TWITTER SENTIMENT ANALYSIS

A Major Project Report
done in partial fulfilment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Anthony K Jose B070054CS
Nipun Bhatia B070272CS
Sarath Krishna S B070062CS

Department of Computer Science & Engineering
National Institute of Technology Calicut
Kerala - 673601
Monsoon 2010

Abstract

Twitter is a micro-blogging site enabling users to send updates (tweets) in the form of messages to a group of friends (followers).Integration with various web applications has made Twitter an obvious choice for a research bed in data and opinion mining, where based on the opinion reflected a tweet can be classified as positive/negative/neutral. In this paper we introduce a cleaner training data for subjective/objective classification to filter out as many neutral tweets as possible in order to obtain better query-based results. We propose different heurestics to select the neutral training data, and study their effect on the precision,recall and accuracy values of different classifers. For future work we propose the use of data clustering for making this classification domain-specific and thus to improve the accuracy.

Contents

Chapter 1

Introduction

1.1 Problem Definition

Our aim is to design a Twitter Sentiment Engine which will classify a tweet into one of the following three classes : positive/negative/neutral. We aim to improvise on the accuracy of the currently existing engines/analysis algorithms by using a cleaner training data, obtained using an intensive approach for subjective/objective classification to filter out as many neutral tweets as possible.

1.2 Background and Recent Research

Among the 2 main types of textual information - facts and opinions, a major portion of current information processing methods such as web search and text mining work with the former. Opinion Mining refers to the broad area of natural language processing, computational linguis- tics and text mining involving the computational study of opinions, sentiments and emotions expressed in text. A thought, view, or attitude based on emotion instead of reason is often collo- quially referred to as a sentiment. Hence, lending an alternate term for Opinion Mining, namely, Sentiment Analysis. This field finds critical use in areas where organizations or individuals wish to know the general sentiment associated to a particular entity - be it a product, person, public policy, movie or even an institution. Opinion mining has many application domains including science and technology, entertainment, education, politics, marketing, accounting, law, research and development. In earlier days, with limited access to user generated opinions, research in this eld was minimal. But with the tremendous growth of the world wide web, huge volumes of opinionated texts in the form of blogs, reviews, discussion groups and forums are available for analysis making the world wide web the fastest, most comprehensive and easily accessible medium for sentiment analysis. However, finding opinion sources and monitoring them over the Web can be a formidable task because a large number of diverse sources exist on the Web and each source also contains a huge volume of information. From a humans perspective, it is both difficult and tiresome to find relevant sources, extract pertinent sentences, read them, summarize them and organize them into usable form. An automated and faster opinion mining and summarizing system is thus needed. Much before ARPAnet expanded into the World Wide Web, we often asked our friends to recommend a good florist/baker, requested reference letters regarding job applicants from colleagues or to explain who they were planning to vote for in the next local elections or consulted consumer reports to decide what refrigerator to buy. But the Internet and theWeb have now (among other things) made it possible to nd out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics that is, people one has never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. Infact, surveys conducted in the recent times indicate that a large percentage of internet users (bordering 81research for product purchase on the internet atleast once. Out of this number, close to a quarter of them perform this research daily. Among people that look up restaurant,hotel and spa reviews, close to 87% credit internet reviews to influencing their purchase decision. Several consumer surveys indicate that customers are willing to shell out almost 90of the original price of current 5-star rated items. Apart from this apparently heavy reliance on reviews, a rather high percentage of internet users have posted comments and reviews on products, making the web an ideal place for consumer feedback. Sentiment analysis has created increased interest due to its promising and potential applications.

Chapter 2

Motivation

Twitter is a micro-blogging site where users have the ability to send updates (tweets) in the form 140 character long messages to a group of friends (followers).Tweets by default are public, which permits people to follow others and read each others tweets without giving mutual permis- sion.However, senders can restrict delivery to friends only. With the steady growth of Twitter over 4 years, Twitter today is still growing at a staggering 1382% as of May 2009 (OmniMedia Inc.).With this massive expansion in user base and the Twitter application program interface (API) going public, Twitter is now integrated with several Web applications and can be sent via instant messaging, Short Message Service (SMS), Really Simple Syndication (RSS), email, or other social networking platforms. This has made Twitter an obvious choice for a research bed in data and opinion mining for todays computer scientists. Given its robustness, Twitter is increasingly used by news organizations to receive updates during emergencies and natural disasters. A number of businesses and organizations are using Twitter or similar microblogging services to disseminate information to stakeholders [3]. Work conducted by Jansen et al. [3] has shown that 19% of all tweets take mention of a brand of which 20% showed sentiment to the brand justifying it as the electronic Word Of Mouth [3] in the consumer market.

Chapter 3

Related Work (Literature Survey)

3.1 Classifiers for Sentiment Analysis

Among the various machine learning algorithms that have been used for sentiment analysis Naive Bayes, SVM and MaxEnt have shown promising results in movie-review classification and subsequently in recent Twitter sentiment analysis research. Here we look into detail into these three approaches and compare and contrast their efficiencies in the Twitter domain as found by previous work.

Feature Extraction & Performance

The collected dataset/training set is used to extract features that will be used to train our classifier. Selection of an optimal feature extraction technique is essential for accurate sentiment analysis. Initial work conducted by Pang et al.[2] on classification of movie reviews indicate unigrams outperforming bigram feature extraction while the contrary result was found to be true in product review sentiment classification . In the world of microblogs, with prime focus set on Twitter, work done by Pak et al. [5] confirm that a bigram model outperforms both unigram and trigram models while using a Multinomial Naive Bayes classifier. However, the reverse was true in the case of SVM and MaxEnt classifier studies conducted by Go et al.[4] . Introduction of a combination of unigram and bigram in feature extraction promised better results in MaxEnt as well as NB classifiers [4].

Unigrams

The easiest and most used approach. Pang et al. [2] reported an accuracy of 81.0%, 80.4%, and 82.9% for Naive Bayes, MaxEnt and SVM respectively in the movie-review domain. This was found to be closely similar to accuracies obtained in twitter classification which were 81.3%, 80.5%, and 82.2% respectively [4].

Bigrams

Bigrams were found to be a very sparse feature in the Twitter corpus as experienced by Go et al. and caused an overall drop in accuracy for MaxEnt and NB. This is evident from the following tweet :

@stellargirl I looooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right. MaxEnt gave equal probabilities to the positive and negative class for this case because there is not a bigram that tips the polarity in either direction.A better option is to combine bigrams and unigrams.

Unigram+Bigram

Both unigrams and bigrams are used as features.In the movie-review domain, there was a decline for Naive Bayes and SVM, but an improvement for MaxEnt [2]. However, recent research in the twitter research bed found that as compared to unigram features, accuracy improved for Naive Bayes (81.3% from to 82.7%), MaxEnt (from 80.5 to 82.7%) and there was a decline for SVM (from 82.2% to 81.6%).

