SENTIMENT ANALYSIS IN TWITTER DATA

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Master of Technology
in
Software Engineering

Under the esteemed guidance of
Mr. Prashant Giridhar Shambharkar
(Assistant Professor)
Computer Science and Engineering
Delhi Technological University

Submitted By-
Ritu Kalonia
(Roll No. - 2K16/SWE/12)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING DELHI TECHNOLOGICAL UNIVERSITY SESSION: 2016-2018

DECLARATION

We hereby declare that the thesis work entitled "**Sentiment analysis in Twitter Data**" which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of degree of Master of Technology (Software Engineering) is a bonafide report of thesis carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Ritu Kalonia
2K16/SWE/12

CERTIFICATE

This is to certify that Ritu Kalonia (2K16/SWE/12) has completed the thesis titled "**Sentiment Analysis in Twitter Data**" under my supervision in partial fulfilment of the MASTER OF TECHNOLOGY degree in Software Engineering at DELHI TECHNOLOGICAL UNIVERSITY.

Supervisor

Mr. Prashant Gridhar Shambharkar

Assistant Professor

Department of Computer Science and Engineering

Delhi Technological University

Delhi -110042

ACKNOWLEDGEMENT

I am very thankful to Mr. Prashant Gridhar Shambharkar (Assistant Professor, Computer Science Eng. Dept.) and all the faculty members of the Computer Science Engineering Dept. of DTU. They all provided immense support and guidance for the completion of the project undertaken by me.

I would also like to express my gratitude to the university for providing the laboratories, infrastructure, testing facilities and environment which allowed me to work without any obstructions.

I would also like to appreciate the support provided by our lab assistants, seniors and peer group who aided me with all the knowledge they had regarding various topics.

Ritu Kalonia
M. Tech. in Software Engineering
Roll No. 2K16/SWE/12

Table of Contents

List of Figures

CHAPTER. 1
INTRODUCTION

With the recent growth of mobile information systems and the increased availability of smart phones, social media has become a large part of daily life in most societies. This development has entailed the creation of massive amounts of data: data which when analysed can be used to extract valuable information about a variety of subjects. Sentiment analysis (SA), also known as opinion mining is the process of classifying the emotion conveyed by a text, for example as negative, positive or neutral. The data made available by social media has contributed to a burst of research activity within SA in recent times and a shift in the focus of the field towards this type of data. Information gained from applying SA to social media data has many potential usages, for instance, to help marketers evaluate the success of an ad campaign, to identify how different demographics have received a product release, to predict user behaviour, or to forecast election results.

1.1. Twitter Sentiment Analysis

A popular social medium is Twitter,1 a micro-blogging site that allows users to write textual entries of up to 140 characters, commonly referred to as tweets. As of June 2015, Twitter has over 302 million monthly active users according to their homepage, whereof approximately 88 % have their tweets freely readable. Additionally, over 84 % of the users also have their location specified in their profiles [Beevolve, 2012], enabling the possibility of performing drill-down on geographic locations. Data created by Twitter is made available through Twitter's API, and represents a realtime information stream of opinionated data. Tweets can be filtered both by location and the time they were published. This has paved the way for a new sub-field of SA. Performing natural language processing on textual data from Twitter presents new challenges because of the informal nature of this data. Tweets often contain misspellings, and the constrictive limit of 140 characters encourages slang and abbreviations. Unconventional linguistic means are also used, such as capitalization or elongation of words to show emphasis.

Additionally, tweets contain special features like emoticons and hashtags that may have an analytical value. Hashtags are labels used for search and categorisation, and are included in the text prepended by a "#". Emoticons are expressions of emotion, and can either be written as a string of characters e.g., ":-)", or as a unicode symbol. Finally, if a tweet is a reply or is directed to another Twitter user, mentions can be used by prepending a username with "@".

1.2. Motivation and Research Focus Area

In this project we explore the effect of applying sophisticated negation scope detection to Twitter sentiment analysis. To our knowledge, no previous work has been done in this regard. The linguistic phenomenon of negation, described in Section 2.1.2, has been shown to play a significant role in SA. Councill et al. [2010] tested a sentiment classifier and found that including their negation classifier provided a 29.5 % improvement in F1 score when classifying positive sentiment, and an 11.4 % improvement when classifying negative sentiment.

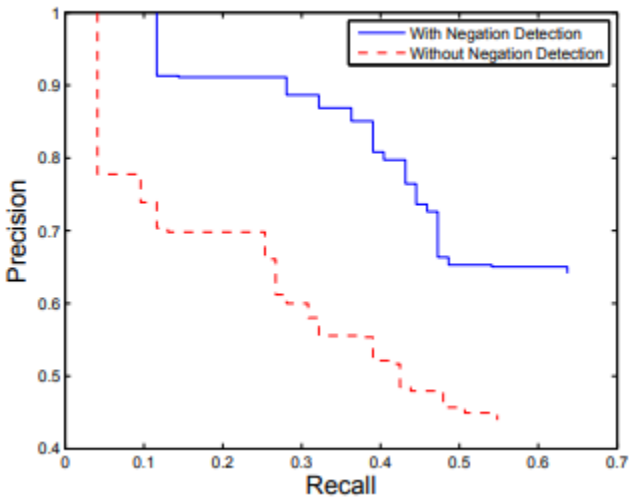

Figure 1.1 The impact of negation handling on positive sentiment prediction

Figure 1.1 graphs the effects on precision and recall of including negation handling when performing positive sentiment prediction, as recorded by Councill et al. Kiritchenko et al. [2014] included a sophisticated solution for handling negated terms in their SemEval-2014 entry by creating tweet-specific sentiment lexica containing individual scores for terms in affirmative and negated contexts, but the state-of-the-art systems in TSA still employ a very simple solution for

identifying which terms are negated, by marking as negated all words from a negation cue term to the next punctuation symbol. In Section 6.2 we present a baseline experiment we have conducted: implementing a naïve negation scope detection solution, commonly used in TSA, in order to compare how it performs on the BioScope Corpus to existing, more sophisticated solutions, and show the potential for improvement in TSA. This naïve solution is then improved upon, and several experiments carried out with a more sophisticated approach to detecting the negation scope, both in isolation and embedded as part of a complete TSA system. The experiments show that naïve classification is slightly outperformed by existing alternative solutions and by our own, improved negation detector.

1.3 Motivation

One of the most popular microbloging platforms is Twitter. Twitter has become a melting pot for all - ordinary individuals, celebrities, politicians, companies, activists, etc. Almost all the major news outlets have Twitter account where they post news headlines for their followers. People with Twitter accounts can reply to or retweet the news headlines. Twitter users who have an account can also post news headlines from any other news outlet. When people post news headlines on Twitter, reply to news posts, or retweet news posts, it is possible that they can express their sentiment along with what they are posting, retweeting or replying to. The interest of this thesis is in what people are saying about news in Twitter. Speci_cally, the interest is in determining the sentiment of Twitter posts about news. This interest was inspired by a local IT company called Heeii in Groningen, the Netherlands. The company develops Twitter applications for web browsers.

We have chosen to work with twitter since we feel it is a better approximation of public sentiment as opposed to conventional internet articles and web blogs. The reason is that the amount of relevant data is much larger for twitter, as compared to traditional blogging sites. Moreover the response on twitter is more prompt and also more general (since the number of users who tweet is substantially more than those who write web blogs on a daily basis). Sentiment analysis of public is highly critical in macro-scale socioeconomic phenomena like predicting the stock market rate of a particular firm. This could be done by analysing overall public sentiment towards that firm with respect to time and using economics tools for finding the

correlation between public sentiment and the firm's stock market value. Firms can also estimate how well their product is responding in the market, which areas of the market is it having a favourable response and in which a negative response (since twitter allows us to download stream of geo-tagged tweets for particular locations. If firms can get this information they can analyze the reasons behind geographically differentiated response, and so they can market their product in a more optimized manner by looking for appropriate solutions like creating suitable market segments. Predicting the results of popular political elections and polls is also an emerging application to sentiment analysis. One such study was conducted by Tumasjan et al. in Germany for predicting the outcome of federal elections in which concluded that twitter is a good reflection of offline sentiment [4].

1.4 Recent Developments Within Sentiment Analysis

Exploring popular opinion on various subjects has always been an important part of humans' information gathering behaviour. Where one in the past needed to conduct surveys to learn about opinion trends, for instance to conduct political polls, the availability of online data expressing sentiment has allowed for non-intrusive data mining to extract this information.

Over the last decade, there has been a substantial increase in the amount of work done in the field of SA. Surveys conducted by Pang and Lee [2008] and Liu and L. Zhang [2012] give a good overview of the state-of-the-art at the respective points in time. The work in the field of SA has largely followed the available data, both in terms of the amount of work done and the focus area. Figure 3.1 shows the amount of hits for queries (3.1) and (3.2) on Google Scholar,1 displaying a shift of the field towards Twitter data in recent years.

1.5 International Workshop on Semantic Evaluation

The International Workshop on Semantic Evaluation (SemEval)2 is a series of evaluations of computational semantic language analysis systems. In recent years, it has been hosted annually. Each iteration of SemEval has a set of tasks. Tasks are hosted by experts within the field of study, who assist participants by providing resources such as training data and facilitating

communication between teams. SemEval-2013 and SemEval2014 both included tasks for TSA, see Nakov et al. [2013] and Rosenthal et al. [2014].

Additionally, SemEval-2015 has two TSA shared tasks, and two TSA-related shared tasks will be included in SemEval-2016. Recent SemEvals have yielded significant improvements to the state-of-the-art of TSA. The TSA tasks in SemEval-2013 and SemEval-2014 included two sub tasks: a term-level subtask (Subtask A) where the aim was to classify the contextual polarity of a term in a tweet and an expression-level subtask (Subtask B) where the aim was to correctly classify the overall polarity of whole tweets. Subtask B is the one relevant to our project, and is the one we will focus on. Throughout the remainder of this report, when we refer to SemEval-2014, we are referring to Subtask B, unless explicitly stated otherwise. In addition to providing training, development, and test data sets of annotated tweets, the task hosts also provide out-of-domain data sets to test the versatility and generalisability of the created submissions.

