Sentimental Analysis on Twitter Database

Revanth Segu
Vikram Suriyanarayanan
Shruthi Deshpande
Sonal Hundekari

Abstract

With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others. Twitter is becoming one of the major online platforms for expressing opinions and thoughts. Sentiment analysis seeks to identify the view-point(s) underlying a text span; an example application is classifying a movie review as thumbs up or thumbs down. To determine this sentiment polarity, we propose a novel machine-learning method that applies text-categorization techniques to just the subjective portions of the document. This paper combines map reduce techniques and machine learning algorithms into a new combined method. This method is tested on US presidential elections in 2012.

Keywords*: Sentiment Analysis, Opinion Mining, Map Reduce, Machine learning.*

INTRODUCTION

Twitter, with nearly 600 million users and over 250 million messages per day, has quickly become a gold mine for organizations to monitor their reputation and brands by extracting and analyzing the sentiment of the Tweets posted by the public about them, their markets, and competitors. Sentiment analysis provides companies with a means to estimate the extent of product acceptance and to determine strategies to improve product quality. It also facilitates policy makers or politicians to analyze public sentiments with respect to policies, public services or political issues.

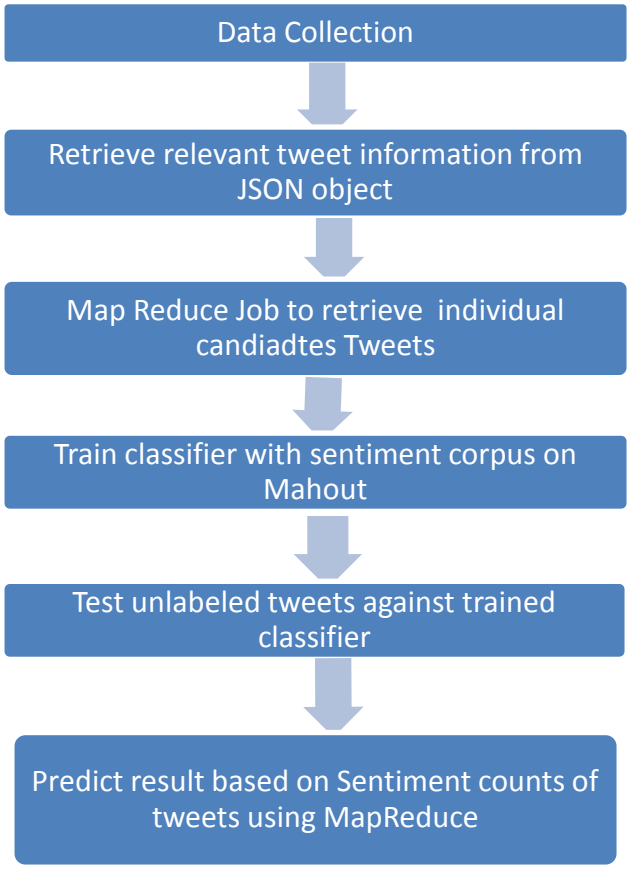
The sentiment found within comments, feedback or critiques provide useful indicators for many purposes. These sentiments can be categorized either into two categories: positive and negative; or into an n-point scale, e.g., very good, good, satisfactory, bad, very bad. In this respect, a sentiment analysis task can be interpreted as a classification task where each category represents a sentiment.

RELATED WORK

Sentiment analysis of tweets data is considered as a much harder problem than that of conventional text such as review documents. This is partly due to the short length of tweets, the frequent use of informal and irregular words,

and the rapid evolution of language in Twitter. A large amount of work has been conducted in Twitter sentiment analysis following the feature-based approaches. Given the character limitations on tweets, classifying the sentiment of Twitter messages is most similar to sentence-level sentiment analysis however, the informal and specialized language used in tweets, as well as the very nature of the micro blogging domain make Twitter sentiment analysis a very different task. It's an open question how well the features and techniques used on more well-formed data will transfer to the micro blogging domain. Just in the past year there have been a number of papers looking at Twitter sentiment and buzz. Other researchers have begun to explore the use of part-of-speech features but results remain mixed. Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data exploit existing Twitter sentiment sites for collecting training data.

This paper attempts to presents a method to extract sentiment from the database in the following steps.



