T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

WIKIPEDIA BASED SEMANTIC SMOOTHING FOR TWITTER SENTIMENT CLASSIFICATION

M.S THESIS

Dilara TORUNOĞLU
200991002

Thesis Advisor:
Assist.Prof. Dr. Murat Can GANİZ

JUNE 2013
ISTANBUL

T.C. DOĞUŞ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

WIKIPEDIA BASED SEMANTIC SMOOTHING FOR TWITTER SENTIMENT CLASSIFICATION

M.S THESIS

Dilara TORUNOĞLU
200991002

Thesis Advisor:
Assist.Prof. Dr. Murat Can GANİZ

JUNE 2013
ISTANBUL

WIKIPEDIA BASED SEMANTIC SMOOTHING FOR TWITTER SENTIMENT CLASSIFICATION

APPROVED BY:

Assist. Prof. Dr. Murat Can GANİZ .....

(Thesis Advisor, Doğuş University)

Prof. Dr. Selim AKYOKUŞ .....

(Doğuş University)

Prof. Dr. Coşkun Sönmez .....

(Yıldız Techical University)

DATE OF APPROVAL: 24.06.2013

PREFACE

In my thesis we develop a semantic smoothing method based on enrichment of Twitter data set with Wikipedia Semantic knowledge which is named Naïve Bayes with Wikipedia Semantic Smoothing (NBWSS). NBWSS makes use of semantic knowledge such as multiword Wikipedia article titles, their categories and redirects. In order to show the efficiency of the proposed system we use Twitter Sentiment 140 Data set which is a large scale sentiment classification dataset consists of tweets in English.

Istanbul, June 2013 Dilara TORUNOĞLU

ABSTRACT

Sentiment classification is one of the important and popular application areas of text classification in which texts are labeled as positive and negative. Moreover, Naïve Bayes (NB) is one of the mostly used algorithms in this area. NB having several advantages on lower complexity and simpler training procedure, it suffers from zero probability problems (Rish, 2001). Smoothing methods are employed for this problem; mostly Laplace Smoothing is used; however in this paper we propose Wikipedia based semantic smoothing approach. Our semantic smoothing formulation is based on the work in (Zhou, 2008). We extend this study by employing Wikipedia to extract topic signatures. Moreover, we also incorporated semantic knowledge in Wikipedia such as categories and redirects. To be more precise, we use Wikipedia article titles that exist in documents, categories and redirects of these articles as topic signatures to enrich the dataset. We apply our approach to sentiment classification of tweets. Results of the extensive experiments show that our approach improves the performance of NB and even can exceed the accuracy of SVM on Twitter Sentiment 140 dataset.

Key Words: Naïve Bayes, Semantic Smoothing, Text Classification, Sentiment Classification, Sentiment Analysis, Wikipedia, Twitter.

ÖZET

Anlamsal sınıflandırma, metin sınıflandırma alanında kullanılan en önemli ve en popüler sınıflandırma yaklaşımlarından biridir ki bu yaklaşımda metinler pozitif ve negatif olarak sınıflandırılmaktadır. Dahası, Naive Bayes (NB) bu alanda en çok kullanılan algoritmadır. NB algoritmasının düşük karmaşıklık, basit öğrenme prosedürü gibi avantajlarının yanında, sıfır olasılık problemiyle uğraşmaktadır (Rish, 2001). Yumuşatma methodları bu probleme uygulanmaktadır, çoğunluklada Laplace yumuşatması kullanılır; ancak bu çalışmada biz Vikipedi tabanlı anlamsal yumuşatma algoritmasını önermekteyiz. Bizim anlamsal yumuşatma algoritmamızın formülleri (Zhou, 2008)'deki çalışmasına dayanmaktadır. Biz bu çalışmadaki anlamsal zenginleştirmeyi Vikipedi kullanarak genişlettik. Ayrıca, Vikipedi kategorilerin ve yönlendirmelerini anlamsal bilgi geliştirme yönünde ekledik. Daha açık konuşmak gerekirse, bu çalışma anlamsal yumuşatma yaklaşımını görülen Vikipedi başlıklarını, bu başlıkların kategorileri ve yönlendirmelerini kullanılarak Twitter veri kümesini zenginleştirmek amaçlı kullanılmıştır. Yaklaşımımızı anlamsal sınıflandırma amacıyla tweet'ler üzerinde uyguladık. Yapılan birçok testin sonucunda görülmüştür ki Twitter Sentiment 140 veri kümesi üzerinde, yaklaşımımız Naive Bayes algoritmasının başarısını arttırmakta ve Karar Destek Makinelerini geçmektedir.

Anahtar Kelimeler: Naïve Bayes, Anlamsal Yumuşatma, Metin Sınıflandırma, Anlamsal Sınıflandırma, Anlamsal Analiz, Vikipedi, Twitter

ACKNOWLEDGMENT

I would like to thank my advisor, Dr. Murat Can Ganiz for his guidance throughout this research. I am also very appreciative to all members of my family; Fatemeh, Hüseyin, Dünya and Doğa Torunoğlu for their love, supports and prayers throughout this process. I would like to especially thank to my fiancé Salih Selamet for his encouragement and understanding.

This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) grant number 111E239. Points of view in this document are those of the authors and do not necessarily represent the official position or policies of the TÜBİTAK.

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

B_{it}	Occurrence of term t in document i
$\|D\|$	Number of labeled training documents
$P(w_t \mid c_j)$	Probability of term w_t in class c_j
$\|V\|$	Vocabulary
N_{it}	Count of the number of times term w_t occurs in document d_i
$P_{c,t}$	Probability of term given class
$N(c_t, D)$	Number of documents in class c that contain term t
$N(c, D)$	Number of documents in class c
$P_{ml}(w_t \mid c_j)$	Probability with maximum likelihood estimate
$P(w_t \mid D)$	Maximum likelihood estimation of term t in collection D
$N(w_t, D)$	Total number of term counts
$N(D)$	Total number of documents in collection D
$P_s(w \mid c_i)$	Unigram class model with semantic smoothing and t_k
$P(t_k \mid c_i)$	Distribution of topic signatures in training documents of a given class
t_k	Topic signature
D_k	Set of documents containing topic signature
$P(w \mid t_k)$	Probability of word given topic signature

ABBREVIATIONS

EM	Expectation Maximization
JM	Jelinek-Mercer Smoothing
k-NN	K-Nearest Neighbors
LibSVM	Support Vector Machines with Linear Kernel
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
MAP	Maximum a Posteriori
MEC	Maximum Entropy Classification
MNB	Multinomial Naïve Bayes
NB	Naïve Bayes
NBWSS	Naïve Bayes with Wikipedia Semantic Smoothing
SGD	Stochastic Gradient Descent
SVM	Support Vector Machines
TF	Term Frequencies
TF-IDF	Term frequency–inverse document frequency
TS	Training Set Size

TABLE OF CONTENTS

1. INTRODUCTION

Text classification is one of the important techniques to automatically organize large amounts of textual data accumulated in organizations, social media and the Internet. Text classification gaining importance with rapid increase in the usage of internet and especially social media sites such as Twitter and Facebook. As a result, a tremendous amount of textual information is generated by individuals as well as the commercial entities, and organizations. One of the important and popular application areas of the text classification is the sentiment classification in which the comment texts are usually categorized as positive or negative.

Commonly used machine learning algorithms in text classification are Naïve Bayes (NB) (Mccallum and Nigam, 1998), k-nearest neighbor (Yang et al., 1999), Support Vector Machines (SVM) (Joachims, 1998). Although SVM is one of the best performing algorithms in this domain, NB can perform better on several cases (Rish, 2001) and additionally it has several advantages such as lower complexity and simpler training procedure. However, NB greatly suffers from sparsity (Rish, 2001) when applied to the particularly high dimensional data as in text classification. This is especially the case when the training data consist of very short documents such as tweets and when the training set size is limited because of the cost of manual labeling processes. In order to avoid zero probability problem smoothing methods are used.

Most commonly used and default smoothing technique is called Laplace Smoothing (LS) which adds one count to all terms in the vocabulary. Though this combination is widely used it proves to be not effective in many applications (Jelinek, 1990). Several other smoothing methods are proposed in order to cope with this problem in the language modeling domain such as Good Turing Smoothing (Gale, 1995), Jelinek-Mercer Smoothing (Chen and Goodman, 1998), Absolute Discounting Smoothing (Vilar et al., 2004) and Linear Discounting Smoothing (Manning et al.,2009) and these can be applied in NB text classification.

There are also more advanced smoothing approaches called semantic smoothing which attempts to distribute probability mass to the using semantic relations (Zhou et al., 2008), (Poyraz et al., 2012). We base our study on the approach introduced in (Zhou et al., 2008) which extracts important concepts called topic signatures from the training documents and calculates term probabilities by statistically mapping terms to topic signatures using Expectation Maximization (EM) algorithm.

To give an example for what is semantic smoothing is that the document containing term "hospital" should return for the query "sanitarium" because both terms are semantically related (i.e. synonymous).

We extend the semantic smoothing method proposed by (Zhou et al., 2008) which is a topic signature based semantic smoothing method to deal with sparsity problem. The idea of our semantic smoothing is to extract explicit topic signatures (e.g. Wikipedia Articles, categories belongs to articles and redirects belongs to articles) instead of implicit ones (collocations) from documents and then statistically map them into single word features. For example, taking the advantage of semantic smoothing we can assign the topic signature "Public Health" to a doc that doesn't include any single word in this topic signature. But clearly includes words about health such as "World Health Organization" and "Insurance medicine".

As in (Zhou et al., 2008) our approach is based on Naïve Bayes (NB) algorithm with the enrichment of the Wikipedia knowledge. We call it Naïve Bayes with Wikipedia Semantic Smoothing (NBWSS).

We significantly extend this approach by using Wikipedia article titles that exist in training documents, and furthermore categories and redirects of these articles as topic signatures. We propose a Wikipedia based semantic smoothing approach since it exploits significant amount of semantic information encoded in the relations between article titles, categories,

and redirects. We conduct experiments on twitter collections taken from Twitter Sentiment 140 (Go et al., 2009) dataset. On extensive experiments show that our approach increases performance on accuracy than compared to NB and in some case, exceeds SVM on Twitter Sentiment 140 dataset.

Our experiments show that when the size of training documents is small, the classifier with Wikipedia Semantic Smoothing (NBWSS) is similar to Bayesian classifier with Laplacian smoothing (MNB). When the size of training increases the proposed algorithm is not only outperforms MNB, but gives better accuracy than SVM classifiers.

We incorporate four types of topic signatures; multi-word Wikipedia Article titles, Wikipedia categories and Wikipedia redirects. Combination of these contextual information enrich the semantic mapping process and helps to deal with sparsity problem in very short documents such as tweets.

2. RELATED WORK

In text classification to avoid sparsity problem, language model smoothing is commonly used. There are several smoothing methods including Jelinek-Mercer (Chen and Goodman, 1998), absolute discounting (Vilar et al., 2004), to smooth unigram language models. Background collection model is mostly used in these methods. On the other hand there is a more effective way to smooth language models called; semantic smoothing. In (Zhou et al., 2008) and (Zhou et al., 2006) several smoothing approaches are been proposed for language modeling.

In (Zhou et al., 2008), they proposed a background collection smoothing model to increase the performance of Naïve Bayes. They used topic signatures which are produced by using XTRACT (Smadja, 1993). They considered three types of topic signatures. These are unigrams (single-word features), multiword phrases, and ontological concepts where available, respectively. After extracting topic signatures and multiword phrases from documents they used them in semantic smoothing background collection model to smooth and map the topic signatures. They concluded that usage of semantic smoothing and topic signatures yield better accuracy results. They implemented semantic smoothing method using the dragon toolkit (Zhou et al., 2007) and conduct experiments on three collections, OHSUMED, LATimes, and 20 newsgroups. They pointed out that when the size of training documents is small; the Bayesian classifier with semantic smoothing outperforms Bayesian classifiers with background smoothing (which is referaed as Jelinek Mercer Smoothing) (Chen and Goodman, 1998) and Laplacian smoothing. We are motivated by this study and adopted same topic signature mapping approach however instead of using multiword phrases that are extracted from documents we employ Wikipedia articles, categories and redirects as topic signatures.

The idea of using multiword phrases or n-grams or topic signatures, is not a new concept. In (Zhou et al., 2008) and (Zhou et al., 2006) the multiword phrases and topic signatures are used for smoothing purposes. On the other hand, the concept of topic signatures is used in (Zhou et al., 2008) which are unigrams (single-word features), multiword phrases that exists in the documents.

A similar technique is using word clustering to group semantically related word groups (Baker et al., 1998). In this study they described the application of Distributional Clustering of words to document classification. They showed that their approach groups similar words together and uses word clusters as document features. This technique uses advantage of semantic relationships between words and works to gain higher classification accuracy. They tested their results on three real-world data sets. And they showed that, better accuracy results obtained compared to Latent Semantic Indexing (LSI).

An expectation–maximization (EM) (Dempster et al., 1977) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models. In EM algorithm there are two iterations; first is for an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and second maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These iterations continue till there is no more change which is convergent to some value or reaches to a given cycle constant. Related to this, expectation-maximization (EM) is widely used in text classification area. In (Zhou et al., 2008) and (Zhu, 2005) EM algorithm is used for make use of unlabeled data for learning multinomial Naïve Bayes (MNB) model (Mccallum and Nigam, 1998).

Today, amount of textual information stored electronically has dramatically increased by the exponential growth in the use of internet and social media. The studies are focused on organizing the information obtained from internet and social media and investigating with a system that automatically labels those with their corresponding topics are known as text categorization. These studies are aimed on topical categorization as sorting the documents with respect to their subjects such as magazine, politics and world etc. A popular application of text classification is the classification of documents according to their sentiment (i.e. positive or negative). Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. Hence sentiment classification is subtask of sentiment analysis.