1.6 State-of-the-Art in Twitter Sentiment Analysis

In this section we present the current state-of-the-art in TSA by breaking the field down into several areas. The typical approach to TSA uses a supervised machine learning system including three main steps: preprocessing, feature extraction, and training the classifier. To reduce noise and remove unnecessary information, the preprocessing step consists of a variety of filters for, e.g., normalizing URLs and elongated words. Features for the classifier are extracted using sentiment scores from polarity lexica, statistics from metacommunicative expressions specific to conversational language such as emoticons and hashtags, as well as natural language processing information including bag-of-words, part-of-speech tags and word clusters. Finally, training the classifier is usually a matter of performing a grid search over the parameter space for selecting the most suitable parameters for a supervised machine learning model.

1.7 Tweets Pre-Processing

Common preprocessing tasks in TSA include filtering out or normalizing URLs and user mentions, because these items have minimal information value in the context of sentiment classification. Agarwal et al. [2011] perform this normalization by substituting user mentions with the tag ||T|| and URLs with the tag ||U||. Another Twitter-specific syntactic feature is prefixing tweets with "RT" to indicate that the following part of the tweet is a retweet — a repost of previous content. A simple way of handling this is to remove the "RT" string from the tweet. It is also common to normalize elongated words, e.g., cooooooll, soooooooo, or happyyyyyy by substituting letters that occur many times sequentially with one or two occurrences of the letter. It was previously quite common to filter out hashtags [Selmer & Brevik 2013]. The assumption behind this is that hashtags when used as intended — i.e., to categorize posts by topic — offer little information of value. Mohammad [2012] show through experiments that hashtags add sentiment-semantic information to tweets by indicating the tone of the message or the writer's emotions.

Go et al. [2009] perform the typical preprocessing steps: URL, user mention and word-elongation normalization, and achieve a reduction of the feature space dimensionality by 45.85 % after constructing their feature vector further down the classification pipeline.

1.8 Twitter-Specific Sentiment Analysis

There are some Twitter-specific sentiment analysis studies. Twitter sentiment analysis is a bit different from the general sentiment analysis studies because Twitter posts are short. The maximum number of characters that are allowed in Twitter is 140. Moreover Twitter messages are full of slang and misspellings (Go et al., 2009). Almost all Twitter sentiment classification is done using machine learning techniques. Two good reasons for the use of machine learning techniques are 1) the availability of huge amount of Twitter data for training, and 2) that there is test data which is user-labeled for sentiment with emoticons (avoiding the cumbersome task of manually annotating data for training). Read (2005) showed that the use of emoticon for training is effective. Below I present some of the most relevant studies on Twitter sentiment analysis.

A Twitter sentiment analysis study by Go et al. (2009) does a two-classed (negative and positive) classication of tweets about a term. Emoticons (for positive ':)', for negative ':(') were used to collect training data from Twitter API. The training data was preprocessed before it was used to train the classier. Preprocessing included replacing user names and actual URLs by equivalence classes of 'URL' and 'USERNAME' respectively, removing repeated letters to 2 (huuuuuuungry to huungry), and removing the query term. To select useful uni-grams, they used such feature selection algorithms as frequency, mutual information, and chi-square method. They experiment with three supervised techniques: multinomial Naive Bayes, maximum entropy and support vector machines (SVM). The best result, accuracy of 84%, was obtained with multinomial Naive Bayes using uni-gram features selected on the basis of their MI score. They also experimented with bi-grams, but accuracy was low. They claim the reason for this low accuracy is data spareness. Incorporating POS, and negation into the feature vector of uni-grams does not also improve results.

The above experiment does not recognize and handle neutral tweets. To take into account neutral tweets, they collected tweets about a term that do not have emoticons. For test data, they manually annotated 33 tweets as neutral. They merged these two datasets with the training data and test data used in the above two-classed classication. They trained a three-classed classier and tested it, but the accuracy was very low, 40%.

Another study by Barbosa and Feng (2010) used a two-phased approach to Twitter sentiment analysis. The two phases are: 1) classifying the dataset into objective and subjective classes (subjectivity detection) and 2) classifying subjective sentences into positive and negative classes (polarity detection). Suspecting that the use of n-grams for Twitter sentiment analysis might not be a good strategy since Twitter messages are short, they use two other features of tweets: meta information about tweets and syntax of tweets. For meta-info, they use POS tags (some tags are likely to show sentiment, eg. adjectives and interjections) and mapping words to prior subjectivity (strong and weak), and prior polarity (negative, positive and neutral). The prior polarity is reversed when a negative expression precedes the word. For tweet syntax features, they use #(hashtag, @(reply), RT(retweet), link, punctuations, emoticons, capital- ized words, etc. They create a feature set from both the features and experiment with machine learning technique available in WEKA. SVM performs best. For test data, 1000 tweets were manually

annotated as positive, negative, and neutral. The highest accuracy obtained was 81.9% on subjectivity detection followed by 81.3% on polarity detection.

A very related study to this thesis was done by Pak and Paroubek (2010). They did a three-classed (positive, negative, neutral) sentiment analysis on Twitter posts. They collected negative and positive classes using emoticons.

For the neutral class, they took posts from Twitter accounts of popular news outlets (the assumption is news headlines are neutral). After the data collection, they did some linguistic analysis on the dataset. They POS tagged it and looked for any dierences between subjective (positive and negative) and objective sentences. They note that there are dierences between the POS tags of subjective and objective Twitter posts. They also note that there are dierence in the POS tags of positive and negative posts. Then they cleaned the data by removing URL links, user names (those that are marked by @), RT (for retweet), the emoticons, and stop words. Finally they tokenized the dataset and constructed n-grams. Then they experimented with several classiers including SVM, but Naive Bayes was found to give the best result. They trained two Naive Bayes Classiers. One of them uses n-gram presence, and the other, POS tag presence.

Pak and Paroubek (2010) achieved best result (highest accuracy) with bigram presence. Their explanation for this is that bi-grams provide a good balance between coverage (uni-grams) and capturing sentiment expression patterns (tri-grams) (Pak and Paroubek, 2010). Negation('not' and 'no') is handled by attaching it to the words that precede and follow it during tokenization. The handling of negation is found to improve accuracy. Moreover, they report that removing n-grams that are evenly distributed in the sentiment classes improves accuracy. Evaluation was done on the same test data used by Go et al. (2009). However, they do not explicitly put their accuracy in number other than showing it in a graph (in which it seems to approach 1) and stating it in words saying a very high accuracy"

1.9 Test Data

1.9.1 Collection

Because the objective of the thesis is to analyze the sentiment of Twitter messages posted in reaction to news, only tweets about news should be collected. However, this is not a simple task. There seems to be no way of obtaining all and only tweets that are posted in reaction to news articles. To overcome this problem, some sort of bootstrapping approach was used. In this approach, rst tweets that contain words of the news headline above some threshold were collected. The threshold used is 3 words and the news headlines are obtained from the news feed of the Denver Post. Once tweets are collected this way, the links from these tweets are extracted. A script uses these extracted links to go to Twitter and to fetch more tweets that contain these links.

There are two assumptions in the bootstrapping approach used above. The rst is that Twitter posts about a particular news articles will always contain a link to the tweet unless the news has been circulated enough to be public knowledge. It does not make sense for somebody to post a reaction to a news article without a link to it. This assumption has a problem because the same news and therefore the same news headline can be used by a dierent news outlet. Therefore the tweet may not be , for example, a reaction to the news article posted by the Denver Post, but instead it is for the same news article posted by New York Times. However this does not aect the sentiment analysis task. What it aects is if somebody wants to know how many people reacted and posted in Twitter, for example, to a news article posted in the Denver Post. This can be solved by deciphering the short URLs to their real URLs. That is if a tweet contains a short URL, and on deciphering it, it gives a real URL that belongs to Denver Post, obviously the Twitter post is meant to be to the news article in The Denver Post.

The second assumption, the reason for going to Twitter to fetch more tweets that contain the link from the initial tweets, is that there will be posts that will not contain words from the news headline but that are still posted in reaction to the news. Such posts will be the types of posts that reect the user's sentiment (negative or positive), unlike the posts that contain words from the headline, which are usually neutral ones. (). This assumption is itself based on another assumption which says that a single real URL will have the same short URL no matter how many people post and repost it on Twitter. However, this is not true for two reasons. One reason is that there are many URL shortening services (186 according to Wikipedia) and thus the short URL will not be the same as a user may use any of them. The second reason is even for one URL

shortening service, there can be more than one short URL for a give real URL because many of the URL shortening services allow users to customize their URLs. Had it not been for this two reasons, it would be possible to fetch all tweets posted in reaction to news article by a certain news outlet, for example, the Denver Post.

1.9.2 News content, Tweets about news and manual annotation

In order to see this, a corpus of 1000 tweets about news was sampled from the test data and manually examined and annotated as negative, positive and neutral. A web interface was built to aid the manual annotation. Where the sentiment of the news was possible to understand from the Twitter posts only, the sentiment was provided for it, where it was di-cult to determine its sentiment, i.e. where it needed a context to assign it a sentiment, it was annotated with 'context'. Here is the procedure I followed in annotating the sentiment of the news :

• A Twitter post that, on reading, sounds to be a news headline is annotated as neutral. This does not mean it does not contain words indicating sentiment

• A Twitter post that contains subordinating conjunctions was annotated the sentiment of the main clause

• A Twitter post that contains subtleties and sarcasms was annotated as one of the three sentiment classes only if it was clearly determinable

• A Twitter post that were di-cult to give a sentiment was annotated 'context' (this is a tweet that needs context to determine its sentiment).

• A sentiment expressed on the content or presentation is taken to be the same, i.e they get the same annotation.

