Soc. Netw. Anal. Min. (2014) 4:181
DOI 10.1007/s13278-014-0181-9

ORIGINAL ARTICLE

A novel sentiment analysis of social networks using supervised learning

Malhar Anjaria · Ram Mohana Reddy Guddeti

Received: 1 September 2013 / Revised: 5 February 2014 / Accepted: 12 February 2014 / Published online: 13 March 2014

Abstract Online microblog-based social networks have been used for expressing public opinions through short messages. Among popular microblogs, Twitter has attracted the attention of several researchers in areas like predicting the consumer brands, democratic electoral events, movie box office, popularity of celebrities, the stock market, etc. Sentiment analysis over a Twitter-based social network offers a fast and efficient way of monitoring the public sentiment. This paper studies the sentiment prediction task over Twitter using machine-learning techniques, with the consideration of Twitter-specific social network structure such as retweet. We also concentrate on finding both direct and extended terms related to the event and thereby understanding its effect. We employed supervised machine-learning techniques such as support vector machines (SVM), Naive Bayes, maximum entropy and artificial neural networks to classify the Twitter data using unigram, bigram and unigram + bigram (hybrid) feature extraction model for the case study of US Presidential Elections 2012 and Karnataka State Assembly Elections (India) 2013. Further, we combined the results of sentiment analysis with the influence factor generated from the retweet count to improve the prediction accuracy of the task. Experimental results demonstrate that SVM outperforms all other classifiers with maximum accuracy of 88 % in predicting the outcome of US Elections 2012, and 68 % for Indian State Assembly Elections 2013.

M. Anjaria (✉) · R. M. R. Guddeti
Department of Information Technology, National Institute of Technology Karnataka, Surathkal, Mangalore 575025, India
e-mail: malhar.anjaria@gmail.com

R. M. R. Guddeti
e-mail: profgrmreddy@nitk.ac.in

Keywords Electoral prediction · Microblogs · Opinion mining · Sentiment analysis · Social intelligence · Social network analysis · Supervised machine learning · Twitter · Twitter analytics

1 Introduction

Microblogging has become one of the most-accepted communication medium used by people across the globe. Millions of text messages are appearing daily on popular websites that provide microblogging services such as Facebook,[1] Google+,[2] Twitter.[3] Very large amount of text messages are published on social networks to express opinions on diversified topics and thereby discussing several current issues. Because of the unrestricted message format and also an easy accessibility of microblogging platforms, internet users have inclined from traditional communication tools such as internet chat, traditional blogs or mailing lists to microblogging services. The exponential rise in numbers of user posts about consumer brands and services or opinions about political and religious events make microblogging sites such as Twitter more popular for carrying out sentiment analysis (Pak and Paroubek 2010).

Twitter is one of the most popular microblogging platforms with more than 25 million visitors monthly. On Twitter, any registered user can publish a short message referred to as tweet with a maximum length of 140 characters, which is visible on the public display with more than one million tweets per hour. Especially during the process of democratic elections, political issues are clearly

[1] http://www.facebook.com.
[2] http://plus.google.com.
[3] http://www.twitter.com.

on the minds of several users. In addition, politicians are tweeting with the electorate and thereby trying to influence Twitter users.

Opinions expressed via tweets are often ambiguous and this in turn makes it difficult for understanding the correct opinion using basic computational models. Hence, natural language processing-based computational models are required for analyzing the tweets response accurately. Natural language models with machine-learning approaches have been employed effectively by O'Connor et al. (2010) and Pak (2010) for finding the sentiment over Twitter.

Majority of the researches concentrate only on the word frequency and thereby finding the related terms that have higher influence on the result. This methodology based on word frequency may not detect the terms that influence the outcome in reality. While predicting the outcome of any event about there is a need to understand major related events with respect to the main event within the context. For example the tweet, "Apple has invented great gadgets, iPhone is the best of all", is surely a positive sentiment about Apple but, it will also provide the information that "iPhone" is the contributing factor for positive sentiment towards "Apple". Let us refer, there terns as an extended or indirect features and details are discussed in detail in Sect. 3.3.

Further, we observed that it is a general human tendency to agree with the similar views of another person-based study of the opinion extraction methodologies in detail. Accordingly, Twitter users will share other user's tweet in case of an agreement with the view presented by the original tweet, referred to as a "retweet". Since Twitter is a social platform, retweets are typically shared with entire user network directly, which is also available on search for the public. Each retweet spreads the sentiment across the user's network and provides similar effect as a tweet itself for the entity. It has also been observed that many users prefer retweeting rather than tweeting the view itself. Not active Twitter users who do not tweet but only retweet will be missed in case of taking only the sentiment analysis of the original tweet. Hence, it is important to leverage the benefit of retweet count to establish an influence factor for the entity. This paper proposes a method to predict outcome of an event incorporating the sentiment score with retweets garnered in favor or against each participating entity. Our method helps taking such scenarios into consideration. To the best of our knowledge this will be the first attempt to establish the influence factor for the entity but not for the user. Further, proposed methodology has a low computation overhead.

In this paper, we try to validate prediction results for the US Presidential Elections 2012 and Karnataka State Assembly Elections (India) 2013 using Twitter data. We utilized extended features for sentiment analysis with different classification methods and combining sentiment score with influence factor for the entity to improve accuracy further.

Rest of the paper is organized as follows: Sect. 2 describes the background and related work; Sect. 3 deals with methodology and framework; Sect. 4 gives the results and analysis; final conclusions are detailed in Sect. 5.

2 Background and related work

2.1 Text sentiment analysis

Sentiment analysis is the computational study of how opinions, attitudes, emotions, and perspectives are expressed in natural language with reference to a subject. It can thus be vital to service providers, allowing them to quickly assess how new products and features are being received. Recent breakthroughs show that this analysis can go beyond a general measure of positive vs. negative, isolating a fuller spectrum of emotions and evaluations and controlling for different topics and community norms. Good amount of research has been carried out in the field of sentiment analysis to predict public sentiment with the use of various techniques.

Sentiment analysis has been carried out with the help of word lexicon databases such as OpinionFinder,[4] SentiWordNet[5] that provide positive and negative orientation of the words with respective scores. Brendon O'Connor et al. (2010) carried out the sentiment analysis using OpinionFinder lexicon scoring mechanism to map Twitter sentiments with public opinion mood for a time-series analysis.

In another attempt, Johan Bollen (2011) used the POMS score for establishing the sentiment values in classifying the moods category and thereby predicting the Dow-Jones stock market. The system measures this sentiment using a syntactic term-based approach, to detect mood signal from very brief Twitter messages.

However, these aforementioned lexicon-score mechanisms limit the flexibility of scoring mechanism for the words encountered in a document. On the other hand, the machine-learning methodology provides more flexibility in selecting scoring mechanism. Sentiment analysis-based experiments using unsupervised learning (Turney and Peter 2002) and supervised (Pang and Lee 2002a) learning have been widely found in the literature. Hence, in the proposed work, we mainly concentrate on the sentiment analysis carried out using supervised machine-learning algorithms for predicting outcome of democratic elections.

[4] http://mpqa.cs.pitt.edu/opinionfinder/.

[5] http://sentiwordnet.isti.cnr.it/.

The system proposed by Pang and Lee (2002b), considered the ratio of positive words to total words to estimate the opinion. Further, they also developed a methodology in 2008 to establish sentiment outcome based on the presence of the term in the tweet rather than just the frequency. This methodology has attracted several researchers concentrating on sentiment analysis, since it works very well for long documents.

Several researchers such as Jiang (2011), Tan et al. (2011) and others have employed support vector machine (SVM), Naive Bayes (NB) and maximum entropy (MaxEnt) such as supervised classifiers whereas, very few researchers have discussed ANN in the study of sentiment analysis.

