A Review of Techniques for Sentiment Analysis Of Twitter Data

Sagar Bhuta
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India.
sagarb311@gmail.com

Avit Doshi
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India.
avit.doshi@gmail.com

Uchit Doshi
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India.
uchit.doshi@gmail.com

Meera Narvekar
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India.
meera.narvekar@djsce.ac.in

***Abstract*—There has been a rapid increase in the use of social networking websites in the last few years. People most conveniently express their views and opinions on a wide array of topics via such websites. Sentiment analysis of such data which comprises of people's views is very important in order to gauge public opinion on a particular topic of interest. This paper reviews a number of techniques, both lexicon-based approaches as well as learning based methods that can be used for sentiment analysis of text. In order to adapt these techniques for sentiment analysis of data procured from one of the social networking websites, Twitter, a number of issues and challenges need to be addressed, which are put forward in this paper.**

***Index Terms*—Sentiment Analysis, Supervised Learning, Social Networks.**

I. INTRODUCTION

Reviews, comments and opinions of the people play an important role in determining whether a given population is satisfied with a product or a service or in judging their response to particular events of interest. Data consisting of such reviews or opinions has a very high potential for knowledge discovery. This data finds its way quickly on to the World Wide Web via personal blogs and Social Networking Websites like Facebook, Twitter, etc. In order to reveal the overall sentiment of the population, retrieval of data from such sources and subsequent sentiment analysis becomes indispensible. Hence, the task in hand is divided into four sub-tasks: (i) extraction, (ii) pre-processing, (iii) analysis and (iv) knowledge discovery.

We focus our attention towards Twitter, a micro-blogging social networking website. On Twitter, users share their views and opinions in the form of text, known as a ‚tweet', which is no more than 140 characters. The topic of the tweet can be anything ranging from movies to international events or criticism of new laws. These tweets hold the key for determining the sentiment of a population as the ideas are original and directly originate from the user's mind. Also, Twitter has witnessed a tremendous increase in the number of users recently. And since the text in a tweet that has to be analyzed is condensed and exceeds no more than 140 characters, sentiment analysis of a tweet is much easier than calculating the sentiment of a large-size document.

In this paper, techniques that have been used for analyzing the sentiment of text, be it a document or a tweet, are reviewed. These techniques range from simple lexicon based approaches to supervised learning methods. The learning based methods include Naïve-Bayes classifiers, Maximum Entropy method and Support Vector Machines. A hybrid technique, label propagation, which makes use of a combination of the above methods and also incorporates a Twitter Follower graph for label distribution is also discussed. In addition to the techniques for sentiment analysis, the paper also highlights a number of issues and challenges that need to be overcome for sentiment analysis of Twitter data. These issues generally include the disadvantages and advantages of different classifiers, adaptability to Twitter conventions and language use and also the relation between structural properties of networks and deriving the sentiment of a population.

II. LITERATURE REVIEW

A lot of research has been done by researchers in the sentiment analysis domain. A few of the many approaches used for sentiment classification are discussed.

A. Lexicon based approach

Lexicon based approaches are used widely to classify text sentiment. Such classifiers attempt to classify data on the number of positive and negative words present in the text, and do not need any training dataset. These words which express

978-1-4799-2900-9/14/$31.00 ©2014 IEEE

opinion are known as "opinion words" and the lexicon is known as "opinion lexicon". A simple subjectivity based lexicon named „opinion finder' has been used previously for sentiment analysis in order to measure consumer confidence [20]. Though the results are noisy, the high error rate can be canceled out by the large volume of data that is investigated. The sentiment score X_t on a day t can be calculated as the ratio of the number of positive words (pos) to the negative words (neg).

$$X_t = \frac{count\ (pos\ \hat{}\ topic)}{count\ (neg\ \hat{}\ topic)} = \frac{p(pos\,|topic, t)}{p(neg\,|topic, t)}$$

The above is one of the scoring functions that can be used to classify the text. In case the scoring function fails, the polarity of the previous sentence can be used as a tie breaker or information from labeled data can be used as well.

The major problem with this approach is that there is no mechanism to deal with context dependent words. For example, the word „long' can be used to convey a positive as well as a negative opinion both depending upon the context in which it is used. For example, "the phone has a long battery life", has a sentiment of opposite polarity when compared with the sentence "the phone takes too long to restart." This problem is addressed using a holistic lexicon based approach [14]. Instead of focusing only on the sentence in question, information and evidences from other reviews and comments are also exploited using conjunction rules. For example, let us again consider the sentence "The camera has a long battery life". Since the adjective „long' can be used to express both positive and negative opinions as explained earlier, we can investigate other reviews that include the word „long'. Let's say, another reviewer has stated „The camera is great and has a long battery life'. By logic, the adjective „long' can only be used to convey a positive opinion in this review. A statement „The phone is great and has a short battery life' is unlikely.