Out of 1000 Twitter posts about news that have been annotated following the above procedure, 31 Twitter posts needed context to understand and determine their sentiment. Thus 96.9% of the test data did not require context to determine their sentiment. So, if context is ignored and if tweets about news are assumed to be context-independent, the accuracy of the assumption becomes 96.9%. This is an assumption worth taking for it is high. But it is important to note here that annotating Twitter posts about news was not easy. Many times, it was difficult to determine

the sentiment. Some Twitter posts seemed both negative and positive. Neutral Twitter posts are difficult to differentiate from either negative or positive Twitter posts. This is because neutral posts can have both negative and positive sentiments. The only litmus for recognizing neutral Twitter posts is to see if they can appear as news headline. Thus determining the sentiment of a Twitter post is not as straightforward as it may seem.

Determining the sentiment without the context of the news is made difficult by other factors too. One factor is sarcasm. What clearly seems to be positive or negative tweet may turn out to be otherwise when seen against the content of the news article. Moreover, the sentiment expressed may be on the news content or the presentation of the news. Twitter posts that contain question marks tend to require context to understand them. For example, "What is Sociology For? Doom, Gloom And Despair http://dlvr.it/DhDss ". This tweet requires reading the content of the news provided in the link to say if it is negative or positive or neutral. A related thing that I tried to examine was whether Twitter posts about news involve third party opinion such as I do not like the article's support for Hamas. Out of 1000 tweets about news, I did not nd a single tweet that involve third party opinion. So, here again, it is safe to assume that tweets about news do not involve third-party opinions.

The high accuracy (97%) above means that context can be ignored in the domain of tweets about news. Thus the whole work of sentiment analysis on tweets about news will assume that the sentiment of a tweet about news is universal and not specic to the content of that particular news. In other words, it is assumed that the sentiment of a tweet about news can be understood without the contents of the news.

CHAPTER. 2
LITERATURE REVIEW

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews [x], documents, web blogs/articles and general phrase level sentiment analysis [16]. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised (e.g., [11] and [13]) and semi-supervised (e.g., [3] and [10]) approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

2.1 Related Work

The bag-of-words model is one of the most widely used feature model for almost all text classification tasks due to its simplicity coupled with good performance. The model represents the text to be classified as a bag or collection of individual words with no link or dependence of one word with the other, i.e. it completely disregards grammar and order of words within the text. This model is also very popular in sentiment analysis and has been used by various researchers. The simplest way to incorporate this model in our classifier is by using unigrams as features. Generally speaking n-grams is a contiguous sequence of "n" words in our text, which is completely independent of any other words or grams in the text. So unigrams is just a collection of individual words in the text to be classified, and we assume that the probability of occurrence of one word will not be affected by the presence or absence of any other word in the text. This is

a very simplifying assumption but it has been shown to provide rather good performance (for example in [7] and [2]). One simple way to use unigrams as features is to assign them with a certain prior polarity, and take the average of the overall polarity of the text, where the overall polarity of the text could simply be calculated by summing the prior polarities of individual unigrams. Prior polarity of the word would be positive if the word is generally used as an indication of positivity, for example the word "sweet"; while it would be negative if the word is generally associated with negative connotations, for example "evil". There can also be degrees of polarity in the model, which means how much indicative is that word for that particular class. A word like "awesome" would probably have strong subjective polarity along with positivity, while the word "decent" would although have positive prior polarity but probably with weak subjectivity. There are three ways of using prior polarity of words as features. The simpler un-supervised approach is to use publicly available online lexicons/dictionaries which map a word to its prior polarity. The Multi-Perspective-Question-Answering (MPQA) is an online resource with such a subjectivity lexicon which maps a total of 4,850 words according to whether they are "positive" or "negative" and whether they have "strong" or "weak" subjectivity [25]. The SentiWordNet 3.0 is another such resource which gives probability of each word belonging to positive, negative and neutral classes [15]. The second approach is to construct a custom prior polarity dictionary from our training data according to the occurrence of each word in each particular class. For example if a certain word is occurring more often in the positive labelled phrases in our training dataset (as compared to other classes) then we can calculate the probability of that word belonging to positive class to be higher than the probability of occurring in any other class. This approach has been shown to give better performance, since the prior polarity of words is more suited and fitted to a particular type of text and is not very general like in the former approach. However, the latter is a supervised approach because the training data has to be labelled in the appropriate classes before it is possible to calculate the relative occurrence of a word in each of the class. Kouloumpis et al. noted a decrease in performance by using the lexicon word features along with custom n-gram word features constructed from the training data, as opposed to when the n-grams were used alone [7]. The third approach is a middle ground between the above two approaches. In this approach we construct our own polarity lexicon but not necessarily from our training data, so we don't need to have labelled training data. One way of doing this as proposed by Turney et al. is to calculate the prior

semantic orientation (polarity) of a word or phrase by calculating it's mutual information with the word "excellent" and subtracting the result with the mutual information of that word or phrase with the word "poor" [11]. They used the number of result hit counts from online search engines of a relevant query to compute the mutual information. The final formula they used is as follows:

$$Polarity(phrase) = \log_2 \frac{hits(phraseNEAR"excellent").hits("poor")}{hits(phraseNEAR"poor").hits("excellent")} \quad (1)$$

Where hits(phrase NEAR "excellent") means the number documents returned by the search engine in which the phrase (whose polarity is to be calculated) and word "excellent" are co-occurring. While hits("excellent") means the number of documents retuned which contain the word "excellent". Prabowo et al. have gone ahead with this idea and used a seed of 120 positive words and 120 negative to perform the internet searches [12]. So the overall semantic orientation of the word under consideration can be found by calculating the closeness of that word with each one of the seed words and taking and average of it. Another graphical way of calculating polarity of adjectives has been discussed by Hatzivassiloglou et al. [8]. The process involves first identifying all conjunctions of adjectives from the corpus and using a supervised algorithm to mark every pair of adjectives as belonging to the same semantic orientation or different. A graph is constructed in which the nodes are the adjectives and links indicate same or different semantic orientation. Finally a clustering algorithm is applied which divides the graph into two subsets such that nodes within a subset mainly contain links of same orientation and links between the two subsets mainly contain links of different orientation. One of the subsets would contain positive adjectives and the other would contain negative.

Many of the researchers in this field have used already constructed publicly available lexicons of sentiment bearing words (e.g., [7], [12] and [16]) while many others have also explored building their own prior polarity lexicons (e.g., [3], [10] and [11]).

The basic problem with the approach of prior polarity approach has been identified by Wilson et al. who distinguish between prior polarity and contextual polarity [16]. They say that the prior polarity of a word may in fact be different from the way the word has been used in the particular context. The paper presented the following phrase as an example:

Philip Clapp, president of the National Environment Trust, sums up well the general thrust of the reaction of environmental movements: “There is no reason at all to believe that the polluters are suddenly going to become reasonable.”

In this example all of the four underlined words “trust”, “well”, “reason” and “reasonable” have positive polarities when observed without context to the phrase, but here they are not being used to express a positive sentiment. This concludes that even though generally speaking a word like “trust” may be used in positive sentences, but this doesn’t rule out the chances of it appearing in non-positive sentences as well.

Henceforth prior polarities of individual words (whether the words generally carry positive or negative connotations) may alone not enough for the problem. The paper explores some other features which include grammar and syntactical relationships between words to make their classifier better at judging the contextual polarity of the phrase.

The task of twitter sentiment analysis can be most closely related to phraselevel sentiment analysis. A seminal paper on phrase level sentiment analysis was presented in 2005 by Wilson et al. [16] which identified a new approach to the problem by first classifying phrases according to subjectivity (polar) and objectivity (neutral) and then further classifying the subjective-classified phrases as either positive or negative. The paper noticed that many of the objective phrases used prior sentiment bearing words in them, which led to poor classification of especially objective phrases.

Table 1: Step 1 results for Objective / Subjective Classification in [16]

Features	Accuracy	Subjective F.	Objective F.
Word tokens	73.6	55.7	81.2
Words + prior polarity	74.2	60.6	80.7
28 Features	75.9	63.6	82.1

It claims that if we use a simple classifier which assumes that the contextual polarity of the word is merely equal to its prior polarity gives a result of about 48%. The novel classification process proposed by this paper along with the list of ingenious features which include information about contextual polarity resulted in significant improvement in performance (in terms of accuracy) of the classification process. The results from this paper are presented in the table above.

Table 2: Step 2 results for Objective / Subjective Classification in [16]

Features	Accuracy	Positive F.	Negative F.	Both F.	Objective F.
Word tokens	61.7	61.2	73.1	14.6	37.7
Word + prior	63.0	61.6	75.5	14.6	40.7
10 Features	65.7	65.1	77.2	16.1	46.2

One way of alleviating the condition of independence and including partial context in our word models is to use bigrams and trigrams as well besides unigrams. Bigrams are collection of two contiguous words in a text, and similarly trigrams are collection of three contiguous words. So we could calculate the prior polarity of the bigram / trigram - or the prior probability of that bigram / trigram belonging to a certain class – instead of prior polarity of individual words. Many researchers have experimented with them with the general conclusion that if we have to use one of them alone unigrams perform the best, while unigrams along with bigrams may give better results with certain classifiers [2], [3]. However trigrams usually result in poor performance as reported by Pak et al. [3]. The reduction in performance by using trigrams is because there is a compromise between capturing more intricate patterns and word coverage as one goes to higher-numbered grams. Besides from this some researchers have tried to incorporate negation into the unigram word models. Pang et al. and Pakl et al. used a model in which the prior polarity of the word was reversed if there was a negation (like “not”, “no”, “don’t”, etc.) next to that word [5], [3]. In this way some contextual information is included in the word models. Grammatical features (like “Parts of Speech Tagging” or POS tagging) are also commonly used in this domain. The concept is to tag each word of the tweet in terms of what

part of speech it belongs to: noun, pronoun, verb, adjective, adverb, interjections, intensifiers etc. The concept is to detect patterns based on these POS and use them in the classification process. For example it has been reported that objective tweets contain more common nouns and third-person verbs than subjective tweets [3], so if a tweet to be classified has a proportionally large usage of common nouns and verbs in third person, that tweet would have a greater probability of being objective (according to this particular feature). Similarly subjective tweets contain more adverbs, adjectives and interjections [3]. These relationships are demonstrated in the figures below:

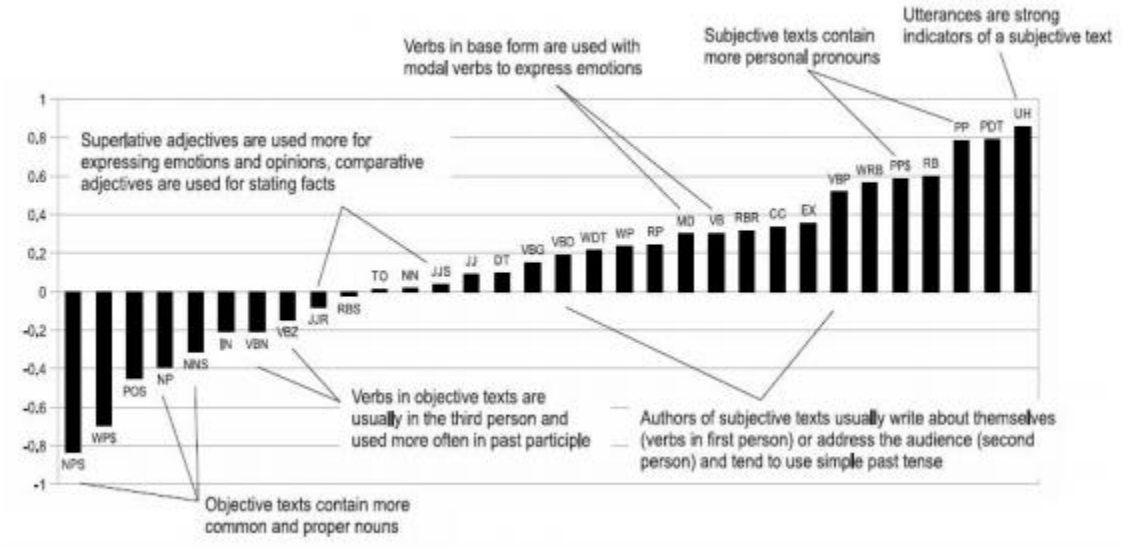

Fig. 2.1 Using POS Tagging as features for objectivity/subjectivity classification

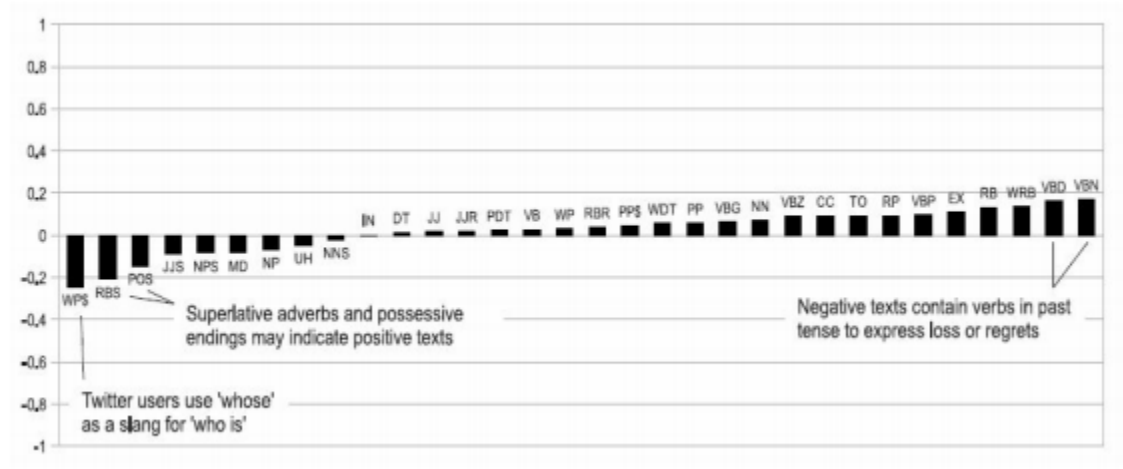

Fig. 2.2 Using POS Tagging as features in positive/negative classification

However there is still conflict whether Parts-of-Speech are a useful feature for sentiment classification or not. Some researchers argue in favour of good POS features (e.g., [10]) while others not recommending them.

Besides from these much work has been done in exploring a class of features pertinent only to micro blogging domain. Presence of URL and number of capitalized words/alphabets in a tweet have been explored by Koulompis et al. [7] and Barbosa et al. [10]. Koulmpis also reports positive results for using emoticons and internet slang words as features. Brody et al. does study on word lengthening as a sign of subjectivity in a tweet [13]. The paper reports positive results for their study that the more number of cases a word has of lengthening, the more chance there of that word being a strong indication of subjectivity.

The most commonly used classification techniques are the Naive Bayes Classifier and State Vector Machines. Some researchers like Barbosa et al. publish better results for SVMs [10] while others like Pak et al. support Naive Bayes [3]. (1-9) and (2-6) also report good results for Maximum Entropy classifier.

It has been observed that having a larger training sample pays off to a certain degree, after which the accuracy of the classifier stays almost constant even if we keep adding more labelled tweets in the training data [10]. Barbosa et al. used tweets labelled by internet resources (e.g., [28]), instead of labelling them by hand, for training the classifier. Although there is loss of accuracy of the labelled samples in doing so (which is modelled as increase in noise) but it has been observed that if the accuracy of training labels is greater than 50%, the more the labels, the higher the accuracy of the resulting classifier. So in this way if there are an extremely large number of tweets, the fact that our labels are noisy and inaccurate can be compensated for [10]. On the other hand Pak et al. and Go et al. [2] use presence of positive or negative emoticons to assign labels to the tweets [3]. Like in the above case they used large number of tweets to reduce effect of noise in their training data.

Some of the earliest work in this field classified text only as positive or negative, assuming that all the data provided is subjective (for example in [2] and [5]). While this is a good assumption for something like movie reviews but when analyzing tweets and blogs there is a lot of objective text we have to consider, so incorporating neutral class into the classification process is now

becoming a norm. Some of the work which has included neutral class into their classification process includes [7], [10], [3] and [16].

There has also been very recent research of classifying tweets according to the mood expressed in them, which goes one step further. Bollen et al. explores this area and develops a technique to classify tweets into six distinct moods: tension, depression, anger, vigour, fatigue and confusion [9]. They use an extended version of Profile of Mood States (POMS): a widely accepted psychometric instrument. They generate a word dictionary and assign them weights corresponding to each of the six mood states, and then they represented each tweet as a vector corresponding to these six dimensions. However not much detail has been provided into how they built their customized lexicon and what technique did they use for classification.

CHAPTER. 3

FUNCTIONALITY AND DESIGN

3.1 Natural Language Processing

Natural language processing (NLP) is a field in the Human-Machine Interaction area concerned with the use of human natural languages for communication with computers. Among the many topics of NLP, the following are particularly relevant in this project.

3.1.1 Bag-of-Words Model

A common way to represent text documents in a simplified manner is by using a bag-of-words model. The technique lists term occurrence and optionally the frequency of term occurrence, disregarding grammar and term order. Machine learning classifiers can use the resulting model directly as feature vectors.

3.1.2 Part-of-Speech Tagging

Part-of-speech (POS) tagging is the process of categorizing the tokens of a sentence into the different parts of speech (such as nouns, verbs, adjectives and adverbs) based on their definitions as well as the contexts. This way, POS tagging attempts to solve the problem of word ambiguity. There are many POS taggers for regular languages trained on treebanks — particularly for the newswire domain — such as the Penn Treebank [Marcus et al., 1993]. However, the conversational language of Twitter causes an out-of-domain problem for these traditional POS taggers, degrading their performance. Gimpel et al. [2011] present a POS tagger tailored to Twitter.

3.1.3 Word2Vec

Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

The process of designing a functional classifier for sentiment analysis can be broken down into four basic categories. They are as follows:

I. Data Acquisition

II. Human Labelling

III. Feature Extraction

IV. Classification

3.2 Data Acquisition

Data in the form of raw tweets is acquired by using the python library “tweestream” which provides a package for simple twitter streaming API [26]. This API allows two modes of accessing tweets: SampleStream and FilterStream. SampleStream simply delivers a small, random sample of all the tweets streaming at a real time. FilterStream delivers tweet which match a certain criteria. It can filter the delivered tweets according to three criteria:

• Specific keyword(s) to track/search for in the tweets

• Specific Twitter user(s) according to their user-id’s

• Tweets originating from specific location(s) (only for geo-tagged tweets).

A programmer can specify any single one of these filtering criteria or a multiple combination of these. But for our purpose we have no such restriction and will thus stick to the SampleStream mode.

Since we wanted to increase the generality of our data, we acquired it in portions at different points of time instead of acquiring all of it at one go. If we used the latter approach then the generality of the tweets might have been compromised since a significant portion of the tweets would be referring to some certain trending topic and would thus have more or less of the same general mood or sentiment. This phenomenon has been observed when we were going through our sample of acquired tweets. For example the sample acquired near Christmas and New Year's had a significant portion of tweets referring to these joyous events and were thus of a generally positive sentiment. Sampling our data in portions at different points in time would thus try to minimize this problem. Thus forth, we acquired data at four different points which would be 17th of December 2011, 29th of December 2011, 19th of January 2012 and 8th of February 2012.

A tweet acquired by this method has a lot of raw information in it which we may or may not find useful for our particular application. It comes in the form of the python "dictionary" data type with various key-value pairs. A list of some key-value pairs are given below:

• Whether a tweet has been favourited

• User ID

• Screen name of the user

• Original Text of the tweet

• Presence of hashtags

• Language under which the twitter user has registered their account

Since this is a lot of information we only filter out the information that we need and discard the rest. For our particular application we iterate through all the tweets in our sample and save the actual text content of the tweets in a separate file given that language of the twitter is user's

account is specified to be English. The original text content of the tweet is given under the dictionary key “text” and the language of user’s account is given under “lang”.