Chen et al. (2011) implemented the feed-forward BPN network and uses sentiment orientation to compute the values at each neuron. The model depends upon the sentiment orientation of terms used in the documents. Moraes et al. (2013) compared the accuracy of ANN and SVM for the document-level sentiment classification using term frequency–inverse document frequency (TF–IDF) technique to extract feature values for unbalance datasets.

Parikh and Movassate (2009), Barbosa and Feng (2010), Davidiv et al. (2010), Go et al. (2010a) followed the machine-learning approach for sentiment analysis of tweets. Davidiv et al. (2010) proposed multiple sentiment types to classify tweets using hashtags and smileys as labels.

Further, Barbosa and Feng (2010), proposed a two-step approach to classify the sentiments of tweets using SVM using abstract features.

Due to the evolution of social networks in general and rise of Twitter in particular has encouraged several researchers, to carry out research on elections and other events sentiment analysis being the main methodology with help of Twitter-based dataset. Along with movie reviews (Doshi 2008), tracking natural calamitous events and utilizing Twitter as a social sensor (Sakaki et al. 2010) or stock market predictions (Bollen et al. 2011), election prediction using Twitter has been centre of attraction and details are given in the following subsection.

2.2 Election prediction-based on Twitter

Davidiv et al. (2010) found that political incumbents and challengers used Twitter in different ways during the US 2010 midterm elections; incumbents focused more on current events, whereas challengers more often targeted incumbents. Examining the same elections, researchers at the Pew Research Center (2010) found that political tweets often were a call to action, encouraging citizens to vote using Twitter as a new medium for citizen movement. Several studies demonstrated that Twitter has become an integral part of the political communication environment in many countries, offering an extremely wealthy source of information for those attracted towards studying public opinion and political behavior.

Skoric et al. (2012) suggested that the word and term frequencies for tweets corresponding to terms related to the names of democratic leaders and organizations with respect to Singapore 2011 elections. Their methodology depends more on the relationship of the users who publish the tweets that contain the keyword terms and thereby deciding the results.

Exploring the keywords and tweeting mechanisms over Twitter Romero et al. (2011) concluded that the Twitter hashtags were persistent across time and that people would use the hashtags for politically controversial topics.

Mejova et al. (2013) have studied popular sentiment analysis on Twitter for the case study of US Presidential nominations, 2011. Nooralahzadeh (2013) have studied 2012 Presidential Elections in France and USA. They compare classification methodologies using word-based polarity, with the Naïve Bayes classifier defined in R-package and AFINN list. Attempt of predicting elections using sentiment analysis has been carried out for various elections such as Australian Federal Elections 2010 (Zhou et al. 2013), Irish General Elections 2010 (Bakliwal 2013), Spanish National Elections 2011 (Aragón et al. 2013), etc.

However, aforementioned methodologies do not take party-based influence factor into consideration while predicting the outcome of the elections, but relies more on the cumulative sentiment score for prediction. Further, sentiment score is not the sufficient criterion to judge the prediction of an electoral event. Gayo-Avello et al. (2011) observed that, studying various methodologies that consider just the tweet sentiment as the vote is not sufficient for prediction of outcome of an electoral event and better prediction models are required.

2.3 Prediction using influence factor

In a social network with large number of users, action of any individual user may influence other users as well. This can be studied with user relations and behavioral analysis of a social network structure. There have been some attempts to develop methods for measuring the influence over the tweets using different parameters such as user mentions, user relations, etc. while incorporating with the sentiment score.

Asur and Huberman (2010) developed a uniform framework to predict the real-world outcome with social media content incorporating sentiment analysis with other social network structural parameters such as user mentions in case of Twitter. Bermingham and Smeaton (2011) used the sentiment polarity volume integrated with the framework of Asur and Huberman (2010) to predict the outcome of Irish General Elections 2011.

Tan et al. (2011) showed that the information about social relationship can be used to further improve the user-level sentiment analysis. They used user relations to predict the possibility of user bonding over Twitter. Their methodology requires a manual user selection and labeling resulting in a heavy computational model when all users are involved. Further, it restricts the influence factor to a particular group of users selected for the prediction analysis.

Several researchers considered the user profile analysis to predict the outcome of an election. Livne et al. (2011) investigated the relation between the network structure and tweets and thereby forecasting US Midterm Elections 2010. Their methodology requires analysis of all participating users in voting over Twitter. Meeyoung et al. (2010) employed the user's bias towards a particular entity and thereby predicting the outcome of any event. Further, they stated that followers-to-following ratio was not a useful parameter to establish the influence factor. However, Boutet et al. (2013) used followers-to-following ratio to predict UK General Elections 2010 with a reasonable amount of success. They used party characteristics with user behavioral analysis to establish the influence model.

Recently, Song et al. (2013) attempted to mine the Twitter data in real time with respect to Korean Elections 2012. Their system is based on the term occurrence and mentions using a social network analogy. Their system calculates the change in opinion trend very well using the term calculation for a given set of tweets. Further, it also takes the user mentions into account to predict the trend behavior which in turn helps in finding the outcome of a topic in a near-time analysis.

Similarly studying the Greek General Elections 2012, Katakis et al. (2013) developed a Voting Advice Application. They concentrated on weighing each unique user found in the dataset and using this weight to decide the party orientation, and the influence user leaves upon social networking voting scenario.

The aforementioned methodologies concentrate more on the user's influential factor, but not on the influence generated for the entity for which tweets were generated. Influence factor for users can be achieved either by selecting a subset of users from a large set of users or by performing a particular set of analysis on all the users participating in the election. But, these techniques are computationally intensive tasks.

Hence, there is a need for a novel model based on sentiment analysis, which can find the influence factor for the entity with less computational cost. We experiment with the hybrid model of unigram + bigram feature vector with extended features extracted from the semantics of the text. We also propose a method to use the influence factor of the participating parties using retweets count and thereby predicting the outcome of the elections.

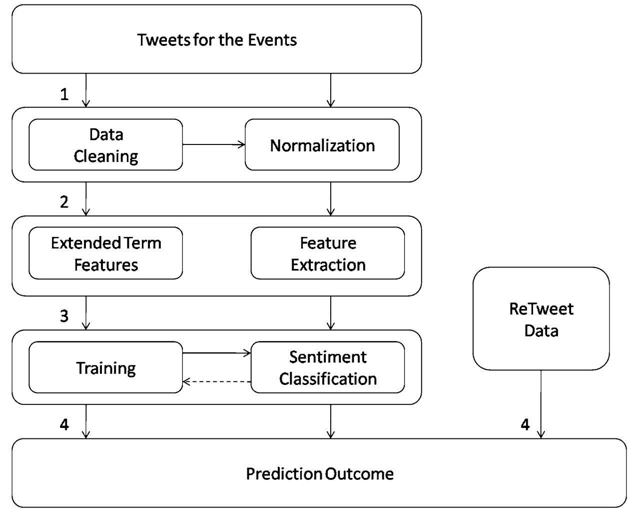

Fig. 1 Proposed high-level system flow

3 Methodology

The proposed system uses salient features of supervised machine-learning algorithms for text-based sentiment classification and thereby incorporating it with social network features, which can be used for analyzing the user behavior to predict the outcome of elections efficiently. Figure 1 depicts the proposed system designed to achieve the prediction with sufficiently higher accuracy. We have used Naïve Bayes, SVM, MaxEnt, ANN classifiers with features extracted from Twitter data using feature extraction methods such as unigram, bigram and hybrid (unigram + bigrams) feature vectors.

We fetched the tweets using Twitter API v1.1[6] and performed data cleaning and normalization as explained in Sects. 3.1 and 3.2, respectively. We employed extended features model of Jiang et al. (2011) and accordingly we modified it to extract extended features related to the electoral event and details are discussed in Sect. 3.3. After achieving extended features for the event, we concentrated on deriving the feature vectors for the classifiers from the remaining tweets. These feature vectors are used in part of chunks to train the classifier as a part of incremental training. After utilizing nearly 2/3rd of the Twitter data we test it with 1/3rd of the Twitter data. In order to achieve sentiment analysis, we perform supervised classification as

[6] https://dev.Twitter.com/.

explained in Sect. 3.4. The results of sentiment analysis are incorporated with the influence factor and thereby predicting the outcome of elections using the model described in Sect. 3.5. Section 3.6 provides the rough time complexity analysis of the complete process of classification and prediction.

