PREDICTING ELECTION POPULARITY

OF A PERSON USING CROWD SENSING IN SOCIAL NETWORKS

Supervisor: Moin Mostakim

Conducted By:

Mashroora Nadi – 11101041

Syed Washfi Ahmad – 11101038

S.M. Saquib Rahman – 11101019

Declaration

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor

_____

Moin Mostakim

Signature of Author

_____

Mashroora Nadi

Signature of Author

_____

S.M.Saquib Rahman

Signature of Author

_____

Syed Washfi Ahmad

Acknowledgement

This work was suggested by Mr. Moin Mostakim, SECS Dept., BRAC University, as a graduation thesis. This is the work of S.M.Saquib Rahman, Mashroora Nadi and Syed Washfi Ahmad students of the SECS department of BRAC University, studying CSE and CS respectively starting from the year 2011.The document has been prepared as an effort to compile the knowledge obtained by us during these four years of education and produce a final thesis which innovatively addresses one of the issues of the current practical world. We tried to relate our work to the present days' revolution which is Microblogging. The idea of our work was proposed by Mr. Moin Mostakim, SECS Dept., BRAC University. The problem was to predicting a person's popularity in election through crowd sensing in social media. We needed to build a scrapper so that our corpus can be enriched. Twitter provided us with an API for the data scrapping. The training set for our Corpus was collected from Thesarus.com. Thanks to Mr. Moin Mostakim, SECS Dept., BRAC University for providing us with such opportunity.

Abstract

Predicting election popularity from social network data is an appealing research topic. This covers all aspects from data collection to data representation through data processing. Although social media may provide a glimpse on electoral outcomes current research does not provide strong evidence to support it can replace traditional polls. Data scrapping could help us with crawling the data and create a database regarding that statistics which can predict the winner. We propose that social networking sites can provide an “open” publish-subscribe infrastructure to sense crowd and efficiently predict an election result for a political party or a political leader. The possibility of winning for a candidate will be predicted by mining representative terms from the social media that people posted before the election or during campaign. Such systems like crowd sensing can cause benefit to both the voters and the nominees. We are working on Twittter as our social media.

Abbreviation

NLP= Natural Language Processing

HMM=Hidden Markov Model

CC = Coordinating conjunction

CD = Cardinal number

DT = Determiner

EX = Existential there

VBD=Verb, past tense

VBG=Verb, gerund or present participle

VBN=Verb, past participle

VBP=Verb, non-3rd person singular present

VBZ=Verb, 3rd person singular present

NN=Noun, singular or mass

NNS=Noun, plural

NNP=Proper noun, singular

NNPS=Proper noun, plural

PDT=Predeterminer

Table of contents

1. Introduction

Social network has introduced a new type of communication tool which can be coined by Microblogging. Twitter [1], Facebook, Tumbler are the leading websites where millions of messages are appearing daily. This has become a very popular means of communication in between the Internet users. People are sharing their lives, views, share opinions and thoughts on different topics and current issues through Microblogging. It can be defined as a blog but in a very precise and short form. People are now shifting from usual and traditional communication systems. The format being free, Microblogging is the most grossing topic of this generation. Users post about various events they participated, persons they like or hate products and services they use or express their political and religious views. The site that supports Microblogging has now become a valuable source of mass people's opinion and sentiments. This data from various websites are open and thus is a very effective source for social studies, product marketing and predicting a possible outcome.

Predicting election popularity from social network data is an appealing research topic. This covers all aspects from data collection to data representation through data processing. Although social media may provide a glimpse on electoral outcomes current research does not provide strong evidence to support it can replace traditional polls. Data scrapping could help us with crawling the data and create a database regarding that statistics which can predict the winner. We propose that social networking sites can provide an “open” publish-subscribe infrastructure to sense crowd and efficiently predict an election result for a political party or a political leader. The possibility of winning for a candidate will be predicted by mining representative terms from the social network's Microblogs that people posted before the election or during campaign. Such systems like crowd sensing can cause benefit to both the voters and the nominees. We are working on Tweeter as our social network. Tweeter uses a hast tag process which will be our key point for sensing the crowd.

Microblogging is growing everyday with its increasing audience and improving platforms. As a result these sources can be efficiently used for mining opinions and

1. www.twitter.com

analyze the public sentiment. For an example, a political party might be interested in these questions:

- What do people think about our party?
- How Positive (or negative) are people about our campaign?
- How would people prefer our agendas to be like?

Social organizations can ask about recent issues or problems, a manufacturer can ask about the quality of their product and how it can be improved. Microblogging services can be a very useful aid for this situations as the Microbloggers use this service every day and let the world know about their liking and disliking in various aspect of their and others life

In our paper, we are going to study how Microblogging can help in predicting an election outcome using Twitter hashtag. After studying some papers we decided to use Microblogging and more precisely Twitter for our study. The Reasons are below.

- Different people use Microblogging platforms in order to post their opinion about different topics. So this is a valuable source of people's opinions.
- Twitter's number text posts are enormous. The collected corpus can be arbitrarily large and will increase every day.
- It is possible to collect text posts from regular users to celebrities, company representatives, politicians, and even country presidents. So different cultural, political and religious views can be collected.