Since human labelling is an expensive process we further filter out the tweets to be labelled so that we have the greatest amount of variation in tweets without the loss of generality. The filtering criteria applied are stated below:

• Remove Retweets (any tweet which contains the string “RT”)

• Remove very short tweets (tweet with length less than 20 characters)

• Remove non-English tweets (by comparing the words of the tweets with a list of 2,000 common English words, tweets with less than 15% of content matching threshold are discarded)

After this filtering roughly 30% of tweets remain for human labelling on average per sample, which made a total of 5,971 tweets to be labelled.

3.3 Human Labelling

For the purpose of human labelling we made three copies of the tweets so that they can be labelled by four individual sources. This is done so that we can take average opinion of people on the sentiment of the tweet and in this way the noise and inaccuracies in labelling can be minimized. Generally speaking the more copies of labels we can get the better it is, but we have to keep the cost of labelling in our mind, hence we reached at the reasonable figure of three.

We labelled the tweets in four classes according to sentiments expressed/observed in the tweets: positive, negative, neutral/objective and ambiguous. We gave the following guidelines to our labellers to help them in the labelling process:

• **Positive**: If the entire tweet has a positive/happy/excited/joyful attitude or if something is mentioned with positive connotations. Also if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant. Example: “4 more years of being in shithole Australia then I move to the USA! :D”.

• **Negative**: If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant. Example: "I want an android now this iPhone is boring :S".

• **Neutral/Objective**: If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information. Advertisements of different products would be labelled under this category. Example: "US House Speaker vows to stop Obama contraceptive rule... http://t.co/cyEWqKlE"

Besides this labellers were instructed to keep personal biases out of labelling and make no assumptions, i.e. judge the tweet not from any past extra personal information and only from the information provided in the current individual tweet.

Over here the strict measure is when classification is between the three categories of positive, negative and neutral. These results reiterate our initial claim that sentiment analysis is an inherently difficult task. These results are higher than our agreement results because in this case humans are being asked to label individual words which is an easier task than labelling entire tweets.

3.4 Feature Extraction

Now that we have arrived at our training set we need to extract useful features from it which can be used in the process of classification. But first we will discuss some text formatting techniques which will aid us in feature extraction:

• Tokenization: It is the process of breaking a stream of text up into words, symbols and other meaningful elements called "tokens". Tokens can be separated by whitespace characters and/or punctuation characters. It is done so that we can look at tokens as individual components that make up a tweet.

• Punctuation marks and digits/numerals may be removed if for example we wish to compare the tweet to a list of English words.

• Lowercase Conversion: Tweet may be normalized by converting it to lowercase which makes it's comparison with an English dictionary easier.

• Stemming: It is the text normalizing process of reducing a derived word to its root or stem [28]. For example a stemmer would reduce the phrases "stemmer", "stemmed", "stemming" to the root word "stem". Advantage of stemming is that it makes comparison between words simpler, as we do not need to deal with complex grammatical transformations of the word. In our case we employed the algorithm of "porter stemming" on both the tweets and the dictionary, whenever there was a need of comparison.

• Stop-words removal: Stop words are class of some extremely common words which hold no additional information when used in a text and are thus claimed to be useless. Examples include "a", "an", "the", "he", "she", "by", "on", etc. It is sometimes convenient to remove these words because they hold no additional information since they are used almost equally in all classes of text, for example when computing prior-sentiment-polarity of words in a tweet according to their frequency of occurrence in different classes and using this polarity to calculate the average sentiment of the tweet over the set of words used in that tweet.

• Parts-of-Speech Tagging: POS-Tagging is the process of assigning a tag to each word in the sentence as to which grammatical part of speech that word belongs to, i.e. noun, verb, adjective, adverb, coordinating conjunction etc.

Now that we have discussed some of the text formatting techniques employed by us, we will move to the list of features that we have explored. As we will see below a feature is any variable which can help our classifier in differentiating between the different classes. There are two kinds of classification in our system (as will be discussed in detail in the next section), the objectivity / subjectivity classification and the positivity / negativity classification. As the name suggests the former is for differentiating between objective and subjective classes while the latter is for differentiating between positive and negative classes.

The list of features explored for objective / subjective classification is as below:

• Number of exclamation marks in a tweet

• Number of question marks in a tweet

• Presence of exclamation marks in a tweet

• Presence of question marks in a tweet

• Presence of emoticons in a tweet

• Number of capitalized words in a tweet

• Number of capitalized characters in a tweet

• Number of punctuation marks / symbols in a tweet

3.5 Classification

Pattern classification is the process through which data is divided into different classes according to some common patterns which are found in one class which differ to some degree with the patterns found in the other classes. The ultimate aim of our project is to design a classifier which accurately classifies tweets in the following four sentiment classes: positive, negative, neutral and ambiguous.

There can be two kinds of sentiment classifications in this area: contextual sentiment analysis and general sentiment analysis. Contextual sentiment analysis deals with classifying specific parts of a tweet according to the context provided, for example for the tweet "4 more years of being in shithole Australia then I move to the USA :D" a contextual sentiment classifier would identify Australia with negative sentiment and USA with a positive sentiment. On the other hand general sentiment analysis deals with the general sentiment of the entire text (tweet in this case) as a whole. Thus for the tweet mentioned earlier since there is an overall positive attitude, an accurate general sentiment classifier would identify it as positive. For our particular project we will only be dealing with the latter case, i.e. of general (overall) sentiment analysis of the tweet as a whole.

The classification approach generally followed in this domain is a two-step approach. First Objectivity Classification is done which deals with classifying a tweet or a phrase as either objective or subjective. After this we perform Polarity Classification (only on tweets classified as subjective by the objectivity classification) to determine whether the tweet is positive, negative or both (some researchers include the both category and some don't).

We used the following Machine Learning algorithms for this second classification to arrive at the best result:

• K-Means Clustering

• Naive Bayes

• Random Forest

3.5.1 K-Means Clustering

k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\| x_i - v_j \| \right)^2 \quad (1)$$

where,

'$||x_i - v_j||$' is the Euclidean distance between x_i and v_j.

'c_i' is the number of data points in i^{th} cluster.

'c' is the number of cluster centers.

It is fast, robust and easier to understand and gives best result when data set are distinct or well separated from each other.

3.5.2 Naive Bayes

The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Bayes theorem provides a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$. Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \quad (2)$$

$$P(c \mid x) = P(x_1 \mid c) * P(x_2 \mid c) * ... * P(x_n \mid c) * P(c) \quad (3)$$

- $P(c|x)$ is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- $P(c)$ is the prior probability of *class*.

- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

In ZeroR model there is no predictor, in OneR model we try to find the single best predictor, naive Bayesian includes all predictors using Bayes' rule and the independence assumptions between predictors.

3.5.3 Random Forest

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks. In this post, you are going to learn, how the random forest algorithm works and several other important things about it. Random Forest is a supervised learning algorithm. Like you can already see from it's name, it creates a forest and makes it somehow random. The „forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.

One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. I will talk about random forest in classification, since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with two trees:

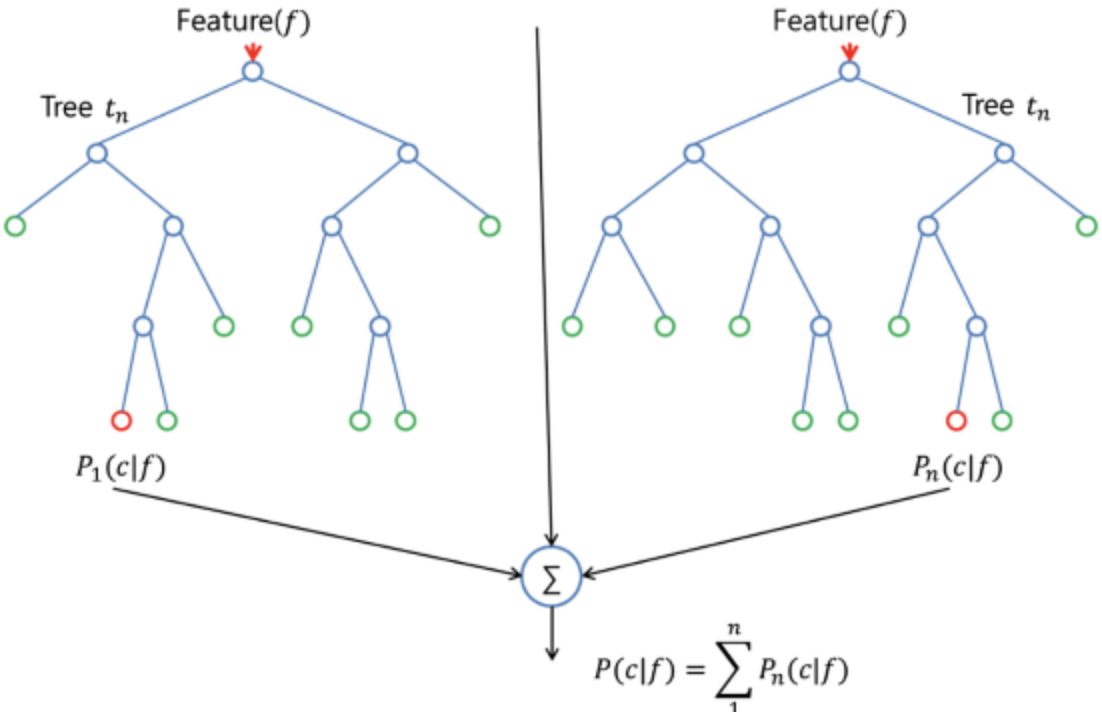

Fig. 3.1 Graphical representation of Random Forest Distribution Algorithm

With a few exceptions a random-forest classifier has all the hyperparameters of a decision-tree classifier and also all the hyperparameters of a bagging classifier, to control the ensemble itself. Instead of building a bagging-classifier and passing it into a decision-tree-classifier, you can just use the random-forest classifier class, which is more convenient and optimized for decision trees. Note that there is also a random-forest regressor for regression tasks.

The random-forest algorithm brings extra randomness into the model, when it is growing the trees. Instead of searching for the best feature while splitting a node, it searches for the best feature among a random subset of features. This process creates a wide diversity, which generally results in a better model.