Also, another obstacle in this approach is that there might be multiple entities addressed within a single sentence, and the opinion for each entity can vary.

For each entity in the sentence, we can compute an orientation score. Semantic orientation represents polarity and strength of a negative word and is a measure of subjectivity and opinion in text [23]. A positive word is assigned the semantic orientation score of +1, and a negative word is assigned the semantic orientation score of −1. All the scores are then summed up using the following function:

$$Score(e) = \sum_{wi:wi \in S \wedge wi \in V} \frac{wi.SO}{d(wi, e)}$$

where *wi* is an opinion word, *V* is the set of all opinion words (including idioms), that is the lexicon, and *S* is the sentence that contains the entity *e*. *d(wi, e)* is the distance between entity e and opinion word *wi* in the sentence s. wi.SO is the semantic orientation of the word wi. The multiplicative inverse in the formula is used to give low weights to opinion words that are far away from the entity *e*.

B. Naïve Bayes Approach

Naïve-Bayes classifier is a simple probabilistic classifier which uses Bayes Theorem. The model can be combined with a decision rule, a common rule being, to pick the hypothesis that is most probable which is known as the maximum a posteriori model or the MAP decision rule. According to this rule, the document d can be classified into class:

$$c^* = \text{argmax}_c\ P(c|d).$$

The classifier is derived from the Bayes rule which is given as:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

where P(c) and P(d) are prior probabilities of the class and the document. P(d) does not play any part in selecting c*.

Let *{f*1*, . . . ,fm}* be a predefined set of *m features* that can appear in a document; examples include the word "still" or the bigram "really stinks". Let n_i (*d*) be the number of times f_i occurs in document *d*. Then, each document *d* is represented by the document vector $d := (n1(d), n2(d), \ldots, n_m(d))$. Since *P(d)* remains constant, the focus is generally on the numerator. Assuming that features are conditionally independent, the classifier is given as:

$$P_{NB}(c|d) = \frac{P(c)(\prod_{i=1}^{m} P(f_i|c)^{n_i(d)})}{P(d)}$$

To avoid zero probabilities, add-1 smoothing can be used. Thus, words which do not appear in a document can be given non zero probabilities.

There are two first order probabilistic models for Naïve Bayes classification, both of which make the assumption that features are independent of each other. One is the Bernoulli model and other is the Multinomial model. The Bernoulli model is a Bayesian Network with no word dependencies and binary word features whereas the Multinomial model is a unigram language model with integer word counts. Multinomial model is used when the frequency of a word occurring in a document counts. Since, the topic of discussion is sentiment classification and not topic classification, just the fact that a word occurs is enough. Hence, a binarized version of the Multinomial version is used which only takes in to account the presence of a word and not its frequency. The Bernoulli model on the other hand generates a Boolean indicator for each term of the vocabulary depending upon its presence or absence. Thus, the Bernoulli model also takes words that do not appear in the document into account. It is found that the multivariate Bernoulli performs well with small vocabulary sizes, but the multinomial model usually performs even better at larger vocabulary sizes, providing on an average 27% reduction in error over the multivariate Bernoulli model at any vocabulary size [5].

1) Unigram Naïve Bayes

For unigram Naïve Bayes, the probability of a term belonging to a class is given as the empirical counts of that term in messages with same class. In multinomial model, the probability is given as:

$$P(t_k|c) = \frac{T_{ct_k}}{|V_c|}$$

Where T_{ct_k} is the number of times the term is associated with the class and V_c is the total number of terms seen for the class. In contrast to the above model, the Bernoulli multivariate model deals with the number of documents containing the term for that class divided by the total number of documents for the class. The binarized variation of the multinomial model clips the word count in each document as one.

The use of χ^2 feature selection method with the Bernoulli model has also been investigated and it showed that using the most discriminative features can improve the performance of the classifier significantly [11]. The χ^2 feature selection method associates a class with the input feature. Assume F has two values {0,1} and C has two values {0,1}. The association between a feature and a class can be calculated as:

$$\chi^2(F,C) = \frac{N(N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11}+N_{10})(N_{11}+N_{01})(N_{00}+N_{10})(N_{00}+N_{01})}$$

The equation is derived using a contingency matrix. N represents the number of times a class and the feature are associated. Higher the score, higher is the association. Hence, terms with the largest χ^2 are used for classifying.

