Election Result Prediction Using Twitter sentiment Analysis

Jyoti Ramteke
Department of Computer Engineering
Sardar Patel Institute of Technology
Mumbai – 400058, India
jyoti_ramteke@spit.ac.in

Darshan Godhia
Department of Computer Engineering
Sardar Patel Institute of Technology
Mumbai – 400058, India
darshan1810@gmail.com

Samarth Shah
Department of Computer Engineering
Sardar Patel Institute of Technology
Mumbai – 400058, India
samarthshah10@gmail.com

Aadil Shaikh
Department of Computer Engineering
Sardar Patel Institute of Technology
Mumbai – 400058, India
aady95@gmail.com

Abstract - **The proliferation of social media in the recent past has provided end users a powerful platform to voice their opinions. Businesses (or similar entities) need to identify the polarity of these opinions in order to understand user orientation and thereby make smarter decisions. One such application is in the field of politics, where political entities need to understand public opinion and thus determine their campaigning strategy. Sentiment analysis on social media data has been seen by many as an effective tool to monitor user preferences and inclination.**

Popular text classification algorithms like Naive Bayes and SVM are Supervised Learning Algorithms which require a training data set to perform Sentiment analysis. The accuracy of these algorithms is contingent upon the quantity as well as the quality (features and contextual relevance) of the labeled training data. Since most applications suffer from lack of training data, they resort to cross domain sentiment analysis which misses out on features relevant to the target data. This, in turn, takes a toll on the overall accuracy of text classification. In this paper, we propose a two stage framework which can be used to create a training data from the mined Twitter data without compromising on features and contextual relevance. Finally, we propose a scalable machine learning model to predict the election results using our two stage framework.

Keywords— sentiment analysis, text classification, training data, labeling, Vader, Twitter

I. INTRODUCTION

The proliferation of social media in the recent past has provided end users a powerful platform to voice their opinions. Platforms like Facebook, Twitter and Google+ are being actively used to share ratings, reviews and recommendations. The authors in [1] suggest how this vast array of information can be actively used for marketing and social studies. Political campaigns have exploited this vast array of information available on the above platforms to draw insights about user opinions and thus design their marketing campaigns. Huge investments by politicians in social media campaigns right before an election along with arguments and debates between their supporters and opponents only enhance the claim that views and opinions posted by users have a bearing on the results of an election. Various sentiment analysis algorithms can be used to identify the attitude of the author w.r.t. an election candidate or a political party.

Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". It is particularly an interesting platform because of its concept of hash tags. Along with the short messages, users can use the hash tag symbol '#' before a relevant keyword or phrase in their Tweet to categorize those Tweets and help them show more easily in Twitter Search. The use of hash tags makes the problem of text classification relatively easier since the hash tag itself can convey an emotion or opinion. For instance, #MakeAmericaGreatAgain is the official hash tag for Republican Presidential Candidate Donald Trump. All tweets consisting of this hash tag would indicate support for this candidate.

In [2], the authors compare the performance of two popular sentiment analysis algorithms, namely Naive Bayes and SVM. Both the algorithms belong to the category of Supervised Learning. Supervised Learning is a branch of Machine Learning which needs a training data set to perform classification. A training data set comprises of training examples which is basically a pair consisting of an input object and a desired label (output value). The quality of classification is dependent upon the quantity as well as the quality of the training set. While performing classification, machine learning models should be provided with a training set which not only has sufficient training examples but also is contextually relevant to the problem. The authors in [3] note that procuring such a training data set is extremely difficult and is a major hindrance to classification problems.

Major supervised learning text classification algorithms rely on extracting features from training data set, assigning weights to the features (depending on their frequency or some user criterion) and then using the weighted features to classify test data set. Due to a lack of contextually relevant training set, researchers generally use a cross domain training set for performing text classification as illustrated in [4]. The most common example of this technique is using the popular IMDB data set which consists of 25000 manually labeled movie reviews. This technique however misses out on an important aspect of contextual relevance because the features extracted from movie reviews need not necessarily match the features of the target data set. Moreover, when tweets come into picture, hash tags themselves become important features. And no other data set can provide hash tags as features except the data that has been mined from Twitter for that specific application. Hence, it becomes necessary to devise a labeling technique for the mined Twitter data which can strike a balance between speed and accuracy.