Therefore when you are growing a tree in random forest, only a random subset of the features is considered for splitting a node. You can even make trees more random, by using random thresholds on top of it, for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

3.6 Limitations of Prior Art

Sentiment analysis of in the domain of micro-blogging is a relatively new research topic so there is still a lot of room for further research in this area. Decent amount of related prior work has been done on sentiment analysis of user reviews [x], documents, web blogs/articles and general phrase level sentiment analysis [16]. These differ from twitter mainly because of the limit of 140 characters per tweet which forces the user to express opinion compressed in very short text. The best results reached in sentiment classification use supervised learning techniques such as Naive Bayes and Support Vector Machines, but the manual labelling required for the supervised approach is very expensive. Some work has been done on unsupervised (e.g., [11] and [13]) and semi-supervised (e.g., [3] and [10]) approaches, and there is a lot of room of improvement. Various researchers testing new features and classification techniques often just compare their results to base-line performance. There is a need of proper and formal comparisons between these results arrived through different features and classification techniques in order to select the best features and most efficient classification techniques for particular applications.

3.7 Difficulty of sentiment analysis

Research shows that sentiment analysis is more difficult than traditional topic based text classification, despite the fact that the number of classes in sentiment analysis is less than the number of classes in topic-based classification (Pang and Lee, 2008). In sentiment analysis, the classes to which a piece of text is assigned are usually negative or positive. They can also be other binary classes or multivalued classes like classification into 'positive', 'negative' and 'neutral', but still they are less than the number of classes in topic-based classification. The main reason that sentiment analysis is more difficult than topic-based text classification is that topic-based classification can be done with the use of keywords while this does not work well in sentiment analysis (see Turney, 2002). Other reasons for difficulty are: sentiment can be expressed in subtle ways without any ostensible use of negative words; it is difficult to determine whether a given text is objective or subjective (there is always a ne-line between objective and subjective texts); it is difficult to determine the opinion holder (example, is it the opinion of the author or the opinion of the commenter); there are other factors such as dependency on domain

and on order of words (Pang and Lee, 2008). Other factors that make sentiment analysis di-cult are that it can be expressed with sarcasm, irony, and/or negation.

CHAPTER. 4

RESULT AND DISCUSSION

The flow diagram of our process which we have used is given in the figure 4.1 .In our experiment we had performed four operations which are discussed.

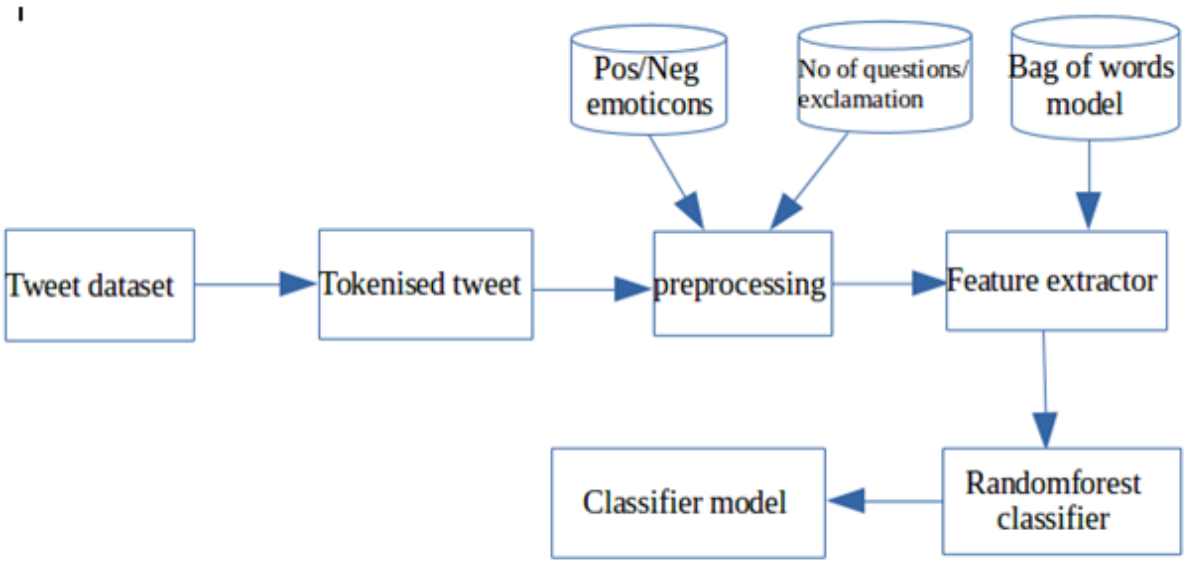

Fig.4.1 Flow Diagram of the process

4.1 Algorithm Design Steps

The complete process is carried out in steps given below

1.First we pickup a dataset from the internet.

2.We preprocessed the data to tokenisd the tweet.

3.After preprocessing we pass the preprocessed data to our feature extractor which extract the feature using Bag of Word model.

4.After this we pass the extacted feature to our classifier which is randomforest classifier to classify our model.

We took dataset of 9500 tweets from which we have used 5471 tweets to train our classifier and the rest to test the classifier. Firstly we perform pre-processing of our dataset using natural language processing in which we tokenise the tweet, tag the part of speech, collect all the stopwords etc,After the pre-processing we extract some extra features from the tweets like number of exclamation marks, number of question mark, number of positive emoticons and number of negative emoticons and pass it to the classifier which than classify the data. The result of experiments are discussed below.

4.2 Results

4.2.1 Sentiment type distribution in training set:

In the first step we classify the tweets without using any extra features. Results are shown in figure 4.2

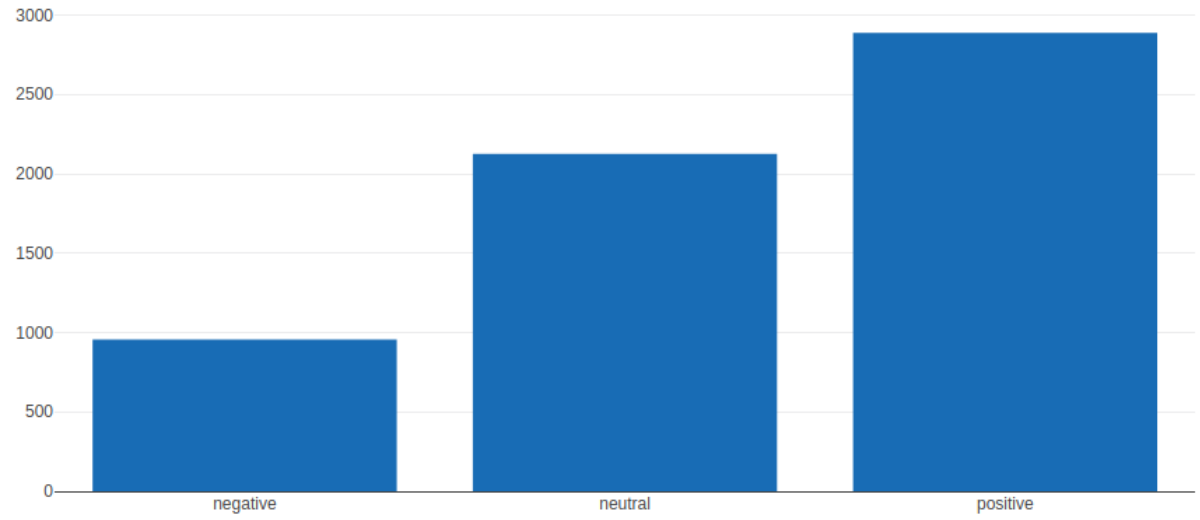

Fig. 4.2 Graph showing Sentiment distribution

Analysing the above graph we can conclude that out of 5900 tweets in training set:

- No of positive tweets are 2800 approx,
- No of negative tweets are 2200 approx.
- No of neutral tweets are 900 approx.

4.2.2 Top words in buid wordlist:

In this step we collected the information of most appeared words in the dataset which you can see in figure 4.3

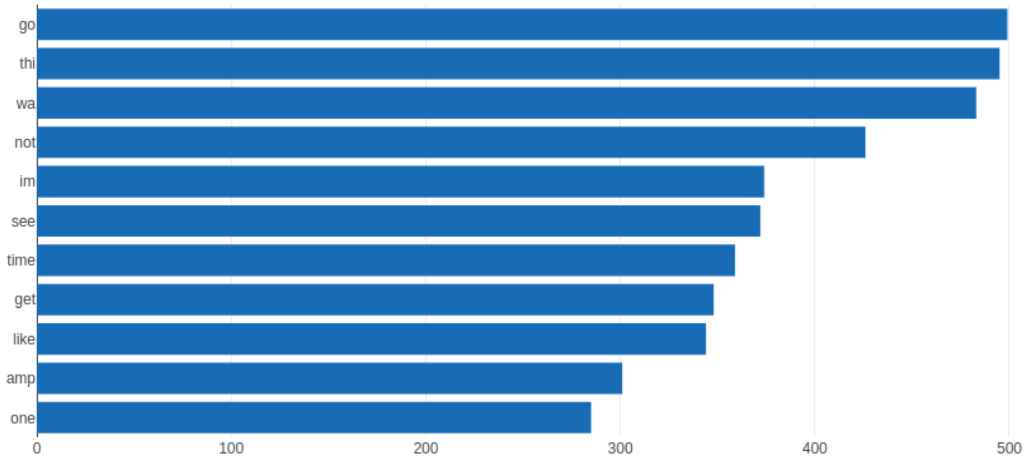

Fig. 4.3 Top words in build wordlist

The above graph shows the count of top words in the build wordlist and from it we can easily analyze that the word "go" is having the highest occurrence and that is nearly 500 in the given set of data

4.2.3 Sentiment distribution using extra features:

To improve the accuracy of the classifier we extract some extra features from the dataset the distribution on the basis of extra features is given below

