Sentiment Analysis of Twitter data using multiple data dictionaries

Rashmi Narvekar[*], **Prasad Marla**[*], **Mayur Ghogale**[*]

[*] Department of Computer Science, Stony Brook University, 11790, New York

Abstract- Increase in social media reach has increased the number of people expressing their ideas on the various social media platforms. However, most of the sentiment analysis focuses on using a single dictionary that is built from a certain domain of data (like facebook posts, tweets, online reviews) to classify the input stream into positive and negative sentiments. We propose an approach of using multiple dictionaries that are built from different domains of data to perform sentiment analysis of twitter data.

I. Introduction

The approach of using data dictionaries for sentiment analysis is mostly done by creating a dictionary in a domain and training the data over the dictionary in the same domain where the initial dictionary was built. Since the dictionary is built in a domain, there are possibilities that the results are biased. It can also be possible that a dictionary built in one domain when used in other domain for sentiment analysis may produce very different results. We approach this problem by using multiple dictionaries built across multiple domains to predict an averaged-out results of sentiment analysis over twitter data collected in different context. By this approach, we assume to better predict the results over the data gathered in any domain.

II. Background

Sentiment analysis is done to gather knowledge about the people conveying the emotions, like age, gender, ethnicity, or for the categorization of the emotion into fear, anger, anticipation, sadness, joy, surprise, anticipation, disgust, trust, etc. However, these findings have always been done by first constructing the data dictionary in the domain under consideration and then the data of the same domain is further used to train the dictionaries.

We see the following 4 dictionaries constructed in various domains and use it later for our combined dictionary approach for twitter data analysis.

1) NRC Word-Emotion Association Lexicon (also called EmoLex)[1]
   This lexicon is created using Amazon’s Mechanical Turk service that provides a platform to obtain hand curated large-scale emotion annotations. These emotions are then weighted according to their average ratings from all the online annotators.

2) NRC Hashtag Emotion Lexicon[2]
   This lexicon is created for twitter data using hashtags to gather data. This lexicon classifies the words in the tweets depending on how close it is related to the emotion of the hashtag of the tweet.

3) Sentiment Composition Lexicon of Opposing Polarity Phrases (SCL-OPP)[3]
   The lexicon is created for predicting the sentiments of phrases made of words with opposite sentiments. The lexicon is built on unigrams, bigrams and trigrams.

4) Affect and Intensity Lexicon[4]
   This lexicon is created by manually curating a large set of facebook data by physiologists. These annotations were then run through linear regression to predict the final score of the words. The words have score in the range of -1(most negative) to 1(most positive).

III. DATA

The data used for the analysis are twitter streams gathered using Twitter4j API. The data input for the analysis is a continuous stream of tweets gathered based on a hashtag.

A. Number of features

We compute the score of every individual word in the tweet. The overall score for the tweet is the combined score of all the words in the tweet. Hence, every word in the tweet is a feature. Thus, there is no fixed or predefined features. This be seen as taking all the words in English dictionary to be the features.

B. Size of data

Since the data is continuously being streamed, there is no limit on the size of data for analysis. The features are the words in each tweet as they would each contribute to the overall sentiment of tweet. The data is pulled at an interval of every 10 seconds using Apache spark.

IV. Methods

The combined results of all the above mentioned four dictionaries are used to predict the category of the tweet. We use positive, negative, and neutral as the three categories for analyzing the data. The main components of the analysis are listed as below:

A. Streaming data

For data streaming we use Apache spark to gather data from twitter using the twitter4j API. To perform the analysis on streaming data, we are using Apache Spark Streaming. We first registered with twitter developers api that allows us to fetch the twitter data using search queries. We first set up a apache-spark-streaming with twitter stream. Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches. Spark Streaming provides a high-level abstraction called discretized stream or DStream, which represents a continuous stream of data. DStreams can be created either from input data streams from sources such as Kafka, Flume, and Kinesis, or by applying high-level operations on other DStreams. Internally, a DStream is represented as a sequence of resilient distributed datasets (RDDs) and all the transformations that are applied are applied on each RDDs.

B. Data Pre- processing

The raw tweet text cannot be directly used for analysis. We filtered out retweets as they appear more than once in the stream and could skew the results. We are filtering by English language. Another set of filtered out items are hyperlinks, tweet mentions. After that we break the tweet into token. We filter out some common stop words that don't add any value for sentiment of the tweet. After cleaning up the tweet of unwanted data we calculate score.

