TweetBuzz

(http://tweet-buzz.appspot.com)

By
Anshul Mehra (0507597)
Srujan Saggam (0508749)

Abstract

TweetBuzz is a data analytics web application that focuses on a new web trend called microblogging. TweetBuzz attempts to generate a relevant summary of microblogs related to the topic phrase provided by the user and analyses sentiments of the microblogs and plots them on map based on their geolocation. The plotted pins are color-coded based on their emotional quotient. The summary generated and graph based statistics are shown. This is done by fetching the latest tweets from one such microblogging website called Twitter

1 Introduction

1.1 Data Analytics

Data analytics has been the buzzword for a while in this decade.

Data analytics (DA) is the science of examining raw data with the purpose of drawing conclusions about that information. Data analytics is used in many industries to allow companies and organization to make better business decisions and in the sciences to verify or disprove existing models or theories. Data analytics is distinguished from data mining by the scope,
purpose and focus of the analysis. Data miners sort through huge data sets using sophisticated software to identify undiscovered patterns and establish hidden relationships. Data analytics focuses on inference, the process of deriving a conclusion based solely on what is already known by the researcher.

This science is generally divided into exploratory data analysis (EDA), where new features in the data are discovered, and confirmatory data analysis (CDA), where existing hypotheses are proven true or false. Qualitative data analysis (QDA) is used in the social sciences to draw conclusions from non-numerical data like words, photographs or video.

TweetBuzz is based on the Qualitative Data Analysis model which is used to draw conclusions about the twitter user's sentiment related to a search topic provided .to the application.

Twitter Data is analyzed for this project because of its colossal user base across the world.

1.2 Sentiment Analysis

One of the major features of this project is that it categorizes tweets into 7 different categories based on the emotional quotient of the tweet. The categories include: Anger, Happiness, Sadness, Disgust, Surprise, Fear and Neutral. Such a categorization helps to understand how the twitter population feels about the search topic.

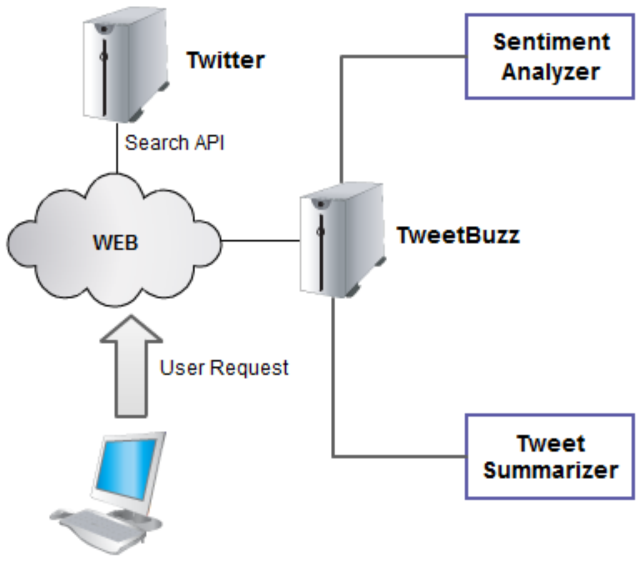
1.3 Tweet Summarization

When a user searches for a topic on twitter the results are returned according to the time that they were tweeted and not by relevancy. This sort of behavior forces the user to go through all the tweets in order to get the essence of the topic. TweetBuzz attempts to automate this process using one of the microblog summarization algorithms.

1.3.1 Related Work

Previous works on tweet summarization focused on summarizing results of database queries for presentation during natural language interactions. Summaries are usually generated for the purposes of providing a “gist” of a document or a set of documents to human readers (e.g., Luhn, 1958; Brandow et al., 1995). Summaries are sometimes also used as inputs to machine learning approaches, say for categorization. Kolcz et al. (2001) summarize textual documents in order to classify them, using the summaries as a feature to be input to a classifier. Most early studies used a news corpus like the Reuters dataset. As the Web started growing in size, the focus moved to Web pages.