The rest of the paper is organized as follows. In section II, we discuss papers which propose a few solutions to the lack of training data. Section III discusses our methodology to create a training data set relevant to elections. Section IV discusses the final machine learning model and the metric to determine the winning candidate. Finally, the conclusion and future scope is discussed in section V.

II. Literature Review

This section summarizes the various approaches for text classification used in [3] for scenarios where training data is not available. This includes unsupervised learning and cross domain sentiment analysis.

For unsupervised learning, the approaches by Turney [5], Harb et al. [6] and Taboada et al [7] are discussed. Turney [5] calculates the semantic orientation (Point wise mutual information w.r.t a positive and a negative seed word) of adjectives and verbs in a sentence and determines the overall polarity by adding up the independent values for semantic orientation. They achieved accuracy of 74% by using this technique. Harb et al. used Google search engine to define associations for positive and negative words. They then counted the total positive and negative words to determine the overall polarity of a blog. Taboada et al. [7] used dictionaries of positive and negative words to and integrated intensifiers and negation words to determine the polarity. On an average, 68% accuracy was achieved using this technique.

For cross domain sentiment analysis, the approaches by Wu and Tan [8] and Liu and Zhao [9] are discussed. Wu and Tan [8] use a two stage framework as follows: At the first stage, an association is created between the source and the target domain by applying a graph ranking algorithm [10]. Then some of the best seeds from the target domain were selected. At the second stage, they used the essential structure to calculate the sentiment score of each documents and then the target-domain documents were labeled based on these scores. Liu and Zhao [9] also propose a two stage framework. At the first stage of their method, they used a feature translator to translate a feature in source domain to a feature in target domain. In the second stage, they used the source domain data to train a classifier and used it to classify the unlabeled data in the target domain.

In both the techniques used above, the overall accuracy has been roughly 70% which is less in comparison to supervised learning methods. Thus, it reinforces the claim that an accurate and contextually relevant training data set is vital to achieve highly accurate results for text classification as illustrated in [13]. However, in [13] the authors achieve only a sparse data set of 1000 tweets which fails to satisfy the quantitative aspect required by a supervised learning algorithm.

III. Data Set Creation

A. Data Collection

Twitter data for two candidates – namely Donald Trump and Hillary Clinton were collected for the dates March 16th, 2016 and March 17th, 2016.

We used the Twitter Streaming API to fetch data relevant to the presidential candidates. The Streaming APIs give developers low latency access to Twitters global stream of Tweet data [11]. The input parameters to the streaming functions were the names of Presidential candidates and other keywords like “Democrats”, “Republicans”. Tweets corresponding to the given parameters were returned in JSON format. The JSON result basically comprised of key-value pairs. Some keys were created at, id, re-tweeted, screen name, location etc. The JSON responses were culled to extract only the body of the tweet and stored in a CSV file.

TABLE I

APPROXIMATE DATA FOR PRESIDENTIAL CANDIDATES

Candidate	***Total Tweets***
Donald Trump	61, 473
Hillary Clinton	60, 121

B. Data Preprocessing

In this stage, the tweets were stripped off special characters like '@' and URLs to overcome noise. Additionally, in the Machine Learning modules, to improve the classifier accuracy, we employ the TF-IDF (term frequency - inverse document frequency) technique, to identify terms which are more relevant to sentiments.

C. Data Labeling

This section provide our two stage framework for creating a labeled training data set This two stage framework helps in achieving a data set that is both contextual and not sparse at the same time which aligns with the requirements of a supervised learning algorithm.

