Application of Location-Based Sentiment Analysis Using Twitter for Identifying Trends Towards Indian General Elections 2014

Omaima Almatrafi
George Mason University
oalmatra@gmu.edu

Suhem Parack
George Mason University
sparack@gmu.edu

Bravim Chavan
George Mason University
bchavan@gmu.edu

ABSTRACT

Location based sentiment analysis is the use of natural language processing or machine learning algorithms to extract, identify, or characterize the sentiment content of a 'text unit', according to the location of origin of the text unit. In this paper, we study the application of location based sentiment analysis using Twitter for identifying trends and patterns towards the Indian general elections 2014. We perform data (text) mining on 650,000 tweets collected over a period of 5 days pertaining to two political parties in India, during the campaigning period. We make use of Naive Bayes algorithm to build our classifier and classify the test data (as positive or negative) according to it. We identify the sentiment of Twitter users towards each of the two Indian political parties, by location and plot our findings on an Indian map. In the end, we present our observations and conclusions and how certain "social events" influence the sentiments of Twitter users on the social network. We also discuss the issues related to geo-location using the data obtained from the Twitter API.

Categories and Subject Descriptors

J.4 [Computer Applications]: Social and Behavioral Sciences

General Terms

Algorithms, Experimentation, and Human Factors.

Keywords

Twitter, sentiment analysis, Indian Elections, and Naïve Bayes.

1. INTRODUCTION

Sentiment analysis is a type of natural language processing for tracking the mood of the public about a particular product or topic. Sentiment analysis, which is also called opinion mining, involves in building a system to collect and examine opinions about the product made in blog posts, comments, reviews or tweets. Sentiment analysis can be useful in several ways. For example, in marketing it helps in judging the success of an ad campaign or new product launch, determine which product or service are popular and even identify which demographics like or dislike particular features.

Sentiment analysis has been an attractive area for many research communities such as machine learning, data mining, and natural processing language. Starting from being a document level classification task, it has been handled at the sentence and more recently at the phrase level. One of the popular methods adopted is a two-step classification. In the first step, the sentence is classified as either subjective or objective (neutral). In the second step, it is classified to be positive or negative; this step is called polarity classification step [7]. 978-1-4503-3377-1

Microblog data like Twitter, on which users post real time reactions to and opinions about events or products they encounter, poses newer and different challenges. First, tweets are much shorter and contain much less content than, for instance, news articles and traditional blogs. Hence, their informational value is less clear-cut. It has been indicated in [1] that up to 40% of all Twitter messages are "pointless babble". Second, only part of the information conveyed is found in the words themselves because 19% of all messages contain links to other websites or photos. Another concern while dealing with tweets is, it consists of a lot of informal language including words such as 'wanna', 'wassup', etc [3]. Therefore, it needs to be processed before applying any classification technique on them.

In this paper, we look at one such popular microblog called Twitter and build models for classifying "tweets" into positive or negative sentiment based on geo-location. We build a model for a binary task of classifying sentiment into positive and negative classes. Although our application can be used to classify the tweets about any topic based on the location, we experiment on the data we collect during the Indian election period April 14th - April 21, 2014. We want to explore the influence of real life social events such as a candidate statement to people's opinion on Twitter, and weather the reaction differs based on the location of the tweeter.

In II, we present the findings of some related work in the literature survey. In section III, we explain the twitter data that we obtain from the twitter API along with the infrastructure required to collect and store these tweets. In section IV, we explain the pre-processing steps performed on the obtained tweets in order to convert it to a form that is fit for performing sentiment analysis. We then explain the process of sentiment analysis in section V. In section V1, we present our findings and results. Finally, we present our conclusion and future work in section VII and VIII respectively.

IMCOM '15, January 08 - 10 2015, BALI, Indonesia
Copyright 2015 ACM 978-1-4503-3377-1/15/01…$15.00
http://dx.doi.org/10.1145/2701126.2701129

2. LITERATURE SURVEY

This section describes the sentiment analysis, how it evolves, its challenges in the micro-blogging environment and the techniques that have been used. Then, highlight some of the researches that incorporate sentiment analysis or the location based in their researches.