1.1 Motivation

Microblogging sites are getting popular day by day. A huge number of people is joining in this community day by day and started to become active Microblogger. They have opinion on almost every aspect. Any person or a company can understand their popularity and public demand from a huge amount of data over there. This can be a huge source of peoples' opinion. From that thought, we figured out this can work as a dataset to find a celebrity's popularity. Being a huge source of data of people's opinion it can provide opinion about that certain celebrity also.

So, we figured out this could be essential application of analyzing sentiment. If we can analyze sentiment behind those opinions we can find out their popularity among people.

For celebrities or political persons it must be hard for them to go through all those micro blogs and analyze sentiment of each of those. Our project can help doing that for them. It saves time and energy. They can also figure out what to change and by how they can increase popularity.

So, if this project gets the much highly appreciated as Google [3] or Facebook [2] or other trusted sources, people would trust and use it to predict their popularity.

2. www.facebook.com
3. www.google.com

1.2 Thesis Outline

Section 1 deals with the I*ntroduction* and *Motivation* in. Section 2 deals with the *Background Researches*. We have included the proposed system in section 2.

Section 3 is comprised of the *Corpus*..In section 4 we have provided the *Technical Overview* where the technical overviews of *Hidden Markov Model ,Machine Learning* has been provided.

Section 5 includes *POS Tagging* which is followed by the *Analysis of the Corpus.*

Analysis of the Corpus is in section 6 and *Training the Classifier* in section 7.Also, it is followed by *Testing the Accuracy* which is in section 8.

Result in section 9,*Future Work* is in section 10 and *References* in section 11

2. Background research

2.1 Previous works

(Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining)

This article deals with micro blogging, the most used communication tool in current world. The way adopted by thousands of users to share and express opinions on different aspects of incidents. This makes the micro blogging websites a rich resource to get opinion by data mining and analyze the sentiment hidden inside it. This paper chose twitter to deal with micro blogging to get opinion and analyze sentiments through it. This article showed the procedure of collecting corpus for sentiment analyzing and opinion mining. The contributions they made were:

1. They made a method to collect the corpus from which they could get the opinion.

2. Language is complex. Directly analyzing it using just a fixed structure does not give accurate result. So, here the writers performed linguistic analysis from the corpus they found

3. By using that they explained the phenomena they discovered. According to this article, sentiments are of three types. They are positive, negative and neutral. They used this to make a sentiment classifier using the corpus.

4. They conducted experimental evaluations to prove the efficiency of their article. Even though many other languages could be chosen but they chose English language to analyze and take opinion from. The purpose of choosing microblogging to analyze was also given in their article. They divided the text into two parts.

1. Subjective;

2. Objective

For the classifier they used Na'ıve Bayes classifier and also POS tagging. Although, POS (Parts of speech) is dependent on the n-grams, they made an assumption of n-gram features and POS information to simplify the calculation. They also increased the accuracy by using the formula of Shannon Entropy. This article presented the procedure of corpus collection and analyzing the sentiment of opinion given in it. They use tree Tagger and POS tagging and observed the distinction between distribution among positive, negative and neutral sets.

(Chowdhury, S., & Chowdhury, W. (n.d.). Performing Sentiment Analysis in Bangla Microblog Posts)

These days a lot of work on sentiment analysis is going on. But they are all limited in analyzing English language. Micro blogging being the recent valuable source for publishing a huge amount of information where users express their views and perspectives the language Bangla not being analyzed where making it falling behind. The writers found out that in recent days just one or two user messages on a particular product or service and making a decision on that cannot bring fruitful result. Analyzing those microblogging can help getting many user messages on a certain issue or product which might be laborious for human to do by hand. So, they chose a machine to do that. In this paper from "tweets" of twitter they intended to extract sentiment and polarity automatically from the user opinion. They made a binary classification of sentiments. They chose it to be subjective or objective. They used a semi-supervised bootstrapping approach for the development of the training corpus. From the previous works done on this topic for English language for classification they decided to choose SVM and MaxEnt to do the analysis to compare with. All the experiments they performed while doing this was by using NLTK Python Toolkit.

At first the collected the raw tweet data from Twitter API and then they did the preprocessing by three steps. They are:

1. Tokenization

2. Normalization

3. POS tagging

Now, they found the processed data. They classified this in unlabeled test data and unlabeled training data. With the training data by applying bootstrapping process they sent it to sentiment classifier which helped deciding the positivity or negativity of the sentence. The methodology they followed while doing this was:

1. Dataset

2. Preprocessing

3. Bangla Sentiment Lexicon Construction

4. Training Set Construction

5. Feature Extraction

6. Illustration with sample tweet

By following this methodology they decided the experimental results and evaluated it.

(Tumasjan, A., Sprenger, T., Sandner, P., & Welpe, I. (n.d.). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment)

Micro blogging done inside twitter has become people's daily activity. A lot of messages on several topics are expressed via it every day. This paper took German federal election as the context to analyze if twitter used as a forum for political deliberation and these online messages written inside 140 characters can work as a mirror to decide political sentiment. They made the aim of their article study as threefold.