For example, Mahesh (1997) examines the



effectiveness of Web document summarization by sentence extraction. Recently, there has been work on summarizing blogs (e.g. Zhou and Hovy, 2006; Hu et al., 2007). Most techniques focus on extraction: the selecting of salient pieces of documents in order to generate a summary. Applying extraction on microblogs at first appears irrelevant since a microblog post is already shorter than most sum- maries. However, extraction is possible when one considers extracting from multiple microblogs posts that are all related to a central theme.[1]

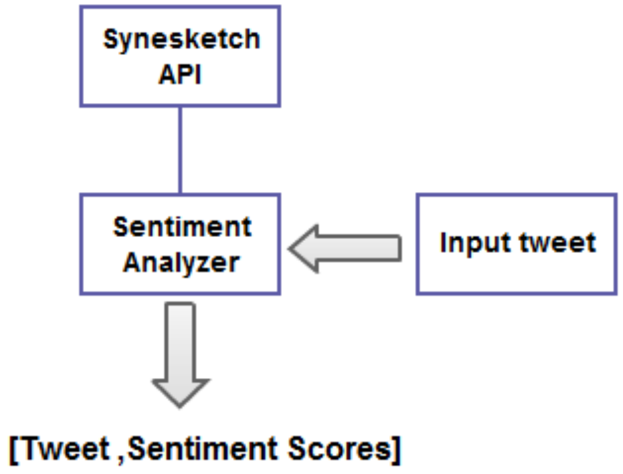
2 Architecture

TweetBuzz has two main components:
1) Sentiment Analyzer
2) Tweet Summarizer

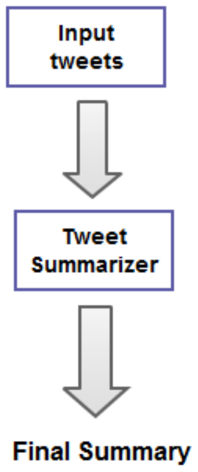
2.1 Sentiment Analyzer

To perform the sentiment analysis of the tweets, TweetBuzz uses an open-source third-party API called Synesktech. Synesketch analyses the emotional content of text sentences in terms of emotional types (happiness, sadness, anger, fear, disgust, and surprise), weights (how intense the emotion is), and a valence (is it positive or negative). The recognition technique is grounded on a refined keyword spotting method which employs a set of heuristic rules, a WordNet-based word lexicon, and a lexicon of emoticons and common abbreviations.[2]

2.2 Tweet Summarizer



Tweet Summarizer processes the list of tweets returned by the Twitter Search API and applies a variation Phase Reinforcement algorithm from the Summarizing Microblogs Automatically paper by Sharifi et. al.[3] and generates a one line summary of the topic.



Before passing the input tweet to the Tweet Summarizer the tweet should be normalized by removing emoticons, hashtags, user mentions , urls etc. This is done using an open source library "twokenize" [3]

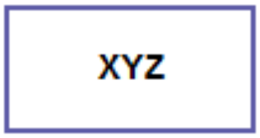
2.3 Phrase Reinforcement Algorithm

The Phase Reinforcement(PR) algorithm is based on the assumption that users tend to use similar words while describing a specific topic and that there is an overlapping of words adjacent to the root word or the topic phrase in either directions. The PR algorithm builds a graph that contains the overlapping sequences and then it extracts the most relevant overlapping sequence of words and projects it as the summary of the topic.

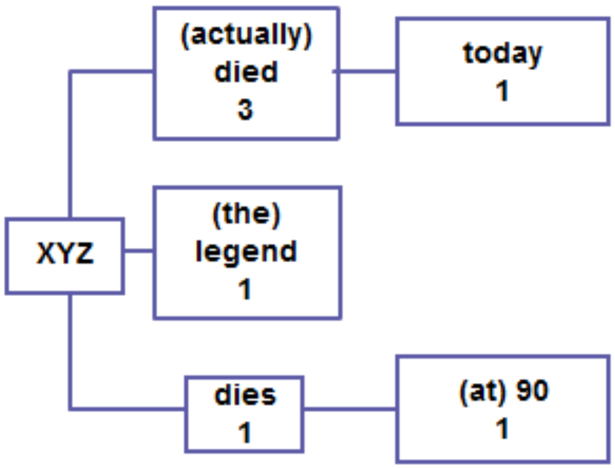
We have used a variation of PR algorithm which uses a greedy approach to find the overlapping sequence of words.

Consider the following sentences with the topic phrase as XYZ:
1) The legend XYZ died today.
2) I can't believe XYZ actually died today.
3) The legend XYZ died today at the age of 90.
4) RIP XYZ the legend
5) XYZ dies at 90 .

The modified PR algorithm works as follows:
Step 1:
Build a tree with root node as the topic phrase i.e "XYZ".



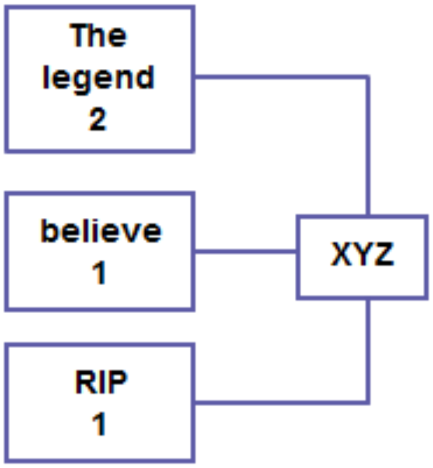
Step 2:
For each tweet find the root phrase and build a tree by inserting each the words that occur after the root phrase "XYZ" in the respective order. If the node is already present in the respective path at the desired position just increase the count variable of the node. Call this as the right summary tree.