The following algorithm provides the workflow of entire proposed methodology.

Algorithm 1: Predicting outcome of an electoral poll with modified extended features

Input: Enter keywords related to electoral event

Output: Outcome in terms of predictions

1. Accumulate tweets using Twitter search API with the queries given as an input. Store tweets to dataset T with attributes {tweetId, tweetxText, userId, retweetCount, tweetTimeStamp}.
2. Perform normalization and cleaning on each instance of T, remove the non-required terms and store it as T'.
3. Extract the extended targets from T' with modified extended feature model; store extended targets and features; create map M = < TweetId,Query_feature > .
4. Provide an incremental training with subset of T' as T_i with output class C for supervised classifier; C ε {positive, negative} and i is iteration.
5. Use trained dataset to perform classification using Supervised Learning; Store class C with tweet; Store sentiment values for each participant U_i as < U_i,Score_Pos,Score_neg > .
6. Evaluate the user influential factor using evaluation model for a participant U, for class P using associated tweet and retweetCount.
7. Feed achieved values from step 5 and step 6 to influence model to achieve the final output.

Figure 1 explains the proposed high-level system flow, and the following subsections will describe each of the blocks of Fig. 1 in detail.

3.1 Data collection approach

We searched using Twitter search API v 1.1 to collect data with various hashtags such as #USElections2012, #USElections, #Elections2012 for US Presidential Election 2012 for the time period of August, 2012 to October, 2012. We collected approximately 100,000 tweets for the duration. In order to collect tweets with Candidate's name, we used the terms "Barack Obama", "John Biden", "Demoeractic" and "Mitt Romney", "Paul Ryan", "Republican".

Similarly, we used hashtags such as #BJP, #Congress, #KJP, #KarnatakaElections for Karnataka State (India) Assembly Elections 2013 between April 2013 and May 2013 along with politicians' names related to all four major political parties.

We considered UTC time-offset of the tweet to ensure that tweets are fetched only from the corresponding nations. Further, we did not check for the location of the tweet due to the lack of information in majority of the dataset. However, the tweets with the names of the states are collected, but it is natural that a tweet containing name of the state is not necessarily be originated from that particular state. For example, a tweet containing "California" might have been originated from the state of "Texas".

Table 1 Example of tweet tokenization with algorithms

Tokenizer	Tokens
Whitespace	I, can't, agree, more, with, @Oprah, about, #debates!!, http://t.co/9zXlfrbe
Treebank	I, ca, n't, agree, more, with, @, Oprah, about, #debates, !, !, https, :,//t.co/9zXlaveFrb
Tweetaware	I, ca, n't, agree,more, with, @Oprah, about, #debates,!,!, http://t.co/9zXlfrbe

3.2 Normalization and feature reduction

In order to obtain feature vectors from tweetsm, we need to first tokenize and clean the data that are populated with internet language. For the process of tokenization, we employed whitespace, treebank and customized tweet tokenizers. We found whitespace tokenizer to be the fastest algorithm but, it is widely used for tokenizing Twitter data. Treebank tokenizer works well for extracting negations and overlapping feature, but fails to extract URL and HTML features in a proper manner. Tweet tokenizer is a customized tokenizer that leverages the salient features of both Whitespace and Treebank tokenizers. It parses the overlapping features such as negations and URLs in a required manner. Table 1 provides the example of tokenization for the following tweet: "I can't agree more with @Oprah about #debates!! https://t.co/9zXlaveFrb".

We removed the stop words that contain general terms such as "a", "the", "etc." and emoticons and other features to get the data in the usable format and the details are as follows:

- Username: We remove the usernames from tweets for parsing purpose; we embed "USER_NAME" in place of the username.
- URLs: We removed image or article urls as well as embedded tweet URLs from the original tweet. However, embedded tweets were stored if it contains the original search query.

- Repeated words: If a word is being repeated in a tweet more than two times, consecutively, then the occurrence of the word can be limited to two. For example, "super super super super super awesome" can be replaced by "super super awesome".
- Repeated characters: If a character is being repeated more than two times consecutively, and if the making of occurrence with a length of two does make a meaning to some word then we can restrict the character occurrence to two. For example, "verrryyyyy" has been replaced by "very" after smoothing.
- Duplicate tweets: We also check for duplicate tweets from a sliding window of the last 100 tweets. The duplicate tweets occur due to sharing a tweet as a retweet using "RT" or quotation marks or plagiarism. Then, all the duplicate tweets can be scrapped.
- Candidate accounts: We also omit the tweets from official accounts of Presidential election candidates and their party office, since these ought to be biased and do not count as general public opinion.
- Word expansion: We expand the famous acronyms as well. The expansion is considered for the Standard English words and specific to the electoral terms as well. As an example, AFAIK—As Far As I Know; BO—Barack Obama.
- Slang removal: We removed internet slang and abbreviations from the tweets such as "God" to "Good" with the help of the internet abbreviation library.

3.3 Feature extraction and extended terms

We used the unigram, bigram and a unigram + bigram (Hybrid) feature extraction method for experimental purpose. Hybrid features are taken for absolute positive and negative words. To ensure negation: abort, ugly, not, no, never, neither, hardly, seldom, etc., were taken.

To extract the extended features, we used the model originally established by Jiang et al. (2011) with different parameters and change of a value in original parameters. Any term that occurs a minimum of 500 times has been taken as an extended target. We used $K = 20$ in extracting top K nouns from the terms that appeared for more than the specified threshold value of 500.

For example, the user expresses a positive sentiment for a target "Obama" by explaining the "healthcare" in the given tweet: "I absolutely love Obama's policies, especially the healthcare". We ignore the future transitive verbs when followed by the query terms. e.g. "Obama will be swearing in January" does not add any extra value to the prediction.

It is very normal to have a comparative opinion providing the sentiment for multiple targets in a single tweet. As an example, "Barack Obama was far better than Romney in second debate" gives a positive review about Obama while providing a negative review for Romney. If we follow only the words polarity then it does not allow fetching the sentiment for multiple targets. Hence, we take all the nouns into consideration for the sentiment analysis.

We ensure that generic terms such as "USA", "voters" are ignored during extended features extraction and thereby ensuring the exclusion of non-subjective data. We use Stanford POS[4] tagger for extracting the significance of the word in that particular sentence.

Terms extracted during this phase indicate all the factors that influence the output of sentiment analysis for each party or the candidate. We analyze top terms that affect sentiment for each candidate using direct and extended features.

3.4 Machine-learning methods

We applied polarity-based classification methods using a set of positive and negative words provided by NLTK API[5] for python.

$$\text{Polarity} = \frac{P(\text{Positive\_Words})/\text{P}(\text{Total\_Words})}{P(\text{Negative\_Words})/P(\text{Total\_Words})} \quad (1)$$

However, this method works only for independent features based on the Standard English dictionary but fails to capture the query specific sentiments. Further, this polarity-based method also fails to capture the sentiments expressed in terms of comparison. Hence, we employed machine-learning-based techniques to meet the requirements of classification. We used the supervised algorithms such as Multinomial Naïve Bayes, SVM, MaxEnt and feed-forward

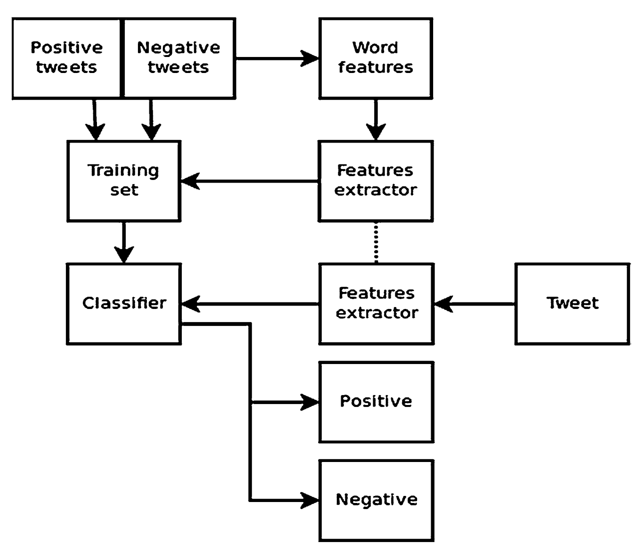

Fig. 2 Flow of supervised learning classifier

Table 2 Example of tweet categorization

Sentiment	Tweet
Positive	@john #Obama's policies are really progressive
Negative	I completely disagree with #Barack Obama's economic policy
Neutral	@BarackObama speech live in 15 min

ANN. Figure 2 shows the training and classification workflow for any supervised classification algorithm.