1. They examined that if Twitter can work as a vehicle for online political deliberation. To do this they looked at peoples use micro blogging for the purpose of exchanging information about the current issues going on inside or outside politics.

2. They evaluated the collected data if that actually works in a meaningful way to decide any sentiment going on that issue they found from the data.

3. They used the data to predict the popularity of the parties they are working upon and they analyzed if this activity can help predicting the popularity and also if it can decide the collations in the real world.

Inside the methodology they used few dimensions to decide the political sentiment. Among those Future orientation, past orientation, sadness, anxiety, positive emotions, negative emotions, anger, achievement, work, tentativeness, certainty and money were prominent. They followed the methodology by using Yu, Kaufmann, and Diermeier, they also parsed all tweets which were published during the timespan they chose to analyze into one sample text so that it can be evaluated by LIWC. As result of their evaluation they found twitter:

1. as a platform for political deliberation

2. as a reflection of political sentiment

3. as a predictor of the election result

In contrast to Sunstein according to whom the sphere of micro blogging cannot be a source to serve information as it lacks a pricing system, the writers found the information inside the blogs of Twitter can definitely be used in a meaningful way.

2.2 Review of Sentiment Analysis

Sentiment analysis is a combination of natural language processing, text analysis and computational linguistics. This is to extract and analyze the sentiment of a content to determine the positivity, negativity or neutral behavior of the content. This can be determined from a sentence or a whole content. Usually positivity is decided by the number of positive keywords of a sentence and negativity by the number of negative keywords. The comparison and winning among the number of each type of contents

decides the positivity and negativity of the whole content. This analyzing is done to bring out the opinion expressed inside the content and bring out the sentiment inside of that. An opinion is an expression that consists of two key components:

1. A topic

2. A sentiment on the topic.

So, “The new food place is amazing”, here food place is the topic and by using the word “amazing” it makes it positive. But, the two types of opinions can co-appear. The sentence “I like X-party but I think they are not likely to win considering the current position” has both positive and negative opinion. The first approach shows the positivity towards X-party by using the word “like” but “unlikely” valence towards the event of “X-party wins”. In order to accurately identify and analyze each type of opinion, different approaches are desirable. Analysis can be done by breaking the whole content into sentences or taking the content as a whole. At times different sentences create an expression which by one sentence could not be done. Here the combined form of sentences creates expression. Now, language cannot be understood by a structure. In fact, it would be too naive to expect the sentiment of the language can always be accurately examined by a machine or an algorithm. Currently four main factors are there to help us from not depending blindly on tools for sentiment analysis:

1. Context: “That person does a great job at stealing money”, in spite of having the word “great”, this cannot be a positive sentence. Depending on context the value of sentence changed.

2. Sentiment Ambiguity: “Please let me know if you find some good articles”, this sentence does not express any sentiment in spite of having the word “good” so this cannot decide the value. In some other sentences, even after not having any sentiment word sentiment can be expressed. For example, “This costs a lot of money”. In spite of not having any sentiment word this expressed a negative sentence.

3. Sarcasm: Sarcasm changes the whole expression of the sentence. For example, “the weather forecast happily informs the news of no rain during this heat”. Even the word “happily” could not make it a positive sentence because of sarcasm.

4. Language: Slang, dialects and language variations can completely change the meaning of the sentence based on context, tone and language. Automated sentiment analysis will improve more over time. Language cannot be analyzed in complete accurate form but by thinking carefully about its use benefits must come.

2.3 Proposed System

The basic purpose of this project would be to determine the popularity of some certain party in an election. In current world, micro blogging took a huge hold in expressing people’s thoughts. Among those 140 characters which is the limit of micro blogging, people express their opinions regarding various issues. Starting from current good music to current political situation every discussion has become a part of this micro blogging. Seeing this, we chose twitter to help determine the prediction from where we could take peoples opinion and decide what party they r supporting or what sort of opinions they are having regarding that certain party. Because from those opinions even if we cannot decide which party they are supporting, we can at least know what they are thinking of that certain party and if they are expecting them to be good or bad and the sentiment they have regarding that party.

We divided our work into four parts. At first we collect the corpus. By doing web crawling we got the opinions we wanted and scrapped those data. We presented the method of doing that. At the beginning we collected all the data regarding the issue we want to predict. Our method allows collecting both positive and negative sentiments of that certain issue. It allows scrapping large data as well.

Now, language is complex. The tweets we collected acted as a data source of further work. We analyzed the data we collected. Those data helped us to find peoples opinion regarding that certain party. From the opinion we got we had to decide the sentiment

hidden inside of it. We tried to found if from those tweets, people have shown any support or opposition to that certain party.

Thirdly we analyzed those sentiments. We tried to divide it into positive and negative opinions. We built a classifier to decide that. The amount of words inside the corpora and in proportion to that the amount of positive or negative words helps deciding the prediction.

And after everything is done by using the classifier we predict the popularity of certain party.