POS tagging

POS tags would be a useful feature since usage for several words varied. For example, over as a verb has a negative connotation whereas over as the noun,would refer to the cricket over which by itself doesnt carry any negative or positive connotation. However past experiments with POS tagging in feature extraction for sentiment analysis have yield little improvements. The accuracy improves slightly for Naive Bayes but declines for SVMs, and the performance of MaxEnt is unchanged while classifying movie-reviews with their individual accuracies being 81.5%,81.9% 80.4% respectively. Limiting to adjectives, following up on previous successful efforts in sentiment detection,failed to produce any results while infact reducing overall accu- racies in all 3 methods despite the intuitive idea of expecting adjectives to carry a great deal of information regarding overall sentiment in a document [2].As is obvious from figure, the 2633 most frequent unigrams are a more optimal choice. These findings were found to be consistent in the area of twitter sentiment analysis as well [4].

Naive Bayesian Classifier

The Naive Bayesian classifier is a straightforward and frequently used method for supervised learning. It provides a flexible way for dealing with any number of attributes or classes, and is based on probability theory.

Maximum Entropy Classifier Maximum entropy classifiers are commonly used as alternatives to Naive Bayesian classifier because they do not require statistical independence of the features that serve as predictors.

Support Vector Machines

An SVM is a kind of large-margin classifier: it is a vector space based machine learning method where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data.

Chapter 4

Work Done & Design

4.1 Twitter Domain (Requirements)

Twitter messages have certain unique attributes that make it complex to analyze and classify while training the data sets. Some of them are as follows:

1. Length: The maximum length of a Twitter message is 140 characters, thus the limited length of tweet, might be comprising of one or two sentences. Thus our task is the simple breakdown of the tweet to extract the polarity from it.

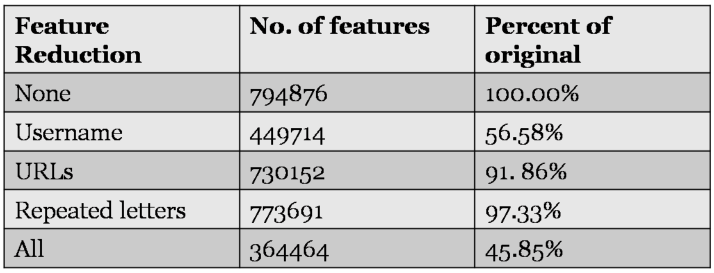
2. Available data: the twitter API supports searching tweets pertaining to a query thus we can obtain a large training set using the twitter API.

3. Language model. The language under consideration is English. Basic problem faced is the limited length which encourages the user to use slang and acronyms. Moreover users post tweet from different devices including mobile phones, thus increasing the rate of misspelling words.

4. Negative sentences: many people would write their tweets with negation before the adjec- tive or verb, which complicates the training set. For example : a sentence such as Not satisfied with the situation of the Senate in the U.S. has the adjective satisfied which assigns a polarity positive without considering the negation in the sentence.

5. Confusing polarity: for certain tweets there will be a confusion or disagreement for the polarity to be assigned. For instance, Sania defeats Martina is positive when taken from Sanias perspective while its negative when Martina is the search query.

6. Dealing with emoticons: Our training data should contain clean labels. The emoticons serve as a noisy label. There are some cases in which the emoticon label would normally not make sense to a human evaluator. For example user ayakyl tweeted, hurrah :) wont have class today!!!! If we remove the emoticon from this phrase, it becomes hurrah wont have class today!!!! in which a human evaluator would normally assess as negative.



Feature Reduction	No. of features	Percent of original
None	794876	100.00%
Username	449714	56.58%
URLs	730152	91. 86%
Repeated letters	773691	97.33%
All	364464	45.85%

Figure 4.1: Feature Reduction & Preprocessing [3]

7. Casual language: Tweets contain very casual language. For example, a user may want to right the word happy as: happpppyyy happpiieee happy hap-e Besides showing that people are happy, this emphasizes the casual nature of Twitter and the disregard for correct spelling.

8. Usage of links: Users very often include links in their tweets. Thus there is a need to classify this type of tweet by using keywords such as URL. But even then its difficult to extract the sentiment sometimes. for example :
Cannot believe Ratan Tata doing this :http://bit.ly/waer3ER
while the URL speaks about volumes of help donated by Ratan Tata to the Mumbai victims, the tweet makes it sound negative.

9. Usernames: Users often include usernames in their tweets, in order to address messages to particular users. A de facto standard is to include the @ symbol before the username (e.g. @natalie).

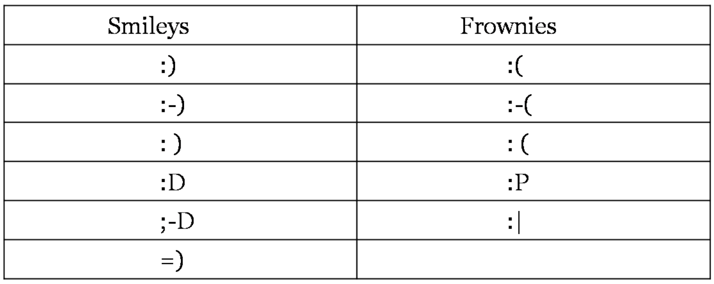
10. Use of special symbols: such as # to denote main subject of tweet and RT to denote that the tweet was forwarded from a previous tweet of a different user. eg. RT the best culfest site i have come across in my collegiate life. springfest by iitkgp Amazing work and really impressive looks. goo.gl/jpvW

4.2 Feature Reduction & Preprocessing

Language For our work, we only consider english language tweets and have specified to retrieve only 'en' tweets in our Tweet scrapper code.

URL In order to reduce the feature size during training, all URLs in the training tweets were replaced with an equivalence class <url> [3] This is shown by Go et. al to considerably reduce feature size.

Retweets Retweeting is the process of copying another users tweet and posting to another account.



Smileys	Frownies
:)	:(
:-)	:-(
:)	: (
:D	:P
;-D	:|
=)	

Figure 4.2: Smileys and Frownies removed

This usually happens if a user likes another users tweet. Retweets are commonly abbreviated with RT. For example, consider the following tweet: Awesome! RT @rupertgrintnet Harry Potter Marks Place in Film History http://bit.ly/Eusxi :). In this case, the user is rebroadcasting rupertgrintnets tweet and adding the comment Awesome!. Any tweet with RT is removed from the training data to avoid giving a particular tweet extra weight in the training data [3].

Emoticons As outlined in [3] , all emoticons are stripped from the training set, as this puts extra weight on emoticons during classification.

Usernames & Hashtags Usernames in twitter are given in the '@username' format. And similarly, people tag tweets pertaining to a category in twitter, using ″ as in NITC. We replace usernames with <puc> and tags with <hashtag>. This replacement of usernames and hashtags helps reduce the feature size by a large margin [3].

Removal of repeated tweets Repeated tweets are removed. Occasionally, the Twitter API returns duplicate tweets. Similar to Retweets, repeated tweets would add extra weight to a particular tweet during training.