Table 2 provides an example of positive, negative and neutral tweets each.

3.4.1 Tweet labeling mechanism

We initially labeled only 500 positive and 500 negative tweets for both the candidates for training set with. We added correctly classified tweets to the training dataset at the end of every iteration and thereby increasing dataset size up to 10,000 tweets gradually. At the end, these labeling were cross-checked by three different members. We used accuracy matrix to measure the performance of sentiment classification. We calculated classification for the final iteration with tenfold cross-validation.

3.4.2 Supervised classifiers

1. Naïve Bayes: Naïve Bayes is a simple probabilistic model based on the Bayes rule with independent feature selection and works very well on text categorization (Chen et al. 2011). It allows a flexible way for dealing with any number of attributes or classes. Further, Naïve Bayes is an asymptotically fastest learning algorithm that examines all its training input. In this paper, we used a multinomial Naive Bayes model (Kibriya et al. 2004) for sentiment classification. A class variable C^* which defines the sentiment for a given tweet d is given by

$$C^* = \arg \mathrm{Max}_c P_{\mathrm{NB}}(C|D) \quad (2)$$

where P_{NB} (C/D) is the Bayes probability defined as

$$P_{\mathrm{NB}}(C/D) = \frac{\left(P(c)\sum_{i=1}^{m} P(f|c)^{n_i(d)}\right)}{P(d)} \quad (3)$$

In this formula, f represents a feature and n_i (d) represents the count of feature f_i found in tweet d. There are a total of m features. Parameters $P(c)$ and $P(f \mid c)$ are obtained through maximum likelihood estimates (Go et al. 2010b).

If we encounter a word during the classification phase, but this word was not found during the training phase, then the probability of both positive and negative classes will be zero. To resolve this ambiguity, we need to allow equal probability which is achieved using Laplacian smoothing constant $k = 1$.

$$\frac{\mathrm{term\_count} + k}{\mathrm{Total\_Terms} + k|c|} \quad (4)$$

2. Support vector machines: SVMs are proven to be highly efficient at text categorization, generally outperforming Naive Bayes (Joachims 1998). We considered large margin classifier to achieve better accuracy of the classification process (Christinani 2000).

$$\vec{w} = \sum_i \alpha_i c_i d_i, \quad \alpha_i \geq 0 \quad (5)$$

We used the NLTK SVM$^{\mathrm{light}}$ API[7] with a linear kernel and all the parameters were set to default values. SVM inputs are data vectors which are fed in sets. Each vector entry is of size m and each entry in the vector represents a feature. For example, with a unigram feature extractor, each feature is a single word found in a tweet. If the feature is present then the value can be set to 1; but if the feature is absent then the value can be set to 0. We use feature presence, as opposed to a count, so that we do not have to scale the input data, which speeds up the overall processing (Go et al. 2010b; Bermingham and Smeaton 2011).

3. Maximum entropy: MaxEnt classification is another technique which has been widely used in a number of natural language processing applications (Berger et al. 1996). Nigam et al. (1999) showed that at times, but not always, MaxEnt outperforms Naïve Bayes for standard text classification. The basic idea behind MaxEnt model is that one should prefer the most uniform model that satisfies a given constraint for the classification (Romero et al. 2011). In a two-class scenario, it is the same as using logistic regression to find a distribution over the classes. Further, MaxEnt does not make any assumption regarding feature independence. This means we can add features such as Bigrams and phrases to MaxEnt without worrying about features overlapping. For example, if we encounter a feature "best" and another "world's best", then their probabilities would be multiplied as though independent, even though the two are overlapping in case of Naïve Bayes but not for MaxEnt.

[7] http://svmlight.joachims.org/.

The model is represented by the following:

$$P(C|D) = {}^{1}\!/_{Z(d)} \exp\left(\sum_{i} \lambda_{i,c} F_{i,c}(d,c)\right) \tag{6}$$

where, c is a class, d is the tweet, λ is the weight vector, $Z(d)$ is a normalization function. $F_{i,c\psi}$ is a feature/class function for feature f_i and class c, defined as follows:

$$f_{i,c}(d,c') = \begin{cases} 1, & n_i(d) > 0 \text{ and } c' = c \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The weight vectors decide the significance of a feature in classification. A higher weight means that the feature is a strong indicator for that particular class and vice versa. The weight vector is found by numerical optimization of the λ so as to maximize the conditional probability. We used Gaussian-prior with ten iterations to gain the sufficient weight accuracy as suggested by Pang and Lee (2002b).

4. Artificial neural network (ANN): the central idea of a neural network which is one of the most popular and effective forms of learning system is to derive features from linear combinations of the input data, and then model the output as a nonlinear function of these features (Hastie 2001).

Neural networks offer several advantages such as adaptive learning, parallelism, fault tolerance, and generalization. In general, neural networks can be classified into two categories, feed-forward and feedback networks. In this study, we considered the feed-forward ANN because of its superior classification ability (Chen et al. 2011).

Among the feed-forward networks, back-propagation network (BPN) is one of the best known networks. BPN is based on an iterative gradient algorithm which is designed to minimize the mean square error between the actual output of a multilayer feed-forward perceptron and the desired output. Based on the rule of thumb and available literature, the number of hidden layers could be one or two (Chen et al. 2011).

The back-propagation algorithm includes both forward and backward passes. The purpose of the forward pass is to obtain the activation value, and the backward pass is to adjust the weights and biases based on the difference between the desired and the actual network outputs. These two passes will continue iteratively until the network converges. The pseudocode of the feed-forward network training by back-propagation algorithm as prescribed by (Chen et al. 2011) is given in Algorithm 2.

Here, all the optimal settings, number of hidden neurons, learning rate, etc. for the ANN are achieved by trial-and-error method. We achieved the output in terms of positive and negative values from the output layer.

Algorithm - 2: Feed Forward Network Training by Back Propagation

While error ε is too large

for each training input Ti:

1.1. Apply the inputs to the network.
1.2. Calculate the output for every neuron from the input layer, through the hidden layer hi, to the output layer Oi.
1.3. Calculate the error at the output layer Oi.
1.4. Use the output error to compute error signals for pre-output layers.
1.5. Use the error signals to compute weight adjustments.
1.6. Apply the weight adjustments.

end for

2. Periodically evaluate the network performance.

3.5 Incorporating sentiment analysis

For an event like an election, the relative sentiment between the parties is essential for predicting the outcome of the event. However, sentiment distribution for the tweets corresponding to a single party exhibit the sentiment towards that particular party but, it is not sufficient to predict an outcome of an entire event. Further, on social networks like Twitter it is difficult to achieve other ground level parameters such as age, race, religion, etc. Hence, the user behavior can be exploited to predict electoral events with other parameters such as the user bonding, connectivity, ego-networks, etc.

