CSE 535: Information Retrieval

Project 04:

Complete Search & Analytics based on dissecting twitter data

Anish Gadekar (50291289)
Parth Shah (50291125)
Saiyam Shah (50291714)
Shruti Bendale (50289048)
Utkarsh Bansal (50290395)

Introduction:

We crawl tweets from multiple cities, for multiple topics and languages using the Twitter API. The collected data is cleaned, preprocessed and then indexed using Solr. A BM25 information retrieval model is initialized and optimized to improve recall. We developed a website that passes the search query to this model in Solr and retrieves the search results. The search results display the user name, tweet text, topic, city and the sentiment of the tweet. A sentiment analyzer identifies the overall sentiment of each tweet and classifies the tweet into happy, sad or neutral. We also generate word clouds for all topics. The website also displays a tweet map that integrates location wise analysis of data.

Implementation:

1. **Data crawling and preprocessing:**

   We collected about 320,000 tweets distributed over 9 cities – NYC, Mexico City, Delhi, Bangkok, Paris, Dublin, California, Sydney and Buenos Aires over the period of 3 months using a script for crawling the data using Twitter REST API.

   A preprocessing script was created to add the location, language and the coordinates of the location and to separate the hashtags, emojis, kaomoji's, mentions, text, URLs from each tweet. This cleaned data is useful for the data analysis part of this project. The text_en, text_es, text_hi, text_th fields were also added using the preprocessing script.

2. **Configuration and optimization of Solr**

   We experimented with the various models and discovered that the BM25 model gives the best recall rate.

   So, BM25 is used as the default similarity for luceneMatchVersion 6.0 and above. We add the following snippet into the schema.xml file to further tweak the values of the model parameters b and k1:

```
<similarity class="solr.BM25SimilarityFactory">
   <float name="b">0.8</float>
   <float name="k1">0.4</float>
 </similarity>
```

   BM25 has two parameters, k1 and b, that can be tuned to improve the performance of the information retrieval system. After a lot of trial and error, k1=0.4 and b=0.8 were found to be the best values for the parameters.

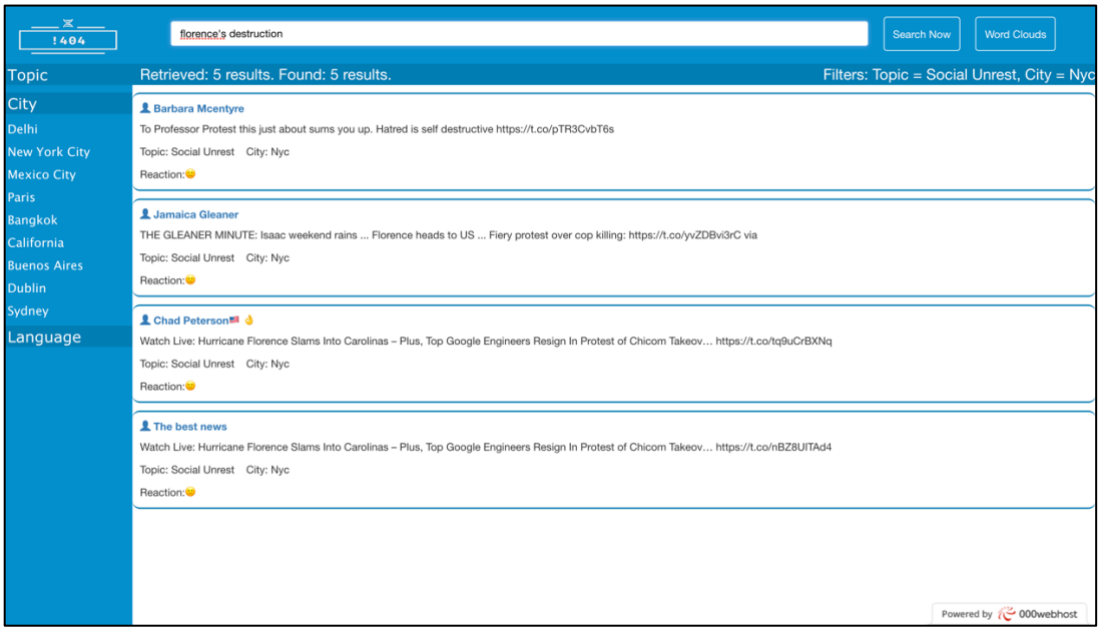
   Using the dismax query with different weightage to different text fields, the best set of weights that were found are:

```
weight_en=1.4
weight_es=1.6
weight_hi=1.3
weight_th=1.3
```