3. Corpus

For predicting election from microblogs, it is needed to gather a huge amount of data as corpus so that the data mining can be done and predict the outcome. In this context this data is collected from Twitter. Now, twitter is not an open source microblogging site and requires to log in with a valid ID and password to get access. To get access and gather our required corpus it was needed to get the Twitter API for python which requires an authorization key. The key was provided by Twitter. Now as the Twitter API was ready, the python based scrapper that we constructed now collect corpus from the microblogs.

3.1. Hashtag

A key was selected to gather corpora. Twitter, Facebook and other well-known microblogging services have introduced a new kind of label known as Hashtag. A hashtag is a type of label or metadata tag which enables the users to share and find a Microblog, post, image or messages with a specified theme and content has been used on social network and microblogging services providing a great amount of accessibility to the content or topic. By placing the hash character (#) before the topic or theme which is a usually a word or unspaced phrase in the beginning, middle or at the very end of a main text of a message, comment, post or caption users create and use hashtags. E.g. #food, #love, #RoarOfTigers, #BangladeshCricetTeamForTheWin etc.

3.2. Gathering the Corpora

In Twitter people use Hashs (#) before a relevant keyword or phrase (no spaces) in their Tweet as Hashtags to categorize their Tweets. It also gives the advantage of showing the Tweet more easily in Twitter Search. Also Searching for that hashtag will categorize and present each message that has been tagged with it. Clicking on a hashtagged word in any message also shows all other Tweets marked with that keyword. Hashtag is the smartest and simplest key to choose for any kind of corpora collection for data mining from a microblogging service. So with the help of hashtag we gathered corpora to train our

classifier. The two types of collected corpora will be used to train a classifier to recognize positive and negative sentiments. In order to collect a corpus of objective posts, we retrieved text messages from Twitter accounts of popular political persons, fan pages and hate pages two get a thorough learning.

As by the rules of the microblogging platform, each message cannot exceed 140 characters. That is a reason for which it is usually composed of a single sentence. Therefore, we assume that positive and negative word that exists within a message represents an emotion for the whole message and all the words of the message are related to this emotion.

In our research, we have used English language as it is the most commonly used in any microblogging services especially in Twitter. However, our method can easily be modified to adapt to other languages' posts since Twitter API allows specifying the language of the retrieved posts.

Data filtering was also done because now a days' microblogging system has changed the way of communication and people uses many abbreviations and signs in their text. The abbreviations were kept and later handled with Thesaurus but the unknown signs were discarded. Also some of the microblogs were discarded due to lack of understandability of language as they use mixture of languages. Modified and filtered Tweets were stored in a database from where later on the classifier will be trained.

4. Technical overview

4.1 Hidden Markov Model:

Hidden Markov model is an application for Bayesian network. In the simplest dynamic Bayesian network a Hidden Markov model can be represented. It is actually a statistical Markov model. The only exception is the Hidden Markov model is percept to be an unobserved states Markov process. It consists of some hidden states. To acquire the understanding of Hidden Markov model, one of the vital models of Markov Model which is Markov Chain is discussed from where the Hidden Markov Model is derived.

4.1.1. Markov Chain

Markov chain is the simplest Markov model. The modeling is done by the change of random variables which changes through time. These changes are recognized as states. The Markov chain suggests that the distribution of states of variable is dependent on the distribution of previous states. Markov model allows a sequence of random variables visiting a set of states. an important element is known by Transition probability which specifies the probability of transiting of the variables from one state to the other. Markov model's assumption is that in the process of transition the next state of transition depends only on the current state and independent of previous history. So only observing the current state the next decision is made. It needs to possess a property which is characterized as ""memoryless" which is required.”Memoryless" refers to the probability distribution of the next state is only dependent on the current state, not on the sequence of events that preceded it. This specific kind of "memorylessness" is called the Markov property. Markov chains have many applications as statistical models of real-world processes.

For a formal definition A Markov chain is given the present state of a probabilistic model, the future and past states are independent. If there is a sequence of random variables X1, X2, X3 ... with the Markov property,

Pr (Xn+1 = x | X1 = x1, X2 = x2,...., Xn = xn) = Pr (Xn+1 = x | Xn = xn).

If both conditional Probabilities are well defined then,

Pr(X1 = x1 ... Xn = xn) > 0.

4.1.2. Hidden Markov Model

A Hidden Markov model partially observable rather than Markov Chain which are fully observable. In Hidden Markov model "Hidden" means the exact sequence of states that generated the observation are hidden. This simple Bayesian network based model uses simplest dynamic Bayesian network for statistical prediction. It is one of the most popular models for sequential statistics. It is widely used because it is simple enough to set the parameter and also rich enough to handle real world problems.