Stage 1: Manual Labeling using hash tag clustering

The first stage of this framework comprises of manually labeling the Twitter data. However, the entire Twitter data set need not be labeled manually. We introduce a technique called

hash tag clustering. Often while mining data from Twitter, users can find multiple tweets consisting of the same hash tag. For instance consider the hash tag #MakeAmericaGreatAgain which is the official hash tag for the U.S. Presidential Candidate, Donald Trump. Now since this is the official hash tag for Donald Trump, it is obvious that any person who tweets with this hash tag is in favor of Trump. Hence, all tweets consisting of the hash tag #MakeAmericaGreatAgain must be labeled positively for Trump. So just by associating a label with a hash tag, thousands of tweets consisting of the same hash tag can be automatically labeled via a code. However before using this technique, it is necessary to sort the hash tags in their decreasing order of frequency. This will make sure that higher frequency hash tags get labeled prior to lower frequency hash tags. Depending on the application, developers or analysts can even choose to not label lower frequency hash tags or hash tags which are ambiguous unlike #MakeAmericaGreatAgain since they will be handled in our next section.

The emphasis on manual labeling is bolstered by the fact that a candidate maybe linked to a scam or an initiative. In the case of Benghazi controversy, which is negatively linked to Hillary Clinton, tweets consisting of #Benghazi cannot be labeled negatively for Hillary by using a cross domain data set. Such contextual data pertaining to scams, controversies or political initiatives, which is a quite common in politics, can be handled only by human intervention.

Stage 2: Using VADER to label remaining tweets

Vader (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It is basically a sentiment intensity polarizer developed by Hutto and Gilbert [12]. Vader takes a sentence as input and provides a percent value for three categories - positive, neutral and negative and compound (overall polarity of the sentence).

TABLE II

VADER SENTEMINT ANALYSIS EXAMPLES

Sentence	***comp***	***pos***	***neg***	***neu***
He is smart and funny	0.83	0.75	0.0	0.254
A horrible book	-0.82	0.0	0.791	0.20
It sux, but I will be fine	0.22	0.274	0.195	0.53

The above table provides three examples of sentences analyzed using Vader. The first sentence is highly positive, second highly negative and third one neutral. For performing sentiment analysis, a training data set should consist of sentences that are unambiguously either positive or negative. Hence, based on our observations, only those sentences having compound value >=0.8(highly positive) or compound value <=-0.8(highly negative) should be included in the training data set and the remaining sentences can be discarded. Python implementation of Vader is readily available on GitHub as an open source project.

Thus, the two stage framework proposed above can be used to create a training data set for Twitter. Based on the nature of application or the degree of fault permissible, the above two stages can be altered. For instance, the threshold of 0.8 in stage 2 can be increased or decreased depending on the needs of the user. Similarly, in cases where the frequency of sentences for individual hash tags is extremely less (like 10 or 20 sentences) then stage 1 can be directly eliminated and all sentences can be labeled using Vader.

TABLE III

LABELED DATASET USING 2 STAGE FRAME WORK

Candidate	***Stage I***	***Stage II***	***Total***
Donald Trump	17166	7321	24487
Hillary Clinton	17115	14435	31550

The total data set collected initially was roughly 60 thousand tweets for individual politicians. However, after labeling the data set using the 2 stage framework, we get roughly 30 thousand tweets for the training data set. This is because the threshold set for Vader in two stage framework is very high, 80%. As a result, ambiguous tweets or highly neutral tweets get eliminated thereby enhancing the quality of the training data set.

D. Algorithms

Various algorithms for natural language processing and more specifically sentiment analysis are available today. We used two algorithms, Multinomial Naive Bayes and Support Vector machines to determine the polarity of tweets. Python provides two packages for implementation of the above two algorithms scikit learn and nltk. We tested both the packages for performing sentiment analysis on the manually labeled data set as described in 2.3.1. The accuracy for both the algorithms has been tabulated below:

TABLE IV

ACCURACY FOR SENTIMENT ANALYSIS ALGORITHMS

Package	***Algorithm***	***Accuracy***
nltk	MNB	0.54
nltk	SVM	0.58
Scikit-learn	MNB	0.97
Scikit-learn	SVM	0.99

As seen in the table above, SVM algorithm from the Scikit-learn package provides the best accuracy for classification. Hence, we use the SVM algorithm (specifically from Scikit-learn) for our final model.