Note: If the word encountered is a stop word then merge it with the next word in the sentence which is not a stop word.

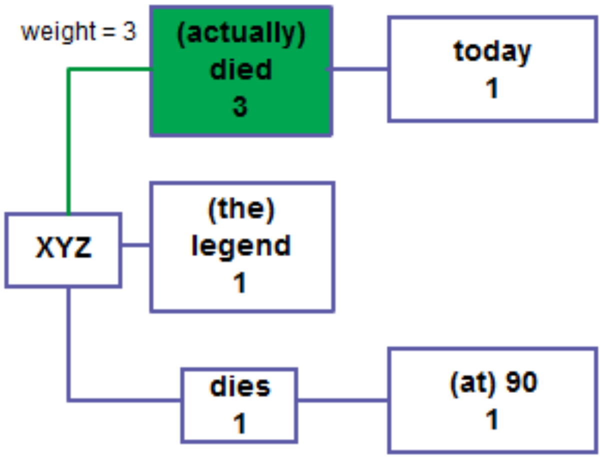
Example: XYZ had died becomes a tree with nodes "XYZ" and "had died". had is not treated as a separate node. If the node already has a stop word then the new stop word is ignored.

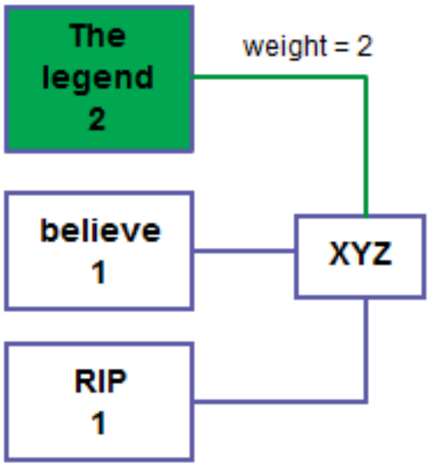
Step 3:
Similarly build the left summary tree.
For each tweet find the root phrase and build a tree by inserting each the words that occur before the root phrase "XYZ" in the reverse order i.e the word adjacent to the root comes first. If the node is already present in the respective path at the desired position just increase the count variable of the node. Call this as the left summary tree.



Step 4:
Now find the highest weighted path by making a greedy choice of selecting the neighbor with highest count (>1) at each node in the left summary tree and right summary tree separately. Call this as Lw and Rw respectively. Ignore the nodes with weight < 2.

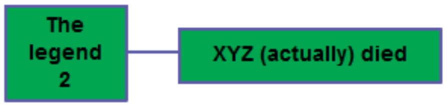
If Lw > Rw:
Merge highest weight sequence of Left Summary Tree with the root and generate the Right Summary Tree again this time by considering only the tweets that formed the highest weight sequence in the Left Summary Tree.



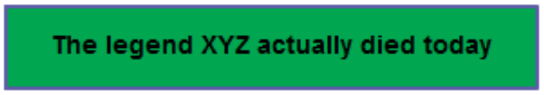


I f
Lw <= Rw:
Merge highest weight sequence on the Right Summary Tree with the root and generate the Left Summary Tree again this time by considering only the tweets that formed the highest weight sequence in the Right Summary Tree.
In our case Rw > Lw so we consider only such sentences which has the highest weighted path Rw i.e

1) The legend XYZ died today.
2) I can't believe XYZ actually died today.
3) The legend XYZ died today at the age of 90.



Step 5:
Print the final summary by extracting the overlapping sequence with the highest weight and merging it with the root.



2.3.1 Observations

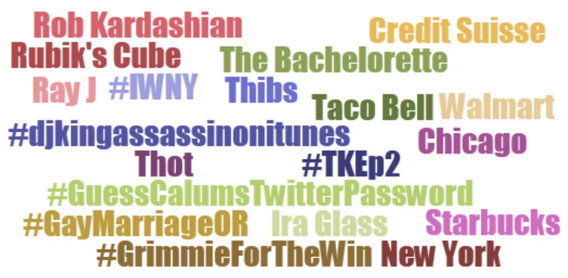
1) If the input set contains lot of retweets the generated summary will be the tweet with most number of retweets. We can modify this to perform summary only on the original tweets by removing the duplicate tweets.