In terms of electoral competition, every tweet adds to the positive or negative influence towards the participating party one tweeted about. In addition to the tweet's original sentiment, user's individual influence via ones' mentions, retweets and followers can be considered to establish the influence factor (Bermingham 2001; Skoric et al. 2012).

It is observed that mention and retweets share a high correlation coefficient when tested with Spearman's rank correlation coefficients on Twitter-based social network analytics (Meeyoung et al. 2010) and the details of Spearman's rank correlation coefficient given by Eq. (8).

$$\rho = 1 - \frac{6\sum (x_i - y_i)^2}{N^3 - N} \tag{8}$$

It is also observed that high number of user mentions or followers do not necessarily signify the influence of a user over Twitter-based social network analytics (Rosenman Evan 2011). Hence, there is a need to experiment with retweets to develop the influence factor-based model.

3.6 Why retweets

While studying tweet and retweet behavior, Stieglitz and Dang-Xuan (2012) observed that 47 % of the users prefer retweet rather than publishing a tweet. If we ignore retweets there is a possibility of missing a high amount of sentiments and thereby getting low prediction accuracy. Hence, we need to consider retweets while performing prediction using sentiment analysis based on Twitter data.

Retweets can be carried out by two different methods:

1. Retweets that are done with a text "RT" or quotation mark preceding the original text. Such retweets are generally done with a comment appended before the term "RT" to add meaning to the tweet.

Let us consider a negative tweet about a term "Romney" as per the following details:

"@shazman economic policies of Romeny sucks"

However, a retweet with RT and a comment at the start will change the entire sentiment as per the following details:

"oh really? he rocks man RT @shazman economic policies of Romeny sucks"

In such cases we need to consider the sentiment of the comment as a tweet sentiment.

2. Retweets can also be done by the default "retweet" action allowed by Twitter via a "retweet" button. In this scenario, the original tweet is shared without any modification with the user's network and a retweet count for original tweet is increased by 1.

Retweets done with the first method generally adds up to the sentiment of the tweet as described above. Hence, a retweet with comment can be considered as a different sentiment. However, if a retweet is done by the first method and if it lacks the comment and also represents the original tweet without any modifications then we increment the count for retweets by 1. On the other hand, the retweets carried by the second method generally shows the agreement with the content and this retweet count is appended with the original tweet meta-data.

3.7 Establishing the influence factor

Assuming social network scenario as a graph; where, each candidate is represented as a node and the number of mentions depicts the in-degree of the node and retweets as the extended out-degree. Retweet helps us understanding the overall connectivity of the candidate and the number of people who agree or disagree by closeness factor. Closeness factor explains how fast a person can reach others in the network.

Closeness factor allows in analyzing the reach and expanded reach of the node in a graph. To establish the centrality factor with retweets, we use the ratio of summation of positive/negative tweets and it's retweets to the total positive/negative tweets as positive/negative influence.

Let us consider all original positive tweets for a party denoted by $\mathrm{T_{pos}}$ and negative tweets by $\mathrm{T_{neg}}$. Let us also consider the corresponding retweets denoted by $\mathrm{RT_{pos}}$ and $\mathrm{RT_{neg}}$. Further, the total positive and negative tweets are represented by $\mathrm{N_{pos}}$ and $\mathrm{N_{neg}}$, respectively. Using these notations, we define the influence factors for positive and negative polarities given in Eqs. (9) and (10), respectively.

$$\mathrm{Inf_{pos}} = \frac{\sum\left(\mathrm{T_{pos_i}} + \mathrm{RT_{pos_i}}\right)}{\mathrm{N_{pos}}} \tag{9}$$

$$\mathrm{Inf_{neg}} = \frac{\sum\left(\mathrm{T_{neg_i}} + \mathrm{RT_{neg_i}}\right)}{\mathrm{N_{neg}}} \tag{10}$$

We also use the sentiment volume-based prediction method as described by Asur and Huberman (2010) and accordingly we modified this model with the help of the influence factor. We evaluate the influence factor with the linear regression method as,

$$y = \beta t_i(A - B) + \beta t_j(C - D) \tag{11}$$

where, βt_j represents the weight coefficient for a sentiment score of tweets; βi_j represents the influence coefficient for a party j; further, A and B represents the actual positive and negative sentiment scores, respectively; C and D represent the positive and negative influence score obtained by Eqs. (9) and (10), respectively.

3.8 Time complexity analysis

In general, supervised learning algorithms increase the time complexity for the entire process of predicting the outcome of the electoral events. However, the model presented here does not cost more than the classification algorithms and the details of time complexity analysis for each stage are given below.

- Data cleaning stage: This stage takes iteration over all the tweets in the dataset and all the words of the tweet. Hence, it requires up to O (total tweets × average word count).
- Feature extraction stage: This task is performed with the data cleaning stage and the extended feature model may require $\mathrm{T}(n) = \mathrm{O}(n^2)$ + parsing time, where n is the number of tweets in the dataset.
- Sentiment evaluation stage: In majority of the cases with supervised learning algorithms, time complexity depends upon the number of iterations or the number of classes taken into consideration. Naive Bayes takes $\mathrm{O}(|D|\mathrm{Lave})$ where Lave is the average length of a document D. On the other hand, MaxEnt and ANN

Table 3 Effects of data cleaning and normalization

Process step	Time complexity
Data cleaning stage	O (|T| × average WordCount)
Feature extraction	O (n^2) + parsing time
Sentiment evaluation stage	O (|D|L_{ave}) or O (n^3 × number of iterations)
Prediction stage	O (1)

Table 4 Effects of data cleaning and normalization

Reduction method	# of features	% of original
None	1,468,285	100
User names	462,069	31.47
URLs	280,148	19.08
Repeated words	49,921	3.4
Repeated characters	25,841	1.76
Duplicate tweets	174,579	11.89
Candidate account tweets	256,656	17.48

depend on the number of iterations taken into consideration or the rate of convergence whereas, SVM may require a time complexity of O (n^3 × number of iterations).

- Prediction stage: The proposed model requires the tweet sentiment value and the retweet count of the same tweet, which are established during the sentiment evaluation and data collection stages in the order. This stage requires the time complexity of O (1) to evaluate the prediction stage.

Neglecting the lower order values, overall time complexity turns out to be $\theta(n^3)$ on approximation. Table 3 summarizes the time complexity of all the stages in the process.

3.9 Simulation environment

We used Intel® core™2 Duo, 3.0 GHz processor with 3 GB of RAM on Ubuntu 11.10 (Oneiric Ocelot) operating system to perform the experiments.

4 Results and discussion

4.1 US Presidential Elections 2012

In the first experiment, we considered an election scenario of a developed country like USA and thereby considering US Presidential Elections 2012 into an account for sentiment-based prediction analysis over Twitter. We collected the dataset with the use of the official Twitter API containing tweets in English language, for the original terms and extended features. We collected approximately 150,000 tweets for basic terms for US Presidential Elections 2012.

Table 5 Effects of data cleaning and normalization

Algorithm	Total time (s)	Average time/text (s)
Whitespace	1.405	0.0001
Treebank	8.315	0.001
Tweet Aware	15.915	0.0014

Data cleaning and normalization process leaves us with 49.1 % of original data for total tokenized feature vectors. Table 4 shows the results after performing normalization step and it is observed that urls are found in many tweets which are not meaningful for sentiment analysis. It is also observed from Table 5 that tweets regarding political events are carried out in the form of conversations among users via tweet replies which in turn increase the usernames in tweets.

We employed whitespace, treebank and custom tweet aware tokenizers for the process of tokenizing data for feature vector extraction. Table 5 shows the comparison of time taken by various tokenizers for the process of tokenization achieved for approximately 6,000 tweets and 50 rounds.

We used unigram, bigram and hybrid (unigram + bigram) to achieve the sentiment values for each model. Hybrid model provides the best results with all the classifiers when mixed with modified extended feature vector model. We also used negative words as part of Bigram vectors in our model.