IV. PROPOSED MODEL

Now that we have a dataset, and the dataset is labeled in two stages, it can be used to train a supervised machine learning model to perform public sentiment analysis and predict election outcome. We split the dataset in 80:20 ratios to prepare the training and testing sets.

TABLE V

TRAINING AND TESTING DATA FOR CANDIDATES

Candidate	*Training*	*Testing*	*Total*
Donald Trump	19589	4898	24487
Hillary Clinton	25240	6310	31550

A. Design

For our proposed model, we perform multistage classification and identify whether the sentiment of a tweet is positive or negative w.r.t. one of the election candidates. In this regard, we first classify the tweet on the basis of the candidate that it is addressing or is relevant to. The first classifier is an 'entity classifier' which classifies a general stream of data into the respective entities. In the next stage, the classification is performed on the basis of the sentiment of the text w.r.t. that particular candidate. Thus each candidate has a classifier associated with him/her. The entity classifier is trained with the entire data set labeled by the entities. The sentiment classifier is trained with data set pertaining to only its candidate. The input to the model shown below is the test data set.

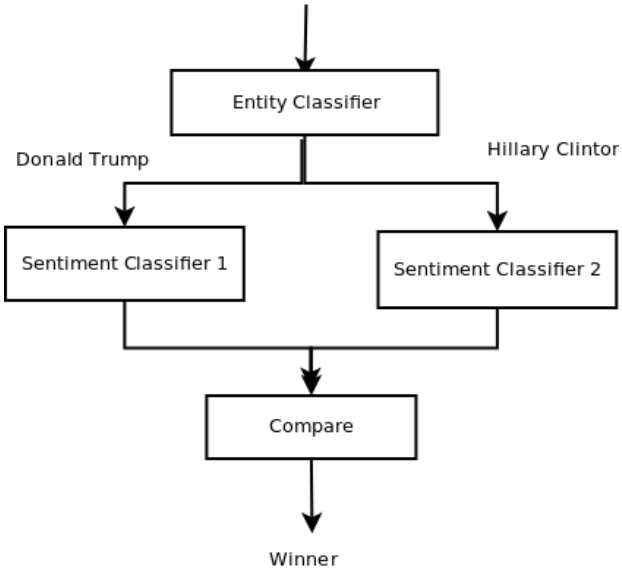

Fig. 1 Model for Election Result Prediction

B. Implementation

To implement the supervised classification model design, we compared the performance of the following classifiers on our preprocessed labeled data set. These classifiers are popularly used for text - based classification as follows:

TABLE VI

SUPERVISED CLASSIFICATION TECHNIQUES COMPARISON

Classification Technique	*F-1 Score*
SVM Linear Kernel	0.97
SVM – rbf kernel	0.39
SVM – liblinear	0.97
Naïve Bayes – MultinomialNB	0.94

Based on the metric of F-1 score, we selected the SVM with linear kernel as our entity and sentiment classifier.

- The entity classifier gave an accuracy of 0.98 when we used a training data of 50,433 tweets and testing data 5,603 tweets for classifying 'Hillary Clinton' and 'Donald Trump'
- The sentiment classifier gave an accuracy of 0.99 when we used a training data of 19,589 tweets and testing data of 4,898 tweets of 'Donald Trump'
- The sentiment classifier gave an accuracy of 0.97 when we used a training data of 25,240 tweets and testing data of 6,310 tweets of 'Hillary Clinton'.
- The outputs of the testing data from both the sentiment classifiers were as in Table VI

C. Aggregration

The winner was decided as the person having the higher Positive versus Total count ratio (PvT Ratio), calculated as

$$Ratio = |P| / |T \quad (1)$$

Here, P constitutes the tweets classified to be positive for the candidate (by the candidate's sentiment analyzer),
T constitutes all the tweets classified as related to the candidate (by the entity classifier).