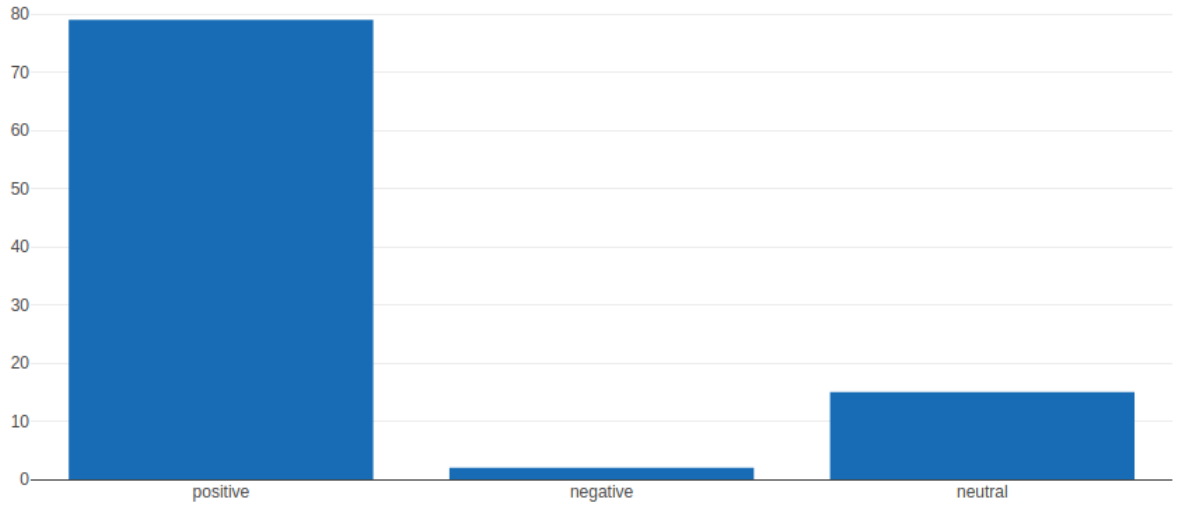

Fig. 4.4 Sentiment distribution using no of positive emoticons

The above graph shows the sentiment distribution using no. of positive emoticons and it shows the various positive, negative and neutral feeds.

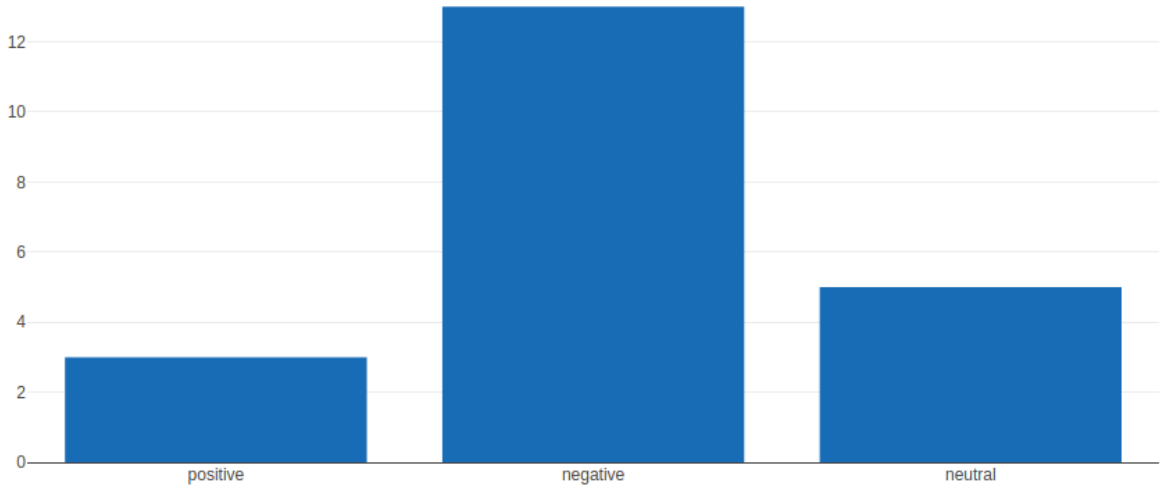

Fig. 4.5 Sentiment distribution using no of negative emoticons

The above graph shows the sentiment distribution using no. of negative emoticons and it shows the various positive, negative and neutral feeds.

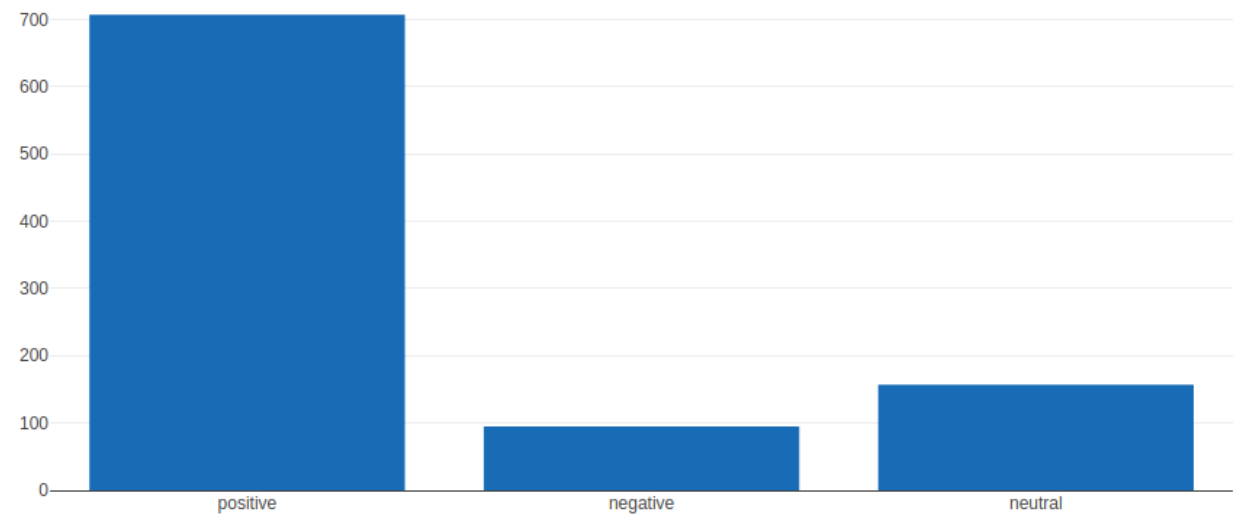

Fig. 4.6 Sentiment distribution using no of exclamations

The above graph shows the sentiment distribution using no. of exclamations and it shows the various positive, negative and neutral feeds.

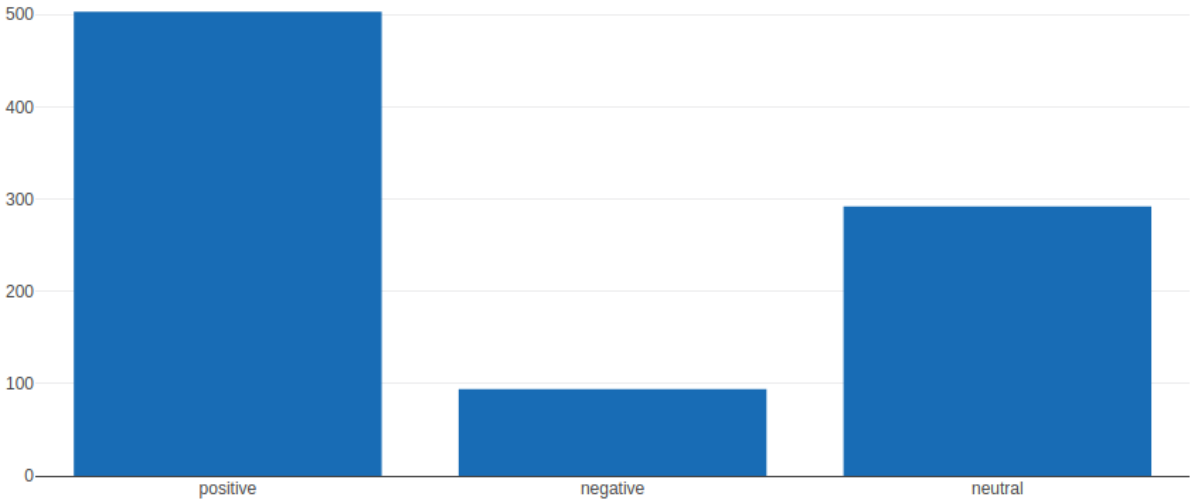

Fig. 4.7 Sentiment distribution using no of hashtags

The above graph shows the sentiment distribution using no. of hashtags and it shows the various positive, negative and neutral feeds.

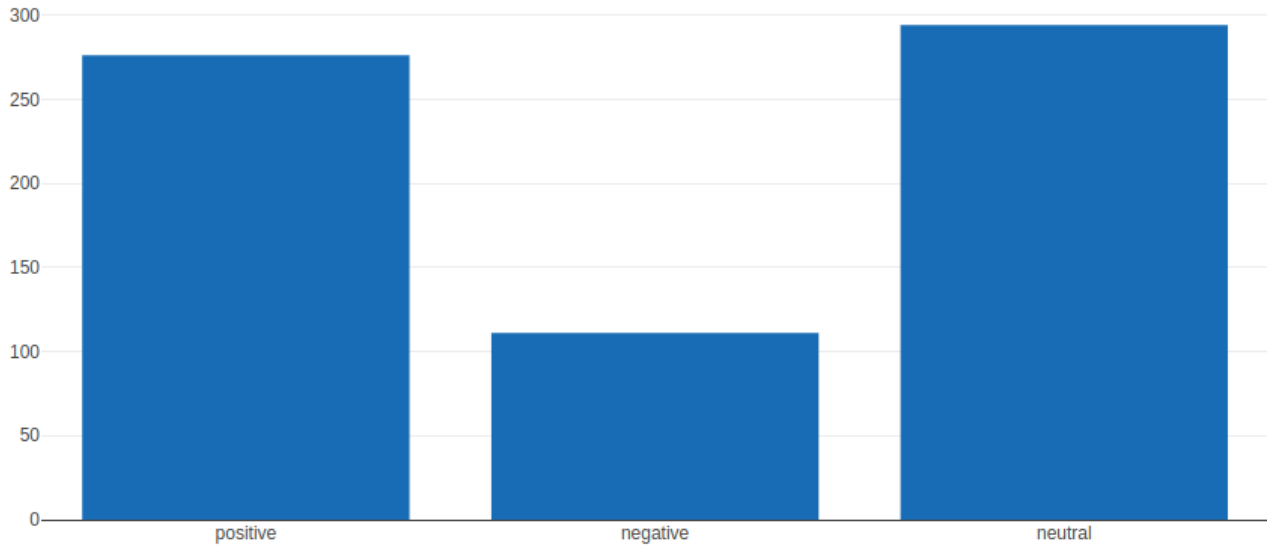

Fig. 4.8 Sentiment distribution using no of question-marks

The above graph shows the sentiment distribution using no. of question marks and it shows the various positive, negative and neutral feeds.