2) Bigram Naïve Bayes

The bigram Naïve Bayes classifier calculates the probability that a document belongs to a class on the basis of the number of times word ‚pairs' are seen for the class.

But since the training set becomes sparse, linear interpolation as well as back off model can be used. The linear interpolation weighs the unigram as well as bigram probabilities to calculate the overall probability of the document.

$$P(c|d) = wP_{unigram}\ (c|d) + (1-w)P_{bigram}\ (c|d)$$

The back off model uses the bigram probability if its seen with the class or else backs off to the unigram probability. The fig 1 shows the performance of variants of Naïve Bayes classifiers.

A few inferences that can be made from the graph in figure 1:

- The classifier performance is better in case of classification into positive-negative as compared to other class modules.

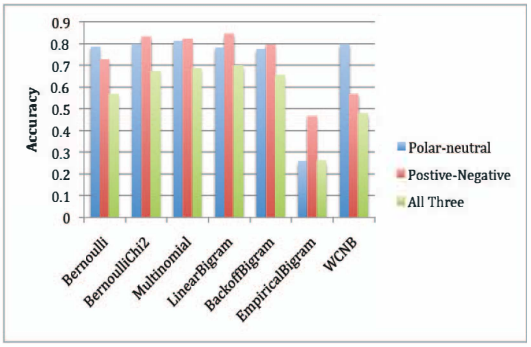

Figure 1: Classifier Performance [10]

- Bigram Naïve Bayes classifier using linear interpolation model performed the best for positive-negative classification.
- The accuracy of the classifier increased significantly when X^2 feature selection method is incorporated.
- As expected, the empirical bigram model performed poorly owing to sparseness of dataset.
- Also, a Weight Normalized Complement Naïve Bayes classifier was also introduced to counter the problem of sampling bias which performed poorly [11].The poor accuracy was attributed to less number of classes and small data set.

C. Maximum Entropy

The maximum entropy technique is a probability distribution estimation technique. It is used for various natural language processing tasks, one of which is text classification. The underlying principle of maximum entropy is if not much is known about the data, the distribution should be as uniform as possible, that is, have the maximum entropy. Constraints allow the distribution to be minimally non uniform. They are derived from the labeled training data and are represented as expected values of features. The solution to the maximum entropy formulation can be found out by the Improved Iterative Scaling Algorithm [16].

For example, let us consider the sentiment analysis of tweets retrieved which contain the word ‚jobs', for the purpose of measuring the consumer confidence. If we are told on an average 50% of the tweets have a positive sentiment score, we would say the number of tweets having negative and neutral sentiment score will be 25% each. It is easy to create such a model, but with increase in number of constraints, the task becomes more complex.

When using maximum entropy, the first step is to identify the features to be included in the model. Then the expected value of the feature is calculated which can be used as a constraint on the model. Thus, maximum entropy allows us to restrict the distribution to have the same value as expected of a feature in the model distribution. It can be shown that the distribution is always of the exponential form [3].

$$P(c|d) = \frac{1}{Z(d)}\exp\left(\sum_i \lambda_i f_i(c,d)\right)$$

Where $f_i(c,d)$ is a feature, λ_i is a parameter to be estimated and $Z(d)$ is just a normalization function.

$$Z(d) = \sum_c \exp(\sum_i \lambda_i f_i(c,d))$$

Feature class functions can be represented as binary functions that fire only if a tweet contains a particular feature. Maximum entropy can thus successfully include features involving bigrams. For example, the feature class function fires only when the bigram ‚still hate' is present and the tweet is then hypothesized to be negative.

$$F_{i,c}(d,c') = \begin{cases} 1, n(d) > 0 \; and \; c = c' \\ 0 \; otherwise \end{cases}$$

One of the advantages of Max Entropy method is that it does not suffer from the ‚independence assumption'. For example, in the phrase ‚bon voyage', the two words almost always co-occur. So instead of counting the evidence of such an occurrence twice, Max Entropy will discount the weight towards classification by half. Thus, bigrams and phrases can easily be added as features.

But since the constraints are estimated from the labeled training data, there is a chance of data being sparse and thus this method may suffer from over fitting. In such cases, a prior has to be used. To integrate a prior, a maximum a posteriori estimation for the exponential model is used. It is found that introducing a prior for each feature improves the performance significantly [21].

D. Support Vector Machines

Support vector machines are high margin classifiers. The main idea underlying SVM for sentiment classification is to find a hyper plane which divide the documents or in our case, tweets as per the sentiment, and the margin between the classes being as high as possible.