Sentiment classification aims to find the opinion of the user where a crucial characteristic of the posted articles is their sentiment or overall opinion towards the subject matter for example whether a product review is positive or negative. Labeling these articles with their sentiment would provide succinct summaries to readers (Pang et al, 2002).

Sentiment classification is commonly a two-class classification problem, labeling positive and negative (Liu, 2012). The data usually consists of product reviews. To determine a class as negative or positive, it is simply looked at the review rating scores from 1-5 stars. It's been given in (Liu, 2012) that for example, a review with 4 or 5 stars is considered a positive review, and a review with 1 to 2 stars is considered a negative review. Without a notr class label, the classification problem becomes much easier. It's been pointed out that sentiment classification is a text classification problem. In sentiment classification, sentiment words are considered to be more important such as, "great", "excellent", "amazing", "horrible", "bad", "worst". So sentiment classification approaches usually make use of these sentiment words explicitly. Machine learning algorithms are also commonly used in sentiment classification such as, Naïve Bayes (NB) (Mccallum and Nigam, 1998) classification, and Support Vector Machines (SVM) (Joachims, 1998).

An example of these work using machine learning techniques in sentiment classification is (Pang et al, 2002). They considered the problem of classifying documents not by topic, but by overall sentiment, e.g., determining whether a review is positive or negative. They use movie reviews data, and find that standard machine learning techniques definitively outperforms human-produced baselines. To compare their results with human produced baselines they asked two graduate students in computer science to (independently) choose good indicator words for positive and negative sentiments in movie reviews. The humans' selections are shown in table 2.1 which is obtained from the paper (Pang et al, 2002).

Table 2.1 Baseline results for human word lists. Data: 700 positive and 700 negative movie reviews (Pang et al, 2002)

	Proposed word lists	Accuracy	Ties
Human 1	Positive: dazzling, brilliant, phenomenal, excellent, fantastic	58%	75%
	Negative: suck, terrible, awful, unwatchable, hideous		
Human 2	Positive: gripping, mesmerizing, riveting, spectacular, cool, awesome, thrilling, badass, excellent, moving, exciting	75%	39%
	Negative: bad, cliched, sucks, boring, stupid, slow		

After obtaining humans selections, they then converted their responses into simple decision procedures that essentially count the number of the proposed positive and negative words in a given document. They focused on applying these procedures uniformly distributed data, because they wanted to obtain 50% for the random-choice baseline result. They also pointed out that the tie rates (percentage of documents where the two sentiments were rated equally likely) are quite high. They chose a tie breaking policy to maximize the accuracy of the baselines.

Because that the tie rates produced by humans are relatively poor in performance results they focused on creating a new list of seven positive and seven negative words including punctuation shown in Table 2.2 obtained from paper (Pang et al, 2002).

Table 2.2 Results for baseline using introspection and simple statistics of the data (including test data) (Pang et al, 2002)

	Proposed word lists	Accuracy	Ties
Human 3 + stats	Positive: love, wonderful, best, great, superb, still, beautiful	69%	16%
	Negative: bad, worst, stupid, waste, boring, ?, !		

And they showed that these words are raised the accuracy to 69%. And they pointed out that, this list has a much lower tie rate of 16%.

On the other hand, they showed that the three machine learning method they used (Naive Bayes (NB), maximum entropy classification (MEC), and Support Vector Machines (SVM) do not perform as well on sentiment classification as on traditional topic-based categorization. They concluded by examining factors that make the sentiment classification problem more challenging. They showed in results that generated via machine learning techniques are almost good in comparison to the human generated baselines. In terms of relative performance they concluded their work with showing, NB tend to do the worst and SVMs tend to do the best, although showed differences aren't very large.

Micro blogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life every day. Therefore micro blogging web-sites are rich sources of data for opinion mining and sentiment analysis. Twitter is a popular micro blogging service where users create status messages called "tweets". With the increase of people using micro blogging sites and expressing their status or feelings of any kind of topic especially with products or companies, researches are tend to work on this rich source. Twitter has become a popular source for sentiment classification. For example (Go et al., 2009), (Jiang et al., 2011), (Pak and Paroubek, 2010) and (Barbosa and Feng, 2010) they use twitter data for sentiment classification. Because micro blogging has appeared relatively recently, there are a few research works that were devoted to this topic. Sentiment classification studies are focused on classifying tweets as positive or negative with respect to a given topic in order to understand the user's feelings of the product or a company. This has an important commercial application since it is critical for companies to understand and get upon the sentiment of their customers towards their products in order to gain competitive advantage.

Among these we use the twitter data set compiled by (Go et al., 2009). They used Twitter's Application Programming Interface (API) for programmatically accessing tweets by query term. They selected the parameter search for English so they have only downloaded English tweets for this purpose.

They expressed the characteristics of smiley icons and these icons can mean a positive or negative emotion which can be used for sentiment classification. For example, ":)" and ":-)" both express positive emotion. Thus they searched for the queries including positive smiley icon ":)", and labeled the obtained tweet as positive. So the query ":(" will return tweets and these tweets will be labeled as negative. The full list of emoticons can be found in Table 2.3.

Table 2.3 List of Emoticons (Go et al., 2009)

Emoticons mapped to :)	**Emoticons mapped to :(**
:)	:(
:-)	:-(
:)	: (
: D	
=)	

The tweets in their training set were from the time period April 6, 2009 to June 25, 2009, totally 1,600,000 tweets. After obtaining tweets with given smiley icon queries they used post-processing. They mainly had 5 processing filters, which are; first, they eliminated smiley icons from Table 2.3 from the obtained tweets; they pointed that with including these icons in tweets the Maximum Entropy model and SVM classifiers would put a large amount of weight on the emoticons, in which the accuracy results could be badly affected. Second, any tweets both containing positive and negative icons were removed to avoid any confusion. For example, "I love my ipad :) but sometimes it is really hard to use :(". In their paper, they omitted these tweets because they didn't want positive features marked as part of a negative tweet, or vice-versa. Third filter was removing Retweets which are processes of coping another user's tweet. To remove these tweets they searched the query term for "RT" which means Retweeted from someone else. Forth filter was removing tweets including icons like ":P". They implied that these icons would not imply any positive and negative sentiment. For last filter, they omitted repeated tweets.

But this is not enough for classifying the tweet. So in (Go et al., 2009) they have created two group sets the words for positive emotions and the words for negative emotions. The word "love" included tweets classified as positive and the word "hate" included tweets classified as negative. The tweets included both words or both smiley icons were eliminated for misclassification issues and given specific product names or person names 1,600,000 tweet were classified.

They collected their test data manually by using the web application. After obtaining 359 total tweets they manually labeled 177 tweets as negative and 182 tweets as positive. Moreover, they pointed that they their test data were not included some of smiley icons not all. To obtain testing collection they followed two steps. Step 1, was searching the Twitter API with special queries which were arbitrarily chosen from different domains. For example, these queries consist of consumer products (40d, etc), companies (at&t, etc), and people (Obama, etc). They obtained these query terms manually to obtain testing tweets. In Table 2.4 list of the queries are shown. And then they worked on tweets obtained by these query searching and labeled them as negative or positive if seen sentiment in tweets. To be more precise, they searched for given smiley icons showed in table 2.3 in tweets and labeled tweets with respect to their meanings. Moreover they pointed out that, the test set was selected independently of the presence of emoticons meaning they ignored smiley icons in test set.

Table 2.4 List of Queries Used to Create Test Set (Go et al., 2009)

Query	Negative	Positive	Total	Category
40d		2	2	Product
50d	13	5	5	Product
aig	7		7	Company
at&t	13		13	Company
bailout	1		1	Misc.
bing	1		1	Product
Bobby Flay		6	6	Person
booz allen	1	2	3	Company
car warranty call	2		2	Misc.
cheney	5		5	Person

comcast	4		4	Company
Danny Gokey		4	4	Person
dentist	9	3	12	Misc.
east palo alto	1	2	3	Location
espn	1		1	Product
exam	5	2	7	Misc.
federer		1	1	Person
fredwilson		2	2	Person
g2		7	7	Product
gm	16		16	Company
goodby silverstein		6	6	Company
google	1	4	5	Company
googleio		4	4	Event
india election		1	1	Event
indian election		1	1	Event
insects	5	1	6	Misc.
iphone app	1	1	2	Product
iran	4		4	Location
itchy	5		5	Misc.
jquery	1	3	4	Product
jquery book		2	2	Product
kindle2	1	16	17	Product
lakers		4	4	Product
lambda calculus	2	1	3	Misc.
latex	5	3	8	Misc.
lebron	4	14	18	Person
lyx		2	2	Misc.
Malcolm Gladwell	3	7	10	Person
mashable		2	2	Product
mcdonalds	1	5	6	Company
naive bayes	1		1	Misc.
night at the museum	3	12	15	Movie
nike	4	11	15	Company
north korea	6		6	Location
notre dame school		2	2	Misc.
obama	1	9	10	Person
pelosi	4		4	Person
republican	1		1	Misc.
safeway	5	2	7	Company
san francisco	3	1	4	Location
scrapbooking		1	1	Misc.
shoreline amphitheatre		1	1	Location
sleep	3	1	4	Misc.
stanford		7	7	Misc.

star trek		4	4	Movie
summize	2		2	Product
surgery	1		1	Misc.
time warner	33		33	Company
twitter		1	1	Company
twitter api	6	2	8	Product
viral marketing	1	2	3	Misc.
visa		1	1	Company
visa card	1		1	Product
warren buffet		5	5	Person
wave s&box		1	1	Product
weka	1		1	Product
wieden		1	1	Company
wolfram alpha	1	2	3	Product
world cup		1	1	Event
world cup 2010		1	1	Event
yahoo	1		1	Company
yankees		1	1	Misc.
Total	177	182	359	-

They have used 359 total tweets for testing purpose included 7 different categories that they have chosen. The testing data with its category names, the total tweets included that categories and the percentages of the given categories in test set are shown in Table 2.5.

Table 2.5 Categories for Test Data (Go et al., 2009)

Category	Total	Percent
Company	119	33.15%
Event	8	2.23%
Location	18	5.01%
Misc.	67	18.66%
Movie	19	5.29%
Person	65	18.11%
Product	63	17.55%
Grand Total	359	

They explored the usage of unigrams, bigrams, unigrams and bigrams, and parts of speech as features. Table 2.6 summarizes the results.

Table 2.6 Classifier Accuracy (Go et al., 2009)

Features	Keyword	Naive Bayes	MaxEnt	SVM
Unigram	65.2	81.3	80.5	82.2
Bigram	N/A	81.6	79.1	78.8
Unigram + Bigram	N/A	82.7	83.0	81.6
Unigram + POS	N/A	79.9	79.9	81.9

The data set is called Twitter Sentiment 140 dataset. We have downloaded this dataset total 1,600,000 tweets 800,000 of which were labeled as positive and 800,000 were labeled as negative class associated. After that we reduced the number of tweets included in this dataset by searching the queries given by (Go et al., 2009). For example, they suggested that for the category company one should search in the query "at&t", "bing", "twitter", "Time Warner", etc. we have searched the given queries included in 7 big categories and obtained 64,204 tweets, in which 34,233 are labeled as negative tweets, 29,971 are labeled as positive tweets. We used these tweets in the dataset for the enrichment of Wikipedia concepts purpose.

In a similar study (Jiang et al., 2011), they focused on target-dependent Twitter sentiment classification with given a query, they classified the sentiments of the tweets as positive, negative or neutral according to whether they contain positive, negative or neutral sentiments about that query. In their study, they used the query as the target of the sentiments. Their approach for solving sentiment classification problem adopts the target-independent strategy, which may assign irrelevant sentiments to the given target. Moreover they pointed out that, these studies were only takes the tweet to be classified into consideration when classifying the sentiment; which its context (i.e., related tweets) were ignored. On the other hand, because tweets were short and ambiguous, they pointed that, it

was not enough to consider only the current tweet for sentiment classification. They were motivated by these mistakes and they proposed to improve target-dependent Twitter sentiment classification by incorporating target-dependent features and taking related tweets into consideration. The problem they addressed was target dependent sentiment classification of tweets. That's why they pointed out that the input of the task was a collection of tweets containing the target and the output was labeling assigned to each of the tweets. They pointed that the tweet does not express any sentiments to the given target however, express sentiments to other things was considered as being opinionated about the target. To be more precise, they give an example of this concept that, this tweet expresses no sentiment to "Bill Gates" but was very likely to be classified as positive about "Bill Gates" by target independent approaches.

"People everywhere **love** Windows & vista. **Bill Gates**"

They have a three steps approach for this purpose. For step one, they focused on subjectivity classification as the first step to decide if the tweet was subjective or neutral about the target. They focused on polarity classification as the second step to decide if the tweet was positive or negative about the target if it was classified as subjective in step one. And for third and last step, they focused on Graph-based optimization as the third step to further boost the performance by taking the related tweets into consideration.

They pointed out that, the third step which; their work differed from most of the studies, they focused on Graph-based Sentiment Optimization. They focused on three kinds of related tweets. First one was the retweets. The second one was the tweets containing the target and published by the same person. And the third one was the tweets replying to or replied by the tweet to be classified.

Based on these three kinds of relations (Jiang et al., 2011), constructed a graph using the input tweet collection of a given target. As shown in Figure 2.1., each circle in the graph was indicated as a tweet. The three kinds of edges indicate being published by the same person (solid line), retweeting (dash line), and replying relations (round dotted line) respectively.