C. Data Analysis

We categorize the tweet into positive, negative, and neutral categories. Emotions corresponding to anger, disgust, fear, sadness sentiments are put into the negative sentiment category. The votes for every token in the tweet is found using the four dictionaries and then counting the maximum votes for any category (positive, negative, or neutral), the overall categorization of the tweet is done. The output of the analysis is continuously appended to a JSON file.

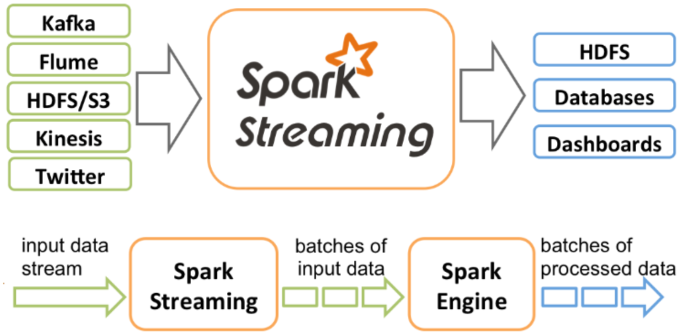

Fig: 1 Shows the Apache Spark pipeline for data streaming. Image source: http://spark.apache.org/docs/latest/streaming-programming-guide.html#initializing-streamingcontext

D. Data Visualization

The results from the analysis are shown with pie charts and bar graphs to better understand the analysis. We use highcharts library to plot these charts. The charts pick data from continuous data analysis sent by the spark server. The data in charts is real-time. The implementation of reading the data from JSON file and plotting to the chart is done using Javascript.

V. Results

To verify the results of this approach we have plotted graphs for various use cases to provide comparison of output for all the dictionaries used and the combined result of all the 4 dictionaries put together.

A. Results of all Dictionaries

The plots in figure 2, 3 and 4 show the combined results of all the four dictionaries put together. In figure 2, the query given to twitter API to filter data is 'google'. The percentage of positive, negative, and neutral tweets are plotted. We observed that there are more positive tweets than the negative tweets.

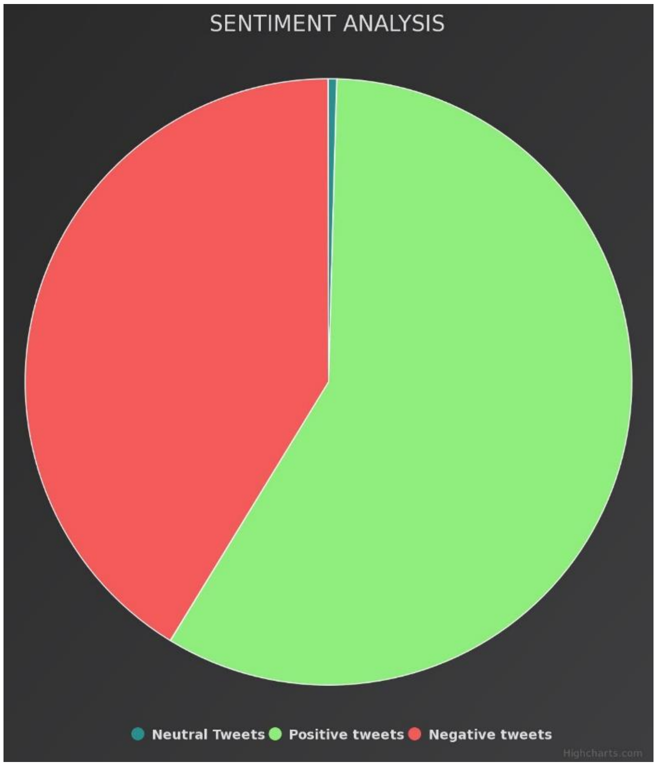

Fig: 2 Shows the percentage of tweets for hashtag ‘Google’.

In figure 3, the tweets were fetched using filter queries ‘#happy’. We observe that percentage of tweets falling into positive tweets are very large. This should be intuitive as generally tweets tagged with #happy are representative of positive sentiment. Thus, the validity of the multiple dictionary approach can be confirmed.

As another test case, we have pulled data for hashtag ‘trump’. The percentage of results can be seen in figure 4.

B. Comparison of Dictionaries

Next test case is the comparison of individual dictionaries and their predictions for the same set of data. In figure 5 the comparison is between the dictionaries ‘NRC Hashtag Emotion Lexicon’, ‘SCL-OPP’ and ‘Affect and Intensity Lexicon’ for data gathered for hashtag ‘Trump’. These results were obtained for 350 tweets.