SVMs are based on the Structural Risk Minimization principle. The objective is to find a hypothesis h for which the true error is the lowest. The true error can be defined as the probability that the h will make an error in categorizing unseen data or a randomly selected test sample.

If we symbolize the hyper plane as $\vec{h}$ and the tweet as $\vec{t}$, and represent the classes into which the tweet has to be classified as $C_j \in \{1,-1\}$ corresponding to the sentiment of the tweet, the solution can be written as:

$$\vec{h} = \sum_i \alpha_i C_i \vec{t_j}, \quad \alpha_i >= 0$$

Here, α_i can be found out by solving a dual optimization problem. Those tweets with α_i greater than zero, are the ones that contribute in finding the hyper plane and are hence known as support vectors.

Feature selection is an important task in machine learning techniques. There are many features that have to be taken into account for text classification, to avoid over fitting and to increase general accuracy. SVMs have the potential to handle large feature spaces with high number of dimensions. Also, the learning ability of a SVM is independent of the dimension of the feature space. SVMs measure the complexity of the hypothesis with which they separate the documents and not the number of features. As long as the text classification problem is linearly separable, the number of features in the feature space is not one of the issues.

To deal with a large number of features, traditional text categorization methods assume that some of the features are irrelevant. But even the lowest ranked features according to feature selection methods contain considerable information [8]. Considering these features as irrelevant often result in a loss of information. Since SVMs do not require us to make such an assumption, information loss can be reduced.

It is found that k-NN works best among the conventional methods for text classification and SVMs are a better option independent of the choice of parameters [8]. There is also an automatic review classifier based on SVM, known as SVM^{light}. This program has been used extensively in the subsequent research involving SVMs.

Though SVM outperforms all the traditional methods for sentiment classification, it is a black box method. It is difficult to investigate the nature of classification and to identify which words are more important for classification [8]. This is one of the few disadvantages of using SVM as a method for document classification.

E. Label propagation

We have seen the lexicon based and machine learning based approaches to sentiment analysis of text. But since Twitter data contains a considerable amount of terseness and informality, adapting the standard supervised learning techniques to sentiment classification of Twitter data is not easy.

Though machine learning methods have improved the accuracy significantly, one of the major drawbacks of almost all supervised learning methods is that they require labeled inputs as training data and adapt very poorly to the changes in language use. Use of language on social networking websites differ greatly from the normal usage, and hence incorporating additional features in these methods is of vital importance to improve the accuracy.

Utilizing the label propagation method using the Twitter Follower graph enables us to exploit the relationships between users, users with tweets and tweets with features in increasing the overall accuracy of determining the sentiment of a tweet is a good option.

A user and his tweets are influenced by the tweets of other users he/she follows [12]. Label propagation is a semi supervised method in which labels are distributed from a small number of nodes which are injected with initial label information. The distribution is through a Twitter Follower Graph G={V,E,W}, where V is the set of n Nodes, E is the