DATA COLLECTION

The database for this project has been collected from twitter API. Around 1 Tera Byte of tweets were collected for 2012 elections.

In order to check sentiments in the tweets, a separate dictionary has been prepared. Below URL has been used a reference for this, as it has large set of sentiment words already collected.

www.sentiment140.com

There are many other ways to retrieve twitter data such as using Twitter streaming/search API. One can write a script to connect to the API and download tweets by providing query. Also a small database is available at Sentiment140.cm for academic purpose.

PREPROCESSING

The extracted database contained tweets as JSON objects. From these objects, text message and id of the tweet were extracted.

Since the database size is too large and we need the system to be scalable, a MapReduce job is used to extract these fields. JSON jar is used in Mapper class to extract "text" and "id" fields of the tweets. The output of this MapReduce job is is then converted to a .tsv format.

The same job can be carried out using JSONSerde package. Once this is installed we can use Hive to convert JSON to txt file.

Sample input JSON file format is shown below:

{"retweet_count":0,"in_reply_to_screen_name":null,"text":"Bijna op stage enzo","in_reply_to_status_id_str":null,"geo":null,"retweeted":false,"in_reply_to_user_id_str":null,"id_str":"154094643720630272","source":"\u0.....","id":154094643720630272,"truncated":false}

Extracted .tsv file format:

153748106129833984 Never new mcdonalds be open this late

153748110324142081 You'll give it up

The dictionary downloaded for sentiments has the format as shown below. In the below format, 4 represents positive sentiment, 2 represents neutral sentiment and 0 represents negative sentiment.

0 1684810338 Sat May 02 22:07:02 PDT 2009 NO_QUERY bobster16 Has broken pixels on his iTouch maybe they'll replace it with a new generation!

4 1960185647 Fri May 29 07:33:39 PDT 2009 NO_QUERY Nicoledyan Good morning people of twitter.

MAP REDUCE TASK

After reading the JSON files and extracting the messages from tweets out next jobs was to check the tweets containing words containing "Obama" or "Romney". To check related words, we have used other keywords such as "Obama", "romney", "democrat", "republican", "mitt", "barrack", "michelle", "#Obama", "#RomneyRyan", "Mitt".

All these keywords are checked against the messages present in tsv file and two new files are generated containing keywords for Obama and Romney.

CLASSIFICATION

The next job is to classify the tweets above for positive, negative and neutral sentiments. We have used Naive Bayes classifier for classification task.

Naive Bayesian Classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. Given a class variable y and a dependent feature vector $x_1$through x_n, Bayes' theorem states the following relationship:

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

Using the naive independence assumption that

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

for all i, this relationship is simplified to

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set.

We have carried out this task in following steps. All the tweets have been used for classification.

1. Convert tsv file into a sequential file

The file extracted from preprocessing is in tsv file format . It is a tab separated file format. This format should be converted into sequential file format, so that Mahout can read it for classification purpose.

A separate Java program is written on Hadoop which takes input as this tsv, reads each tabs and writes the output to a sequential file.

Sentiment corpus which was downloaded for checking emotions in the tweets is also converted from tab to sequential format.

2. Training the classifier with sentiment corpus on Mahout

After generating sequential files, Naive Bayes classifier has been trained in Mahout. Using sequential file for sentiment corpus, a model is

generated in Mahout which will generate weighted vectors for the corpus for 3 different classes i.e positive, negative and neutral.

3. Testing tweets against the trained classifier

The sequential files generated for Obama and Romney tweets were tested against the classifier generated in above step. The classifier uses weighted vectors generated in training phase for classification task.

The output of out classifier will be three classes, positive, negative and neutral, classifying all the tweets for Obama and Romney in different classes.

The output of this class is redirected again to a text file.

4. Checking counts of sentiments using MapReduce

The output files generated from the above classifier are given as input to a Map Reduce Job. The Map reduce job checks the counts of zeros, twos and fours from the output file and displays it for all Obama and Romney.

Based on this output, we can predict the output for election or we will get to know the public opinion for election.