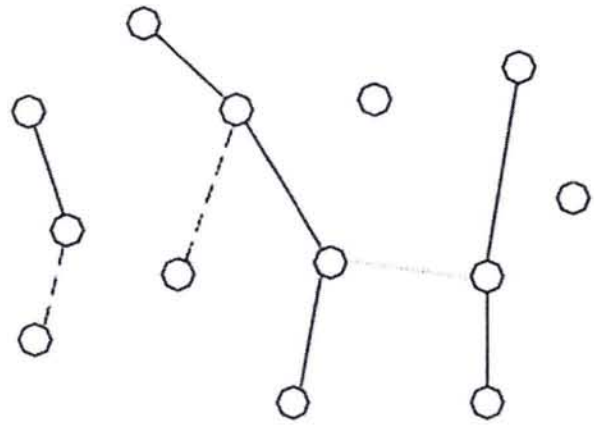

Figure 2.1 An example graph of tweets about a target (Jiang et al., 2011)

They classified each tweet as positive, negative or neutral towards the query with which it was downloaded. They obtained 459 positive, 268 negative and 1,212 neutral tweets in total for testing purpose. They concluded in experimental results, their approach greatly improved the performance of target-dependent sentiment classification.

Table 2.7 Effectiveness of the context-aware approach (Jiang et al., 2011)

		F1-Score(%)		
System	**Accuracy**	**Pos**	**Neu**	**Neg**
Target-dependent sentiment classifier	66.0	57.5	70.1	66.1
Graph-based optimization	68.3	63.5	71.0	68.5

Their results can been seen in Table 2.7. They pointed out that, the overall accuracy of the target-dependent classifiers over three classes was 66.0%. And they showed that the graph-based optimization improved the performance by over 2 points which they have tested in t-test with $p < 0.005$. They pointed that this results was showing the context information was useful for classifying the sentiments of tweets. And from these results, they found that the

context-aware approach was especially helpful for positive and negative classes in terms of accuracy.

Another study done for twitter sentiment classification purpose was (Pak and Paroubek, 2010). In the paper, they focused on using Twitter as well, since Twitter is the most popular micro blogging platform, for sentiment analysis. They showed how to automatically collect a corpus for sentiment analysis and opinion mining purposes. They performed linguistic analysis of the collected corpus. Using the corpus, they built a sentiment classifier that was able to determine positive, negative and neutral sentiments for a document. For the corpus collection they have done same approach as the paper in (Go et al., 2009). They classified the tweets with corresponding smiley icons. In table 2.8 the characteristics of the test set is shown.

Table 2.8 The characteristics of the evaluation dataset (Pak and Paroubek, 2010)

Sentiment	Number of samples
Positive	108
Negative	75
Neutral	33
Total	216

They used Tree Tagger for POS-tagging and observed the difference in distributions among positive, negative and neutral sets. They showed that the proposed classifier was able to determine positive, negative and neutral sentiments of documents. The classifier was based on the multinomial Naive Bayes classifier that uses N-gram and POS-tags as features. They pointed that N-gram based classifier uses the presence of an n-gram in the post as a binary feature. Moreover, the classifier based on POS distribution estimate probability of POS-tags presence within different sets of texts and uses it to calculate posterior probability. They concluded the paper with showing the experimental evaluations that proposed techniques were efficient and performed better than previously proposed methods. Also worked with English, however, they impressed that proposed technique could be used with any other language.

In (Barbosa and Feng, 2010), they proposed an approach to automatically detect sentiments on Twitter messages by exploring several characteristics of how tweets were written and meta-information of the words that compose these messages. Moreever, they have presented an effective and robust sentiment detection approach for Twitter messages, which used biased and noisy labels as input to build their models. Also in the paper, they influenced sources of noisy labels as their training data. These noisy labels were obtained by a few sentiment detection websites over twitter data. They are: retweets; hashtags; replies; link, if the tweet contains a link; punctuation (exclamation and questions marks); emoticons (textual expression representing facial expressions); and upper cases (the number of words that starts with upper case in thetweet). The information about the 3 data sources are shown in table 2.9

Table 2.9 Information about the 3 data sources (Barbosa and Feng, 2010)

Data Sources	URL	#Tweets	Sentiments
Twendz	http://twendz.waggeneredstrom.com/	254081	pos/neg/neutral
Twitter Sentiment	http://twittersentiment.appspot.com/	79696	pos/neg/neutral
TweetFeel	http://www.tweetfeel.com/	13122	pos/neg

(Barbosa and Feng, 2010) show that their features were able to capture a more abstract representation of tweets. As a result the proposed solution was more effective than previous ones and also more robust regarding biased and noisy data, which is the kind of data provided by these sources (i.e. Twitter). They concluded that performance increases due to the fact that, their approach created a more abstract representation of these messages compared to a raw word representation as in previous approaches. Furthermore, although the data is noisy and biased, the data sources provide labels of reasonable quality and, since they have different bias, combining them also brought some benefits. The main disadvantage of their approach was the cases of sentences that contain antagonistic sentiments meaning including no sentiments such as strong or focused.

A similar study was done in (Davidov et al., 2010). In this paper they proposed a supervised sentiment classification framework which was based on data from Twitter as well. They used 50 Twitter tags and 15 smileys's for labeling on sentiment approach purpose. They have asked 2 judges for 3852 most frequent tweeter tags for to obtain annotation results which are shown in the table 2.10. The second column displayed the average number of tags, and the last column shows % of tags annotated similarly by two judges.

Table 2.10 Annotation results for the 3852 most frequent tweeter tags (Davidov et al, 2010)

Category	# of tags	% agreement
Strong sentiment	52	87
Likely sentiment	70	66
Context-dependent	110	61
Focused	45	75
No sentiment	3564	99

They evaluated the contribution of different feature types for sentiment classification and showed that their framework successfully identifies sentiment types of untagged sentences. The quality of the sentiment identification of their approach was also confirmed by human judges. They also explored dependencies and overlap between different sentiment types represented by smiley's and Twitter hashtags.

Another study on Twitter sentiment classification was done by (Diakopoulos and Shamma, 2010). They add television broadcast to Twitter data for sentiment classification purpose. They pointed out that Television broadcasters were beginning to combine social micro-blogging systems such as Twitter with television to create social video experiences around events. Influenced with this they looked at one such event, the first U.S. presidential debate in 2008, in conjunction with aggregated ratings of message sentiment from Twitter. They began to develop an analytical methodology and visual representations that could help a

journalist or public affairs person better understand the temporal dynamics of sentiment in reaction to the debate video. For this reason they demonstrated visuals and metrics that could be used to detect sentiment pulse, anomalies in that pulse, and indications of controversial topics that could be used to inform the design of visual analytic systems for social media events.

(Diakopoulos and Shamma, 2010) demonstrated that the overall sentiment of the debate was negative and that tweeters tended to favor Obama over McCain. They show that interesting events can be detected by looking at anomalies in the pulse of the sentiment signal and that controversial topics can be identified by looking at correlated sentiment responses. They pointed out that this analysis was highly dependent on the polarized structure of a political debate, however they also have tend to explore how other events, (speeches, TV shows, sports), could also be analyzed using sentiment classification. That's why they suggested that a system embedding such metrics and visuals as they have developed in the paper could enable journalists to identify key sections of a debate performance, or could enable public affairs officials to optimize a candidate's performance.

In (Bifet and Frank, 2010), they first discussed the challenges that Twitter data streams pose, focused on classification problems, and then considered the streams for opinion mining and sentiment analysis. To avoid streaming unbalanced classes, they proposed a sliding window Kappa statistic for evaluation in time-changing data streams. Using this statistics they performed a study on Twitter data using learning algorithms for data streams.

Twitter streaming data could enable any user to discover what is happening in the world at any given moment in time. Though, the Twitter Streaming API deliver a large quantity of tweets in real time, they expressed that data stream mining and evaluation techniques are the best solution for the task at hand, but have not considered in previous work. They discussed the challenges that Twitter streaming data poses, focusing on sentiment analysis, and proposed the sliding window Kappa statistic as an evaluation metric for data streams. The evaluation results are shown in table 2.11 and 2.12 and figure 2.2.

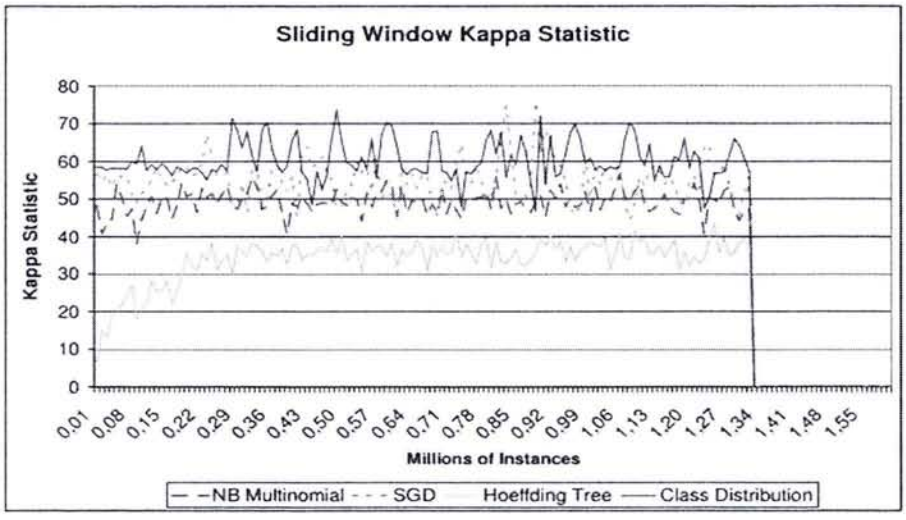

Figure 2.2 Sliding window prequential accuracy and Kappa measured on the twittersentiment.appspot.com data stream (Bifet and Frank, 2010)

Results of experiments are showed in Multinomial NB, Stochastic gradient descent (SGD) and Hoeffding Tree. SGD has experienced a revival since it has been discovered that it provides an efficient means to learn some classifiers even if they are based on non-differentiable loss functions. The most well-known tree decision tree learner for data streams is the Hoeffding tree algorithm. It employs a pre-pruning strategy based on the Hoeffding bound to incrementally grow a decision tree.

Table 2.11 Total prequential accuracy and Kappa obtained on the Edinburgh corpus data stream (Bifet and Frank, 2010)

	Accuracy	**Kappa**	**Time**
Multinomial Naive Bayes	86.11%	36.15%	173.28, sec
SGD	86.26%	31.88%	293.98 sec.
Hoeffding Tree	84.76%	20.40%	6151.51 sec.

Table 2.12 Accuracy and Kappa for the test dataset obtained from twittersentiment.appspot.com using the Edinburgh corpus as training data stream (Bifet and Frank, 2010)

	Accuracy	**Kappa**
Multinomial Naive Bayes	73.81%	47.28%
SGD	67.41%	34.23%
Hoeffding Tree	60.72%	20.59%

They have considered all tests performed and ease of interpretability, and they pointed out that the SGD-based model, used with an appropriate learning rate, could be recommended for this data.

There are several studies which aim to increase classification accuracy by exploiting semantic relations in Wikipedia.

(Genc et al., 2011) employed Wikipedia based transform for classifying tweets. They have mapped twitter messages onto their most similar Wikipedia pages and the distances between pages are used for as a proxy for the distances between messages. In order to do so, they had two steps; in first step they have mapped tweets to Wikipedia pages, and then computed the distance between the Wikipedia pages as a measure of semantic distance between the tweets.

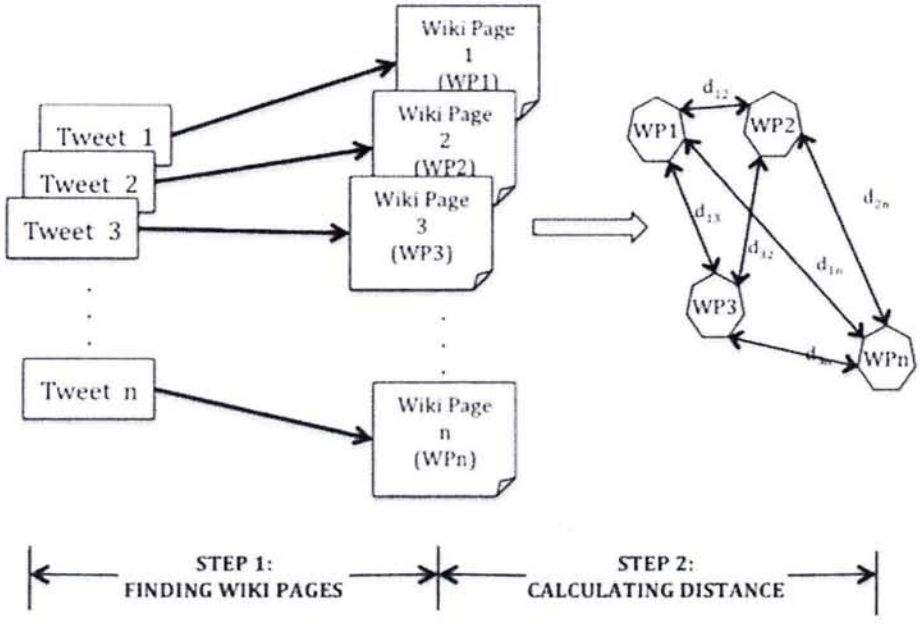

Figure 2.3 Steps of used approach (Genc et al., 2011)

First, they identified a set of words for the given tweets to associate a tweet to a Wikipedia page. They pointed that, after the stop word elimination which they apply Latent Semantic Analysis (LSA) package, all the words were included in the word set. And then for each word, they checked to see if there was a direct page dedicated to the word. And also they have checked for existing of a disambiguation page which was precise mapping to the right page, leading to more accurate distance measures. After that, they founded a list of candidate pages for the tweet by aggregating each page associated with each word of the word set. And a score was calculated for each candidate Wikipedia page by counting the number of occurrences of the words in the word set. Finally, the page with the highest score was selected as the associated Wikipedia page for the tweet. Model of finding a Wikipedia page with associated tweet is shown in figure 2.5.