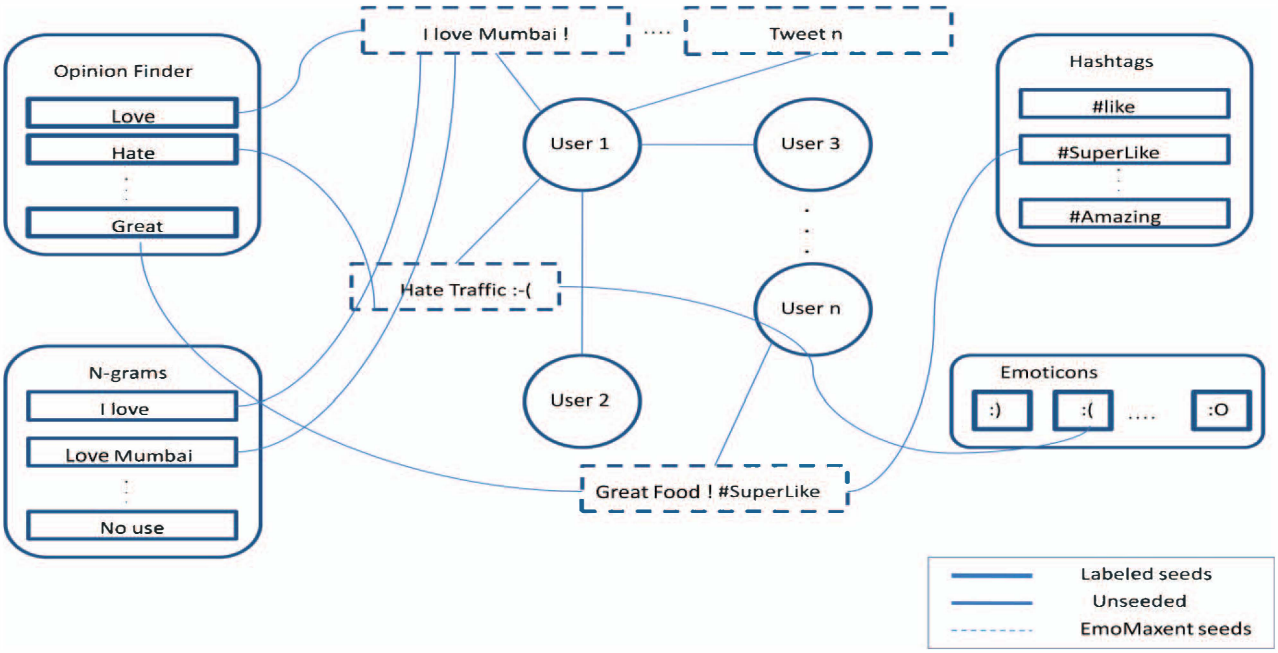

Figure 2: Illustration of graph with All Edges and Noisy seed.

set of m edges and W is a n x n weight matrix, where w_{ij} is the weight of edge(i,j). The spreading of label distributions can be seen as random walks with three possible actions [12]:

- Injecting a seeded node with its seed label
- Continuing the walk from the current node to the neighbor node.
- Abandoning the walk.

Modified Adsorption [22] takes three parameters: μ_1, μ_2, μ_3 which control the relative importance of each of the three actions mentioned above.

Since the graph needs to be seeded, the following are some of the variants for seeding the graph [12]:

- Maxent-seed: OpenNLP Maximum Entropy dataset can be trained on the Emoticon dataset; every tweet node is seeded with its polarity predictions for the tweet.
- Lexicon-seed: Nodes are created for every word in the OpinionFinder lexicon. Positive words are seeded as 90% positive if they are strongly subjective and 80% positive if weakly subjective; similarly and conversely for negative words. Every tweet is connected by an edge to every word in the polarity lexicon it contains, using the weighting scheme discussed with Feature-edges below.
- Emoticon-seed: Nodes are created for emoticons and seeded as 90% positive or negative depending on their polarity. An example of an emoticon dataset is shown in fig 2.

Another aspect of graph construction is to add edges and their weights:

- Follower-edges: We add an edge with weight 1 if user A follows user B. This weight is comparable to the weight of moderately frequent edge in the set of feature edges.
- Feature-edges: Nodes are used to represent hash tags and features and are connected to tweets which contain them. Features can be unigrams or bigrams. An edge connecting a tweet to a feature has weight w_{tf} which is obtained using relative frequency ratios of the feature between the dataset d in question and the emoticon dataset r which is used as a reference corpus.

$$w_{tf} = \begin{cases} log\left(\frac{P_d(f)}{P_r(f)}\right), & if\, P_d(f) > P_r(f) \\ 0, & otherwise \end{cases}$$

All seeds are collectively termed as noisy seeds and All edges are used by combining both the sets of edges.

A max entropy classifier trained with distant supervision (emoticon dataset) in combination with the Twitter Follower Graph performs better than lexicon based ratio predictor and competes favorably with fully supervised approaches.

TABLE 1: Mean square error per user [12]

Classifier	***MSE***
Random	0.167
LEXRATIO	0.170
EMOMAXENT	0.233
LPROP (Follower-edges, Maxent-seed)	0.233
LPROP (All-edges, Lexicon-seed)	0.187
LPROP (Feature-edges, Noisy-seed)	0.148
LPROP (All-edges, Noisy-seed)	0.148

From the results in Table 1, we can infer that the Mean Squared Error in predicting the sentiment is the least when All-edges and Noisy seeds are used. Thus, the Twitter

Follower Graph does contribute to the accuracy. Also, it can be noted that addition of follower edges does not play any role in reducing the error.

III. ISSUES AND CHALLENGES

Though there are a number of techniques that can be used for sentiment analysis, it is difficult to say that a particular technique outperforms the others by a large margin. A number of issues and challenges that must be addressed to increase the overall performance and accuracy of sentiment analysis are discussed.

It is seen that some techniques are lexicon based while some are learning based. The lexicon based techniques make use of dictionaries of words annotated with their semantic orientation to classify text. Generally these methods give a high precision but low recall. On the other hand, the learning based methods classify text on the basis of labeled examples. The relevant features are extracted from the training dataset and then used to train the algorithms.