4.1.3. Formal Definition of a Hidden Markov Model

In a Hidden Markov model we have some random variables,

Z1… Zn ϵ (1… m) [some discrete random variables in some finite set]

And some hidden variables

X1… Xn ϵ X [discrete, R, R^d]

This random variables respects the graph,

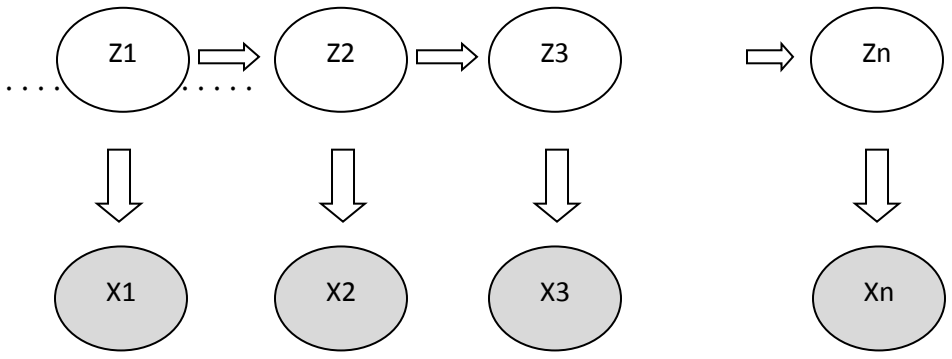

Fig: Trellis Diagram for HMM

It is also known as Trellis diagram. Here Zs are hidden variables and Xs are observed variable. In a Markov chain there only exists observed variables. But here in Hidden Markov model there exists a set of hidden variables.

This Model leads to the equation,

$$P(X_1, \ldots, X_n, Z_1, \ldots, Z_n) = P(Z_1)\, P(X_1|Z_1) \prod_{k=2}^{n} P(Z_k|Z_{k-1})$$

So an HMM consists of

- An *alphabet* $\sum$ which is a set of characters that can be emitted by the model. Various examples of an alphabet include the case of the dishonest casino model, in which the alphabet is the set {1, 2, 3, 4, 5, 6}, a model for coin tosses where the alphabet would be {H, T}, and for a genomic sequence, in which the alphabet would be {A, T, C, G}.
- A *finite set of states* Q that describe interesting properties about the system. In the case of the dishonest casino model, we have the fair and loaded states.
- *Transition probabilities* between any two states, which will be denoted by aij, the transition probability between state i and state j at any point in the sequence. Note that these probabilities do not change with time and stay constant throughout. Furthermore, because these probabilities form a probability distribution, the sum of all transition probabilities from a state i to any other state in the model will be 1; this means that we have to either stay in the current state or transition to another state at each point in the model.
- *Start probabilities* which denote the probability of starting at a given state in the first time point.
- *Emission probabilities* within each state. At every time point we emit a letter, and each state has its own set of emission probabilities, which denote the probability that each letter in the alphabet is emitted when the model is in that state. These probabilities form a distribution of the letters in the alphabet and therefore must also sum up to 1.

4.2 Machine learning

Machine learning involves the study of algorithms and procedures that can learn from a collection of data. Instead of programming some strict instructions machine learning uses effective predictions and decision for building a model from provided sample input. A branch of theoretical computer science is known as computational learning theory where learning algorithm and their performance are measured. A learning theory not always confirms guarantees of the performance of algorithm because of the finite characteristics of training sets and uncertainty of future. On the contrary, probabilistic bounds on the performance are quite common. One way of quantification of generalizing error is the bias–variance decomposition.

4.2.1. Naive Bayes

Being one of the most efficient and effective inductive learning algorithms for machine learning and data mining, Naive Bayes is well known for its competitive performance in classification is surprising. This happens because the conditional independence assumption is rarely true in real world applications.

A set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features is known as Naive Bayes method. Given a class variable and a dependent feature vector through, Bayes' theorem states the following relationship:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Using the naive independence assumption that,

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$$

Naive Bayes classifiers have worked quite well in many real-world situations, In spite of their apparently over-simplified assumptions. They are famously document classification

and spam filtering. A small amount of training data to estimate the necessary parameters are required which will be provided with our corpus.

Compared to more sophisticated methods, Naive Bayes learners and classifiers can be extremely fast. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution. This in turn helps to alleviate problems stemming from the curse of dimensionality.

5. POS Tagging

Corresponding to particular parts of speech, the method of tagging or making up a word in a text is known as POS Tagging. It also marks up the definition of a word and the relationship of words with related as well as adjacent words in a sentence or in a paragraph. The text and sentences are often associated to a specific corpus. POS tagging is now done in the context of computational linguistics. It uses a set of algorithms which is associated with discrete terms. They also consist of hidden parts of speech, in accordance with a set of descriptive tags. POS-tagging algorithms fall into two distinctive groups: rule-based and stochastic.

We are interested in a difference of tags distributions between sets of texts (positive, negative, and neutral). To perform a pair wise comparison of tags distributions, we calculated the following value for each tag and two sets (i.e. positive and negative posts).

$$P_{1,2}^{T} = \frac{N_1^T - N_2^T}{N_1^T + N_2^T}$$

Where NT 1 and N2 T are numbers of tag T occurrences in the first and second sets respectively.