By using these extra feature we are able to increase the accuracy to 60%

```
Predicting time 3.6000521183013916s
=================== Results ===================
            Negative     Neutral     Positive
F1       [0.33595801 0.50823938 0.72674419]
Precision[0.53333333 0.51675485 0.66489362]
Recall   [0.24521073 0.5        0.80128205]
Accuracy 0.6035648432698217
================================================
================================================
```

Fig. 4.9 Accuracy using random forest classifier

4.2.4 Most important feature:

Using Xgb boost classifier we extact some important feature which are shown in figure 4.10

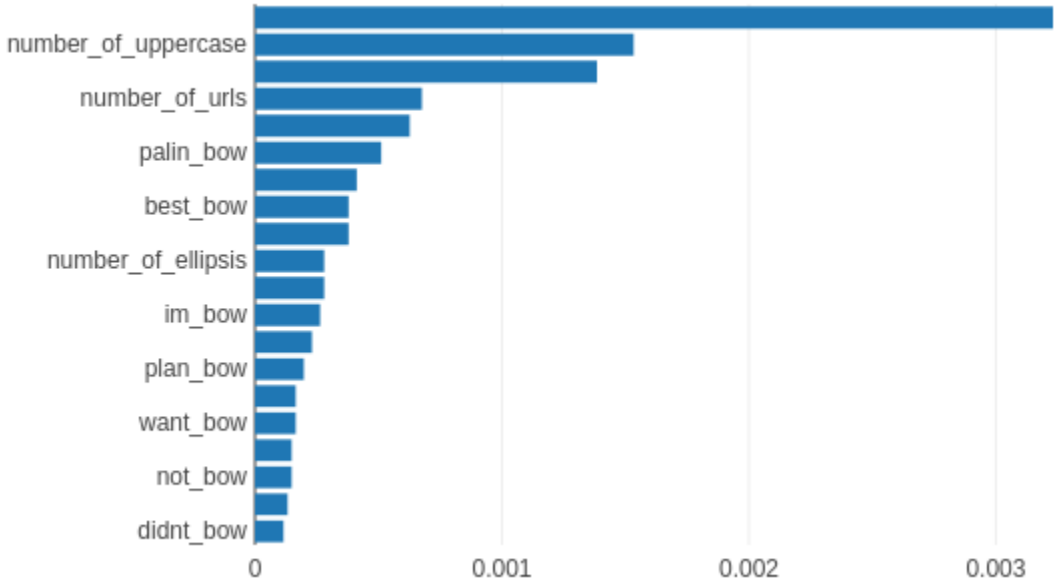

fig. 4.10 Most important features

CHAPTER. 5

CONCLUSION AND FUTURE SCOPE

The task of sentiment analysis, especially in the domain of micro-bloging, is still in the developing stage and far from complete. So we propose a couple of ideas which we feel are worth exploring in the future and may result in further improved performance.

Right now we have worked with only the very simplest models; we can improve those models by adding extra information like closeness of the word with a negation word. We could specify a window prior to the word (a window could for example be of 2 or 3 words) under consideration and the effect of negation may be incorporated into the model if it lies within that window. The closer the negation word is to the unigram word whose prior polarity is to be calculated, the more it should affect the polarity. For example if the negation is right next to the word, it may simply reverse the polarity of that word and farther the negation is from the word the more minimized ifs effect should be.

Apart from this, we are currently only focusing on unigrams and the effect of bigrams and trigrams may be explored. As reported in the literature review section when bigrams are used along with unigrams this usually enhances performance. However for bigrams and trigrams to be an effective feature we need a much more labeled data set than our meager tweets.

Right now we are exploring Parts of Speech separate from the unigram models, we can try to incorporate POS information within our unigram models in future. So say instead of calculating a single probability for each word like P(word | obj) we could instead have multiple probabilities for each according to the Part of Speech the word belongs to. For example we may have P(word | obj, verb), P(word | obj, noun) and P(word | obj, adjective). Pang et al. [5] used a somewhat similar approach and claims that appending POS information for every unigram results in no significant change in performance (with Naive Bayes performing slightly better and SVM having a slight decrease in performance), while there is a significant decrease in accuracy if only adjective unigrams are used as features. However these results are for classification of reviews and may be verified for sentiment analysis on micro blogging websites like Twitter.

One more feature we that is worth exploring is whether the information about relative position of word in a tweet has any effect on the performance of the classifier. Although Pang et al. explored a similar feature and reported negative results, their results were based on reviews which are very different from tweets and they worked on an extremely simple model. The problem with unequal classes is that the classifier tries to increase the overall accuracy of the system by increasing the accuracy of the majority class, even if that comes at the cost of decrease in accuracy of the minority classes. That is the very reason why we report significantly higher accuracies for objective class as opposed to positive or negative classes. To overcome this problem and have the classifier exhibit no bias towards any of the classes, it is necessary to label more data (tweets) so that all three of our classes are almost equal.

In this research we are focussing on general sentiment analysis. There is potential of work in the field of sentiment analysis with partially known context. For example we noticed that users generally use our website for specific types of keywords which can divided into a couple of distinct classes, namely: politics/politicians, celebrities, products/brands, sports/sportsmen, media/movies/music. So we can attempt to perform separate sentiment analysis on tweets that only belong to one of these classes (i.e. the training data would not be general but specific to one of these categories) and compare the results we get if we apply general sentiment analysis on it instead. Last but not the least, we can attempt to model human confidence in our system.

We could develop our custom cost function for coming up with optimized class boundaries such that highest weightage is given to those tweets in which all 5 labels agree and as the number of agreements start decreasing, so do the weights assigned. In this way the effects of human confidence can be visualized in sentiment analysis.

References

[1] Albert Biffet and Eibe Frank. Sentiment Knowledge Discovery in Twitter Streaming Data. Discovery Science, Lecture Notes in Computer Science, 2010, Volume 6332/2010, 1-15, DOI: 10.1007/978-3-642-16184-1_1

[2] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

[3] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.

[4] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner and Isabell M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2010.

[5] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002.

[6] Chenhao Tan, Lilian Lee, Jie Tang, Long Jiang, Ming Zhou and Ping Li. User Level Sentiment Analysis Incorporating Social Networks. In Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), 2011.

[7] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[8] Hatzivassiloglou, V., & McKeown, K.R.. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL, 2009.

[9] Johann Bollen, Alberto Pepe and Huina Mao. Modelling Public Mood and Emotion: Twitter Sentiment and socio-economic phenomena. In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[10] Luciano Barbosa and Junlan Feng. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In Proceedings of the international conference on Computational Linguistics (COLING), 2010.

[11] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002.

[12] Rudy Prabowo and Mike Thelwall. Sentiment Analysis: A Combined Approach. Journal of Infometrics, Volume 3, Issue 2, April 2009, Pages 143-157, 2009.

[13] Samuel Brody and Nicholas Diakopoulus. Cooooooooooooooollllllllllllll!!!!!!!!!!!!!! Using Word Lengthening to Detect Sentiment in Microblogs. In Proceedings of Empirical Methods on Natural Language Processing (EMNLP), 2011.

[14] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In Proceedings of International Conference on Computational Linguistics (ICCL), 2004.

[15] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.

[16] Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In the Annual Meeting of Association of Computational Linguistics: Human Language Technologies (ACL-HLT), 2005.

[17] Ian H. Witten, Eibe Frank & Mark A. Hall. Data Mining – Practical Machine Learning Tools and Techniques.

[18] Ricgard O. Duda, Peter E. Hart & David G. Stork: Pattern Classification.

[19] Steven Bird, Even Klein & Edward Loper. Natural Language Processing with Python.

[20] Ben Parr. Twitter Has 100 Million Monthly Active Users; 50% Log In Everyday https://mashable.com/2011/10/17/twitter-costolo-stats/#IHj1S.bGvGqo

[21] L. Barbosa and J. Feng. Robust sentiment detection on Twitter from biased and noisy data. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 3644. Association for Computational Linguistics, 2010.

[22] M. Bautin, L. Vijayarenu, and S. Skiena. International sentiment analysis for news and blogs. In Proceedings of the International Conference on Weblogs and Social Media (ICWSM), 2008

[23] K. Dave, S. Lawrence, and D.M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classication of product reviews. In Proceedings of the 12th international conference on World Wide Web, pages 519528. ACM, 2003. ISBN 1581136803.

[24] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. In Proceedings of the International Conference on Weblogs and Social Media (ICWSM). Citeseer, 2007

[25] V. Hatzivassiloglou and K.R. McKeown. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computa tional Linguistics, pages 174181. Association for Computational Linguistics, 1997.

[26] B. Liu, X. Li, W.S. Lee, and P.S. Yu. Text classication by labeling words. In PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, pages 425430. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[27] P. Melville, W. Gryc, and R.D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classication. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 12751284. ACM, 2009.

[28] J.C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou. Eectiveness of simple linguistic processing in automatic sentiment classication of product reviews. ADVANCES IN KNOWLEDGE ORGANIZATION, 9:4954, 2004. ISSN 0938- 5495.

[29] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of LREC 2010, 2010.

[30] B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2):1135, 2008. ISSN 1554-0669.

[31] T. Wilson, J. Wiebe, and P. Homann. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 347354. Association for Computational Linguistics, 2005.