Removal of redundant/repeated letters Due to the casual nature of Twitter language, several words (in many cases opinion words) are misspelt or often over emphasized due to which the classifer may not attribute polarity of this word (eg. looooooove) to the actual word (eg.love) during training. We considered using stemming algorithms for this purpose, but this did not help, in fact reduced accuracies in case of MaxEnt and Naive Bayes. As a work-around, we used the approach followed by [3]. In words containing more than 3 occurences of the same letter together, these occurences were replaced with 2 instances of the letter. eg. haaaaaaaappy would be replaced by haappy , gooooooooood would be replaced by good.

Lowercasing All the characters were lowercased to ensure that all tokens map to the corresponding feature irrespective of casing.

Special Characters we use the equivalent class <puc> for special characters occuring in a tweet such as ′?′,′!′ etc. Punctuations are separated from the words and treated as separate tokens for unigrams for feature extraction.

4.3 Twitter API

Our basic need for an API is for two purposes: To obtain training data which we would use in order to test results of our classifier and to obtain tweets pertaining to a particular query(once the twitter sentiment engine is running and working). Twitter exposes its data via an Application Programming Interface, (API). Twitter itself offers the functionality where researchers can search for tweets pertaining to a query. However we chose to collect our data for training sets from certain twitter sentiment analyzers already existing for this purpose, as they have already classified the test data based on the polarity of the tweets and thus we get an advantage to verify our results with those of the existing analyzers. Twitter Search provides a REST API so that we can search for tweets in an automated fashion. REST (Representational State Transfer), which enables developers to access information and resources using a simple HTTP invocation. Using REST we can obtain domain-specific data simply by pointing a URL to a specific location. So, the Twitter Search API is a REST service that enables users to point to a specific URL and retrieve a variety of tweets that meet the criteria specified in the URL. This enables us to accept input within a Web application and dynamically query Twitter based on that input, using a simple URL that encodes the input into a format that the API understands. This results of the search can be retrieved in Atom format, RSS (.rss) and JavaScript Object Notation (JSON.json).

In our work we wrote a code for the Search API which when given a query retrieves the results in JSON format and stores them in an Object form. We used this API to search for tweets containing URL, random stream of tweets for parsing through subjectivity lexicons(MPQA and AFINN), and for obtaining the test tweets.

4.4 Subjectivity/Objectivity Classifier

Mihalcea et al.[5] summarize the evidence of several projects on subsentential analysis as follows: the problem of distinguishing subjective versus objective instances has often proved to be more difficult than subsequent polarity classification, so improvements in subjectivity classification promise to positively impact sentiment classification [5]. In the Twitter senti- ment domain, work conducted by Go et al. involves using an NB classifier with 2 classes : objective/subjective to classify tweets as neutral or polar. However, due to lack of sufficiently clean training data, accuracies in classification were low. As a solution to this, we used 4 different approaches in order to obtain a cleaner training data for our neutral class:

1. MPQA

   The MPQA(multi-purpose question answering system) opinion corpus [8]contains news articles from a wide variety of news sources manually annotated for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, etc.). The wordlist we use is a list 'opinion words' with indicated polarity. Our method of creating a neutral training set involved passing random tweets through the MPQA word list and filtering out tweets that contain these opinion words. We used the Twitter stream API to generate random tweets and passed it through the MPQA word list filter to give our neutral training set.

   With dipping accuracies, we noticed that the MPQA while containing several relevant opinion words, it did not include commonly used internet opinion lingo. So, we made use of the AFINN word list which is a manually labelled word list made by Finn rup Nielsen. This word list contained more relevant twitter opinion words and was shown to be successful in its use in sentiment analysis for microblogs (A new ANEW: Evaluation of a word list for sentiment analysis in microblogs,[7]) We retrieved 3,50,000 tweets out of which we could filter 14,000 tweets after passing it through MPQA lexicon , and 35,000 after passing through AFINN.

2. Wikipedia

   We considered using wikipedia as a source of neutral training tweets. This was based on the notion that a tweet may be considered neutral if it could appear as a news headline or a wikipedia sentence [3]. Wikipedia has a statement of neutrality which is stated as follows :

   "Neutral point of view (NPOV) is a fundamental principle of Wikipedia and of other Wikimedia projects. All Wikipedia articles and other encyclopedic content must be written from a neutral point of view. This means representing fairly, proportionately, and as far as possible without bias, all significant views that have been published by reliable sources. This is non-negotiable and expected of all articles and all editors."

   For our requirement, we needed a collection of random wikipedia sentences for use as our neutral training set. We wrote java code to retrieve random pages from wikipedia by using wikipedia 'Random Page'(http://en.wikipedia.org/wiki/Special:Random) feauture and subsequently selected a line from the page as one of the training tweets.We collected a total of 60,000 random wikipedia sentences.

3. URL

   We noted that most often tweets containing URLs were objective tweets and considered using such tweets in our neutral training set. Again, using the search API, we retrieved tweets containing the query term 'http'. We collected 79,000 tweets containing http links, which were used for training and testing purposes.

4. Hybrid Approach

   A possible improvement to subjectivity classification accuracies could by achieved by using a combination of the above mentioned neutral training sets.

4.5 Main Classifier

Once neutral tweets are filtered out of the stream of selected tweets, the remaining sentiment-laden tweets are passed through the main (positive/negative) classifier for classification into positive and negative tweets. Based on our literature surveys, we model a Multinomial Naive Bayes Classifier with Mutual Information feature selection and add-1 smoothing coupled with a

Unigram+Bigram approach for feature extraction.

4.5.1 Naive Bayesian Classifier

The Naive Bayesian classifier is a straightforward and frequently used method for supervised learning. It provides a flexible way for dealing with any number of attributes or classes, and is based on probability theory. It is the asymptotically fastest learning algorithm that examines all its training input. It has been demonstrated to perform surprisingly well in a very wide variety of problems in spite of the simplistic nature of the model. Furthermore, small amounts of bad data, or noise, do not perturb the results by much. The Naive Bayesian classification system is based on Bayes rule and works as follows.

P(c|t) = P(c)*P(t|c)/P(t)

It assumes each feature is conditional independent to other features given the class. That is, where c is a specific class and t is text we want to classify. P(c) and P(t) is the prior probabilities of this class and this text. And P(t|c) is the probability the text appears given this class. In our case, the value of class c might be POSITIVE or NEGATIVE, and t is just a sentence. The goal is choosing value of c to maximize P(c|t): Where P(wi|c) is the probability of the ith feature in text t appears given class c. We need to train parameters P(c) and P(wi|c). It is simple for getting these parameters in Naive Bayes model. They are just maximum likelihood estimation (MLE) of each one. When making prediction to a new sentence t, we calculate the log likelihood log P(c) + log P(wi|c) of different classes, and take the class with highest log likelihood as prediction. In practice, it needs smoothing to avoid zero probabilities. Otherwise, the likelihood will be 0 if there is an unseen word when it making prediction. Using add-1 smoothing is a solution to that problem.