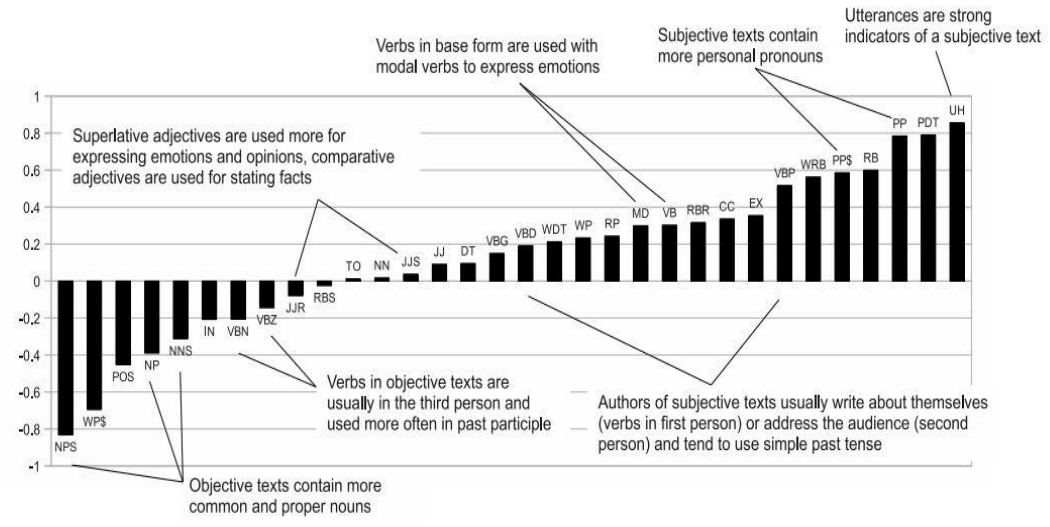

Fig: P^T Values for Subjective and objective

After plotting the equation, the values of P^T across all the tags where set 1 is a subjective set which is a mixture of the positive and the negative sets. The set 2 is an neutral objective set.

These sets are shown in figure 2. From the graph we can observe that POS tags are not distributed evenly in two sets, and therefore can be used as indicators of a set. Next, The observed Phenomena is explained. It can be observed that objective texts tend to contain more common and proper nouns (NPS, NP, NNS), while authors of subjective texts use more often personal pronouns (PP, PP$). Authors of subjective texts usually describe themselves as a first person or address the audience as a second person (VBP), while verbs in objective texts are usually in the third person (VBZ). As for the tense, subjective texts tend to use simple past tense (VBD) instead of the past participle (VBN). Also a base form of verbs (VB) is used often in subjective texts, which is explained by the frequent use of modal verbs (MD). In the graph, we see that superlative adjectives (JJS) are used more often for expressing emotions and opinions, and comparative adjectives (JJR) are used for stating facts and providing information. Adverbs (RB) are mostly used in subjective texts to give an emotional color to a verb. Figure 3 shows values of P T for negative and positive sets. As we see from the graph, a positive set has a prevailing number of possessive wh - pronoun 'whose' (WH$), which is unexpected. However, if we look in the corpus, we discover that Twitter users tend to use 'whose' as a slang version of 'who is'. For example: I have some Bangladesh-Pakistan match tickets :) whose ready for the Bangladesh?? Another indicator of a positive text is superlative adverbs (RBS), such as "most" and "best". Positive texts are also characterized by the use

of possessive ending (POS). As opposite to the positive set, the negative set contains more often verbs in the past tense (VBN, VBD), because many authors express their negative sentiments about their loss or disappointment. Here is an example of the most frequent verbs: “missed”, “bored”, “gone”, “lost”, “stuck”, “taken”. We have compared distributions of POS-tags in two parts of the same sets (e.g. a half of the positive set with another half of the positive set). The proximity of the obtained distributions allows us to conclude on the homogeneity of the corpus. (Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining)

6. Analysis of the corpus

The system we proposed to achieve for that the very first work we had to do was creating the scrapper. We had to use twitter API in order to do that. We searched through it. The data we wanted to collect was through hash tag. For that we crawled through the page by web crawler. We inserted a hash tag in our machine which we created and found every tweet related to that tag scrapped those data by using our scrapper machine. Now these data would work as our source of data or corpora.

Here comes the work of parser. Here we did HTML parsing to parse. We went into thesaurus.com and parsed words from that website. We set a value for every word. In order to creating the database while searching for positive words we set positive value of it manually. So the words we get from web, related to the word we searched for, all of those get a positive value and get inserted to our database. We did the same with negative words by putting negative values to it. This is how we enriched our database and created our knowledge base.

Now in every tweet we find the value of that tweet comparing to our database. The words the tweet contains, for positive words it gets a count of +1 and for negative words -1. So after getting the corpora we find the summation of count. If the count remains greater than we consider the sentiment as positive and for lesser than one we considered it to be negative.