We used precision matrix as explained in Eq. (12) to calculate accuracy of the supervised classifiers. These results are incorporated with influence factor as explained in Sect. 3.5. We use tenfold cross-validation method to find the accuracy of the prediction model by choosing tweets from dataset at random,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

where, TP, true positive; TN, true negative; FP, false positive and FN, false negative.

We considered datasets of 10,000 tweet feature vectors selected at random to carry out prediction. Figure 3 provides the results for sentiment analysis using different supervised machine-learning algorithms and it is observed that SVM achieves the maximum sentiment accuracy of ~88 % when used with hybrid (unigram + bigram) feature extraction model. On the other hand, Naïve Bayes achieves the minimum accuracy of ~68 % when used with Bigram model and this is due to the feature overlapping

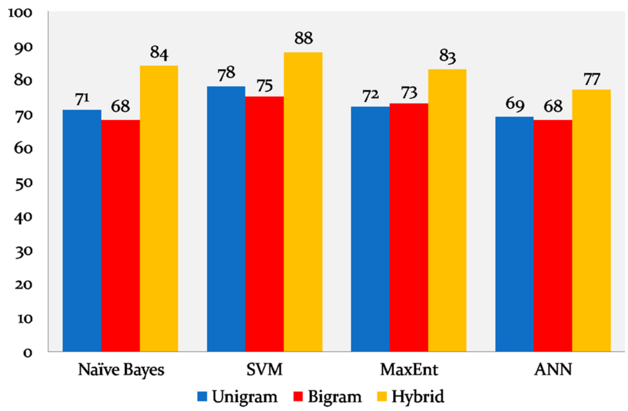

Fig. 3 Results for machine-learning algorithms

cases as explained in Sect. 3.4. Further, it is also observed that MaxEnt achieves reasonable accuracy of ~83 % with the hybrid model by properly addressing the feature overlapping. ANN achieves accuracy of ~77 % with Hybrid model but it is found to be the most time consuming for the training period and also requires a very high amount of training data for convergence for tweets with diversified topics.

Feature presence verification works more accurately for sentiment analysis using SVM when compared to MaxEnt and Naïve Bayes (Pang and Lee 2002a). Despite structural similarities in the output function of SVM and ANN, the models differ in the way the solutions are obtained. An important advantage of SVM over ANN lies in the optimization approaches. SVM obtains the support vectors in a convex optimization problem, which always find the global minimum and thereby finding a unique solution. Whereas, ANN is trained with the gradient descent method and it may not converge to the optimal solution (Chen et al. 2011). Figure 4 represents the sentiment scores for both the US Presidential candidates over the time.

The prediction model explained in Eq. (11) is independent of the feature extraction methods and classifiers employed for sentiment analysis. However, the prediction model depends on the outcome of the sentiment analysis. Since, SVM achieves the maximum sentiment accuracy we decided to utilize the results for prediction with influence factor in terms of retweet data. We allowed an error rate of 2 % when compared to the ground-truth to accept the successful results of prediction as a success and accordingly achieved average accuracy of ~88 % over tenfold accuracy matrix. It is notable that with sentiment analysis the maximum accuracy was 88 %. The details are as explained in the following Fig. 5.

The proposed method provides better accuracy with growing data as it can be seen in Fig. 6. *X*-axis shows the number of tweets in thousands for training datasets whereas; *Y*-axis shows the accuracy in percentage. The gradual improvement in exponential accuracy is due to the extended term feature model; however this performance will be stabilized once the threshold values ($K = 500$, $N = 20$) are achieved. Thus, extended features will increase in the initial training phase and gets stabilized after a few iterations. With the help of extended feature model we derived the top 4 terms related to both the US Presidential candidates as shown in Table 6. We

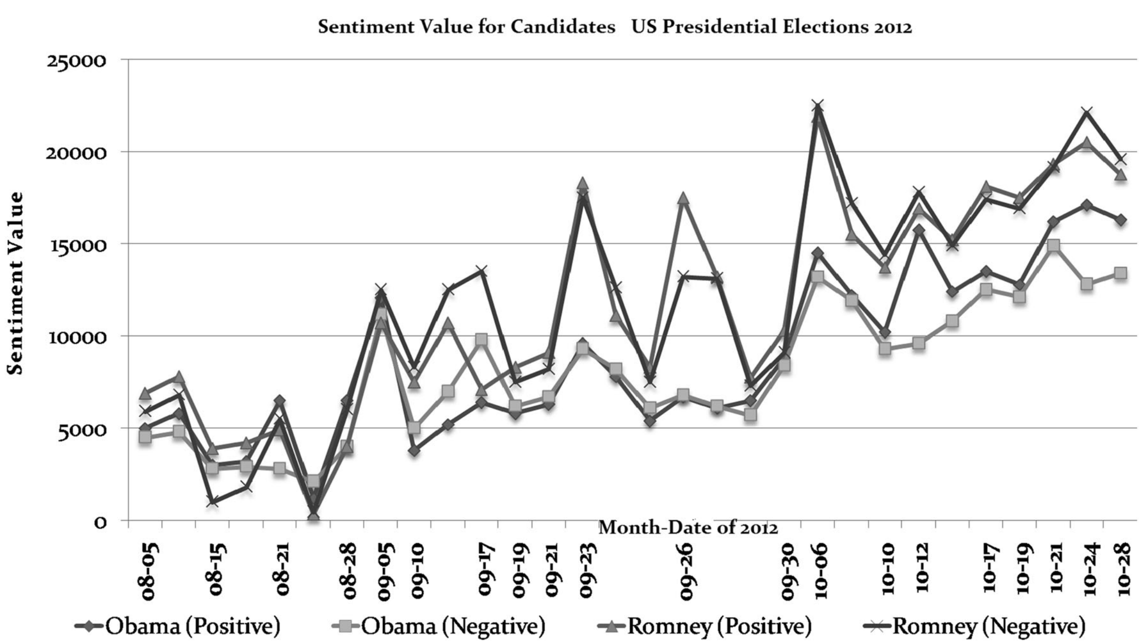

Fig. 4 Sentiment score of candidates

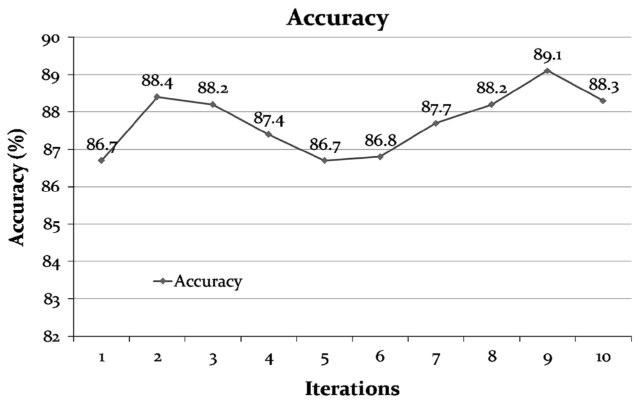

Fig. 5 Tenfold prediction accuracy

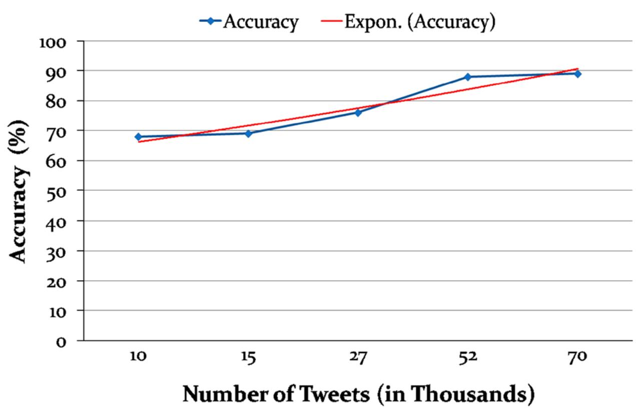

Fig. 6 Accuracy growth with extended feature model

Table 6 Major terms derived using extended features

Candidate	Major terms
Barak Obama	#obamarcare, $lgbt, Job, benghazi
Mitt Romney	#taxPlan, debate, massachusets, banker

considered the terms that occur at least 500 times in the entire dataset in a required manner as explained in Sect. 3.3 of methodology.