TEST RESULTS

We have used 80 percent of the data for training and 20 percent for the training. The test results found were as follows:

Training Accuracy:

```
---------------------------------------
---------------
Correctly Classified Instances
:         381        96.2121%
Incorrectly Classified Instances
:          15         3.7879%
Total Classified Instances
:         396

=======================================
===============
Confusion Matrix
---------------------------------------
---------------
a       b       c       <--Classified
as
132     4       2        |  138
a     = 0
2       113     3        |  118
b     = 2
1       3       136      |  140
c     = 4
```

Testing Accuracy:

```
---------------------------------------
---------------
Correctly Classified Instances
:          68        66.6667%
Incorrectly Classified Instances
:          34        33.3333%
Total Classified Instances
:         102

=======================================
===============
Confusion Matrix
---------------------------------------
---------------
a       b       c       <--Classified
as
26      7       6        |  39
a     = 0
3       12      6        |  21
b     = 2
5       7       30       |  42
c     = 4
```

Counts of the positive, negative and neutral classes for Obama and Romney are shown below:

Obama Count:

NEGATIVE 0 5085K

NEUTRAL 2 5117K

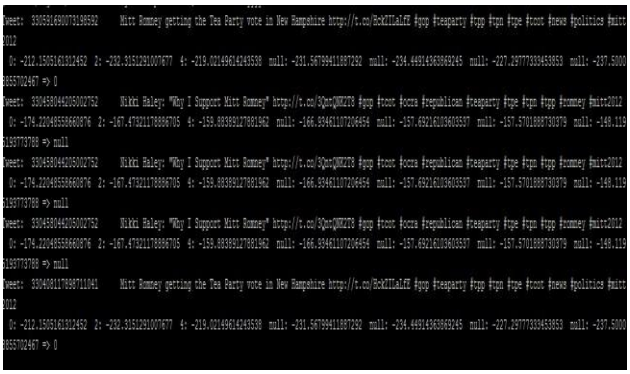
POSITIVE 4 3631K

Romney Count:

NEGATIVE 0 1300K

NEUTRAL 2 800K

POSITIVE 4 585K



FUTURE WORK

The current work for classifying sentiments is carried out only by extracting sentiment words from Tweets. This can be extended to extract other feature such as Emotion Icons, Exclamation Marks, Negation words, Intensity words. In addition to this, extracting features can be extended to extract bigrams, trigrams and POS taggers from tweets to get more accurate results.

The same project can be extended used to detect reviews for a product/movie or for predicting elections in independent states.

PROBLEMS FACED

While converting tsv file to sequential file, we were getting exception such as file not found. We needed to set classpath for Java on Windows Linux for that. Classpath in Java is path to directory or list of directory which is used by ClassLoaders to find and load class in Java program.

CONCLUSION

A new scalable approach of extracting sentimental features from large database of authors writing style has been proposed. The developed architecture using Naive bayes is robust for extracting the sentiments.

REFERENCES

[1] Feature Selection andWeighting Methods in Sentiment Analysis by Tim O'Keefe, Irena Koprinska *In Proceedings of 14th Australasian Document Computing Symposium (December 2009) Key: citeulike:7915897*

[2] Predicting The US Presidential Election using Twitter data by *Swathi Chandrasekar, Emmanuel Charon, Alexandre Ginet*

[3] Opinion mining and sentiment analysis Bo Pang1 and Lillian Lee2 *Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135 c 2008 Bo Pang and Lillian Lee.*

[4] Sentiment Analysis and Opinion Mining *Bing Liu AAAI-2011 Tutorial Sentiment Analysis*

and Opinion Mining Morgan & Claypool Publishers, May 2012.
[5] Twitter dataset from http://www.twitter.com/
[6] Big Data lecture notes from http://www.utdallas.edu/~lkhan/Spring2013/
[7] Hadoop tutorial from http://developer.yahoo.com/hadoop/tutorial/
[8] MapReduce tutorial from http://hadoop.apache.org/docs/r1.0.4/mapred_tutorial.html
[9] Mahout tutorials from https://cwiki.apache.org/confluence/display/MAHOUT/Quickstart
[10] Harry Zhang "The Optimality of Naive Bayes". FLAIRS2004 conference
[11] Caruana, R. and Niculescu-Mizil, A.: "An empirical comparison of supervised learning algorithms". *Proceedings of the 23rd international conference on Machine learning, 2006.*