The issue with the lexicon based approach is the availability of lexicons, which are not available in all languages. Learning based methods also achieve good results. Nevertheless, they require training and a training dataset.

Both statistical as well as syntactic techniques can be used. Syntactic techniques can yield good results since syntactic rules of language can be applied to identify nouns, adjectives, adverbs, etc. But again, the major problem is these techniques depend heavily on the language and thus cannot be ported to other languages.

The statistical techniques have a probabilistic background and focus on words and their categories. An important advantage of statistical techniques over the syntactic techniques is that these techniques are independent of language.

As per our survey, the performance and accuracy of classifiers depends on a number of factors.

- The features used influence the classifier accuracy. For example, while using n-gram framework, using a high value for n degrades the performance. For sentiment analysis, the value of n should be at the most 2 or 3. Also, the number of occurrences of a word does not matter much, the mere fact that it does is enough. Apart from n-gram features, one can even use emoticons for labeling. It is seen that the accuracy of learning algorithms increases when trained using emoticons.
- Not all features that are returned by the tokenization algorithm must be used, because the list contains a lot of irrelevant features. Hence, the feature selection method used also determines the accuracy of a classifier. Generally for sentiment analysis, the Chi-square test, which is discussed earlier in this paper, and the Mutual Information method is used. Each algorithm evaluates the keywords in a different manner and thus leads to different selections. Moreover, each algorithm requires different configuration such as the level of statistical significance, the number of selected features, and so on.
- Apart from the features and the feature selection method, the classifier accuracy depends a lot on the domain in question. Different classifiers yield different results when applied to different domains. For twitter, the binarized version of Naïve Bayes along with Mutual Information feature selection method is known to outperform even the Support Vector Machines.

Adapting to the Twitter domain poses a real challenge for the methods that are discussed. The learning methods, especially, need to be adapted to the language use on Twitter and also to structural properties of large scale networks. One important point to be noted is that sentiment analysis of tweets involves sentence-level classification and not document-level, unlike classification of online reviews. There are certain conventions that are followed on Twitter that need to be addressed.

- Twitter users sometimes direct their tweets to certain other users, who they follow. This is known as a „Twitter mention'. The general convention is using the „@' symbol followed by „Twitter Handle' of the user.
- Many a times, external links are included as part of the tweets.
- When users like a tweet, they share it with their followers using the retweet button, or by using the acronym „RT' followed by the tweet which they want to share.

In addition to the above, usage of slang and casual use of language are other characteristics that are seen in general on social networking websites. People generally tend to use „gr8' for „great', or „ya' to mean „yes'. Also, presence of exaggeration is also common. For example, we come across „goooood' instead of „good'. Hence, the existing methods have to be modified to adapt to the language use on social networking websites.

Social networks contain a few high degree vertices known has „hubs' or „connector vertices'. Such vertices or nodes have a degree number much greater than the average. Existence of such hubs in networks was demonstrated by the famous Travers and Milgram's experiment [5]. In this experiment of tracking the route of the letters between a number of vertices and a target, it was seen that out of the 64 letters that actually reached the target, 16 were delivered to the target by the same person. That is, 25% of the edges to the target were from the same vertex. Such vertices are known as „hubs' of social traffic. In our domain, these correspond to well known personalities on twitter who have a large follower count. Hence, the views or opinions of such individuals are capable of influencing a larger number of people at once.

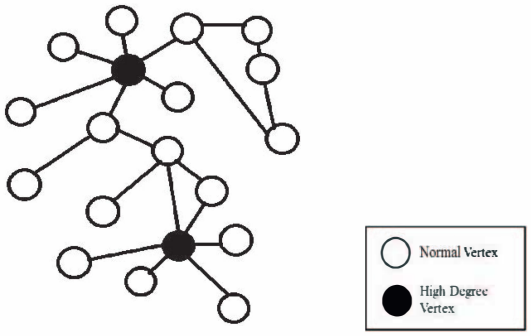

Figure 3: A sample social network to show high degree connecter vertices (represented by black nodes).

Another property of social networks that does not work well for the purpose of sentiment analysis is Preferential Attachment. This property states that higher the degree of a vertex, higher the probability of a new edge to be added to the vertex. This property is also termed as the „Rich-get-Richer' process in social networks.

A sample code to demonstrate the same was designed using MATLAB. The problem statement is as follows:
"Let us say, there are 1000 restaurants in a city and 10000 people. The probability of a person visiting a restaurant should be proportional to the number of people already in the restaurant".