7. Training the Classifier

7.1 Feature extraction

The dataset we have collected from Twitter is used to extract the features, which was used to train our sentiment classifier. We have used the presence of an n-gram as our binary feature, while for general information retrieval purposes, the frequency of a keyword's occurrence is a more suitable feature, since the overall sentiment may not necessarily be indicated through the repeated use of keywords. Pang et al. have obtained better results by using a term presence rather than its frequency (Pang et al., 2002). We have experimented with bigrams. Pang et al. (Pang et al., 2002) reported that unigrams outperform bigrams when performing the sentiment classification of movie reviews, and Dave et al. (Dave et al., 2003) have obtained contrary results: bigrams and trigrams worked better for the product-review polarity classification. We have tried to determine the best features for the Twitter data. On one hand high-order n-grams, such as bigrams, should better capture patterns of sentiments expressions. On the other hand, unigrams should provide a good coverage of the data. (Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining)

7.2 Classifier

We have built a sentiment classifier using the Naïve Bayes classifier. Naïve Bayes classifier is based on Bayes' theorem.

$$P(s|M) = \frac{P(s).P(M|s)}{P(M)} \text{ … … … … … … … ….(1)}$$

Where s is a sentiment, M is a twitter message. We can simplify the equation because of the number of dataset. So, we simplify the equation,

$$P(s|M) = \frac{P(M|s)}{P(M)} \text{ … … … … … … … … … (2)}$$

$$P(s|M) \sim P(M|s) \text{… … … … … … … (3)}$$

We train Bayes classifier, which uses features like presence of n-grams. N-gram based classifier uses the presence of n-gram in the different set of texts to calculate the posterior probability. We make an assumption of conditional dependence of n-grams to simplify the equation.

$$P(s|M) \sim P(G|s) \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where G is a set of n-grams representing the message. We have assumed the n-grams are conditionally independent.

$$P(G|s) = \prod_{g \in G} P(G|s) \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Finally, we have calculated the log-likelihood of each sentiment.

The algorithm for our Naïve Bayes classifier is:

1. $N\_grams(S) \leftarrow \{words | \forall \{words\} \in \{valid\ words\}$
2. W_SC ← for bigram(S)
3. Srt(W_SC) ← $\{\forall_{words} \in (Valid\ words)\}$
4. $T_s \leftarrow \{S_1, S_2, \ldots, S_N\}$
5. Classifier ← NaiveBayesClassifier(Probdist, TestProbdist)
6. for s in TestSample {

   P_i ←Classifer.classify(s); P_i = Probability of i

   t_i +=1; where t is the total number of dataset

   }

7. For Aggregated Probability:
8. Sort(P_i):
9. *for i=1 to i=N* {

   $P_i * \leftarrow \sum_{i=1}^{N} (P_i * \frac{t_i}{N})$

   }

So, the new equation for aggregated probability which we have derived to add an extra weight for comparison for N number will be,

$$P_i *= \sum_{i=1}^{N} (P_i * \frac{t_i}{N}) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

8. Testing Accuracy

We have tested our classifier on real sets of tweets. We have used our own collected corpora for that. We have computed the accuracy ((Manning and Sch¨utze, 1999)) by this equation:

$$\text{Accuracy} = \frac{\text{N(correct classifications)}}{(all\ classifications)}$$

9. Result

We have tested the impact of an n-gram order on the classifier's performance. The results of this comparison are presented in Figure-1. As we see from the graph, the best performance is achieved when using bigrams. We explain it as bigrams provide a good balance between a coverage (unigrams) and an ability to capture the sentiment expression patterns (trigrams).

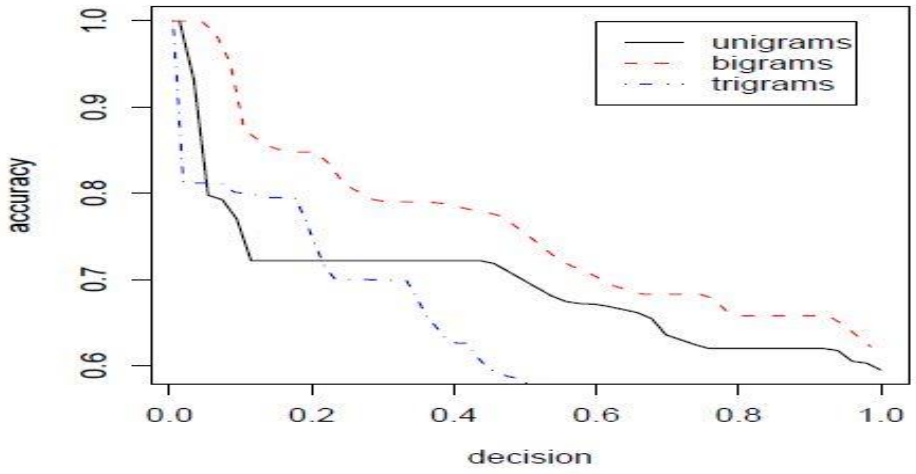

Figure-01

Figure-02, represents the positive probability vs. number of tweet count which shows that the increase in dataset changes the amount of probability.