The second set of comparison was done on two dictionaries ‘NRC Hashtag Emotion Lexicon’ and ‘NRC Word-Emotion Association Lexicon’. The analysis was done for 12,000 tweets with hashtags ‘Obama’. We observed that these two dictionaries gave almost similar results.

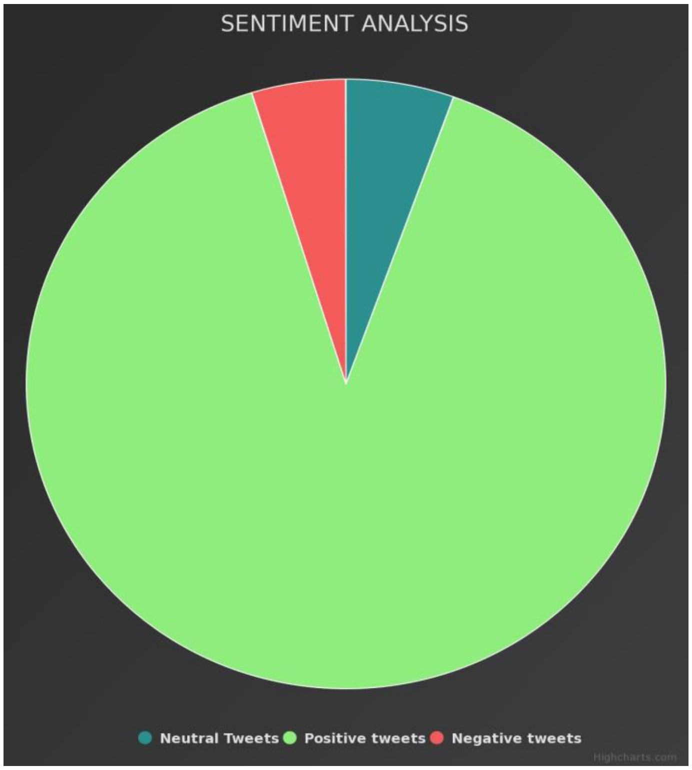

Fig: 3 Shows the percentage of tweets for hashtag ‘happy’.

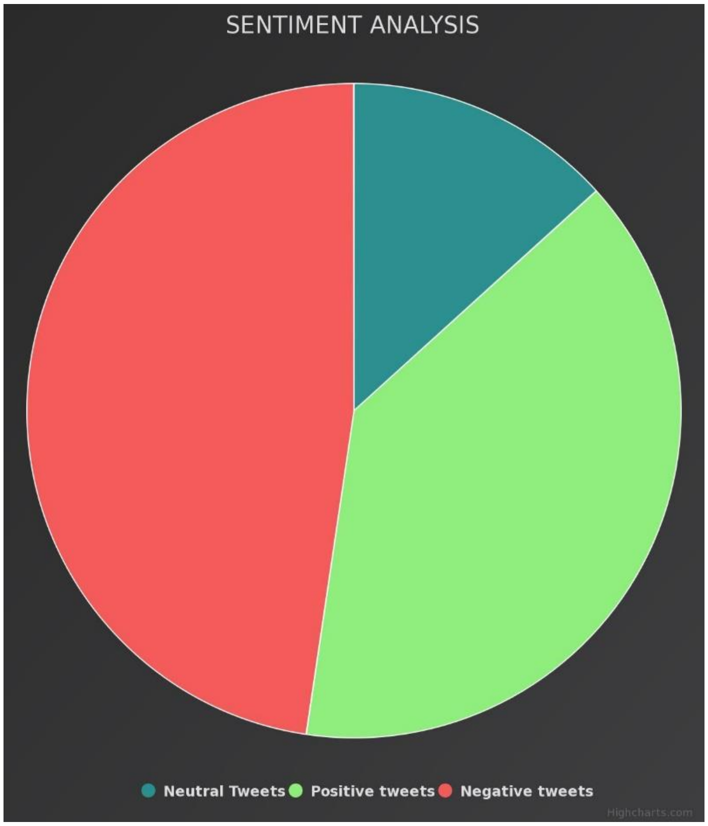

Fig: 4 Shows the percentage of tweets for hashtag ‘Trump’.