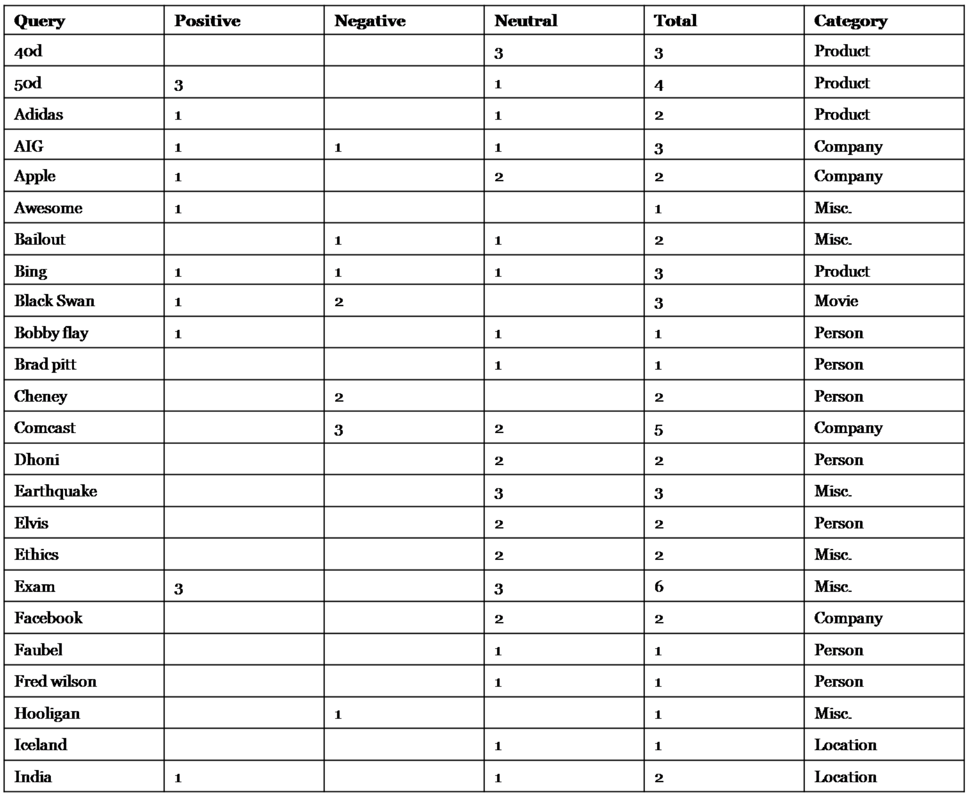
4.6 Experimental Setup & Results

For our experimental setup, we made use of the following open source java libraries - OpenNLP, MALLET and Weka. For our Naive Bayes classifier, we made use of the instances, normalisation and StringtoWordVector classes in Weka to create a word vector from the tweets. We then wrote our own version of the Multinomial Naive Bayes classifier and used the StringtoWordVector utility to toggle tokenisation and feature selection in the form of unigrams and bigrams. However, accuracies and F-measures were quite low with this implementation, so we moved to MALLET for our Naive Bayes classifier. We use the Csv2Classify class to load our data and NaiveBayes class to classify the instances. Code to create unigrams, bigrams and combinations of both were written by us using the StreamTokenizer class in Java.

Maxent was first implemented by us in OpenNLP, and while our tests with positive and negative tweets were matching with previous work [3], subjective classification showed rather poor results. We wished to try a different implementation, and so we moved to MALLET's MaxEnt implementation to plot curves and data for our training sets.

4.6.1 Test Set

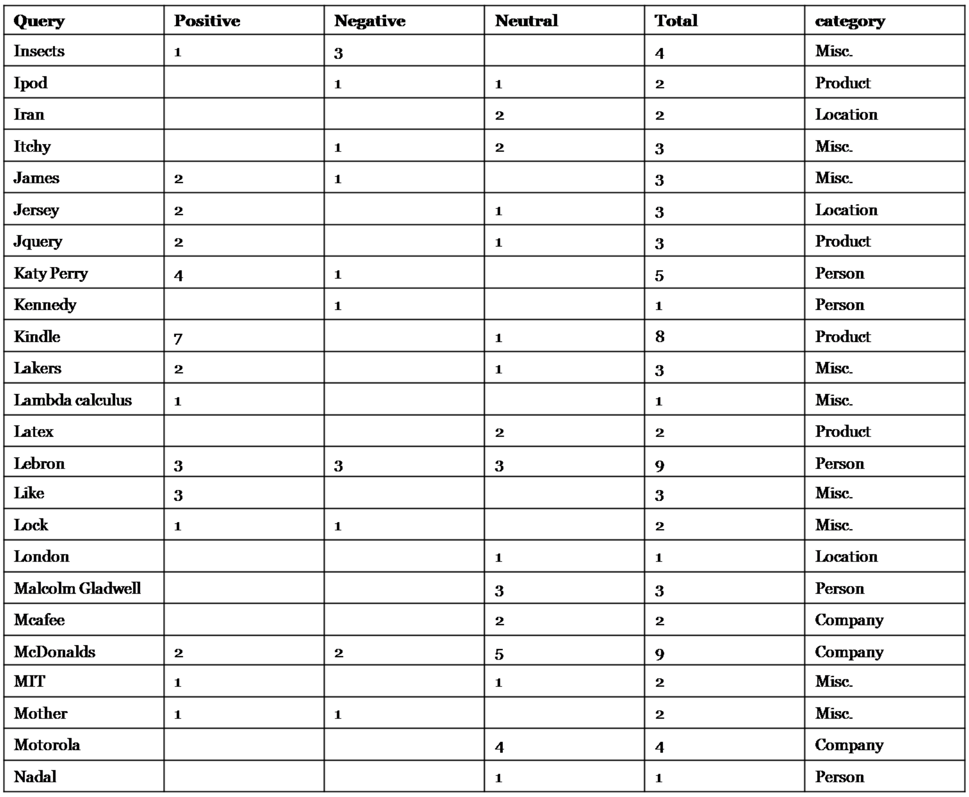
For our test set, we modelled our test set as follows : The Twitter API was searched with specific queries. These queries were arbitrarily chosen from different domains. For example, these queries consist of consumer prod- ucts (40d, 50d, kindle2), companies (aig, att), and people (Bobby Flay, Warren Buffet). The query terms used are listed in Table.