It can be inferred from fig.4 that there are a few restaurants with a large number of customers. This explains the preference for popularity. An individual having a large follower count has a higher probability of getting followed in the future. Hence, if we use the label propagation method, the label distribution via such nodes will be much more than the average. This property follows a power law distribution. Hence on a log-log scale, we get a linear relation.

IV. PROPOSED SYSTEM

Our system uses data extracted from social networking websites, mostly Twitter for decision-making. This system will act as a secondary data source for corporate as well as government officials. We breakdown the process into the following subtasks:

- Extraction: Involves capturing of real time data from twitter streams and storing it in a suitable format. The user is asked to enter a keyword. Tweets containing the keyword are extracted from the streams.

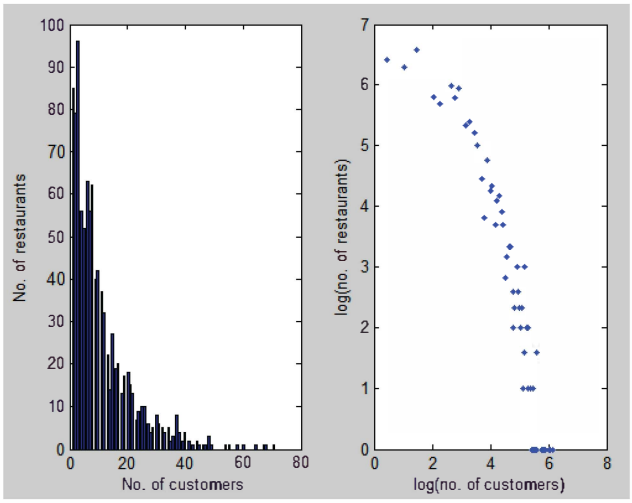

Figure 4: A Histogram from a MATLAB sample code to demonstrate the „Rich-get-Richer' process in social networks.

- Preprocessing: The extracted data is then subjected to preprocessing. Preprocessing involves filtering out of non-English tweets for simplicity. Another aspect of preprocessing is feature reduction. An attempt to reduce the number of features is made. Words in tweets that represent twitter mentions are discarded. That is, the symbol „@' followed by the twitter handle of a user. Also the acronym „RT' is removed from the tweets in the training dataset. Interpretation of slang and casual use of language also is incorporated. For example, restriction of exaggeration is brought about by replacing words like „goooooood' with good.
- Analysis: The preprocessed data is then subjected to sentiment analysis. Along with the sentiment, the corresponding number of tweets and the geographical location of the tweets is also determined.
- Knowledge discovery: In order to gauge the opinion of a population in response to a particular event, or a product or service, it is necessary to store data relevant to the topic of discussion. The system allows storage of the not only the sentiment of the tweets but also the count and the geographical location. Capturing such information before an event and then after the event allows the user to gauge the sentiment of a given range of population in response to that event. We make use of statistical graphs and geographical charts to derive inferences.

  We propose to tailor our system for a number of applications. One of which is acting as a data source for strategic decision-making. For example, tracking the effectiveness of a particular business strategy, by recording the number of positive and negative tweets before and after the strategy is put to use.

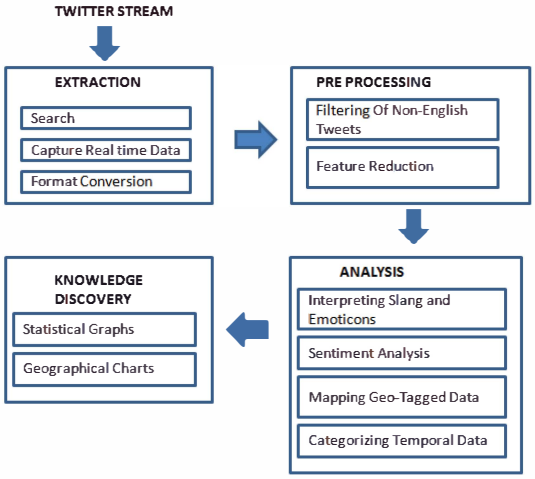

Figure 5: Block diagram of the proposed system.

This will help business officials determine the feasibility and efficiency of their promotional campaigns. This tool can also help the media decide the reaction of the citizens of the country to a national topic of interest, like a new bill being passed by the Government. The system also uses the geo-tagging feature which helps derive inferences regarding the geographical location of positive and negative tweets, thus enabling the user to come up with strategies specific to a region.

V. CONCLUSION AND FUTURE SCOPE

Thus, it has been found out that a number of techniques can be used to perform sentiment analysis of text. But the methods are domain specific. Moreover the techniques need to be adapted to the source from which the data is extracted. If the source is a social networking website, the language use and specific conventions need to be addressed.

