhou, Y.(2004). *Effectiveness of simple linguistic processing in automatic sentiment classification. In the proceedings of the eighth International ISKO conference. Wurzburg, Germany*
- [2]Yang, H. , Si, L. , Callan, J. (2006) *Knowledge transfer and opinion detection in the TREC 2006 Blog track.In Proceedings of TREC.*
- [3]Wiebe, J. , Wilson, T. , Bell. M. (2001) . *Identifying collocations for recognizing opinions. In the proceedings of ACL/EACL Workshop on collocation, Toulouse, France.*
- [4]Pang, B. , Lee, L. Vaithyanathan, S. (2002).*Thumbs up? Sentiment classification using Machine Learning Techniques. In the Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*
- [5]Pang, B. , Lee, L. (2005) *Seeing starts, Exploiting class relationships for sentiment categorization with respect to rating scales.In the Proceedings of the ACL.*
- [6]Dave, K. , Lawrence, S. , M. Pennock, D. (2003) *Mining the peanut gallery: Opinion Extraction and semantic classification of product reviews.In Proceedings of the 12th international conference on World Wide Web*
- [7]D. Turney, P. (2002)*Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In the proceedings of the 40th Annual Meeting of the Association for computational Linguistics (ACL) , Philadelphia.*
- [8]Liu, B. , Hsu, w., Ma, Y. (1998)*Integrating Classification and Association rule Mining.In Proceedings of the 4th international conference on Knowledge Discovery and Data mining*
- [9]Agrawal, R.,Srikant, R. (1994)*Fast Algorithm for Mining Association rules . In Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile,Chile*
- [10] Popescu, A.M., Etzioni, O. (2005)*Extracting Product Features and Opinions from Reviews,In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*
- [11] Hu, M., Liu, B. (2004)*Mining Opinion Features InCustomer Reviews,In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, NY , USA*
- [12]Qiu, G. ,Liu, B. , Bu, J., Chen, C. (2009)*Expanding Domain Sentiment Lexicon Through Double Propagation,In Proceedings of the 21st international joint conference on Artificial intelligence, San Francisco, CA, USA*
- [13] Hu, M., Liu, B. (2004)*Mining and Summarizing Customer Reviews,In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, NY , USA*
- [14]Etzioni, M. , Cafarella, D. , Downey, S. , Kok, A. , Popescu, T. , Shaked, S. , Soderland, D. weld, A. Yates (2005)*Un-supervised name-entity extraction from the web.Artificial Intelligence Journal*

- [15] **Tesniere, L. (1959)***Elements de syntaxe structural*, Klincksirck, Paris
- [16] **Pang, B. , Lee, L. (2008)***Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval* 2(1-2), pp. 1–135.
- [17] **Liu, B. (2010)***Opinion mining and sentiment analysis. Handbook of Natural Language Processing*, Second Edition.
- [18] **Lin, D. (1998)** *Dependency-based evaluation of MINIPAR*. In **Workshop on Evaluation of Parsing Systems at ICLRE**
- [19] **Wilson, T. , Wiebe, J. , Hwa, R. (2004)** *Just How Mad are You? Finding strong and Weak Opinion clauses* . In **Proceedings of AAAI**
- [20] **Martin, J. , white, P. (2005)** *The Language of evaluation : Appraisal in English* .Palgrave Macmillan, London, New York
- [21] **Greene, S. (2007)***Spin: Lexical Semantics, Transitivity, and the Identification of Implicit Sentiment*.University of Maryland
- [22] **Morinaga, S. ,Yamanishi, K. ,Tateishi, K. and Fukushima, T. (2002)***Mining product reputations on the Web*,In **Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining(KDD)**, pp. 341–349, (Industry track).
- [23] **Yi, J., Nasukawa, T., Bunescu R., Niblack, W. (2003)***Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques*,In **Proceedings of IEEE International Conference on Data Mining (ICDM)**
- [24] **Fellbaum, C. (1998)***Wordnet: An Electronic Lexical Database*.MIT Press.