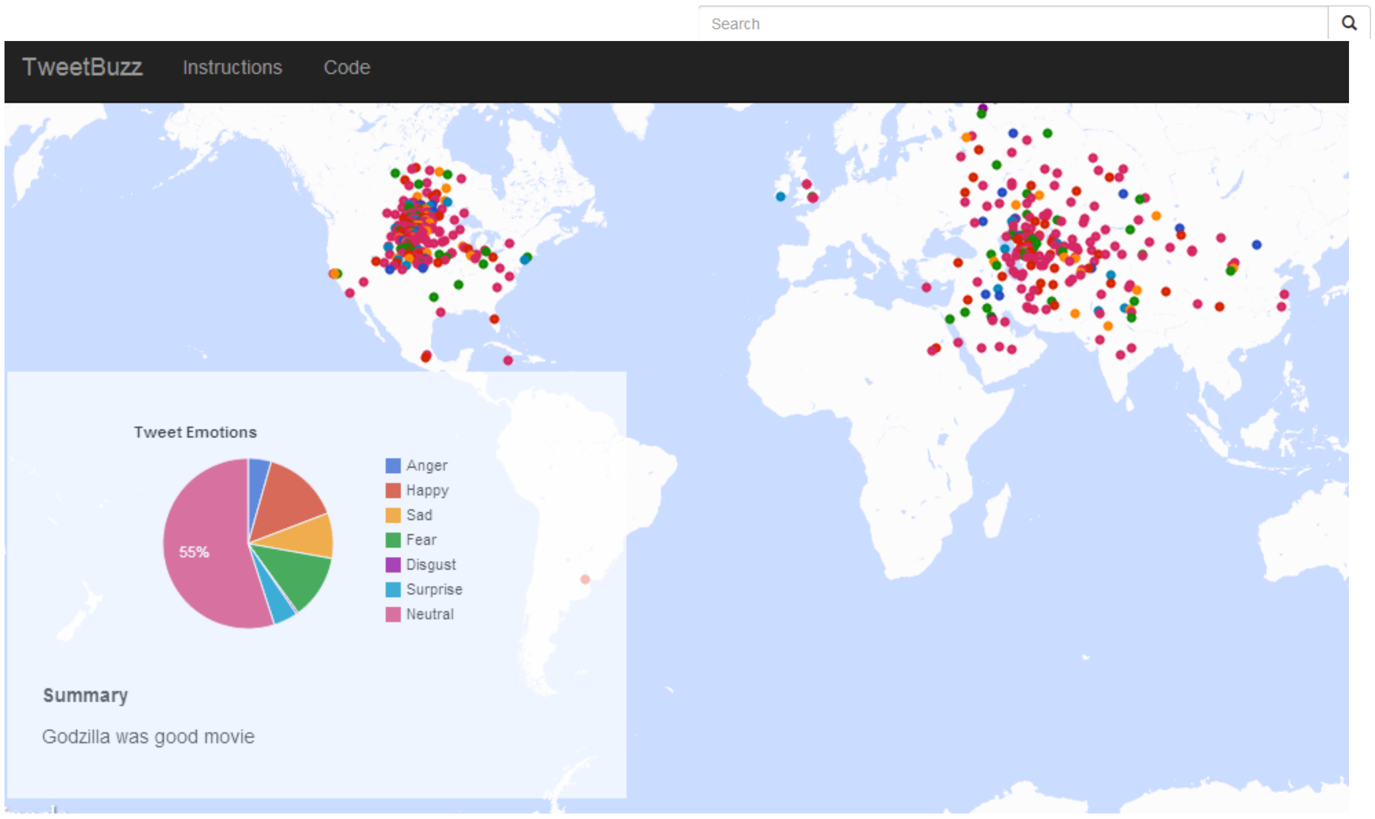
2) There may exist tweets that doesn't contain the topic phrase . This algorithm doesn't work on such tweets.

3) While constructing the tree the tweets are normalized to remove punctuations, commas, emoticons etc. This results in the generated summary to lose its punctuation, commas etc.

3 User Interface

To demonstrate the use of this application a graphical user interface has been created where the user is presented with a page where he will be able to see the current





Summary

Godzilla was good movie

trending topics on twitter and a search box to enter a custom search topic. User can select either of these options to perform a search .

The search results are visualized on a world map based on the geolocation of the tweets and the markers are color coded based on the emotional quotient of the tweets.

There is a panel which displays the statistics of the tweets based on their emotional quotient and a auto generated summary for the search topic .The figure shows the result for the search topic "Godzilla".

This interface gives an overview of the twitter users sentiment analysis across the world based on the search topic.