Hence, future scope in the sentiment analysis domain involves devising a method for sentiment classification that works well when applied to all the domains. Our proposed system, as of now, exploits Twitter for knowledge discovery and the same can be extended to other social networking websites like Facebook, Google Plus, etc.

REFERENCES

[1] Bo Pang and Lillian Lee., "Opinion Mining and Sentiment Analysis", Foundations of Information Retrieval, Vol. 2, Nos. 12, 1-135, 2008.

[2] Pang, L. Lee, and S.Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques", Proceedings of the ACL-02 conference on Empirical methods in natural language processing- Volume 10, pages 79–86, 2002.

[3] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty, "Inducing features of random fields", IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 19, no. 4, April 1997.

[4] Georgios Paltoglou, Mike Thelwall, "Twitter, MySpace, Digg: Unsupervised Sentiment Analysis in Social Media", ACM Transactions on Intelligent Systems and Technology, Vol. 3 Issue 4, Article 66, September 2012.

[5] Andrew McCallum , Kamal Nigam, "A comparison of Event models for Naive Bayes text classification", AAAI-98 workshop on learning for text categorization, 1998.

[6] Jeffrey Travers, Stanley Milgram, "An experimental study of the small world problem", Sociometry, Volume 32 issue 4, Dec 1969.

[7] Thorsten Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features", Machine Learning: ECML-98 lecture notes, Computer Science Volume 1398, pp 137-142, 1998.

[8] Anuj Sharma, Shubhamoy Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis", Proceedings of the 2012 ACM Research in Applied Computation Symposium, 2012.

[9] Haiyang Sui, Christopher Koo, Syin Chan,"Sentiment classification of product reviews using SVM and decision tree induction", 14th ASIS SIG/CR Classification Research Workshop, 2003.

[10] Alistair Kennedy , Diana Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters", Computational Intelligence, Volume 22, 2006.

[11] Suhaas Prasad, "Micro-blogging sentiment analysis using Bayesian classification methods", (http://nlp.stanford.edu/courses/cs224n/2010/reports/suhaasp.pdf), 2010.

[12] Michael Speriosu, Nikita Sudan, Sid Upadhyay, Jason Baldridge, "Twitter polarity classification with label propagation over lexical links and the follower graph", Proceedings of EMNLP'11, Conference on Empirical Methods in Natural Language Processing, pages 53-63, 2011.

[13] Pimwadee Chaovalit, Lina Zhou, "Movie review mining: a comparison between supervised and unsupervised classification approaches", Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[14] Xiaowen Ding, Bing Liu, Philip S. Yu, "A holistic lexicon-based approach to opinion mining", Proceedings of the 2008 International Conference on Web Search and Data Mining, 2008.

[15] Christopher Johnson, Parul Shukla, Shilpa Shukla, "On classifying the political sentiment of tweets", (http://www.cs.utexas.edu/~cjohnson/TwitterSentimentAnalysis.pdf).

[16] Kamal Nigam, John Lafferty, Andrew McCallum, "Using maximum entropy for text classification", IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.

[17] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. "Twitter mood predicts the stock market", Journal of Computational Science, 2(1), March 2011.

[18] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, Isabell M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment", Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, 2010.

[19] Efthymios Kouloumpis, Theresa Wilson, Johanna Moore, "Twitter sentiment analysis: the good the bad and the OMG!", Fifth International AAAI Conference on Weblogs and Social Media, 2011

[20] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, Noah A. Smith, "From tweets to polls: linking text sentiment to public opinion time series", Proceedings of the International AAAI Conference on Weblogs and Social Media, Washington DC, May 2010.

[21] Stanley F. Chen , Ronald Rosenfeld, "A Gaussian Prior for Smoothing Maximum Entropy Models", Technical report CMU-CS-99-108, Carnegie Mellon University,1999.

[22] Partha Pratim Talukdar and Koby Crammer, "New Regularized Algorithms for Transductive Learning", Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science Volume 5782, pp 442-457, 2009.

[23] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll and Manfred Stede, "Lexicon-based methods for sentiment analysis", Computational linguistics, volume 37, number 2: 267–307, MIT Press, 2011.

[24] Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, Bing Liu, "Combining lexicon-based and learning-based methods for twitter sentiment analysis", Hewlett-Packard Laboratories, HPL-2011-89, 2011.

[25] Zornitsa Kozareva, Eduard Hovy, "Insights from network structure for text mining", Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1616–1625, 2011.