We compare our results with prediction results of uselectorals.org[8] for popular votes. Like any other method, our proposed method also faces the constraints of real voting compared to the social media voting scenario.

As per Twitter data collected for US election specific terms, Democratic candidate Mr. Barack Obama was ahead by a margin of 10.8 % on popular votes by getting 55.4 % votes on Twitter to Republican candidate Mr. Mitt Romney's vote share of 44.6 %. The results deviates with ±5 %, compare to real results is due to many real-time on field parameters such as, other independent candidates, huge number of users not clear about their voting opinion, lack of clarity on Twitter, missing location details on Twitter, etc.

4.2 Karnataka (India) State assembly elections 2013

We also considered Karnataka (India) State assembly elections 2013 which were held in April–May 2013 to study an entirely different electoral scenario in a developing country like India. The results show an entirely different picture of the Twitter-based prediction method with an accuracy of 68 % over 10 iterations and the vote share results swinging in favor of Bhartiya Janta Party (BJP) with other competitive parties being Indian National Congress (INC), Janta Dal Secular (JDS) and Kanataka Janta Party (KJP).

We followed a similar procedure of US Presidential Elections 2012 for Karnataka (India) State Elections 2013 to predict election results by gathering over 23,000 tweets. However, only 8,000 tweets from the general public were found in a proper format based on the filtering process carried out over the tweets from both the accounts of official and press/media. Further, cleansing and normalization phase has removed duplicated tweets with "RT" and plagiarized tweets. This process reduced the tweets from 8,000 to 5,000 for final processing and thereby providing a better chance to predict the elections using influence factor. We employed Hinglish (Hindi words written in English fonts) word lists to parse Indian tweets written in regional language (Hindi). We adopted the same labeling and training strategy of US Presidential Elections 2012 for Karnataka (India) State Elections 2013.

Since, SVM with Hybrid feature extraction model produced the maximum sentiment accuracy for US Presidential Elections 2012; hence, we considered the same procedure for Karnataka (India) State Elections 2013. After initial experiments, the prediction accuracy is found to be 58 % for sentiment analysis without taking retweet data into account. Further, this prediction accuracy is raised from 58 to 62 % by incorporating the influence factor with the sentiment analysis. This 4 % improvement in the prediction accuracy signifies the role of the influence factor generated by retweet data which is present in large numbers in the dataset of Karnataka Elections 2013. Still there is a difference of 26 % between prediction results of US Presidential Elections 2012 and Karnataka (India) Elections 2013. This discrepancy is due to the skewed tweets resulting in unbalanced data in large numbers towards two major political parties participated in Karnataka Elections 2013. Table 7 shows the results of the political parties in terms of vote share percentage based on Twitter data in comparison to the actual results announced by Election Commission of the Government of India for Karnataka State Elections 2013.

[8] http://www.uselectorals.org.

Table 7 Vote share based on Twitter sentiment analysis

Party name	Experiment results (%)	Original results (%)
BJP	37	20
INC	23	37
JDS	12	20
KJP	28	11
Other	0	12

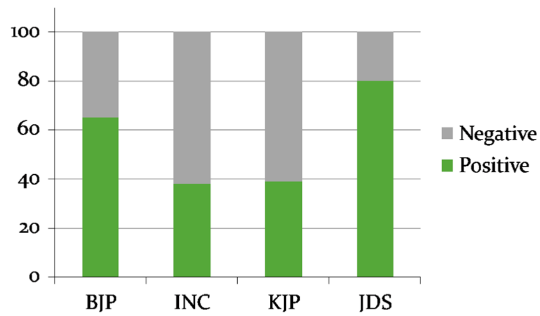

Fig. 7 Sentiment variation for parties of Karnataka

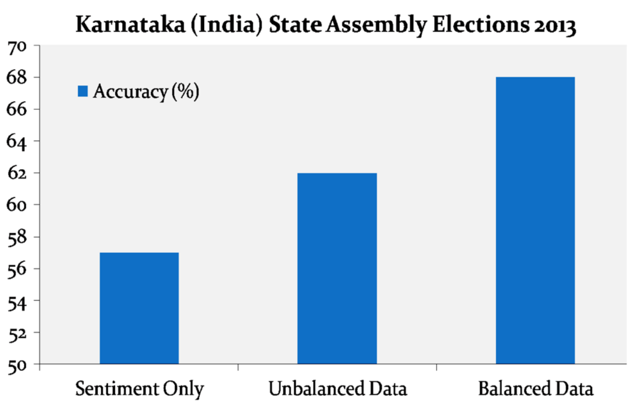

Fig. 8 Accuracy comparison for SVM with datasets

A large chunk of tweets skewed towards two parties BJP and INC on Twitter affects the accuracy for Twitter votes; in addition to the lack of presence of other participating parties such as JDS, KJP, etc., resulted in reducing the accuracy. The overall more number of negative tweets and corresponding retweets add up to negative value and lesser prediction accuracy towards the results as shown in Fig. 7.

To provide better prediction accuracy results for Karnataka Elections 2013 we need to overcome this aforementioned unbalanced data by balancing the Karnataka dataset. Hence, we considered 1,000 tweets for each participating political party and thereby increasing the prediction accuracy from 62 to 68 % as shown in Fig. 8.

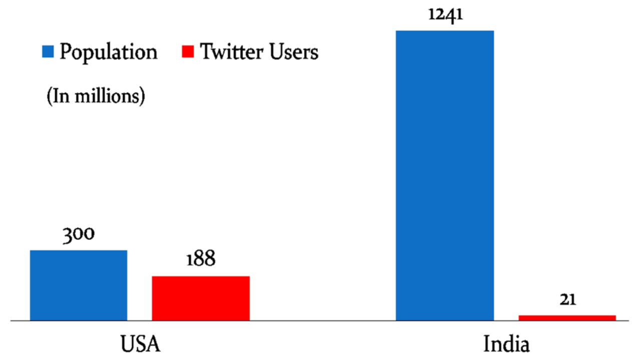

Fig. 9 Population and Twitter users of USA and India

Table 8 Major terms Derived using extended features

Candidate	Major terms
BJP	Development, #pinkRevolution, IT, #NammaMetro
INC	#MinningScam, Yeddy, #JusticeHegde, #corruption
JDS	#MinningScam, Hassan, #JusticeHegde, #corruption
KJP	Yeddy, Vishwashgatuka, #JusticeHegde

Twitter-based poll prediction results also depend on what percentage of the actual population that is going to cast their vote compared to the actual opinions expressed over Twitter. It is evident from Fig. 9 that United States has very active Twitter population count when compared to that of India.

Further, the lack of accurate locations does not allow extracting ground-truth results for state level information. This will also lead to miscalculation when a Twitter user residing in state A tweets about state B and vice versa but in an actual election scenario Twitter user from state A will not be able to cast his(her) vote in state B and vice versa. As an example, a person residing in any other Indian state like Delhi may also tweet related to Karnataka (India) State Assembly Elections, but the user will not be allowed to cast his vote in actual elections since he(she) is not actually residing Karnataka state.

Table 8 shows the influencing terms related to Karnataka State Assembly Elections 2013 extracted using the extended feature model with ($K = 150$, $N = 15$).