TABLE VI

PvT RATIO FOR CANADIDATES

Candidate	*Positive*	*Negative*	*Total*	*PvT Ratio*
Donald Trump	2681	2170	4851	0.553
Hillary Clinton	1378	2410	3788	0.364

Direct count of positive tweets cannot be used as a metric to determine the winner since the data set count may be biased towards a particular candidate. Consider a scenario where 50,000 tweets for Donald Trump are mined out of which only 10,000 are positive and 30,000 tweets are mined for Hillary Clinton out of which 9,000 are positive then direct comparison of positive tweets would yield incorrect results since the percentage of positive tweets for Clinton is much higher.

Hence determining the percentage of positive tweets for every politician i.e. PvT Ratio will give a fair idea of the popularity of every candidate.

VI. Conclusion and Future Scope

The use of social media for prediction of election results poses challenges at different stages. In this paper, we first tackle the scarcity of training data for text classification by providing a two stage framework. Finally we propose our model for election result prediction which uses the labeled data created using our framework. While our model alone may not be sufficient to predict the results, however it becomes a crucial component when combined with other statistical models and offline techniques (like exit polls).

We implemented the proposed model on a dataset which was created by mining Twitter for 3 days. However, this model can be extended in the future to create an automated framework which mines data for months since election result prediction is a continuous process and requires analysis over long periods of time. Features should be extracted from newly mined data and compared with existing set of features. Some similarity metric can be used to compare the new and old features. Only in cases where the metric value crosses a threshold, the newly mined data should be labeled using the two stage framework. Thus we recommend creating an Active learning model wherein the model itself recommends what data should be labeled. This would minimize the efforts for labeling while making sure that there is no compromise on contextual relevance.

References

[1] Alexander Pak and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining". In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, may 2010.

[2] Y. Yang and F. Zhou, "Microblog Sentiment Analysis Algorithm Research and Implementation Based on Classification", *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)* ,2015.

[3] M. Hajmohammadi, "LACK OF TRAINING DATA IN SENTIMENTCLASSIFICATION: CURRENT SOLUTIONS", IJRCCT, vol. 1, no. 4, pp. 133-138, 2012.

[4] K. Mao, J. Niu, X. Wang, L. Wang and M. Qiu, "Cross-Domain Sentiment Analysis of Product Reviews by Combining Lexicon-Based and Learn-Based Techniques", *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, 2015.

[5] P. D. Turney, "Thumbs up or thumbs down?: semantic orientationapplied to unsupervised classification of reviews," presented at theProceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania,2002.

[6] A. Harb, M. Planti, G. Dray, M. Roche, Fran, o. Trousset, and P.Poncelet, "Web opinion mining: how to extract opinions from blogs?," presented at the Proceedings of the 5th internationalconference on Soft computing as transdisciplinary scienceandtechnology, Cergy-Pontoise, France, 2008.

[7] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede,"Lexicon-based methods for sentiment analysis," *Comput. Linguist.*, vol. 37, pp. 267-307, 2011.

[8] Q. Wu and S. B. Tan, "A two-stage framework for cross-domainsentiment classification,"*Expert Systems with Applications*, vol.38, pp. 14269-14275, Oct 2011.

[9] K. Liu and J. Zhao, "Cross-domain sentiment classification using atwo-stage method," presented at the *Proceedings of the 18th ACMconference on Information and knowledge management, HongKong, China,* 2009.

[10] Q. Wu, S. Tan, H. Zhai, G. Zhang, M. Duan, and X. Cheng,"SentiRank: Cross-Domain Graph Ranking for SentimentClassification," presented at the Proceedings of the 2009IEEE/WIC/ACM International Joint Conference on WebIntelligence and Intelligent Agent Technology-Volume 01, 2009.

[11] "The Streaming APIs | Twitter Developers", *dev.twitter.com*, 2016. [Online]. Available: https://dev.twitter.com/streaming/overview. [Accessed: 25- Apr- 2016].

[12] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[13] Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference* on. IEEE, 2013.