Query	Positive	Negative	Neutral	Total	Category
40d			3	3	Product
50d	3		1	4	Product
Adidas	1		1	2	Product
AIG	1	1	1	3	Company
Apple	1		2	2	Company
Awesome	1			1	Misc.
Bailout		1	1	2	Misc.
Bing	1	1	1	3	Product
Black Swan	1	2		3	Movie
Bobby flay	1		1	1	Person
Brad pitt			1	1	Person
Cheney		2		2	Person
Comcast		3	2	5	Company
Dhoni			2	2	Person
Earthquake			3	3	Misc.
Elvis			2	2	Person
Ethics			2	2	Misc.
Exam	3		3	6	Misc.
Facebook			2	2	Company
Faubel			1	1	Person
Fred wilson			1	1	Person
Hooligan		1		1	Misc.
Iceland			1	1	Location
India	1		1	2	Location

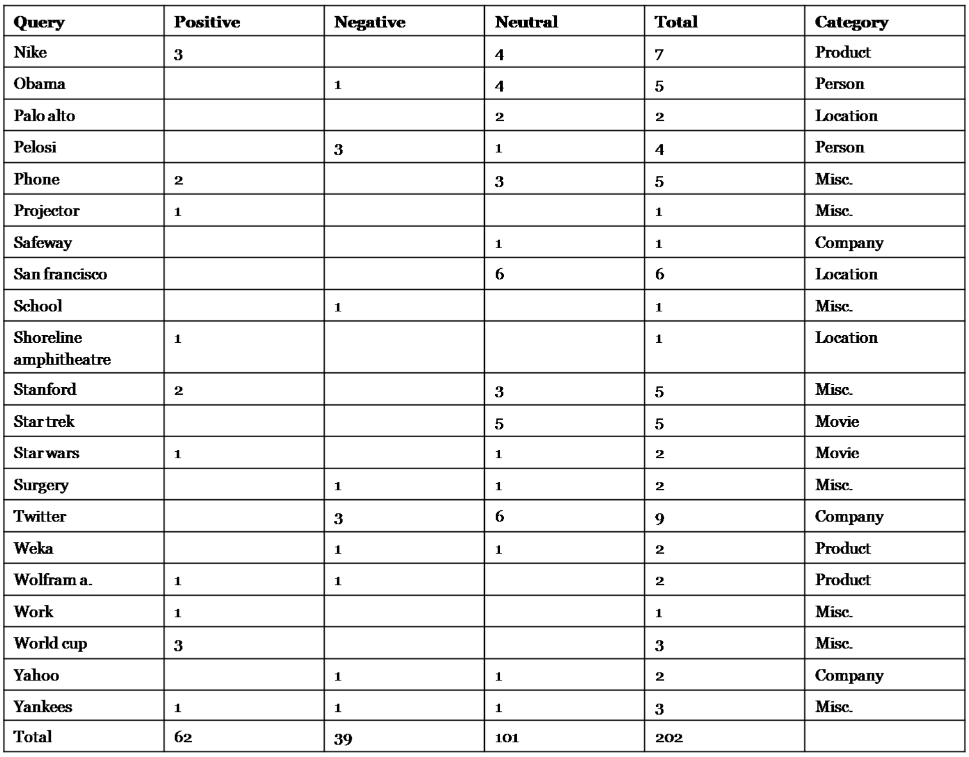
Figure 4.3:

The different categories of these queries are listed in figure 4.4.



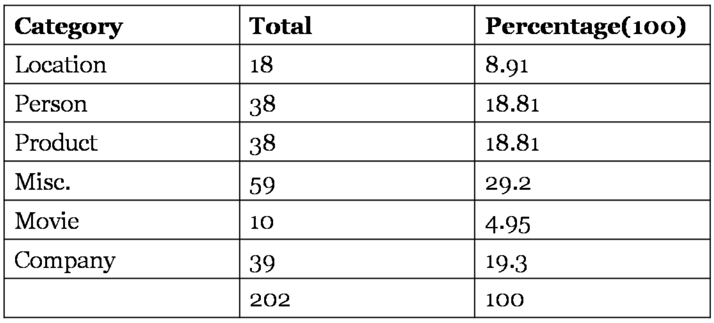
Query	Positive	Negative	Neutral	Total	category
Insects	1	3		4	Misc.
Ipod		1	1	2	Product
Iran			2	2	Location
Itchy		1	2	3	Misc.
James	2	1		3	Misc.
Jersey	2		1	3	Location
Jquery	2		1	3	Product
Katy Perry	4	1		5	Person
Kennedy		1		1	Person
Kindle	7		1	8	Product
Lakers	2		1	3	Misc.
Lambda calculus	1			1	Misc.
Latex			2	2	Product
Lebron	3	3	3	9	Person
Like	3			3	Misc.
Lock	1	1		2	Misc.
London			1	1	Location
Malcolm Gladwell			3	3	Person
Mcafee			2	2	Company
McDonalds	2	2	5	9	Company
MIT	1		1	2	Misc.
Mother	1	1		2	Misc.
Motorola			4	4	Company
Nadal			1	1	Person

Figure 4.4:



Query	Positive	Negative	Neutral	Total	Category
Nike	3		4	7	Product
Obama		1	4	5	Person
Palo alto			2	2	Location
Pelosi		3	1	4	Person
Phone	2		3	5	Misc.
Projector	1			1	Misc.
Safeway			1	1	Company
San francisco			6	6	Location
School		1		1	Misc.
Shoreline amphitheatre	1			1	Location
Stanford	2		3	5	Misc.
Star trek			5	5	Movie
Star wars	1		1	2	Movie
Surgery		1	1	2	Misc.
Twitter		3	6	9	Company
Weka		1	1	2	Product
Wolfram a.	1	1		2	Product
Work	1			1	Misc.
World cup	3			3	Misc.
Yahoo		1	1	2	Company
Yankees	1	1	1	3	Misc.
Total	62	39	101	202	

Figure 4.5: List of Queries used for Test Set



Category	Total	Percentage(100)
Location	18	8.91
Person	38	18.81
Product	38	18.81
Misc.	59	29.2
Movie	10	4.95
Company	39	19.3
	202	100

Figure 4.6: Distribution of Test Set over categories

The result set for a query is then looked at. If a result is seen that contains a sentiment, it is marked positive or negative. Thus, this test set is selected independently of the presence of emoticons.

We make use of the MaxEnt and Naive Bayes classifier for analysis. Previous works on subjectivity classification such as [10] [9] make use of MaxEnt,SVM Naive Bayes.

We make use of java based MALLET (Machine Learning for Language Toolkit) from University of Massachusetts,Amherst. MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text.

For our polar training set, we constructed an equal distribution of positive and negative tweets as defined by smileys and frowneys as outlined by (Go et. al) [3]. For each analysis, an equal number of neutral and non-neutral tweets were used in training.

Naive Bayes with MPQA training showed peak accuracies of 65.5%, 61% and 56% respectively. Accuracies for unigram although initially dipped , began to increase beyond a certain training size.

In case of MaxEnt, accuracies were considerably lower for the unigram case , while slight decrease for bigram unigram + bigram case.

The AFINN word list filtering proved to show better accuracies with unigrams for Naive Bayes peaking at 76.5% accuracy while MaxEnt accuracies dipped drastically to 49.5% for unigrams. NB accuracies were also higher for the bigram and unigram+bigram case as well.

We considered filtering our training data through both the selected lexicons (MPQA AFINN) as we were interested in seeing the results, however, we were only successful in retrieving 5000 tweets on passing 3,50,000 tweets through this filter.