5 Conclusion

A social network like Twitter that allows public opinion provides a great platform in measuring the public opinion with the reasonable accuracy. We studied various supervised classifiers for sentiment analysis over Twitter data with various feature extraction models such as unigram, bigram, hybrid (unigram + bigram). We proposed a novel methodology to extract the terms influencing the event by modifying the existing extended features model. Further, we also proposed a model to establish the influence factor generated with retweet over Twitter. We used this influence factor model to predict the electoral event by incorporating it with sentiment analysis. To study opinion extraction over Twitter, we considered two different election scenarios of developed nations (USA) and developing nations (India). We achieved the prediction accuracy of $\sim 88~\%$ in the case of US Presidential Elections 2012 using SVM with Hybrid feature vector model for sentiment analysis. On the other hand, we achieved $\sim 68~\%$ prediction accuracy for Karnataka (India) State Assembly Elections 2013. We conclude that Twitter does appear to display a predictive quality which is improved by incorporation of the influence factor based on the user behavior analysis or other social network parameters for carrying out prediction using sentiment analysis. Further, the inclusion of more influential factors based on the personal details such as gender, age, educational background, employment, economic criterion, rural and urban and social development index will further improve the poll prediction accuracy to even higher levels. Our proposed model weighs all the Twitter user sentiments equally and thereby achieving reasonable prediction accuracy for a participating party. It may be possible to weigh selected celebrity users' sentiments more than other Twitter users and it may lead to higher prediction accuracy of any event using the influence factor. Since the Twitter data is increasing exponentially, the future requirement of text-based sentiment analysis should be efficient and fast. Further, hybrid classifiers need to be parallelized to provide real-time high performance computing results for prediction of any event.

References

Aragón P et al (2013) Communication dynamics in Twitter during political campaigns: the case of the 2011 Spanish national election. Policy & Internet 5(2):183–206. doi:10.1002/1944-2866.POI327

Berger AL, Della Pietra VJ, Della Pietra SA (1996) A maximum entropy approach to natural language processing. J Comput Linguist 22(1):39–71

Asur S, Huberman B (2010) Predicting the future with social media. In: WI-IAT'10 Proceedings of the 2010 IEEE/WIC/ACM International Conference on web intelligence and intelligent agent technology, vol 01, pp 492–499

Boutet A et al (2013) What's in your tweet: I know who you supported in the UK 2010 general elections. In: Association for the Advancement of Artificial Intelligence

Bakliwal A (2013) Sentiment analysis of political tweets: towards an accurate classifier. In: Proceedings of the Workshop on language in social media (LASM 2013), Atlanta, Georgia, pp 49–58

Barbosa L, Feng J (2010) Robust sentiment detection on Twitter from biased and noisy data. In: Proceedings of COLING

Bermingham A, Smeaton AF (2011) On using Twitter to monitor political sentiment and predict election results. In: Proceedings of the Workshop on sentiment analysis where AI meets psychology (SAAIP), IJCNLP 2011, Chiang Mai, Thailand, pp 2–10

Bollen J, Pepe A, Mao H (2011) Modeling public mood and emotion: Twitter sentiment and socioeconomic phenomena. In: Proceedings of the Fifth International AAAI Conference on weblogs and social media (ICWSM 2011), Barcelona, Spain

Chen LS, Liu CH, Chiu HJ (2011) A neural network based approach for sentiment classification in the Blogosphere. Elseveir J Inf 5:313–322

Cozma R, Chen K (2011) Congressional candidates "use of Twitter during the 2010 Midterm Elections: a wasted opportunity?". In: 61st Annual Conference of the International communication association

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Davidiv D, Tsur O, Rappoport A (2010) Enhanced sentiment learning using Twitter hashtags and smileys. In: Proceedings of the 23rd international conference on computational linguistics, 2010, pp 241–249

Doshi L (2008) Using sentiment and social network analyses to predict opening-movie box-office success in fulfilment of Master's degree proceedings at Department of Electrical and Computer Engineering, Massachusetts Institute of Technology

Gayo-Avello D, Metaxas PT, Mustafaraj E (2011) Limits of electoral predictions using Twitter. In: International conference on weblogs and social media. IAAA, Barcelona, Spain

Go A et al (2010a) Sentiment analysis of Twitter posts about news. Natural language processing. Stanford University

Go A et al (2010b) Twitter sentiment classification using distant supervision. Natural language processing. Stanford University

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer, New York

Jiang L et al (2011) Target-dependent Twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the association for computational linguistics

Katakis I et al (2013) Social voting advice applications—definitions, challenges, datasets and evaluation. In: IEEE Transactions on cybernetics

Kibriya A et al (2005) Multinomial Naive Bayes for text categorization revisited—AI 2004. Adv Artif Intell Lect Notes Computer Sci 3339:488–499

Livne A, Simmons MP, Adar E, Adamic LA (2011) The party is over here: structure and content in the 2010 election. In: International conference on weblogs and social media. IAAA, Barcelona, Spain

Meeyoung C et al (2010) Measuring user influence in Twitter: the million follower fallacy. In: Fourth International AAAI Conference on weblogs and social media

Mejova Y, Srinivasan P, Boynton B (2013) GOP primary season on Twitter: "popular" political sentiment in social media. WSDM'13, February 4–8, 2012, Rome, Italy

Moraes R, Valiati JF (2013) Document-level sentiment classification: an empirical comparison between SVM and ANN. Elseveir Trans Expert Syst Appl 40:621–633

Nigam K, Laverty J, Mccallum A (1999) Using maximum entropy for text classification. In: IJCAI-99 Workshop on MACHINE learning for information filtering, pp 61–67

Nooralahzadeh F, Arunachalam V, Chiru C (2013) 2012 Presidential Elections on Twitter—an analysis of how the US and French Election were reflected in Tweets. In: CSCS '13 Proceedings of the 19th International Conference on Control Systems and

Computer Science, 2013, pp 240–246. doi:10.1109/CSCS.2013.72
O'Connor B, Balasubramanyan R et al (2010) From tweets to polls: linking text sentiment to public opinion time series. In: Proceedings of the International AAAI Conference on weblogs and social media, Washington, DC
Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the Seventh International Conference on language resources and evaluation (LREC'10)
Pang B, Lee L (2002a) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp 115–124. doi:10.3115/1219840.1219855
Pang B, Lee L (2002b) Sentiment classification using machine learning techniques. In: Proceedings of the Conference on empirical methods in natural language processing (EMNLP), Philadelphia
Parikh R and Movassate M (2009) Sentiment analysis of user-generated Twitter updates using various classification techniques. Stanford University, Stanford
Pew Research Center (2010) Parsing Election Day media: how the midterms message varied by platform. Pew
Romero DM, Meeder B, Kleinberg J (2011) Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In: Proceedings of the 20th International Conference on World Wide Web (WWW). doi:10.1145/1963405.1963503
Rosenman Evan TR (2011) Retweets but not just retweets. Submitted to Applied Mathematics in partial fulllment of the honors requirements for the degree of Bachelor of Arts, Harvard College, Cambridge, Massachusetts March 30, 2012. http://people.seas.harvard.edu/~mruberry/erosenmanthesis.pdf
Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web 2010, ACM, pp 851–860. doi:10.1145/1772690.1772777
Skoric M, Poor N, Achananuparp P, Lim E, Jiang J et al (2012) Tweets and votes: a study of the 2011 Singapore general election. In: Proceedings at 2012 45th Hawaii International Conference on system sciences
Song M, Kim Chul M (2013) RT2M: real-time Twitter trend mining system. In: 2013 International Conference on social intelligence and technology. doi:10.1109/SOCIETY.2013.19
Stieglitz S, Dang-Xuan L (2012) Political communication and influence through microblogging: an empirical analysis of sentiment in Twitter messages and retweet behavior. In: 2012 45th Hawaii International Conference on system sciences. doi:10.1109/HICSS.2012.476
Tan et al (2011) User-level sentiment analysis incorporating social networks. In: Proceedings of the Sixteenth ACM SIGKDD International conference on knowledge discovery and data mining
Turney PD (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL
Zhou X, Tao X, Yong J, Yang Z (2013) Sentiment analysis on tweets for social events. In: Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design, CSCWD 2013, pp 557–562. http://dx.doi.org/10.1109/CSCWD.2013.6581022