2.1 Twitter Sentiment Analysis

Some of the early and recent researches on sentiment analysis of Twitter data use distant learning to acquire sentiment data. Pak and Paroubek in [8] use tweets ending in positive emoticons like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative; examples of emoticons are shown in (Table I). They build models using Naive Bayes, Max Entropy and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a unigram, bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. They collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. in [4]. For objective data they crawl twitter accounts of popular newspapers like "New York Times", "Washington Posts" etc. They report that POS and bigrams both help (contrary to results presented in [4]). Both these approaches, however, are primarily based on n-gram models.

Bermingham and Smeaton experiment two classification techniques namely SVM and Multinomial Naïve Bayes (MNB) for both blog and micro blog sentiment analysis. They found that MNB technique outperforms SVM on micro blogs with short text [2].

In terms of political sentiment analysis past researches were post-hoc on static and small samples, as indicated in [6]. They built a real-time system for 2012 US elections to understand political practices at work through the use of Twitter. In our application, we added to the real-time sentiment analysis the location factor to particularly understand the users opinion based on their location.

2.2 Location-based Analysis

With the spread of mobile devices equipped with GPS, location based social networking becomes an attractive area for research. In fact, it opens an opportunity to study different aspects of human behavior and enable a variety of services. Many applications take advantage of such services to establish a relationship between the user's location and the content, which in turn help the people of interest to make decisions.

Hassan et al in [5], have examined millions of web search queries taking into account the location to predict the news intent of the user. However, they don't take into account the time when the query is issued.

In our application, we use Naïve Bayes as our classifier method. In addition, we take the location and time into consideration while collecting and analyzing the data.

2.3 Naïve Bayes classifier

Naïve Bayes Classifier is a simple probabilistic technique based on a naïve independent assumption. It assumes the presence or absence of a certain feature of a class is unrelated to any other feature. In other words, every feature contributes in determining which label should be assigned to a given input vector. To choose a label for an input vector, the Naïve Bayes classifier calculates the prior probability of each label by checking the frequency of each label in the training set. Then, the contribution from each feature is combined with the prior probability, to get a likelihood estimate for each label. The label that has the highest likelihood estimate is then assigned to the input vector. The algorithm uses Bayes' rule:

$$P\ (label \mid features) = \frac{P(label) * P(features \mid label)}{P\ (features)}$$

Label in the equation represents the sentiment class (positive or negative), and the features are the words in the tweet after extraction.

3. DATA DESCRIPTION

A tweet consists of 140 characters. A username starts with the '@' sign while a 'hashtag' starts with '#'. Each 'hashtag' is used to link a tweet to a certain topic. So, a search with '#gmu' will return all the tweets about George Mason University (GMU) published by twitter users. These tweets may also contain acronyms (e.g. 'nvm' for never mind etc.) along with emoticons (such as ':-)',':-D' etc.). Each tweet can also contain embedded links in them as well.

3.1 Data Collection

In order to gather the twitter data (tweets), we first set up 'OAuth'. 'OAuth' is a protocol that allows applications and specifically user profiles, secure authorization in order to access an API. We have to specify our app's consumer_token, consumer_secret, and a callback_url in our code in order to authenticate it to collect twitter data (tweets). Once our app is authenticated, we can begin collecting data from twitter including tweets, followers, re-tweets, favorite tweets etc. For topic based sentiment analysis, we collect tweets on the fly based on the search term in the hash tag. However, for monitoring a certain event or topic, we set up an AWS EC2 instance with Windows OS that runs a python script periodically to collect tweets, clean and store the data in a database. In order to interact with Twitter's API, we use Tweepy which is a python library for interacting with Twitter.

3.2 Tweepy API

In order to use the Tweepy API in our program (in Python), we simply use: 'import tweepy'

A sample code snippet for getting tweets from a timeline by specifying a keyword and a location is shown below:

```
twitterStream = Stream(auth, listener())
        twitterStream.filter(track=["#apple"],
languages=['en'],        locations=[-70.93044414,
41.65526800, -70.9221071,41.66216000])
```

One of the challenges with using the Twitter API is that not all users turn their location as 'on' on twitter, which makes it difficult to extract locations associated with a tweet. To overcome this for our project, we extract the location information from their twitter profiles that the user themselves specify.