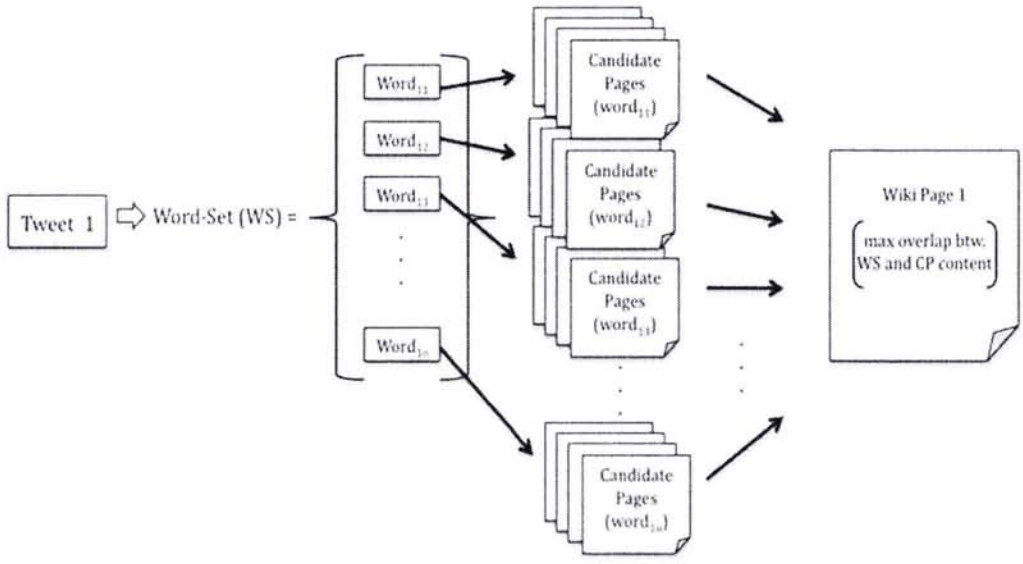

Figure 2.4 Finding a Wikipedia page associated with a tweet (Genc et al., 2011)

For calculating the distance of two Wikipedia pages, their work was based on the linking between the categories associates with these two pages. The categories of the Wikipedia pages were linked to one another in a graph structure. They have captured the network structure of categories for each Wikipedia page for five levels and computed the semantic distance between the two Wikipedia pages by finding the length of the shortest path from a category of one page to a category of the other page. This calculation approach is shown in figure 2.5

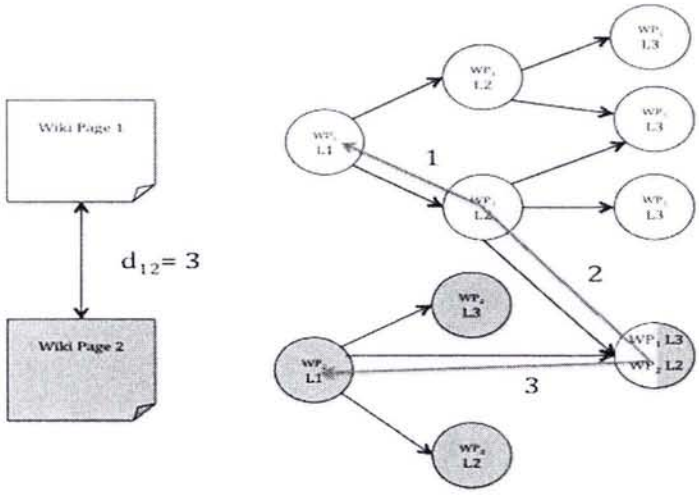

Figure 2.5 Calculating the distance between two Wikipedia pages (Genc et al., 2011)

Finally, they have showed that technique was more accurate than alternative techniques of string edit distance and Latent Semantic Analysis (LSA) for classification of a set of twitter messages.

While Wikipedia has been commonly used in text classification for semantic purpose, there are studies which focus on to transform Wikipedia into a structured thesaurus. In (Wang at al., 2009), they first find the Wikipedia concepts in given document. When candidate concepts have been found these were added into the document along with their related concepts with the usage of synonymy or hyponymy or hierarchical relation or associative relation. They have used Reuters, Ohsumed and 20 Newsgroups datasets and SVM algorithm for classification. For measuring the classifiers performance Precision and Recall were used. The results show 4.5% improvement over baseline with the help of associative concepts and hyponyms on documents.

A similar study to our approach was done by (Hu et al., 2009). To add semantic knowledge they enriched document representation with the background knowledge. In the paper, they presented a novel text clustering method to use semantic knowledge by enriching document representation with Wikipedia concept and category information. They have developed two approaches, exact match and relatedness-match, to map text documents to Wikipedia concepts, and further to Wikipedia categories. The framework of the approach of the study in is shown in figure 2.6.

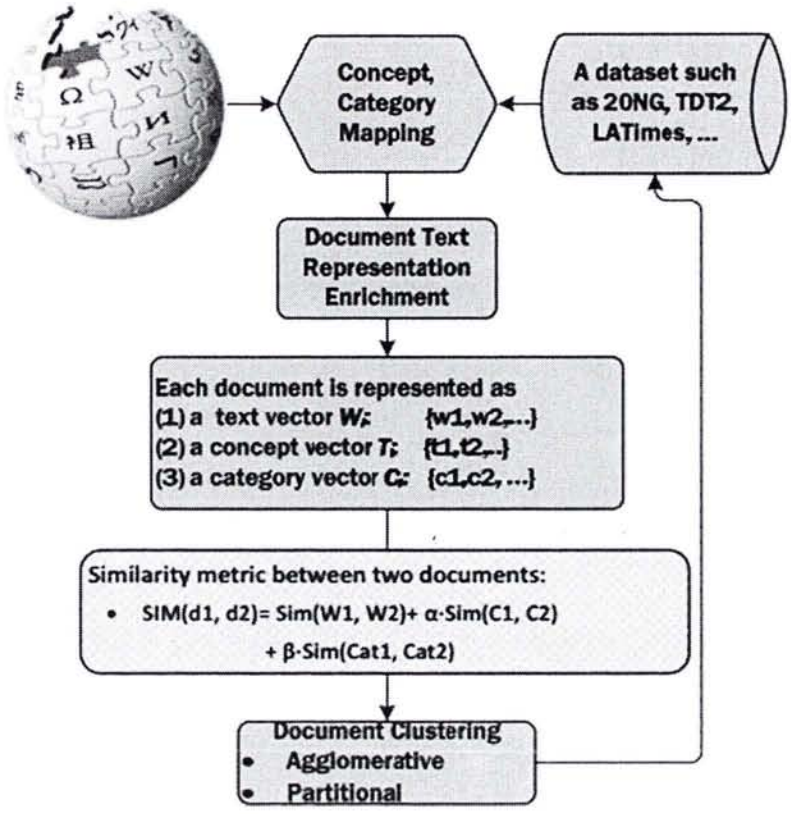

Figure 2.6 The framework of leveraging Wikipedia for document clustering (Hu et al., 2009).

They tested their approach on three datasets; 20-newsgroup, TDT2, and LA Times. They showed that clustering performance improves significantly by enriching document representation with Wikipedia concepts and categories.

Similarly, in (Huang et al., 2009) they showed how Wikipedia based semantic knowledge can be exploited for document clustering. They created a concept-based document representation by mapping the terms and phrases within documents to their corresponding articles (or concepts) in Wikipedia. Using similarity measure they evaluated the semantic relatedness between concepts sets for two documents. Finally, they have tested the concept-based representation and the similarity measure on two standard text document datasets as Reuters and Ohsumed. Their results, show that the use of Wikipedia based semantic knowledge improves the clustering performance.

In (Luo et al., 2011), they proposed a novel term weighting scheme by exploiting the semantics of categories and indexing terms. Specifically, the semantics of categories are represented by senses of terms appearing in the category labels as well as the interpretation of them by WordNet. Also, the weight of a term is correlated to its semantic similarity with a category. They tested results on three commonly used data sets; Reuters, 20News Groups, WebKB. They showed that the proposed approach outperforms term frequency–inverse document frequency (TF-IDF) in the cases that the amount of training data is small or the content of documents is focused on well-defined categories. Also they pointed out that, the proposed approach compared favorably with two previous studies.

A corpus-based thesaurus and WordNet were used to improve text categorization performance in (Li et al., 2012). They employed the k-Nearest Neighbor (k-NN) algorithm and the Back Propagation Neural Network (BPNN) algorithms as the classifiers. The k-NN is a simple yet effective algorithm for text categorization and the BPNNs has been widely used in the categorization and pattern recognition fields. On the other hand, the standard BPNN has some generally acknowledged limitations, such as a slow training speed and can be easily trapped into a local minimum. To alleviate the problems of the standard BPNN, two modified versions, Morbidity neurons Rectified BPNN (MRBP) and Learning Phase Evaluation BPNN (LPEBP), were considered and applied to the text categorization. In WordNet, words are organized into taxonomies where each node is a set of synonyms (a ''synset'') representing a single sense. In experiments they use only a noun taxonomy with hyponymy/hypernymy relations (or an is-a relation). They conducted the experiments on both the standard Reuter-21578 data set and the 20 Newsgroups data set. They showed that their methods achieved high categorization effectiveness as measured by the precision, recall and F-measure protocols.

Although there are numerous studies using English Wikipedia in semantic analysis, there are limited numbers of studies using Turkish Wikipedia (Vikipedi) for text mining. Among those Poyraz et al. (2012) employ a similar approach as ours. They used bag of words (BOW) model and Wikipedia enrichment on Turkish data sets where obtained from Turkish newspapers articles. They have used 1150Haber, AA Haber, and Hurriyet_6c1k dataset. For the classification both multinomial Naïve Bayes (MNB) and Support Vector

Machine (SVM) classifiers were used and the baseline BOW representation and wiki enrichment were discussed and compared. Accuracy improvement results are shown in table 2.12.

Table 2.13 Accuracy Improvement over baseline (Poyraz et al., 2012)

DATA SET	NB	SVM
1150 HABER	0,09%	0,60%
AAHABER	0,45%	0,08%
AAHABER-18428	0,30%	0,47%
HÜRRİYET-6C1K	0,66%	1,13%

They showed that there was a slight improvement on accuracy when they enrich the data with Wikipedia concepts. On the other hand, they have only used multiword Wikipedia Article titles as Wikipedia concepts which differ from our approach as we enriched the twitter data not only with Wikipedia articles titles but also with categories and redirects of the articles. Our approach differs from this work from three ways. First one is by adding not only multiword Wikipedia articles but also categories and redirects of related article titles. Second, we used Wiki Semantic Smoothing (WSS) motivated from (Zhou et al., 2008) and obtained better accuracy results. Third, we use English twitter data set to be enriched with respect to Wikipedia knowledge.

3. APPROACH

In this section, we first give brief explanation of Naïve Bayes Algorithm (McCallum and Nigam, 1998), smoothing methods Laplace Smoothing, Jelinek Mercer Smoothing (Chen and Goodman, 1998) and semantic smoothing approach used in (Zhou et al., 2008). Following this we focus on our approach which is based on the algorithm (Zhou et al., 2008). Our approach is called Wikipedia based Semantic smoothing (WSS) model we provide details about the components used in our model which are namely; Freebase Wikipedia Extractor, Term Extractor and Wiki concept extractor. Then we give explanation about the topic signatures (TS) and advantages of using them motivated by the work (Zhou et al., 2008). Finally we give detailed information about our topic signatures; Wikipedia Article titles, categories and redirects.

3.1. Naïve Bayes Algorithm

In text classification, one of the most commonly used and popular machine learning algorithms is Naïve Bayes (McCallum and Nigam, 1998) due to its easy implementation and low complexity. For demonstration text document purpose there are two used event models used in Naïve Bayes; binary and multinomial Naïve Bayes.

In our approach we use multinomial Naïve Bayes model (MNB). Multinomial Naïve Bayes uses term frequencies instead of "1" or "0" binary values. Compared to multivariate Bernoulli model, in MNB documents are denoted by a vector of term counts. The class conditional probability of the term w_t in class c_j is given by:

$$P(w_t|c_j) = \frac{1+\sum_{i=1}^{|D|} N_{it} P(c_j|d_i)}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{|D|} N_{is} P(c_j|d_i)} \quad (1)$$

Where $|V|$ is vocabulary (total number of words) and N_{it} is count of the number of times word w_t occurs in document d_i.

3.2. Smoothing Methods

Missing terms (unseen events) in the documents can cause zero probability which is simply called sparsity problem in NB (Zhou et al., 2008). To eliminate this problem, smoothing methods are used. This is simply distributing some probability mass by decreasing the probabilities of existing terms and assigning the extra probability mass to unseen terms. Two of commonly used smoothing algorithms are Laplace Smoothing and Jelinek-Mercer Smoothing. Motivated by (Zhou et al., 2008) we used Semantic Smoothing approach for to deal with sparsity problem in Naïve Bayes algorithm.

3.2.1. Laplace Smoothing

To eliminate zero probability problem one solution could be Laplace Smoothing. Laplace smoothing is the most common smoothing method used NB. This method is simply adding a pseudo count to every unseen term counts to avoid zero probability problems. Although, this method is highly used its main disadvantage is to give too much probability mass to previously unseen events for sparse sets of data over large vocabularies. Probability of term given class with Laplace for multinomial Naïve Bayes (Manning and Schütze, 1999):

$$P_{c,t} = \frac{1 + N(c_t, D)}{|V| + N(c, D)} \tag{2}$$

where $P_{c,t}$ is the probability of term given class. $N(c_t, D)$ represents number of documents in class c that contain term t and $N(c, D)$ denotes number of documents in class c.