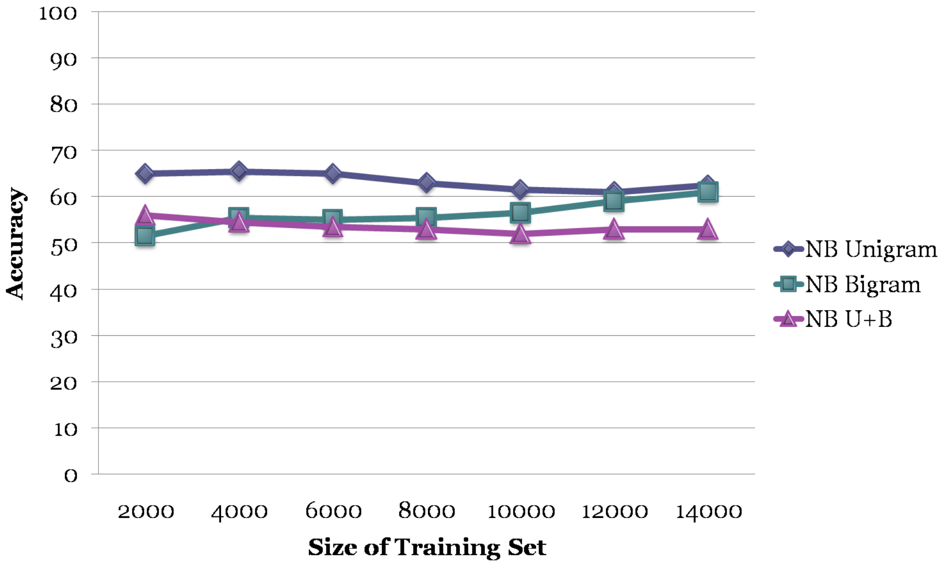


Figure 4.7: Naive Bayes with MPQA

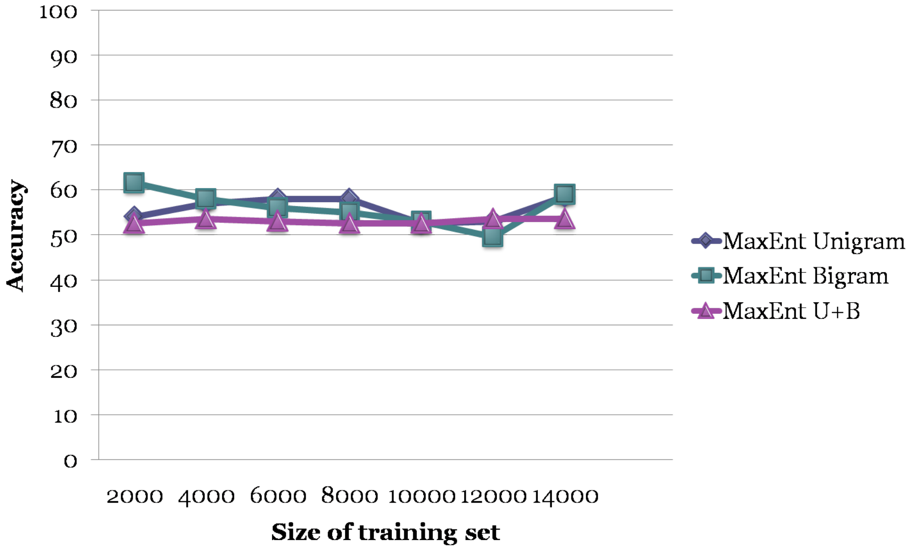


Figure 4.8: MaxEnt with MPQA

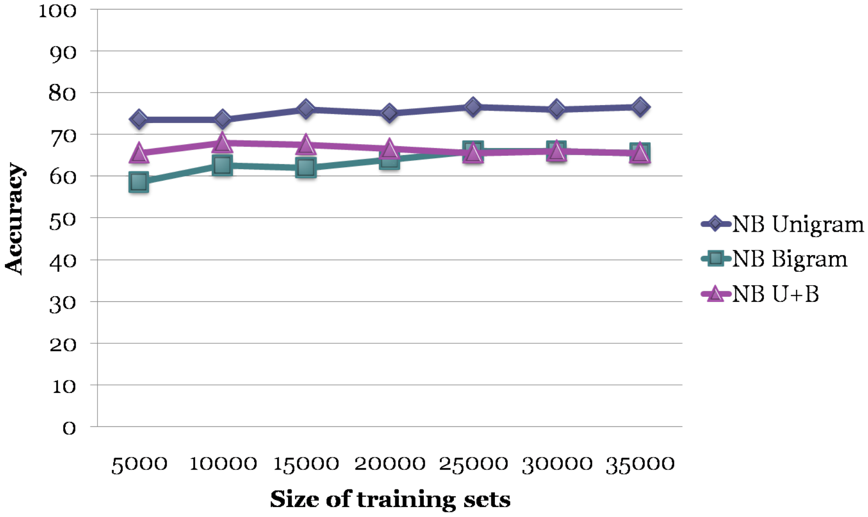


Figure 4.9: Naive Bayes with AFINN word list

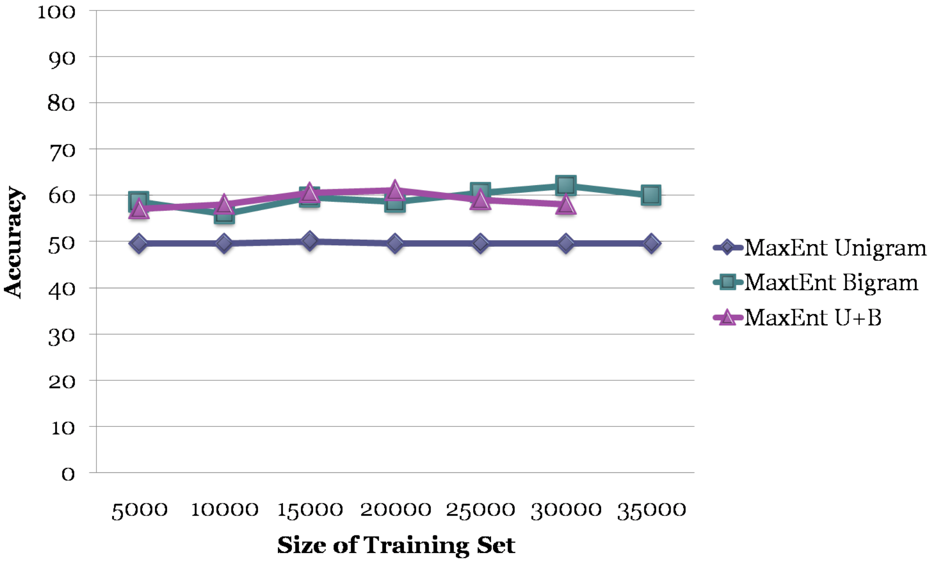


Figure 4.10: MaxEnt with AFINN word list

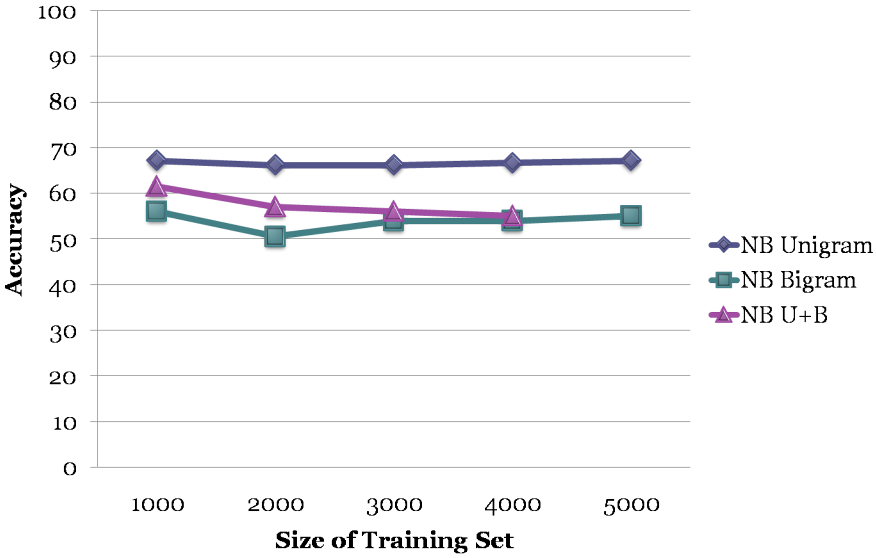


Figure 4.11: Naive Bayes with MPQA+AFINN

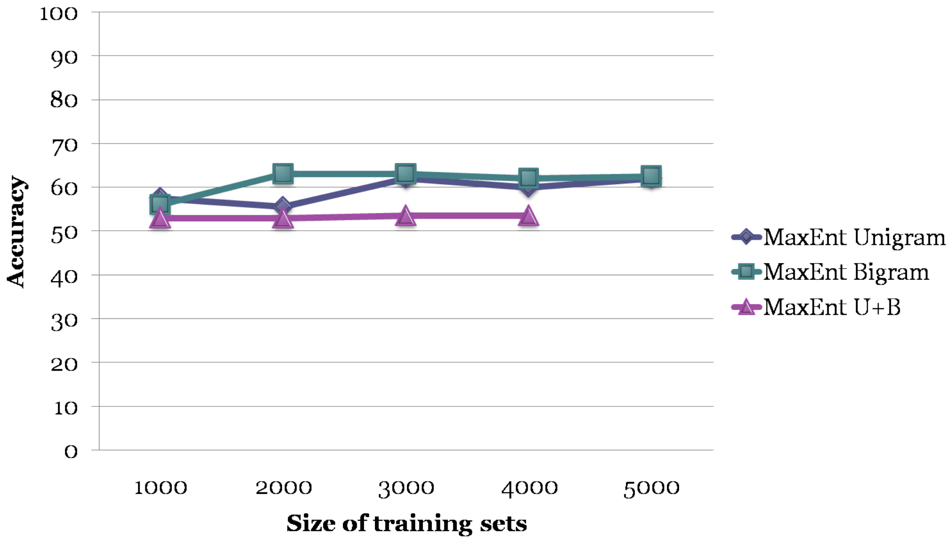


Figure 4.12: MaxEnt with MPQA+AFINN

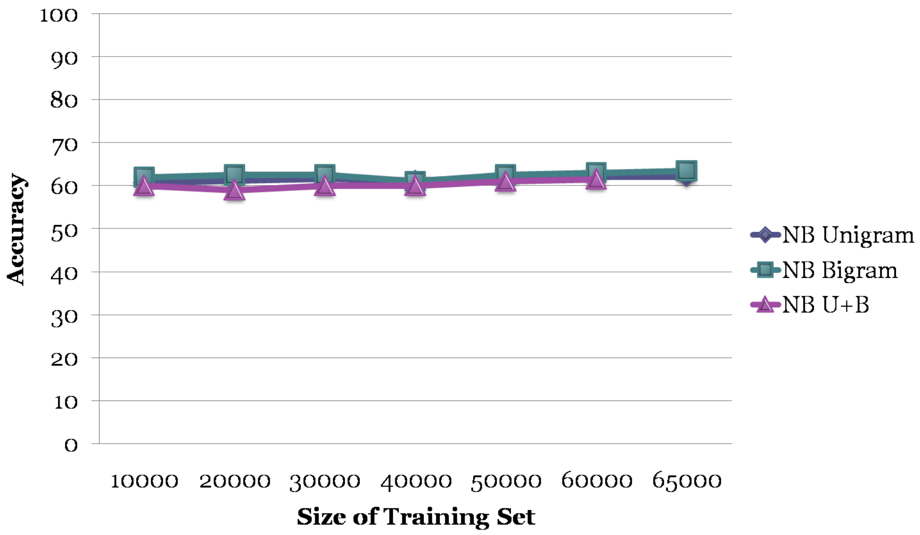


Figure 4.13: Naive Bayes with Wiki sentences

Except in the bigram case, NB fared better with unigram giving highest accuracies at 67.16% peak. In this case, our study was perhaps limited by the size of the training set as the number of tweets returned on filtering were quite low. However, with a larger MPQA+AFINN filtered training set, accuracies might be better in NB.

Wikipedia as a neutral training set as well showed higher accuracies for the NB model over the MaxEnt classifier, however, similar to MPQA, it fared worse compared to other training sets. This is probably due to the lack of grammatical structure and the style of writing used in twitter, which wikipedia sentences could perhaps not 'catch' during training.

Neutral training sets using URLs showed considerably high accuracies in NB for unigram, bigram and unigram+bigram. Accuracies however dropped for Maxent.

We considered using a 'hybrid' of all these training sets in NB in unigram since accuracies in NB were found to be consistently higher in case of unigrams.With a neutral training set of 25000 AFINN/5000 MPQA/25000 URL/15000 wiki we were able to achieve an accuracy of 73.5