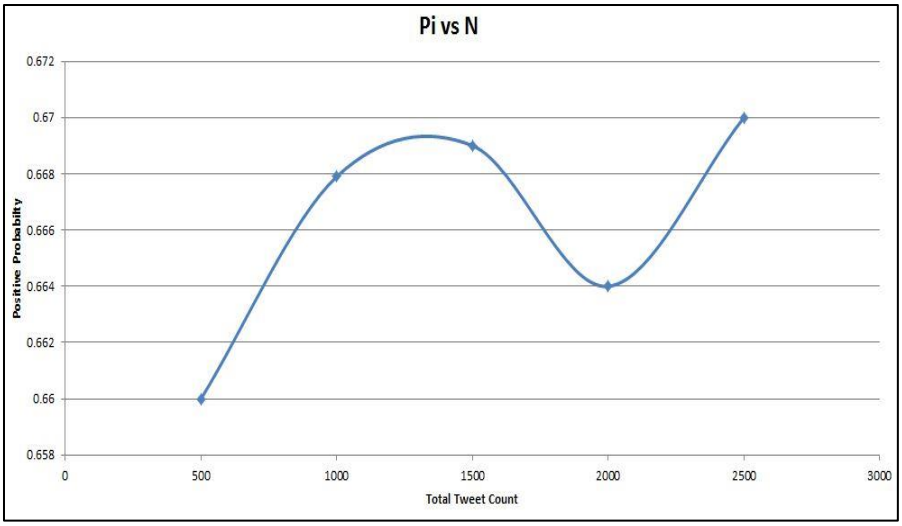

Figure-02

Figure-03 shows the positive probability vs. the total tweets of that individual. If the positive probability of a person increases then it puts an impact on the graph.

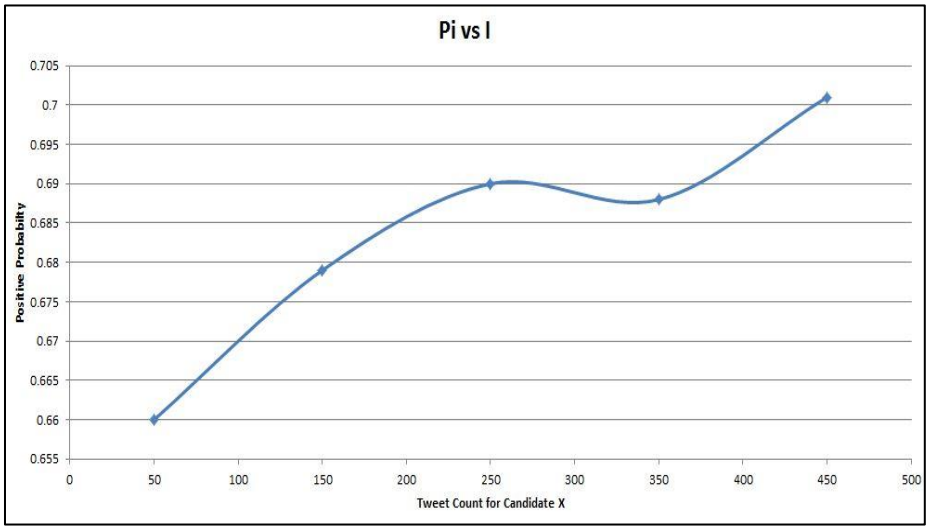

Figure-03

10. Future Work

Microblogging is a demanding prospect in todays technology based virtual society. Everyday more and more people are getting involved with Microblogging sites; the more and more data is available on any aspects and themes. So anyone can express their opinion and suggestion through a microblogging sites. It may be associated with culture, religion, politics, demography, sports etc. So people from different fields can be benefited by the huge data for the purpose of assessment. People may evaluate their popularity theoretically, but practically it is impossible to go through this much of data by their own. Our crowd sensing system, if modified can help with this situation. We have planned some future works with purpose of making our system a universal one. It can be an online application where people can search for their own popularity through the web as well as others. With so much incoming data our Corpus will be richer and more accurate. Our vision is to make our system a trusted one like Facebook, Twitter, Wikipedia or Google where our site can certify one person's popularity and can also present a sorted list with popular people on the same category. We have only selected English as our analysis language. In future we are going to include more language and also enable mixed language to include as Corpus and analyze the sentiment. The list of our future work,

- Universalization of the system to predict the popularity of a person.
- Creating a Celebrity hit list.
- Online application for determining popularity throughout the web.
- Multi-language support.
- Product review.

11. References:

1. G. Adda, J. Mariani, J. Lecomte, P. Paroubek, and M. Rajman.1998. The GRACE French part-of-speech tagging evaluation task. In A. Rubio, N. Gallardo, R. Castro, and A. Tejada, editors, LREC, volume I, pages 433–441,Granada, May.
2. Ethem Alpaydin. 2004. Introduction to Machine Learning (Adaptive Computation and Machine Learning). The MIT Press
3. Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining
4. Tumasjan, A., Sprenger, T., Sandner, P., & Welpe, I. (n.d.). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment
5. Chowdhury, S., & Chowdhury, W. (n.d.). Performing Sentiment Analysis in Bangla Microblog Posts
6. S. Asur, B. A. Huberman,(2010) Predicting the Future With Social Media referencing
7. J. Zhang, Z. Cai, Y. Gan, B. Zhang, L. He,(2007) Prediction Algorithms for User Actions
8. B. Pan, Y. Zheng, D. Wilkie, C. Shahab, (2013)Crowd Sensing of Traffic Anomalies based on Human Mobility and Social Media
9. P.T. Metaxas,(2011) How (Not) To Predict Elections
10. M. Demirbas, M.A. Bayir, C.G. Akcora, Y.S. Yilmaz, H. Ferhatosmanoglu,(2010) Sourced Sensing and Collaboration Using Twitter