3.2.2. Jelinek-Mercer Smoothing

Smoothing methods are separated in two main methods whether the smoothing is interpolated or backed-off (Chen and Goodman, 1998). Jelinek-Mercer smoothing method is fall under interpolated models since the maximum estimate is interpolated with the smoothed lower-order distribution (Chen and Goodman, 1998).

$$P_{ml}(w_t|c_j) = \frac{N(c_t)}{N(c,D)} \tag{3}$$

where $P_{ml}(w_t|c_j)$ is the probability with maximum likelihood estimate and $P(w_t|D)$ is the maximum likelihood estimation of term t in collection D where D is the total number of documents.

$$P(w_t \mid c_j) = (1-\beta) \times P_{ml}(w_t \mid c_j) + \beta \times P(w_t \mid D) \tag{4}$$

where β can be set to some constant. Due to zero probability problems, we again need to smooth maximum likelihood estimation of the term t in collection D.

For multinomial event model $P(w_t \mid D)$ is:

$$P(w_t \mid D) = \frac{1 + N(w_t, D)}{|V| + N(D)} \tag{5}$$

where $N(w_t, D)$ is the total number of term counts, and $N(D)$ represents the total number of documents in collection D.

3.2.3. Semantic Smoothing

The semantic smoothing approach statistically maps topic signatures in all training documents of a class into single-word features (Zhou et al., 2008). They linearly interpolated the semantic mapping component with a simple language model as described in equation 6 from the paper (Zhou et al., 2008):

$$P_s(w|c_i) = (1-\delta)P_b(w|c_i) + \delta \sum_k P(w|t_k)\, P(t_k|c_i) \quad (6)$$

In equation 6, $P_s(w|c_i)$ is unigram class model with semantic smoothing and t_k is for k-th topic signature and $P(t_k|c_i)$ are stands for distribution of topic signatures in training documents of a given class. Also, in paper (Zhou et al., 2008) they pointed out that, equation 6 can be estimated by maximum likelihood estimate. δ is the coefficient for to control the influence of semantic mapping component in mixture model. If this coefficient was set to zero it can be turned to simple language model. They pointed out that here the problem was how to compute $P(w|t_k)$. For each topic signature t_k, they obtained a set of documents (D_k) containing the signature. Additionally, they used the document set D_k to approximate the semantic mapping from t_k to single-word features in the vocabulary. They suggested that this would be unrealistic that assuming that all words appearing in $D_k$would be including in that topic signature t_k. Because of that reason they did not just apply maximum likelihood estimate some words would address topics corresponding to other topic signatures while some were background words of the whole collection. Therefore they employ a mixture language model to remove the noise:

$$P(w|D_k) = (1-\alpha)P(w|t_k) + \alpha P(w|D) \quad (7)$$

When the mixture model was used for text generation, it was unknown regarding what model a word was exactly generated by. They said that, it was instead a hidden variable.

On the other hand, the chance of selection either model was known. In the formula α was denoted as the coefficient accounting for the chance of using the background collection model to generate words. Moreover, the log likelihood of generating the document set D_k was:

$$\log P(D_k) = \sum_w c(w, D_k) \log P(w|D_k) \quad (8)$$

Here $c(w, D_k)$ denotes the document frequency of term w in D_k, i.e., the occurrence count of w and $t_k$shows occurrence count of term w in all collection. They estimated $P(w|t_k)$ by EM algorithm (Baker et al., 1998) with following;

The Expectation step was set initial values,

$$\acute{P}^{(n)}(w) = \frac{(1-\alpha)P^{(n)}(w|t_k)}{(1-\alpha)P^{(n)}(w|t_k)+\alpha P(w|C)} \quad (9)$$

For Maximization step they have continued this process till it converged.

$$P^{(n+1)}(w|t_k) = \frac{c(w,D_k)\acute{P}^{(n)}(w)}{\sum_i c(w_i,D_k)\acute{P}^{(n)}(w)} \quad (10)$$

And finally they showed, EM algorithm was initialized by the maximum likelihood estimator with regarding setting the background coefficient α. They concluded that the larger α gets the more specific the trained parameters were.

Motivated by this study (Zhou et al., 2008) we implement the same approach and used the same formulas with EM algorithm to obtain baseline results of this study. We extend their

work by adding Wikipedia Articles, Categories and Redirects as topic signatures thus adding semantic knowledge for sentiment classification of twitter messages.

3.3. Wikipedia Based Semantic Smoothing Model

"Wikipedia is a collaboratively edited, multilingual, free Internet encyclopedia supported by the non-profit Wikimedia Foundation. Its 25 million articles, over 4.1 million in the English Wikipedia alone, are written collaboratively by volunteers around the world"[1]. In Wikipedia Articles there are rich information about the concept and more as synonymy and hyperlinks. To gain excess the Wikipedia resource we used Freebase Wikipedia Extraction[2] and its processed dump of English Wikipedia in xml format. We extracted the Wikipedia Article titles as Wikipedia concepts and categories that belong to article titles and redirects respectively. We have used English Twitter data as the main source and enriched the data with topic signatures by using semantic smoothing approach used in (Zhou et al., 2008), not only with English Wikipedia Articles titles but also with Wikipedia Articles Categories and Wikipedia Articles Redirects. Our system is similarly summarized and explained in details in the following figure 3.1

[1] (2013) The Wikipedia website [Online] Available: http://en.wikipedia.org/wiki/Wikipedia

[2] Google, Freebase Wikipedia Extraction (WEX), http://download.freebase.com/wex/, <08> <06>, <2012>

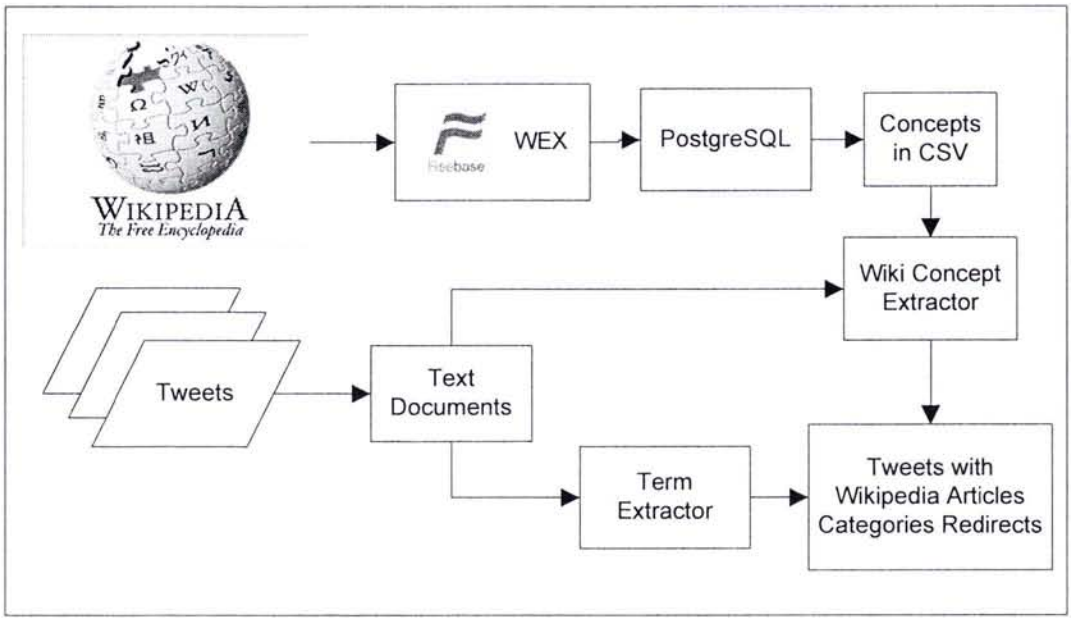

Figure 3.1 Design of the System

3.3.1. Freebase Wikipedia Extractor

The Freebase Wikipedia Extraction[3] (WEX) is a processed dump of the English language Wikipedia. Each article is transferred to readable XML format common relational features like templates, categories, article sections, and redirects are extracted in tabular form. Freebase WEX is provided as a set of database tables in TSV format for PostgreSQL[4], along with tables providing mappings between Wikipedia articles and Freebase topics, and corresponding Freebase Types.

Wikipedia dump was retrieved in August 6th, 2012 and in the PostgreSQL there were 6,108,629 Wikipedia articles, 5,587,540 Wikipedia redirects and 17,356,454 Wikipedia categories which are shown in table 3.3.1 in detailed.

[3] Google, Freebase Wikipedia Extraction (WEX), http://download.freebase.com/wex/, <08> <06>, <2012>

[4] http://www.postgresql.org/

Table 3.1 Wikipedia Dump Size

Wikipedia Dumps	# of size
Wikipedia article titles	6,108,629
Wikipedia categories	17,356,454
Wikipedia redirects	5,587,540

Each of article titles describes a topic which we call concepts. These concepts can be single or multiple consecutive words that can be named entity, a compound word or a commonly used term in a specific domain.

We have used WEX to obtain Wikipedia tables and added them to PostgreSQL. From database which necessary queries we have added Wikipedia article titles, Wikipedia categories with respect to their article titles, Wikipedia redirects with respect to their article titles and Wikipedia article titles with categories and redirects into 4 different comma-separated values (csv) form.

3.3.2. Term Extractor

Term Extractor creates an array vector that includes words term frequencies that occur in given text. These terms are added to their term-frequency vector. In Term Extractor, each text document is represented as term-frequency vector.

3.3.3. Wiki Concept Extractor

Obtained from Wikipedia dumps we have all the information on Wikipedia article titles, using PostgreSQL database. Wiki Concept Extractor searches for Wikipedia article titles, categories and redirects. These concepts could be one, two or three word phrases. We have limited this search starting with two word phrases as occurrence of one word phrase would

not add a semantic knowledge as it is already included in tweets. Then, given concepts are searched of occurrence in given tweets. If all the separated words of wiki concepts occur in given tweets, they are added to term frequency as if they are a unique attribute entity. By this way we exploit semantic relationships between terms in the tweets. To give an example "The White House" is the official residence and principal workplace of the President of the United States, is a Wikipedia article title. After preprocessing we obtain, "White" and "House", we check this two separated word occurrences in given tweet. Without this approach in let's say the tweet is, "My dream is to see the white house" all these words will be represented as separate terms in a vector space (Bag of Words approach) and their semantic relationship will be disregarded. On the other hand with our approach Wiki Concept Extractor will add semantic knowledge of the multiword phrase of "White house" which is associated with single words using EM algorithm in the further steps.

3.3.4. Topic Signatures

In paper (Zhou et al., 2008) they suggested that there was no strict definition for topic signatures. And any topic carrier could be viewed as a topic signature. For this reason in (Zhou et al., 2008) they considered three types of topic signatures which were unigrams (single-word features), multiword phrases, and ontological concepts, respectively. They pointed out that a concept was a unique meaning in a specific domain which represents a set of synonymous terms in the domain. In their study, they used UMLS concepts as topic signatures for the corpus of OHSUMED.

They suggested that a multiword phrase could consist of two or more words adjacent to each other which were a kind of fixed expressions or collocations. "Space Program", "Third World Debt", and "Machine Learning" are some examples of multiword phrases. They said that these phrases could be viewed as n-grams with syntactic and statistical constraints. They used in their work multiword phrases rather than n-grams because of the forming was making more sense when using its semantics. On the other hand, they concerned about the complexity of extracting these phrases. To obtain lowest complexity they only extracted phrases from training documents. They have used Xtract tool to obtain multiword phrases automatically.

In paper (Zhou et al., 2008), they have introduced ontological concepts and multiword phrases as topic signatures because both were less ambiguous than single-word topic signatures.

We motivated by the concept Topic signatures from the paper (Zhou et al., 2008). And we improved these concepts by using Wikipedia Articles, Categories and Redirects as topic signatures and used them in semantic approach to obtain better accuracy results by enriching the Twitter dataset.

3.3.5. Wikipedia Articles, Categories and Redirects

Wikipedia is a collaboratively edited, multilingual, free Internet encyclopedia supported by the non-profit Wikimedia Foundation. Its 25 million articles, over 4.1 million in the English Wikipedia alone, are written collaboratively by volunteers around the world. For example while searching the title "Barack Obama" you will come across to this picture shown in Figure 3.2

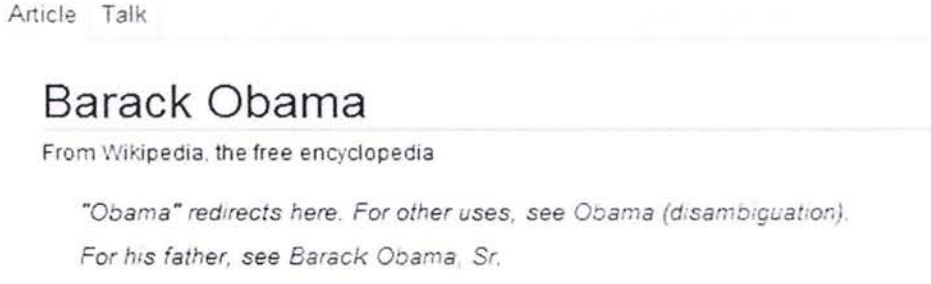

Figure 3.2 Search For "Barack Obama" in Wikipedia (Retrieved from Wikipedia website)

After, the searches of "Barack Obama" in down of the same page the categories of "Barack Obama" are shown in Figure 3.3.

Figure 3.3 Categories of "Barack Obama" in Wikipedia (Retrieved from Wikipedia website)

"Barack Obama" Wikipedia Article has 43 distinct categories which all are added as topic signatures to enrich the Twitter data. And this topic has 8 redirect pages that retrieve the same page "Barack Obama". The redirects of this article are shown in Figure 3.4.

Figure 3.4 Redirects of "Barack Obama" in Wikipedia (Retrieved from Wikipedia website)

In the paper (Poyraz et al., 2012), only multiword Wikipedia article titles are added to the corpus. But in order to get better accuracy Wikipedia categories belongs to given

Wikipedia article and redirects should be added. On the other hand Wikipedia categories and redirects are hard to find in given dataset especially on twitter data set while people are micro blogging their statuses and writing like dialect.