- [25] **Kim, S.-M.,Hovy, E. (2004)** *Determining the sentiment of opinions*, In **Proceedings of the International Conference on Computational Linguistics (COLING)**.
- [26] **Andreevskaia, A., Bergler, S. (2006)***Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses*, In **Proceedings of the European Chapter of the Association for Computational Linguistics(EACL)**.
- [27] **Esuli, A. , Sebastiani, F. (2006)***Determining term subjectivity and term orientation for opinion mining*, In **Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)**.
- [28] **Esuli, A. , Sebastiani, F. (2006)***SentiWordNet: A publicly available lexical resource for opinion mining*,In **Proceedings of Language Resources and Evaluation (LREC)**.
- [29] **Hatzivassiloglou, V., McKeown, K.(1997)***Predicting the semantic orientation of adjectives*, In **Proceedings of the Joint ACL/EACL Conference**.
- [30] **Riloff, E., Wiebe, J. (2003)** *Learning extraction patterns for subjective expressions*,In **Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)**
- [31] **Riloff, E., Wiebe, J., Wilson, T. (2003)** *Learning subjective nouns using extraction pattern bootstrapping*,In **Proceedings of the Conference on Natural Language Learning (CoNLL)**

- [32] Aue, A. , Gamon, M. (2005)*Customizing sentiment classifiers to new domains: A case study*, In **Proceedings of Recent Advances in Natural Language Processing (RANLP)**.
- [33] Blitzer, J. , Dredze, M. ,Pereira, F. (2007)*Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification*,In **Proceedings of the Association for Computational Linguistics (ACL)**.
- [34] Raina, R., Y. Ng , A., Koller, D. (2005) *Transfer Learning by constructing informative priors*, **Workshop on Transfer Learning in the Nineteenth Annual Conference on Neural Information Processing Systems (NIPS)**
- [35] Obozinski, G., Taskar, B., Jordan, M., (2006) *Multi-task Feature Selection*, In the workshop of **structural Knowledge Transfer for Machine Learning in the 23rd International Conference on Machine Learning (ICML)**
- [36] Manning, C.D., Raghavan, P., Schtze, H., (2008)*Introduction to Information Retrieval*.Cambridge University Press.
- [37] R.O. Duda, P.E. Hart, and D.G. Stork,(2001)*Pattern classification*, 2nd edition, John Wiley and Sons, New York.
- [38] A. Webb,(2002)*Statistical Pattern Recognition*, John Wiley & Sons, New York.
- [39] C. Liu and H. Wechsler,(2000) Robust Coding Schemes for Indexing and Retrieval from Large Face Databases, **IEEE Transactions on Image Processing**, vol. 9, no. 1, 132-136.
- [40](1998)*Data Mining and Knowledge Discovery*, 2nd edition, 121–167 . **Kluwer Academic Publishers, Boston. Manufactured in The Netherlands**
- [41]R. Kohavi,(1995)*A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. **IJCAI: 1137-1145**.
- [42] H. Larochelle and Y. Bengio, (2008)*Classification using Discriminative Restricted Boltzmann Machines*, **Proceedings of the 25th International Conference on Machine Learning (ICML)**, 536-543.
- [43] V.Vapkin and A.Sterin,(1977)*On structural risk minimization or overall risk in a problem of pattern recognition*.
- [44] H.William, A. Saul , T. William,(2007)"Section 16.5. Support Vector Machines". **Numerical Recipes: The Art of Scientific Computing (3rd ed.)**. New York: Cambridge University Press
- [45] G.Bouma(2007), *Normalized (Pointwise) Mutual Information in Collocation Extraction*,**Proceedings of GSCL, Tubingen**
- [46] D. A. McAllester, R.Schapire (2000) *On the Convergence Rate of Good–Turing Estimators*, **Proceedings of the Thirteenth Annual Conference on Computational Learning Theory**
- [47]A. Glen, R. W. Douglass, A. Zardecki (2005). *Mesh enhancement*. Imperial College Press. p. 404