For training, we used sentence polarity dataset V.1.0 about movie review collected by Cornell[1]. We used the first 2000 sentences for each class, which means the total training dataset is 4000 labeled instances.

[1] http://www.cs.cornell.edu/ people/pabo/movie-review-data

3.3 Data Storage

We ran our scripts for streaming the tweets on Amazon Web Services (AWS) EC2 instances for a period of 5 days. The tweets were in the CSV format. We loaded these tweets in a MySQL database so that we could perform various queries like obtaining sentiments by location, etc.

4. DATA PRE-PROCESSING

Data obtained from Twitter may contain empty spaces, some special characters that may not be well suited for being processed by data mining algorithms to perform sentiment analysis. Thus, to make this data fit for data mining algorithms, we perform preprocessing on it to make it fit for training and testing our classification algorithms. Pre-processing is also required since Twitter data may contain emoticons and abbreviations.

The major steps performed in the pre-processing step include the removal of whitespaces, converting the '@' character to AT_USER, replacing the keywords preceded by '#' character with only the keywords, replacing all the links present with just the word 'URL'. We also replace all the major emoticons like :), :(etc. and the abbreviations such as LOL, OMG etc. with its corresponding emotion. Tables 1 and 2 provide some examples of the meaning of some emoticons and abbreviation that is used in twitter to help increase the accuracy of the classifier. Nowadays, many users use emoticons and abbreviations in their tweets to put their points across and give it a more personal touch. These emoticons and abbreviations are very useful in data mining in terms of sentimental analysis as they let us know, to a certain extent, the emotional response or sentiments of a person regarding certain issues. Thus, it helps in identifying the positive or negative reactions that people may have with regard to given issues or topics. Slang words also contribute much to the emotion of a tweet. So they can’t be simply removed. Therefore a slang word dictionary is maintained to replace slang words occurring in tweets with their associated meanings.

Table 1. Examples of Emoticons and their corresponding emotion

Emoticons	Meaning/Corresponding Emotion
:) , :D, =), =D,:-)	Happy
:(, :-(, =(, =[,)-:	Sad
:P , =P	Joking/Happy

Table 2. Examples of Slang Acronyms and their corresponding emotion or meaning

Abbreviation/ Slangs	Meaning/ Corresponding Emotion
Lol, Lmao, Rof	Laughing/ Happy
jk	Joking
Wanna	Want to

5. SENTIMENT ANALYSIS PROCESS

In the implementation, we choose Python 2.7 as a programming language for the sentiment analysis. In addition, we use Natural Language Toolkit 2.0 (NLTK) to help in building the classifier. NLTK is an open source platform for building Python programs that works with human language data. It includes a suite for text processing libraries for classification, tokenization, and tagging, parsing, and other functions. In our experiment, we use it mainly for extracting the most informative features and use it to build the classifier model.

The process to assign a sentiment for a tweet is illustrated in Figure 1.

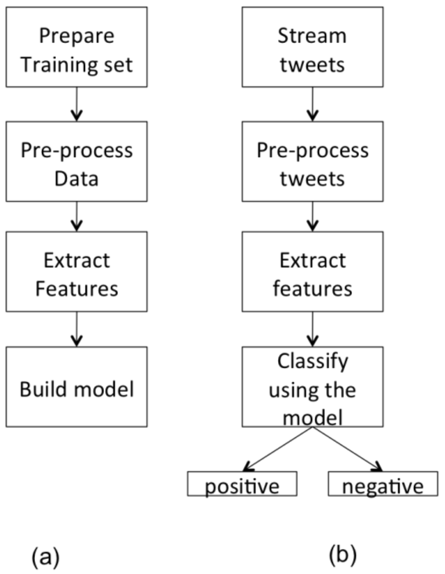

Figure 1. (a) The process of building the classifier model from the training dataset. (b) The process of using the classifier model to classify the tweets

To build the classifier:

1. Collect the training dataset. (Described in the Data Description section).
2. Preprocess each instance in the dataset. (Described in Data Preprocessing section).
3. Extract the important features for each instance and use them to build the classifier.
4. Build the classifier model, and save it into file.

Since the goal is to classify the tweets into two classes “positive”, or “negative”, we used supervised learning technique specifically Naïve Bayes classification, which considers the features (words) to independently contribute to the probability calculation.