Consequently, we only used Wikipedia Term Extractor for checking the occurrence of Wikipedia articles in given tweet. If the given wiki concepts is seen, then its categories are added as if they are seen in the given tweet. In this way we add more semantic relation between the words. And redirects are added as the equal way. To give an example for Wikipedia article "The White House" has 16 categories included "Houses completed in 1800", "Buildings of the United States government in Washington, D.C" and etc. It is hard to find those wiki concepts seen in tweets as status update only allows to write 140 characters. Thus, we add these categories if their respected article is found in given tweet. With this approach we add more semantic knowledge about the "The White House" itself. For the redirects, they are the correct form of articles. If user writes an incorrect article and searches it from Wikipedia, it redirects the intended article. The redirects are used for this purpose. To give an example, "Accessible Computing" is not a Wikipedia article but if user searches for this headline, Wikipedia redirects user to "Computer accessibility". With this way we eliminate the wrong meaning of the given bigrams and add the correct one instead.

To give an example our tweet is given as;

"**Barack** wins the election. As everybody knows **Obama** was the best candidate"

or

"**Barack Obama** wins the election. As everybody knows **Obama** was the best candidate"

After tokenization process we obtain attributes as:

"Barack, wins, the, election, as, everybody, knows, Obama, was, the, best, candidate"

All these words are represented as separate terms in a vector space (Bag of Words approach) and their semantic relationship are disregarded especially "Barack Obama" is unnoticed. To eliminate this problem, we added semantic relations of Wikipedia Articles,

categories and redirects. From Wiki Concept Extractor we obtain "Barack" and "Obama" words separately and we check the existence of both words in given tweets if the both words are found in given tweet then the "Barack Obama" Wikipedia Article is added to corresponding tweet as a topic signature. So the new attributes will be:

"Barack, wins, the, election, as, everybody, knows, Obama, was, the, best, candidate, 'Barack Obama' "

Moreover, after finding the existence of Wikipedia article title we do not check the existence of the categories in the tweet as the limitation of 140 characters.

4. EXPERIMENTAL SETUP

4.1. Twitter Data Set

Recent years, micro blogging become one of very popular and commonly used communication tools among Internet users. Sharing opinions on different aspects become a new trend among millions of users. Moreover, micro blogging cites, being a rich source of data for opinion mining and sentiment analysis, has turned to a subject mostly worked on. Using the good results on via sentiment classifications researches and studies are focused on classifying the twitter with sentiment classification and showing written reviews are positive or negative with respect to the given topic, to understand the user's feelings of the product or a company.

Among all micro blogging sites Twitter is the most popular of all where users create status messages called "tweets". For this reason, Twitter has become the one of the most used source in sentiment classification. In paper (Go et al., 2009) their aim was obtaining twitter data for sentiment classification. We use the same data set obtained from (Go et al., 2009) but we narrowed down the number of tweets which is going to be explained in details.

In (Go et al., 2009), they focused on expressing the characteristics of smiley icons and these icons can mean a positive or negative emotion which can be used for sentiment classification. For example, ":)" and ":-)" both express positive emotion. Thus they searched for the queries including positive smiley icon ":)", and labeled the obtained tweet as positive. So the query ":(" will return tweets and these tweets will be labeled as negative. The full list of emoticons can be found in Table 4.1.

Table 4.1 List of Emoticons (Go et al., 2009)

Emoticons mapped to :)	Emoticons mapped to :(
:)	:(
:-)	:-(
:)	: (
: D	
=)	

The tweets in their training set were from the time period April 6, 2009 to June 25, 2009, totally 1,600,000 tweets. After obtaining tweets with given smiley icon queries they used post-processing. They mainly had 5 processing filters, which are; first, they eliminated smiley icons from Table 4.1 from the obtained tweets; they pointed that with including these icons in tweets the Maximum Entropy model and SVM classifiers would put a large amount of weight on the emoticons, in which the accuracy results could be badly affected. Second, any tweets both containing positive and negative icons were removed to avoid any confusion. For example, "I love my ipad :) but sometimes it is really hard to use :(". In their paper, they omitted these tweets because they didn't want positive features marked as part of a negative tweet, or vice-versa. Third filter was removing retweets which are processes of coping another user's tweet. To remove these tweets they searched the query term for "RT" which means retweeted from someone else. Forth filter was removing tweets including icons like ":P". They implied that these icons would not imply any positive and negative sentiment. For last filter, they omitted repeated tweets.

But this is not enough for classifying the tweet. So in (Go et al., 2009) they have created two sets of words one for positive emotions and one for negative emotions. The word "love" included tweets classified as positive and the word "hate" included tweets classified as negative. The tweets included both words or both smiley icons were eliminated for misclassification issues and given specific product names or person names 1,600,000 tweet were classified.

They collected their test data manually by using the web application. After obtaining 359 total tweets they manually labeled 177 tweets as negative and 182 tweets as positive. Moreover, they pointed that they their test data were not included some of smiley icons. To obtain testing collection they followed two steps. Step 1, was searching the Twitter API with special queries which were arbitrarily chosen from different domains. For example, these queries consist of consumer products (40d, etc), companies (at&t, etc), and people (Obama, etc). They obtained these query terms manually to test tweets. In Table 4.2 list of the queries are shown. And then they worked on tweets obtained by these query searching and marked them as negative or positive if seen sentiment in tweets. Moreover they pointed out that, the test set was selected independently of the presence of emoticons.

Table 4.2 a List of Queries Used to Create Test Set (Go et al., 2009)

Query	Negative	Positive	Total	Category
40d		2	2	Product
50d	13	5	5	Product
aig	7		7	Company
at&t	13		13	Company
bailout	1		1	Misc.
bing	1		1	Product
Bobby Flay		6	6	Person
booz allen	1	2	3	Company
car warranty call	2		2	Misc.
cheney	5		5	Person
comcast	4		4	Company
Danny Gokey		4	4	Person
dentist	9	3	12	Misc.
east palo alto	1	2	3	Location
espn	1		1	Product
exam	5	2	7	Misc.
federer		1	1	Person
fredwilson		2	2	Person
g2		7	7	Product
gm	16		16	Company
goodby silverstein		6	6	Company
google	1	4	5	Company
googleio		4	4	Event
india election		1	1	Event
indian election		1	1	Event
insects	5	1	6	Misc.
iphone app	1	1	2	Product
iran	4		4	Location
itchy	5		5	Misc.
jquery	1	3	4	Product
jquery book		2	2	Product
kindle2	1	16	17	Product
lakers		4	4	Product
lambda calculus	2	1	3	Misc.
latex	5	3	8	Misc.
lebron	4	14	18	Person
lyx		2	2	Misc.
Malcolm Gladwell	3	7	10	Person
mashable		2	2	Product
mcdonalds	1	5	6	Company

naive bayes	1		1	Misc.
night at the museum	3	12	15	Movie
nike	4	11	15	Company
north korea	6		6	Location
notre dame school		2	2	Misc.
obama	1	9	10	Person
pelosi	4		4	Person
republican	1		1	Misc.
safeway	5	2	7	Company
san francisco	3	1	4	Location
scrapbooking		1	1	Misc.
shoreline amphitheatre		1	1	Location
sleep	3	1	4	Misc.
stanford		7	7	Misc.
star trek		4	4	Movie
summize	2		2	Product
surgery	1		1	Misc.
time warner	33		33	Company
twitter		1	1	Company
twitter api	6	2	8	Product
viral marketing	1	2	3	Misc.
visa		1	1	Company
visa card	1		1	Product
warren buffet		5	5	Person
wave s&box		1	1	Product
weka	1		1	Product
wieden		1	1	Company
wolfram alpha	1	2	3	Product
world cup		1	1	Event
world cup 2010		1	1	Event
yahoo	1		1	Company
yankees		1	1	Misc.
Total	177	182	359	-

They have used 359 total tweets for testing purpose included 7 different categories that they have chosen. The testing data with its category names and the total tweets included that categories are shown in Table 4.3.

Table 4.3 Categories for Test Data (Go et al., 2009)

Category	Total	Percent
Company	119	33.15%
Event	8	2.23%
Location	18	5.01%
Misc.	67	18.66%
Movie	19	5.29%
Person	65	18.11%
Product	63	17.55%
Grand Total	359	

The data set is called Twitter Sentiment 140 dataset. We have downloaded this dataset total 1,600,000 tweets 800,000 of which were labeled as positive and 800,000 were marked as negative class only by using emotions. In order to align the test set with the training set of 1,600,000 tweets we use the query terms provided in Table 4.2 to filter the tweets in training set. It turns out that 64,204 out of 1,600,000 tweets include one or more of these query terms. In our experiments we use these 64,204 tweets as our labeled dataset. We do not use the original test set provided by (Go et al., 2009) which consists of 359 tweets. Please note that 64,204 tweets we use in our experiments is labeled only using emotions and therefore is highly noisy.

After that we reduced the number of tweets included in this dataset by using the queries given in table 4.2. For example, they suggested that for the company category one should search the query "at&t", "bing", "twitter", "Time Warner", etc. we have searched the given queries included in 7 big categories and obtained 64,204 tweets, in which 34,233 are labeled as negative tweets, 29,971 are labeled as positive tweets. Then we used these tweets in the dataset for the enrichment of Wikipedia concepts purpose.

- Twitter Data Set (TW): Twitter Sentiment 140 dataset narrowed to 64,204 tweets with given 7 big categories search.

4.2. Twitter Enriched with Wikipedia Articles, Categories & Redirects Data Sets

To see the effect of wiki semantic smoothing on text classification on small training data like twitter corpus we have set 4 different types of datasets. Four of which were enrichment of twitter corpus. These datasets namely, Wikipedia article titles, Wikipedia articles categories, Wikipedia articles redirects, Wikipedia articles categories redirects.

On our tweet corpus we have 64204 tweets, in which 34233 are labeled as negative, 29971 are labeled positive. From tweeter corpus we have obtained 4 different types of dataset.

- Twitter enriched with Wikipedia Articles (TWA): Wiki concept extractor only added Wikipedia articles with respect to seen in tweet with their term frequencies.
- Twitter enriched with Wikipedia Articles and with respect to their categories (TWAC): Tweets categories were added to twitter data with term-frequency of 1 without checking if they actually exist in the tweet or not. If the same category matches a Wikipedia article previously added to the tweet the term frequency of category is increased.
- Twitter enriched with Wikipedia Articles and with respect to their redirects (TWAR): The same approach repeated for redirects of the Wikipedia articles.
- Twitter enriched with Wikipedia Articles and with respect to their categories and redirects (TWACR): Tweets are enriched with articles categories and redirects. Duplicated of categories and redirects were omitted and term frequency is increased.

Table 4.4 shows the description of the datasets with respect to their attributes articles categories and redirect numbers.

Table 4.4 Description of the Datasets

Data Sets	# of Attributes	# of Titles	# of Categories	# of Redirects
TWA	56661	4042	0	0
TWAC	68084	4042	15197	
TWAR	70300	4042		12352
TWACR	83639	4042	25282	

In table 4.4, for TWA dataset we have 4,042 Wikipedia articles seen in tweets. This is a low number compared to the given 6,108,629 Wikipedia articles on the other hand with micro blogging in given limited number of characters as 140, it is hard to write the correct form of the word, instead users write as dialects. We have come up with the words as "nooooooooooooo", "loveeeeeeeeeee" and so on. With this type of written Wiki concept extractor had difficulties to find and add the given articles. In TWAC representation we have added 4,042 Wikipedia articles included with 15,197 categories so the tweets are more enriched compared to TWA representation. In TWAR, we have added only 12,352 redirects with respect to their 4,042 Wikipedia articles. For the final dataset type TWACR we have added total number of 25,282 attributes included categories and redirects with respected articles. Even though the summation of categories and redirects were 27,549 attributes, we can underline that there were duplicate of words between categories and redirects, meaning the same word was both a category and a redirect.

5. EXPERIMENTAL RESULTS

We apply Naïve Bayes Wiki semantic smoothing (NBWSS) and multinomial Naïve Bayes Laplace smoothing (MNB) and Support Vector Machines with linear kernel (LibSVM) algorithms to each of our datasets. For each data set we performed 10-fold cross-validation and report average accuracy.

We want to show performance results in our approach NBWSS compared to MNB and SVM on datasets with Wikipedia enrichments with article titles, categories and redirects. For this purpose we have tested 4 data sets; Twitter with Wikipedia Article titles (TWA), Twitter with Wikipedia Article titles and categories (TWAC), Twitter with Wikipedia Article titles and redirects (TWAR) and Twitter with Wikipedia Article titles, categories and redirects (TWACR) on three algorithms; MNB and SVM and our approach NBWSS. Experiments are done in different training percentages namely, 5, 10, 30, 50 and 70 %.

In Figure 5.1 and table 5.1, we show the accuracies of NBWSS, MNB and SVM on data set TWA, where tweets are enriched with Wikipedia articles only. As shown in the figure, with low training sizes wiki semantic smoothing performs similarly with MNB but gets better accuracy compared to SVM. When training size increases SVM gets higher accuracy, yet this algorithm gives lower accuracy compared to Naïve Bayes algorithms. NBWSS increases accuracy approximately by ~1%.