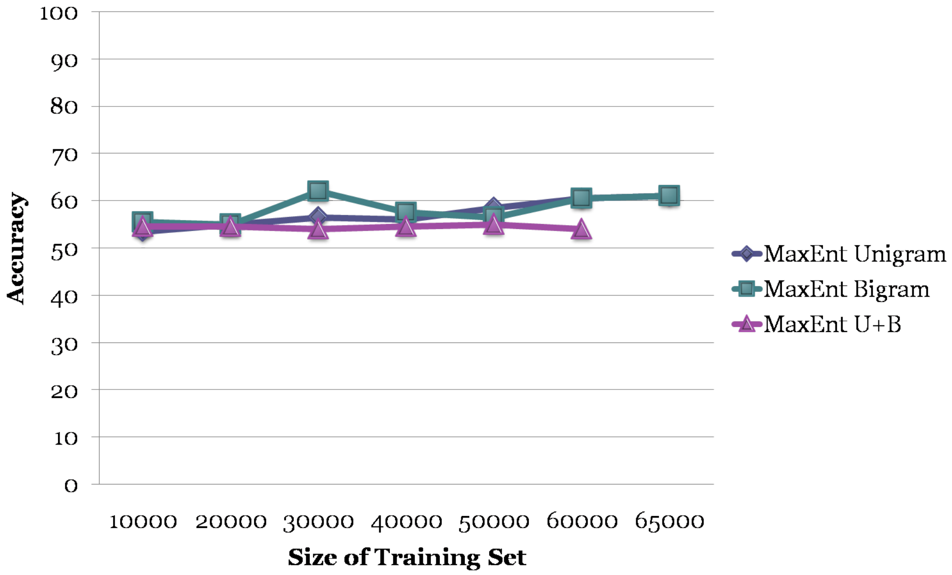


Figure 4.14: MaxEnt with Wiki sentences

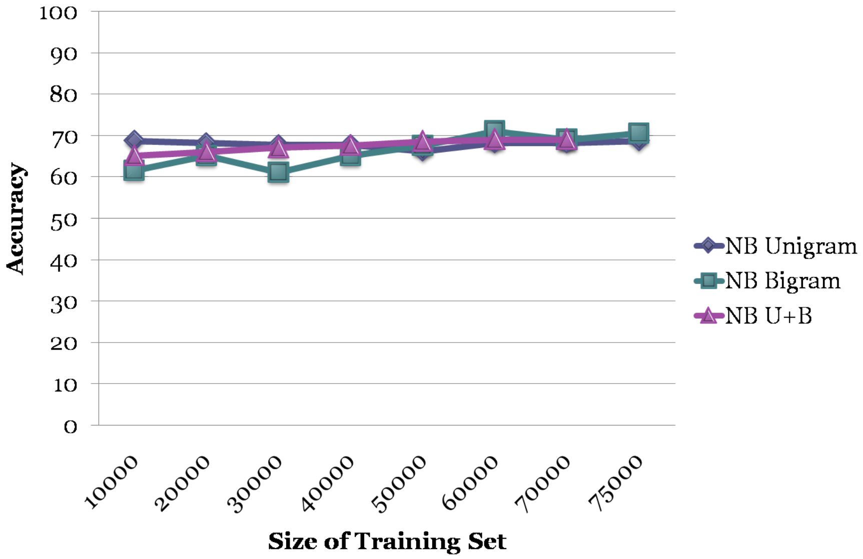


Figure 4.15: Naive Bayes with URL tweets

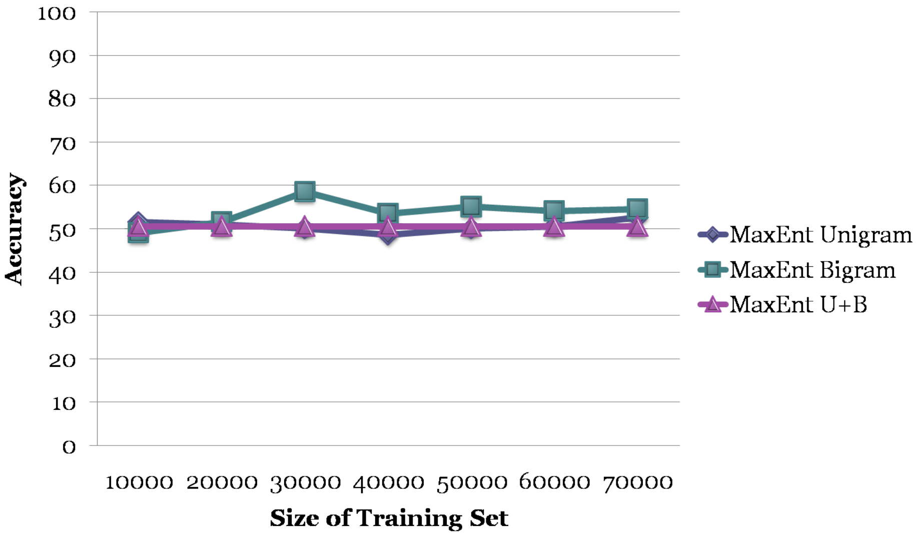


Figure 4.16: MaxEnt with URL tweets

Chapter 5

Conclusion & Future Work

In this work, we have performed a brief literature survey of sentiment analysis in Twitter and analysed the effect of cleaner training data on improving subjectivity classification accuracies in Twitter. We have proposed certain heuristics to select neutral training data and evaluated these heuristics by reporting their accuracies to various classification schemes. We proposed the use of lexicons, neutral Wikipedia sentences and tweets containing URLs in training the subjective classifier with higher accuracies. Having proposed this, we plot accuracies for each of these over parameters including training size and use of unigram,bigram or combination of both as well as over different classifiers. We have found Naive Bayes to perform well for this with accuracies reaching 76.5% with AFINN filtering matching accuracies of previous works on subjectivity such as quoted in [9] as 73.8% . Following is a list of our extensions to this study.

5.0.2 Training Size:

Our work showed particularly promising results for AFINN filtering on neutral training. However, we were unable to harness the immense potential of the Twitter data source in terms of size of the training set. Our training sets generated for AFINN were 35000 from an initial tweet set of 3,50,000. By increasing the filtered tweets to the order of millions, we believe accuracies can be further improved.

5.0.3 Domain Specificity:

Our best classifier achieved subjectivity classification accuracies of 76%. However, with the specification of the domain, accuracies can be further improved by restricting training data relevant to that domain.

5.0.4 Hybrid training sets:

URLs performed second best in our analysis with Naive Bayes. This was probably due to the fact that not all URL containing tweets are necessarily neutral, although a large majority of them are. Moreover, sentiment-laden words have a chance of being included before these URLs despite the fact that they may not convey any opinion of the user itself, but rather stating the opinion of others.

eg. oh dear http://tinyurl.com/dmcnz7 ms pelosi bad ? this too shall pass, u′ll see ... always does

This tweet while although containing a URL, has sentiment-laden tokens like ′bad′. However, during training, this tweet may wrongly train the classifier with the assumption that it is neutral, leading to low accuracies. A solution is to pass URL tweets through the AFINN lexicon to remove tweets containing sentiment-laden words, leaving us with only neutral URL tweets and better classification. Similar hybrid approaches between different heuristics may improve classification.

Bibliography

[1] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan.2002.Thumbs up? Sentiment Classification using Machine Learning Techniques .Proceedings Of EMNLP:79-86.

[2] Bernard J. Jansen, Mimi Zhang, Kate Sobel, Abdur Chowdury.2009.Twitter Power: Tweets as Electronic Word of Mouth.Journal Of the American Society for Information Science and Technology.

[3] Alec Go, Richa Bhayani, Lei Huang.2009.Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford Digital Library Technologies Project.

[4] Alexander Pak, Patrick Paroubek .2010.Twitter as a Corpus for Sentiment Analysis and Opinion Mining .Proceedings of LREC 2010.

[5] Bo Pang and Lillian Lee.2008.Opinion Mining and Sentiment Analysis .Foundations of Information Retrieval.Vol. 2, Nos. 12 (2008) 1135 .

[6] http://www.cs.waikato.ac.nz/ml/weka/

[7] http://fnielsen.posterous.com/afinn-a-new-word-list-for-sentiment-analysis

[8] http://www.cs.pitt.edu/mpqa/

[9] Janyce Wiebe, Ellen Riloff.2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts.Proccedings of the 6th International Conference on Computational Linguistics Intelligent Text Processing (CICLing-05).

[10] Bin Lu , Benjamin Tsou. 2010. Combining a Large Sentiment Lexicon in Machine Learning for Subjectivity Classification. Proceedings of the 9th International Conference on Machine Learning Cybernetics,Quingdao-2010.