4 How it works?

The application consists of html web front-end and java servlets to process the requests from the front-end.

Web Interface

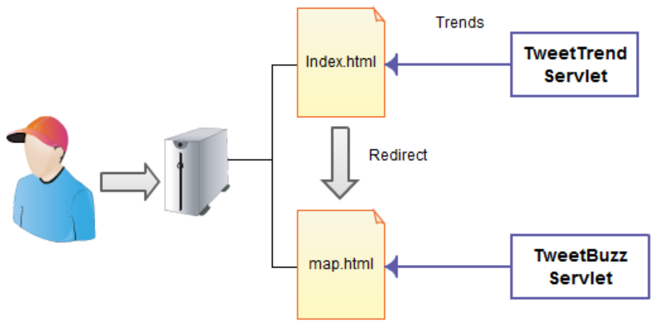
The front-end has two pages:
Index.html : This is the landing page to the application. Users are greeted with a selection of currently trending topics and hashtags and a search bar. They have the option of choosing a trending topic or searching for a topic of their choice.

Map.html : When they select or search a topic from the index page they are redirected to map.html. This page handles the display of map, graph and summary in context of the query.

Servlets

TweetTrendsServlet.java : This servlet queries the Twitter server and fetches the top trending keywords and hashtags. The hashtags are fetched for several locations to give a wide range than just top-10 topics. However, duplicates are removed so that only unique keywords are displayed in the web-GUI.

TweetBuzzServlet.java : When a topic is selected or searched on the landing page, this servlet queries the twitter server and fetches 1500 latest tweets about that topic. The tweets are then passed to 1) Sentiment analyzer which returns wapper tweet objects containing the sentiment of the tweets, 2) Tweet summarizer that summarize the tweets to one or two sentences. The results are then returned to map.html and visualized



in a nice way.

TweetWrapper.java : TweetWrapper is responsible for using the Synesketch API and generate a wrapper tweet containing the text of the tweet and its emotion.

Summarizer.java : Summarizer is the implementation of the phrase reinforcement algorithm and is responsible for sumarizing tweets. StopWordRecognizer.java, KeyTree.java and Utilities.java are helper classes for Summarizer.

5 Features

1) Top trending suggestions on home page.
2) Plotting to tweets to map to identify which regions are talking about a given topic.
3) Sentiment based marker color on map to see what sentiment is prevalent in which region. For eg. After a game of soccer final

winning country might be tweeting happy thoughts and loosing country might be tweeting angry rants.
4) Graph analytics to give a precise estimate of the emotion classifications.
5) A Summary is generated for each topic to give an idea as to what people are saying about that topic.

6 Limitations

Geo-location

A large chunk of user tweets which are returned by Twitter Search API are non-geo tagged which means they cannot be mapped on a map. Due to the restriction of number of tweets from Twitter Search API maximum up to 1500. It is observed that less than 10% of the tweets contain geo-locations. This limitation makes us to difficult to draw conclusion based on the geography.
For demonstration purpose we have randomly assigned geolocations for non-geotagged tweets .

Retweets

If the input set to the Tweet Summarizer contains a large number of retweets. Then the output is biased towards the most retweeted tweet in the set.

Tweet Summarization

Due to nature of the Phrase Reinforcement algorithm where it needs to find the overlapping words around the root phrase, the algorithm may not work for Hashtag as a search topic since hashtags are generally found at the end of the tweet and does not guarantee the condition required by the PR algorithm.

7 Improvements

Sentiment Analyzer

A better machine learned approach can improve the sentiment classification.

Tweet Summarizer

Though the PR algorithm generates good summary it still needs some improvement as manual summaries have a better structure than the auto generated summaries.

Effective spam filtering can reduce the error rate in generating auto summary.

8 Conclusion

Given an massive realtime dataset such as Twitter an application like TweetBuzz can be very useful to analyze the sentiments of population about a topic, product or person. This can be useful for plenty of applications like business intelligence (for studying consumer response) or psychology (for studying human behavior after an important event).

It is interesting to note that most of the tweets are not geotagged. This could be because they do not feel safe sharing their location.

9 References

[1] http://cs.uccs.edu/~jkalita/work/StudentResearch/SharifiMSThesis2010.pdf
[2] http://synesketch.krcadinac.com/blog/
[3] http://www.ark.cs.cmu.edu/TweetNLP
[4] http://www.cs.utexas.edu/~dinouye/papers/inouye2011-comparing-twitter-summarization-algorithms-socialcom2011.pdf
[5] http://www.cs.uccs.edu/~jkalita/papers/2013/SharifiBeauxComputerJournal2013.pdf