Table 5.1 Accuracy of MNB, SVM and NBWSS on TWA Data Set

TS	MNB	SVM	NBWSS
5	0.6954	0.6527	**0.7020**
10	0.7107	0.6830	**0.7169**
30	0.7293	0.7160	**0.7360**
50	0.7365	0.7257	**0.7440**
70	0.7397	0.7311	**0.7482**

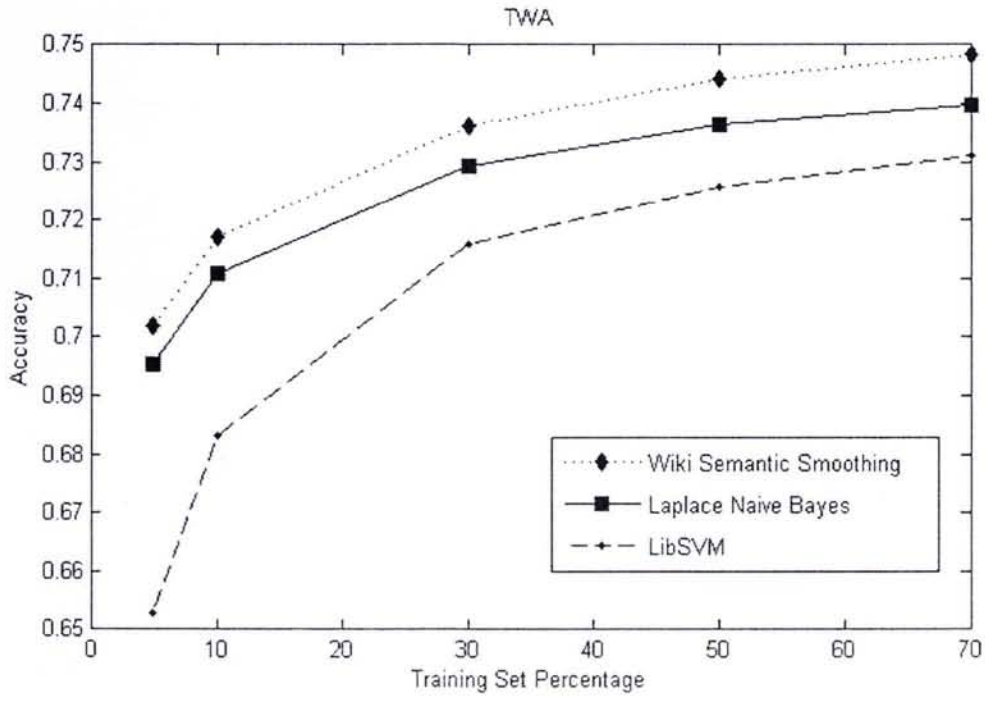

Figure 5.1 Accuracy of MNB, SVM and NBWSS on TWA Data Set

This conclusion is because adding semantic approach resulted in increasing the performance in terms of accuracy. Yet best result obtained by this approach is **0.7482** with NBWSS in 70% training size. MNB is resulted in 0.7397 accuracy with same training set rate, which is low approximately ~1% compared to our approach. Again, in this training size SVM obtained a similar result compared to MNB yet lower than NBWSS.

In Figure 5.2 and table 5.2, the accuracies of algorithms is shown on data set TWAC, where tweets are enriched with Wikipedia articles and related categories. Same results are seen on this data set. On the other hand, SVM reaches Naïve Bayes Laplace Smoothing in training size 30 and beats MNB in terms of accuracy. Yet still, NBWSS gives better accuracy of all compared to both algorithms. Algorithm increases accuracy by % 1.11.

Table 5.2 Accuracy of MNB, SVM and NBWSS on TWAC Data Set

TS	MNB	SVM	NBWSS
5	0.6822	0.6555	**0.6908**
10	0.6955	0.6808	**0.7055**
30	0.7133	0.7159	**0.7228**
50	0.7217	0.7261	**0.7322**
70	0.7262	0.7331	**0.7379**

Same conclusion results on TWAC data set which is adding semantic approach resulted in increasing the performance in terms of accuracy. Best performance result is obtained in TWAC data set is with NBWSS algorithm with 70% training size is **0.7379.** MNB was resulted in 0.7262 in terms of accuracy and SVM is 0.7331. With this data set, SVM gave better result compared to MNB yet could not resulted better performance when compared to our approach NBWSS. SVM performs very low in low training size namely 5%. Moreover, when the training size increases SVM gives better results compared to low training size results.

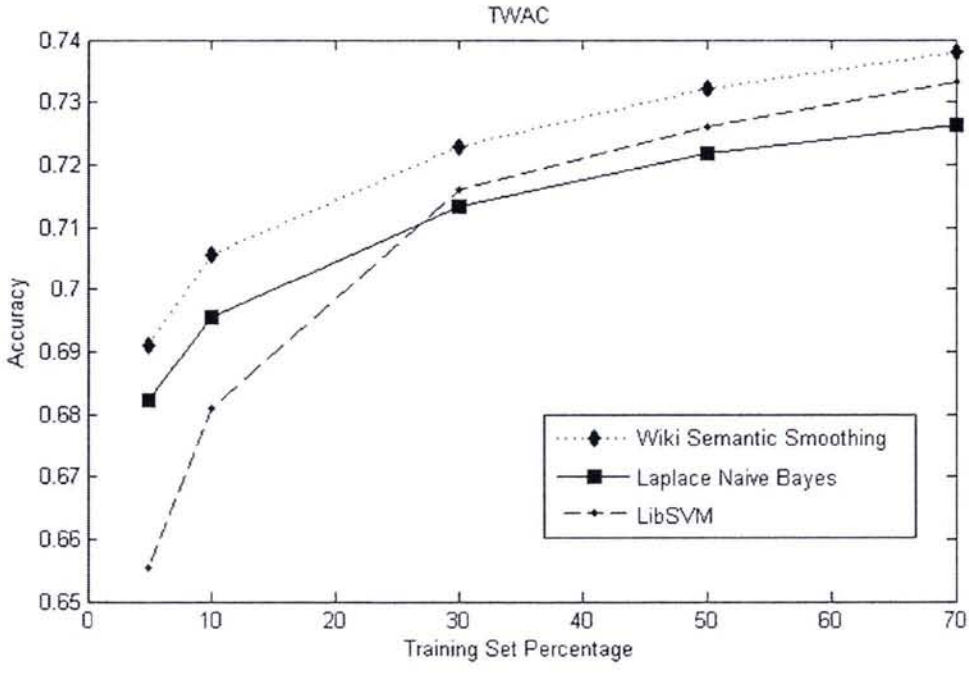

Figure 5.2 Accuracy of MNB, SVM and NBWSS on TWAC Data Set

In Figure 5.3 and table 5.3, the accuracies are shown on data set TWAR, where enrichment was done by redirects. On this test SVM and Naïve Bayes Laplace Smoothing gets same accuracy results on highest train size percent 70. Our approach Wiki semantic smoothing increases accuracy approximately % 1.16.

In figure 5.3 and table 5.3, we conclude with same conclusion same with results on TWA and TWAC datasets. Best performance result is obtained in TWAR data set is with NBWSS algorithm with 70% training size is **0.7479**. MNB was resulted in 0.7318 in terms of accuracy and SVM is 0.7316. With this data set, SVM and MNB give same results in terms of accuracy. Yet both algorithms could not result better performance when compared to our approach NBWSS. SVM performs very low in low training size namely 5%. Moreover, when the training size increases SVM gives better results compared to low training size results, which is also seen in TWA and TWAC datasets.

Table 5.3 Accuracy of MNB, SVM and NBWSS on TWAR Data Set

TS	MNB	SVM	NBWSS
5	0.6949	0.6610	**0.7047**
10	0.7043	0.6794	**0.7151**
30	0.7220	0.7154	**0.7330**
50	0.7289	0.7263	**0.7409**
70	0.7318	0.7316	**0.7479**

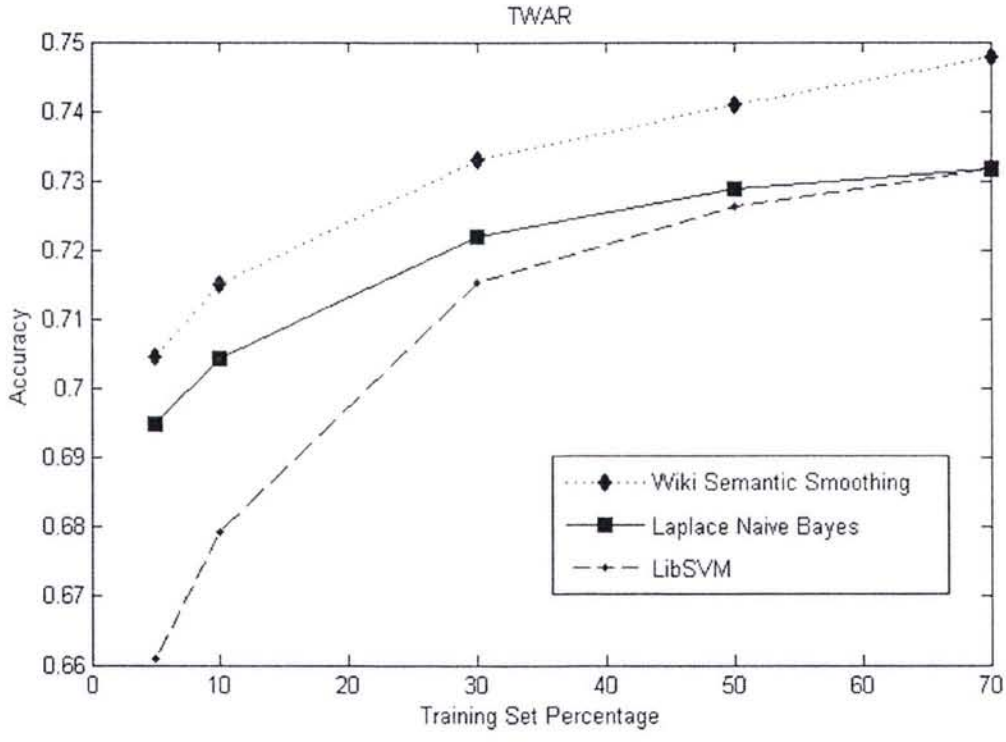

Figure 5.3 Accuracy of MNB, SVM and NBWSS on TWAR Data Set

In Figure 5.4 and table 5.4, the accuracies are shown on data set TWACR, where all wiki concepts were used for enrichment purpose. On this again SVM gives the lowest accuracy of all and Naïve Bayes Wiki semantic smoothing increases accuracy approximately % 1.14.

Table 5.4 Accuracy of MNB, SVM and NBWSS on TWACR Data Set

TS	MNB	SVM	NBWSS
5	0.6965	0.6531	**0.7035**
10	0.7111	0.6769	**0.7206**
30	0.7301	0.7142	**0.7406**
50	0.7375	0.7256	**0.7497**
70	0.7415	0.7317	**0.7555**

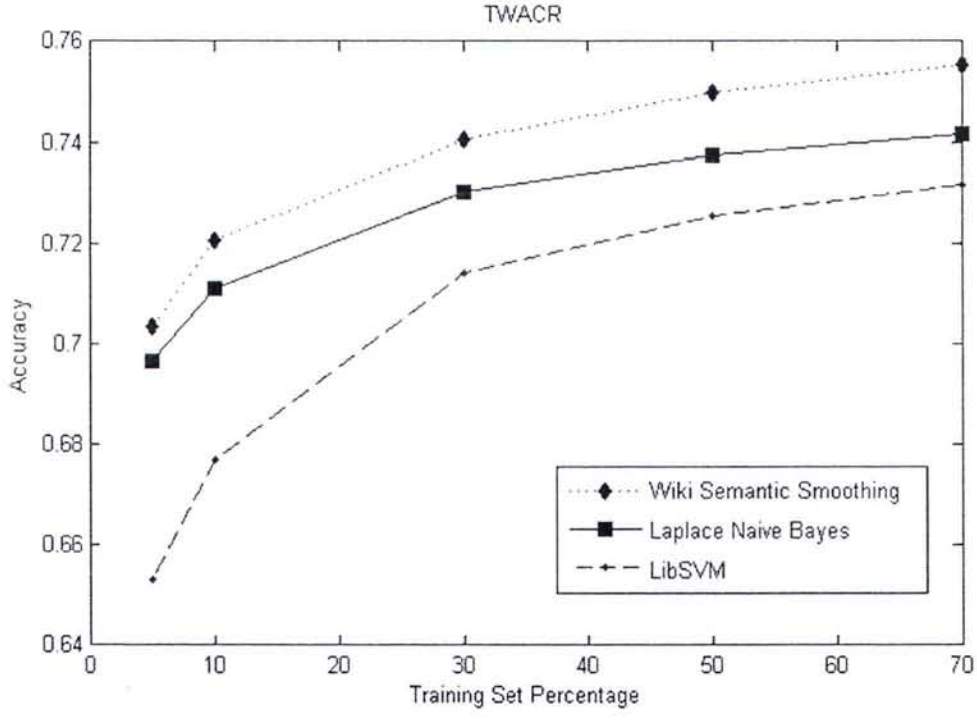

Figure 5.4 Accuracy of MNB, SVM and NBWSS on TWACR Data Set

Seen in figure 5.4 and table 5.4, we conclude with same conclusion same with results on TWA, TWAC, and TWAR datasets. Best performance result is obtained in TWACR data set is with NBWSS algorithm with 70% training size is **0.7555**. MNB was resulted in 0.7415 in terms of accuracy and SVM is 0.7317. With this data set, MNB give better

results compared to SVM in terms of accuracy almost ~1%. Yet both algorithms could not result better performance when compared to our approach NBWSS. SVM performs very low in low training size namely 5%. Moreover, when the training size increases SVM gives better results compared to low training size results, which is also seen in TWA TWAC and TWAR datasets.

Finally, we want to show the differences between TWA, TWAC, TWAR and TWACR datasets in NBWSS algorithm in terms of accuracy. In Figure 5.5 and table 5.5, accuracy change on wiki semantic smoothing on different types of data enrichments are shown. TWACR data gives the better accuracy as expected, semantic relations with categories and redirects when added. On the other hand, the lowest accuracy is observed on TWAC data set. As giving category knowledge as semantic purpose decreased the accuracy. Because, most categories being slightly unrelated to the given article. Such as for the article "Barack Obama" we enriched the tweets with adding the category "United Church of Christ members" decreased not only semantic meaning but also performance on accuracy.