To train the classifier, we use a balanced two datasets representing the positive and negative sets as described in the data description previously. Diving into the process of building the classifier, first, preprocess each instance in the training dataset to get features vector. Then, ignore the stop words, links, and mentions (@user) to extract important features. We use a list of stop words (about 600 words) from Google stop-words project[2] to eliminate any words that adds no meaning (i.e. the, it, will, is, with, wants, etc.). Then, we removed duplicate from features list. Table 3 shows an example of a tweet going through some of the process steps. After that, NLTK comes to play. It takes the extracted features to build the classifier based on Naïve Bayes algorithm. We save the resulted classifier model for later use to provide better performance.

Now, the sentiment analysis classifier model is ready for use. We used the classifier to get the sentiment for the collected tweets. The following steps show how to use the classifier is

[2] https://code.google.com/p/stop-words/2014-02-24

shown in figure 1: (i) Stream the tweets. (ii) Preprocess them. (iii) Load the classifier and classify them into “positive”, or “negative” class.

Task	Result
Original tweet	The concept of space is terrifying … I never wanna see #gravity ☹
Preprocess the tweet	The concept of space is terrifying. I never want to see gravity sad
Extracted feature	Concept, space, terrifying, gravity, sad

Table 3. Steps tweets go through during the sentiment analysis

6. RESULT

Using hand labeled real time tweets for evaluation, we got an accuracy of 70%, which means the classified correctly. Other measurements we used for to evaluate our classifier are precision and recall. The recall is 80%, which means that the algorithm returns most of the relevant results, while the precision is 66% means that the algorithm returns more relevant results than irrelevant. This matches the findings of other researches; negative sentiment is harder to get right due to the negation words, and due the sarcastic nature of the negative or political tweets in particular [6].

For our experiment about location-based sentiment analysis of tweets for the Indian Elections, we collected 650,000 tweets over a period of one week. We collected these tweets on a daily basis and performed sentiment analysis on the tweets. We tried to analyze the trends and patterns about people's sentiments towards these political parties. The 2 political parties include the Aam Aadmi Party (AAP) and the Bharatiya Janta Party (BJP).

Figure 2 below shows a comparison of the positive vs. negative sentiments of the twitter users towards the AAP. It can be seen that there is a decrease in negative sentiment towards this party and an increase in positive sentiment towards them.

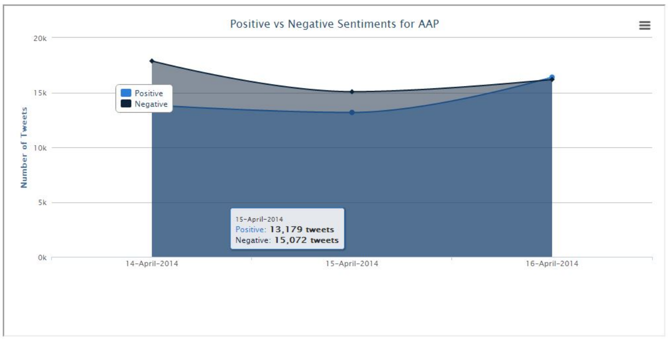

Figure 2. This chart shows a comparison of the positive vs. negative sentiments of the twitter users towards the AAP

Similarly, figure 3 shows a comparison of the positive vs. negative sentiments of the twitter users towards the BJP. It can be noticed that the positive sentiment towards this party is seen decreasing and the negative sentiment towards them is on the rise. We could related this increase and decrease in the sentiments with 'social events' that took place surrounding the political campaigns and elections. For example, a negative statement by a leader of a certain party resulted in an increase in negative tweets and sentiment about them.