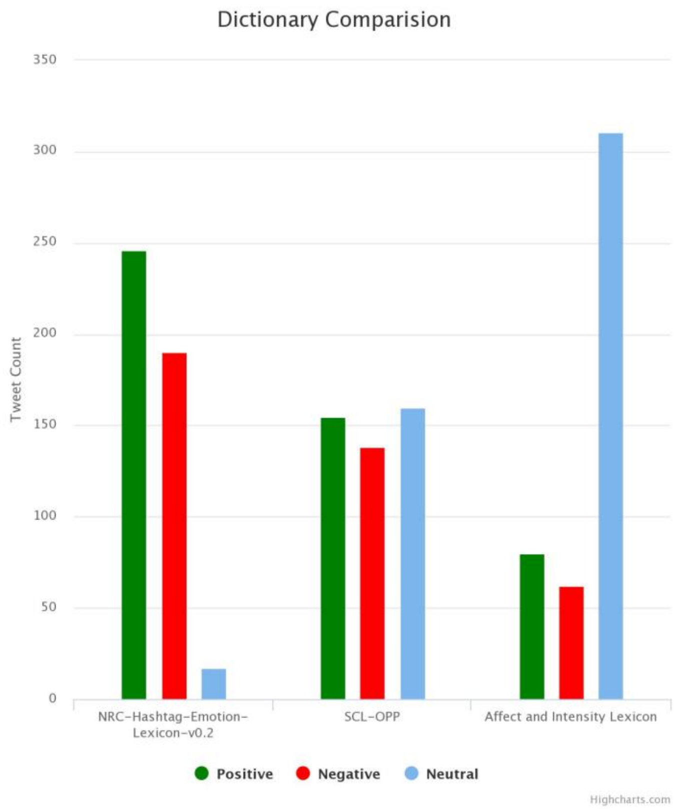

Fig: 5 Shows the comparative results for 'NRC Hashtag Emotion Lexicon', 'SCL-OPP' and 'Affect and Intensity Lexicon' for hashtag 'Trump.'

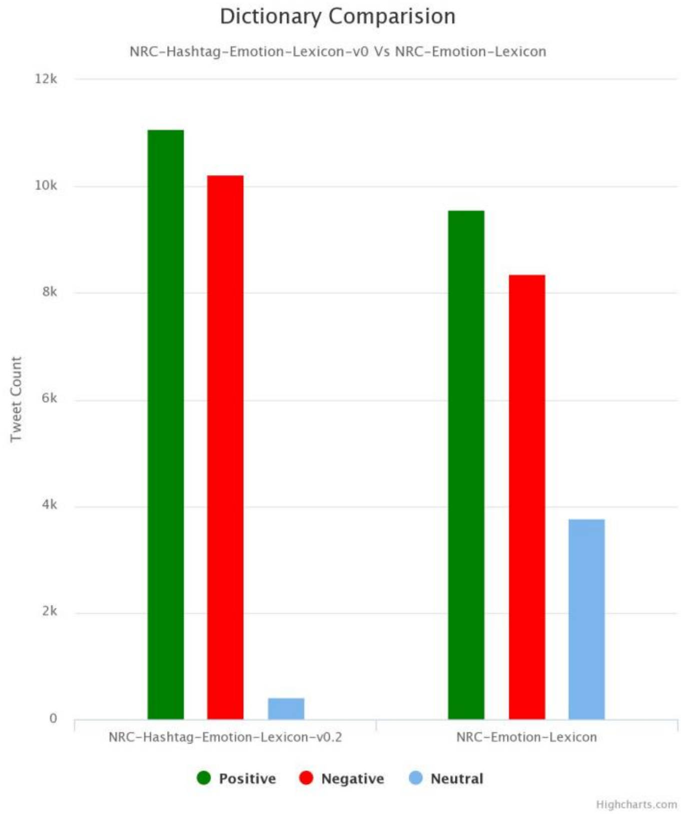

Fig: 6. Shows the comparative results for 'NRC Hashtag Emotion Lexicon' and 'NRC Word-Emotion Association Lexicon' for hashtag 'Obama'.

C. Real-time trend analysis

To demonstrate the online sentiment analysis, we tracked the sentiments of tweets during the EPL match Crystal Palace vs Chelsea on 17th December 2016.

The search query for the input stream was 'crystal palace, #crsytalplacelive'. We tried to track the sentiments of the crystal palace followers by using these search query terms. From the results, we get we can see that at the start of the match general sentiment of the followers was positive towards their team.