Table 5.5 Accuracy of TWA, TWAC, TWAR, TWACR Data sets on NBWSS

TS	TWA	TWAC	TWAR	TWACR
5	0.7020	0.6908	**0.7047**	0.7035
10	0.7169	0.7055	0.7151	**0.7206**
30	0.7360	0.7228	0.7330	**0.7406**
50	0.7440	0.7322	0.7409	**0.7497**
70	0.7482	0.7379	0.7479	**0.7555**

Best result is obtained in 5% training size, is **0.7047** with TWAR dataset. Also TWACR is resulted in 0.7035 which is almost same with TWAR result. For other training sizes namely; 10, 30, 50 and 70 TWACR dataset resulted with highest accuracy result of all which are **0.7206**, **0.7406**, **0.7497**, and **0.7555**. To be more precise TWACR data set

which is Twitter enriched with Wikipedia article titles, categories and redirects give better accuracy results in almost all training sizes. This conclusion is also expected as adding more semantic we suggest that the performance will increase. However, the lowest accuracy resulted in TWAC data set. Which is also accurate, that adding unrelated all categories of given Wikipedia article title decreased the performance on algorithms.

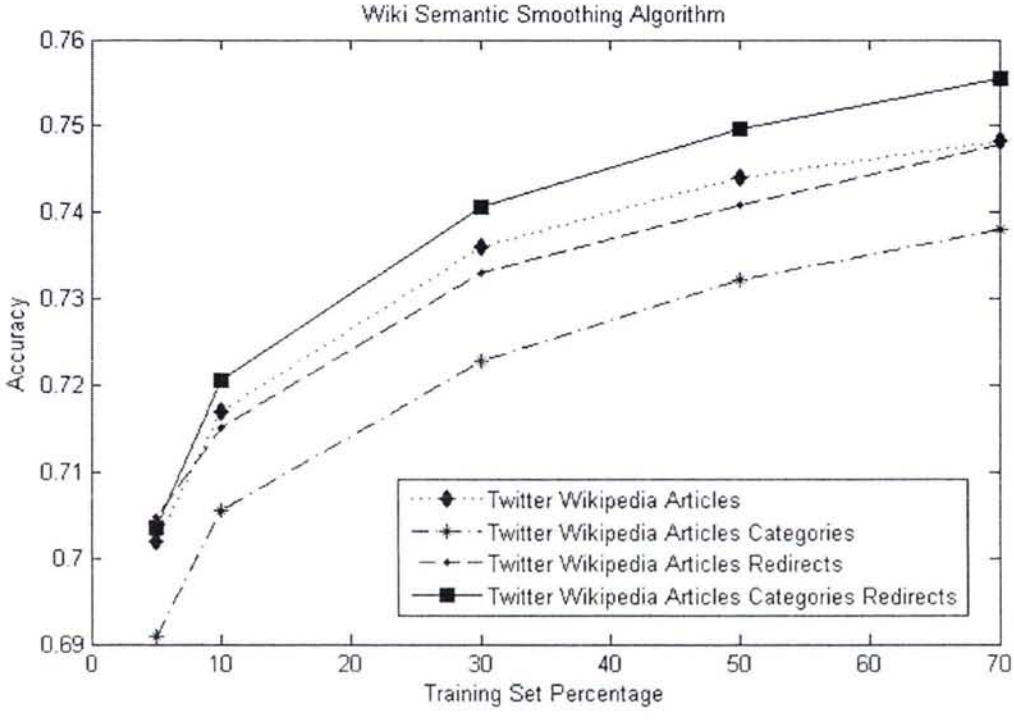

Figure 5.5 Accuracy of TWA, TWAC, TWAR, TWACR Data sets on NBWSS

6. CONCLUSION

Sentiment classification is one of the important and popular application areas of text classification in which texts are labeled as positive and negative. Moreover, Naïve Bayes (NB) is one of the mostly used algorithms in this area. NB having several advantages on lower complexity and simpler training procedure, it suffers from zero probability problems (Rish, 2001). Smoothing methods are employed for this problem; mostly Laplace Smoothing is used; however in this paper we propose Wikipedia based semantic smoothing approach. Our semantic smoothing formulation is based on the work in (Zhou, 2008).

We extend this study by employing Wikipedia to extract topic signatures. Moreover, we also incorporated semantic knowledge in Wikipedia such as categories and redirects. The main idea of the smoothing method is to identify explicit topic signatures as wiki concepts and add the semantic relation to the given word. These concepts can either be a Wikipedia article; Wikipedia articles related categories or Wikipedia articles related redirects. To be more precise, we use Wikipedia article titles that exist in documents, categories and redirects of these articles as topic signatures to enrich the dataset. We apply our approach to sentiment classification of tweets. Numerous of studies are implemented to address this problem by using semantic knowledge of Wikipedia. Most of these studies are conducted on large datasets as, Ohsumed, 20News groups and Reuters. Neither of these studies focused on small data sets such as Twitter. As Twitter is one of the most popular micro blogging cites, there are enormous number of data online. We downloaded 1,600,000 tweets from Twitter Sentiment 140 dataset and reduced its size to 64204 tweets, in which 34233 are labeled as negative tweets, 29971 are labeled positive tweets. We reduced its size by searching the most used categories which are referred in study (Go et.al, 2009).

We obtained used Wikipedia dump which was retrieved in August 6th, 2012 and in the PostgreSQL[5] there were 6,108,629 Wikipedia articles, 5,587,540 Wikipedia redirects and

[5] http://www.postgresql.org/

17,356,454 Wikipedia categories. Using WEX we obtain Wikipedia Articles, categories and redirects. We enriched the tweets in four different ways. At first we only added semantic relations between Wikipedia Articles to tweets which we obtained TWA dataset, next we enriched tweets by adding these articles categories to given tweets which we obtain TWAC dataset. At third the redirects are added with same approach meaning TWAR dataset obtained, and the forth one all the categories and redirects added in terms of their related Wikipedia article in this way TWACR dataset was obtained.

After enriching the tweets we used our proposed model Naïve Bayes Wiki Semantic Smoothing (NBWSS), Naïve Bayes Laplace Smoothing (MNB) and Support Vector Machines (SVM) classifiers to test given data. Our proposed model is very simple extension of multinomial Naïve Bayes (MNB). We conducted comprehensive experiments on testing collections that we enriched with Wikipedia to compare our semantic smoothing algorithm with other approaches. Results of the extensive experiments show that our approach improves the performance of NB and even can exceed the accuracy of SVM on Twitter Sentiment 140 dataset.

Best results in TWA, TWAC, TWAR and TWACR data sets were obtained by NBWSS algorithm. Which are; **0.7482**, **0.7379**, **0.7479** and **0.7555** respectively. We increased the performance of Naïve Bayes with using Wiki Semantic Smoothing approach in Wikipedia enriched datasets.

As expected best dataset collection was resulted in TWACR data set followed by TWAR, TWA and TWAC datasets. Obtaining best performance in TWACR is because adding all the semantic knowledge of Wikipedia which are Wikipedia article titles, categories and redirects increased the accuracy.

To be more precise TWACR data set which is Twitter enriched with Wikipedia article titles, categories and redirects give better accuracy results in almost all training sizes. This

conclusion is also expected as adding more semantic we suggest that the performance will increase. However, the lowest accuracy resulted in TWAC data set. Which means, that adding all categories of given Wikipedia article title decreased the performance on algorithms. We observe that, most categories are slightly unrelated to the given article. Such as for the article "Barack Obama" we enriched the tweets with adding the category "United Church of Christ members" decreased not only semantic meaning but also performance on accuracy.

REFERENCES

[1] A. Mccallum, K. Nigam: "A Comparison Of Event Models For Naive Bayes Text Classification", In: Aaai-98 Workshop On 'Learning For Text Categorization', 1998.

[2] Yang, Yiming, and Xin Liu. "A re-examination of text categorization methods." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

[3] Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." Machine learning: ECML-98 (1998): 137-142.

[4] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[5] Jelinek, Fred. "Self-organized language modeling for speech recognition." Readings in speech recognition (1990): 450-506.

[6] Gale, W. A. Good-Turing Smoothing Without Tears. Journal of Quantitative Linguistics, 2:217-237, 1995

[7] Chen, S. F., and Goodman, J. An Emprical Study of Smoothing Techniques for Language Modeling. Technical Report, Harvard University Center for Research in Computing Technology, 1998

[8] Vilar, D., Ney, H., Juan, A., and Vidal, E. Effect of Feature Smoothing Methods in Text Classification Tasks. Proceedings of the 4th International Workshop Pattern Recognition in Information Systems, pp. 108-117, 2004.

[9] Manning, C. D., Raghavan, P., and Schütze, H. An Introduction to Information Retrieval. Text classification and Naïve Bayes, chapter 13, pp. 263-270, Cambridge UP, Cambridge, England, 2009.

[10] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford (2009): 1-12.

[11] Zhou, Xiaohua, Xiaodan Zhang, and Xiaohua Hu. "Semantic smoothing for Bayesian text classification with small training data." SIAM2008) Proc. Intl. Conf. on Data Mining. 2008.

[12] Zhou, X., Hu, X., Zhang, X., Lin, X., & Song, I. Y. (2006, August). Context-sensitive semantic smoothing for the language modeling approach to genomic IR. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 170-177). ACM.

[13] Smadja, Frank. "Retrieving collocations from text: Xtract." Computational linguistics 19.1 (1993): 143-177.

[14] Zhou, Xiaohua, Xiaodan Zhang, and Xiaohua Hu. "Dragon Toolkit: Incorporating auto-learned semantic knowledge into large-scale text retrieval and mining." Tools

with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on. Vol. 2. IEEE, 2007.

[15] Baker, L. Douglas, and Andrew Kachites McCallum. "Distributional clustering of words for text classification." Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998.

[16] Zhu, Xiaojin. "Semi-supervised learning literature survey." (2005).

[17] Genc, Yegin, Yasuaki Sakamoto, and Jeffrey Nickerson. "Discovering context: Classifying tweets through a semantic transform based on Wikipedia." Foundations of Augmented Cognition. Directing the Future of Adaptive Systems (2011): 484-492.

[18] Wang, P., Hu, J., Zeng, H. J., & Chen, Z. (2009). Using Wikipedia knowledge to improve text classification. Knowledge and Information Systems, 19(3), 265-281.

[19] Hu, X., Zhang, X., Lu, C., Park, E. K., & Zhou, X. (2009, June). Exploiting Wikipedia as external knowledge for document clustering. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 389-396). ACM.

[20] Huang, A., Milne, D., Frank, E., & Witten, I. H. (2009). Clustering documents using a Wikipedia-based concept representation. In Advances in Knowledge Discovery and Data Mining (pp. 628-636). Springer Berlin Heidelberg.

[21] Luo, Qiming, Enhong Chen, and Hui Xiong. "A semantic term weighting scheme for text categorization." *Expert Systems with Applications* 38.10 (2011): 12708-12716.

[22] Li, Cheng Hua, Ju Cheng Yang, and Soon Cheol Park. "Text categorization algorithms using semantic approaches, corpus-based thesaurus and WordNet." *Expert Systems with Applications* 39.1 (2012): 765-772.

[23] Poyraz, M., Ganiz, M. C., Akyokus, S., Gorener, B., & Kilimci, Z. H. (2012, July). Exploiting Turkish Wikipedia as a semantic resource for text classification. In Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on (pp. 1-5). IEEE.

[24] "(2013) The Wikipedia website. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia"

[25] Google, Freebase Wikipedia Extraction (WEX), http://download.freebase.com/wex/, <08> <06>, <2012>

[26] http://www.postgresql.org/

[27] Torunoglu, D., et al. "Analysis of preprocessing methods on classification of Turkish texts." Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on. IEEE, 2011.

[28] Baker, L. Douglas, and Andrew Kachites McCallum. "Distributional clustering of words for text classification." Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998.

[29] Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011, June). Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Vol. 1, pp. 151-160).

[30] Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of LREC (Vol. 2010).

[31] Barbosa, L., & Feng, J. (2010, August). Robust sentiment detection on twitter from biased and noisy data. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 36-44). Association for Computational Linguistics.

[32] Davidov, D., Tsur, O., & Rappoport, A. (2010, August). Enhanced sentiment learning using twitter hashtags and smileys. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 241-249). Association for Computational Linguistics.

[33] Diakopoulos, N. A., & Shamma, D. A. (2010, April). Characterizing debate performance via aggregated twitter sentiment. In Proceedings of the 28th international conference on Human factors in computing systems (pp. 1195-1198). ACM.

[34] Bifet, A., & Frank, E. (2010, January). Sentiment knowledge discovery in twitter streaming data. In Discovery Science (pp. 1-15). Springer Berlin Heidelberg.

[35] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 1-38.

[36] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

[37] Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

[38] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

BIOGRAPHY

Place and Date of Birth		Istanbul - 29.09.1986
High School	2000-2004	Halide Edip Adıvar Anatolian High School
B.S Degree	2004-2009	Doğuş University Computer Engineering Department
M.S Degree	2009-	Doğuş University Computer Engineering Department

Work Experiences:

2009-	Doğuş University Research Assistant

Publications:

Torunoglu, D., Cakirman, E., Ganiz, M. C., Akyokus, S., & Gurbuz, M. Z. (2011, June). Analysis of preprocessing methods on classification of Turkish texts. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on* (pp. 112-117). IEEE.

Torunoglu, D., Telseren, G., Sagturk, O., & Ganiz, M. C. (2013, June). Wikipedia Based Semantic Smoothing for Twitter Sentiment Classification. In *Innovations in Intelligent Systems and Applications (INISTA), 2013 International Symposium on*. IEEE.

Celikkaya, G., Torunoglu, D., & Eryigit, G. (2013, October). Named Entity Recognition on Real Data: A Preliminary Investigation for Turkish. In AICT 2013, The International Conference on Application of Information and Communication Technologies.