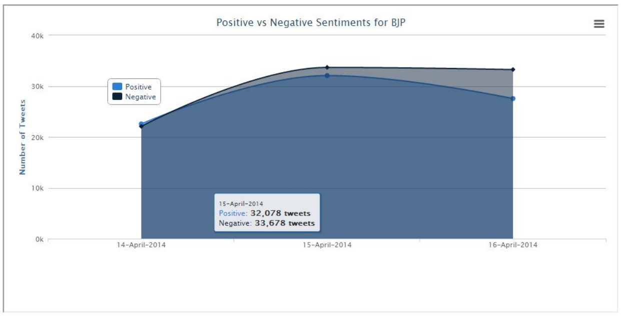

Figure 3. This chart shows a comparison of the positive vs. negative sentiments of the twitter users towards the BJP

Figure 4 below shows positive tweets for the BJP by Indian States. The metropolitan cities such as Mumbai and New Delhi tend to have more tweets towards the BJP. On the other hand, the state of Andhra Pradesh did not seem to have significant tweets about either parties indicating that there are more local parties that get the votes of the people of this state (example Telgu Desam Party etc.).

It can be noticed however, that the BJP is more popular party than the AAP. This is because the number of positive tweets (and total tweets) for the BJP far exceeds those of the AAP. This means that among twitter users who tweet political commentary pertaining to the Indian Elections 2014, there are more mentions (both total and positive) of the BJP. This could mean that based on the tweets of the users about these political parties, the BJP could have a comfortable victory over the AAP.

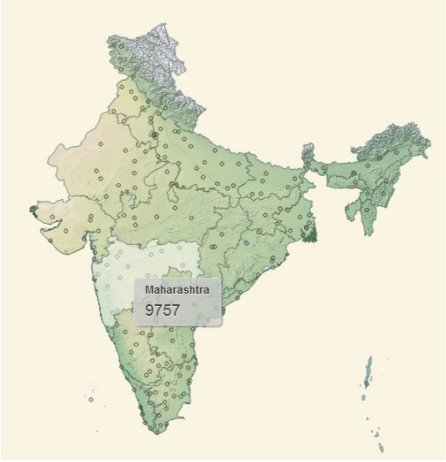

Figure 4. Shows positive tweets for the BJP by Indian States

7. CONCLUSION

The large amount of information contained in microblogging websites makes them an attractive source of data for opinion mining and sentiment analysis.

We performed location based sentiment analysis on 650,000 tweets in order to understand trends and patterns regarding the Indian elections. We saw how sentiments, both positive and negative, change from one location to the other. We also studied how certain 'social events' can trigger a sharp rise in both negative and positive sentiments regarding a political party. Based on the large amount of tweets (both with positive sentiments as well as total in number), we predict that the BJP will emerge as the more successful party based on the sentiment analysis of the tweets.

In our research, we have presented a method for an automatic collection of a corpus that can be used to test a sentiment classifier. Our classifier is able to determine positive, negative sentiments of tweets collected based on location.

8. FUTURE WORK

In the future, we would like to analyze tweets coming from political parties, perform sentiment analysis on them in order to understand their policies better. We would like to focus on major twitter accounts of a certain political party and mine their network in order to understand if there is any existence of 'cliques' among these account users. We would also like to include support for multilingual tweets. Our system would then be able to provide location based sentiment analysis from tweets in other languages such as Arabic, Hindi, etc. to provide better local results. We would also like to take this further, and identify 'influencers' in the twitter network (e.g. a person whose tweets are the most re-tweeted) and reach out to them. The future work can also include neutral sentiment.

9. REFERENCES

[1] Abrol, S., & Khan, L. (2010, February). Twinner: understanding news queries with geo-content using twitter. In *Proceedings of the 6th Workshop on Geographic Information Retrieval* (p. 10). ACM.

[2] Bermingham, A., & Smeaton, A. F. (2010, October). Classifying sentiment in microblogs: is brevity an advantage?. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1833-1836). ACM.

[3] Dijck, J. V. (2011). Tracing Twitter: The rise of a microblogging platform. *International Journal of Media & Cultural Politics*, *7*(3), 333-348.

[4] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1-12.

[5] Hassan, A., Jones, R., & Diaz, F. (2009, November). A case study of using geographic cues to predict query news intent. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 33-41). ACM.

[6] Wang, H., Can, D., Kazemzadeh, A., Bar, F., & Narayanan, S. (2012, July). A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations* (pp. 115-120). Association for Computational Linguistics.

[7] Liu, K. L., Li, W. J., & Guo, M. (2012, July). Emoticon Smoothed Language Models for Twitter Sentiment Analysis. In *AAAI*.

[8] Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC*.