We can clearly observe a sudden spike in the negative tweets in the middle. We can attribute this to the fact that Crystal Palace conceded a goal just at the end of first half. So, it is understandable that general sentiment of the tweets would be negative but as the match went on, the sentiment was again swinging towards positive.

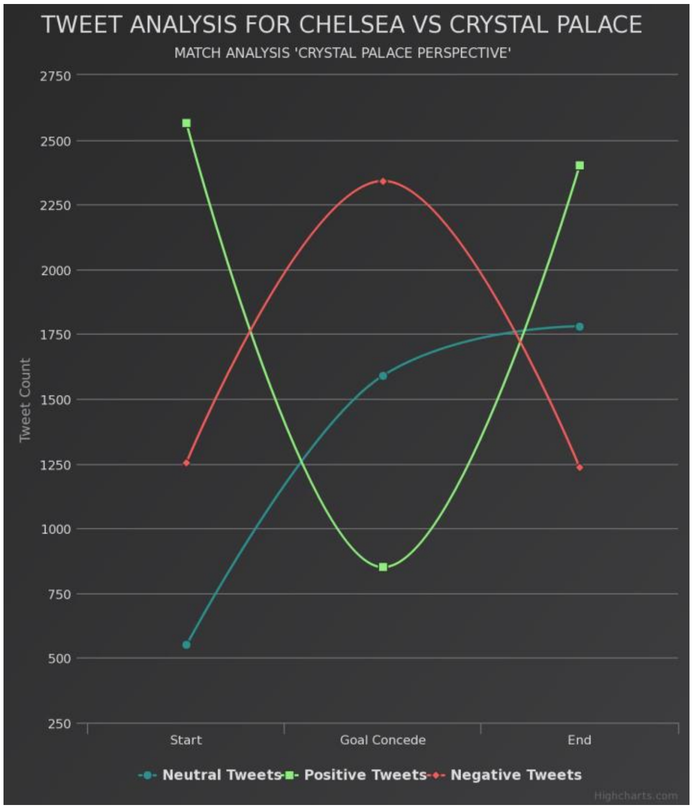

Fig: 7. Shows the Sentiment analysis of tweets gathered during a Football match

VI. Discussion

Our analysis is based on unigrams and the NRC Hashtag Emotion Lexicon provides the maximum accurate results for various test cases of the data as it is built on twitter domain.

Affect dictionary probably would perform better on Facebook posts and bigrams. Hashtags alone could give the sentiment of the tweet. Multiple dictionaries used together provide results with higher accuracy.

VII. Conclusion

We compared the results of the individual dictionaries against stream of tweets. We found that dictionaries like ' NRC Hashtag Emotion Lexicon' and ' NRC Word-Emotion Association Lexicon' perform well and gave fairly accurate results. As these dictionaries were generated from the data of the same domain i.e. twitter, they seem to give good results.

We gave various emotion hashtags as filter to twitter search API, like #sad, #angry. We used 'NRC Emotion Hashtag' dictionary to get the results. We could get almost 80-85% accurate results. The reason being that this dictionary is derived from the hashtags and it can identify sentiments more accurately.

Also, combining multiple dictionaries for calculating sentiments has a benefit that there is no need for complex machine learning algorithms to compute the sentiments and we can get results with good accuracy.

Acknowledgment

The authors would like to thank Andrew Schwartz, professor in the Computer Science department at Stony Brook University for useful guidance regarding various aspects of this work.

References

[1] Saif Mohammad and Peter Turney, Crowdsourcing a Word-Emotion Association Lexicon, Computational Intelligence, 29 (3), 436-465, 2013.

[2] Saif M. Mohammad, Svetlana Kiritchenko, Using Hashtags to Capture Fine Emotion Categories from Tweets, Computational Intelligence, in press.

[3] Svetlana Kiritchenko and Saif M. Mohammad , Sentiment Composition of Words with Opposing Polarities. In Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. June 2016. San Diego, CA.

[4] H. Andrew Schwartz, Daniel Preot¸iuc-Pietro, Gregory Park, Johannes C. Eichstaedt, Margaret Kern, Lyle Ungar, Elizabeth P. Shulman, Modelling Valence and Arousal in Facebook posts.

Authors

First Author – Rashmi Narvekar, Stony Brook University, New York. Email: rashmi.narvekar@stonybrook.edu.

Second Author – Prasad Marla, Stony Brook University, New York. Email: prasad.marla@stonybrook.edu.

Third Author – Mayur Ghogale, Stony Brook University, New York. Email: mayur.ghogale@stonybrook.edu.