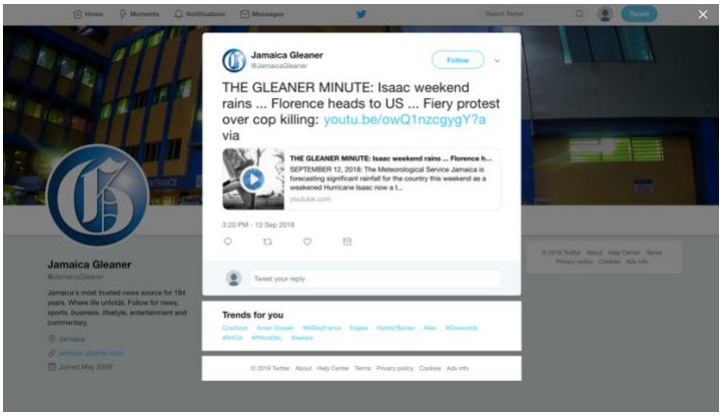
3. UI Creation

For the User Interface, we have used HTML, CSS, Bootstrap, and JavaScript. The homepage of the website has a search option and a button for viewing the data analysis. A screenshot of the homepage is as follows:

The search query is passed on to solr and the username, tweet text, topic, city and the reaction of the retrieved results are displayed. We use “Florence’s destruction” as a sample query. We can filter out the results based on topic, city and as well as language. For our current query, we filter the results based on city New York City, and based on topics Social Unrest. The 5 results retrieved after filtering are as follows:



The username of each result is a clickable link that redirects us to the actual tweet on twitter.

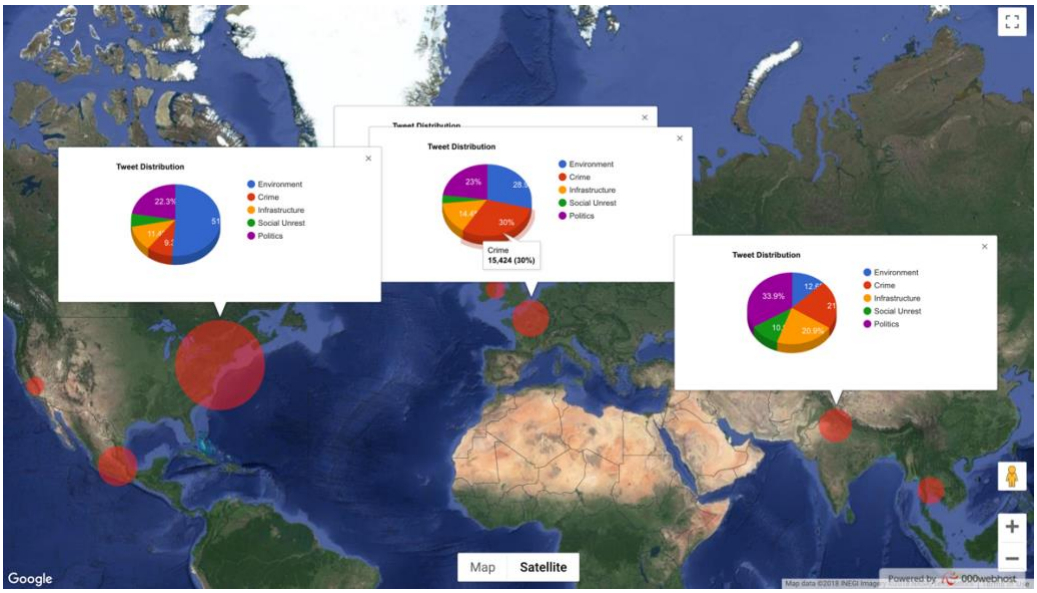


4. Data Analysis

The analytics are important to get a broader view of the search results and further information not easily perceived from plaintext query results. We implemented the following methodologies to analyze the collected data and display it in the form of a tweet map, pie charts and word clouds. We also performed sentiment analysis on the tweets to generate the overall sentiment of each tweet. The data analysis methods we used are as follows:

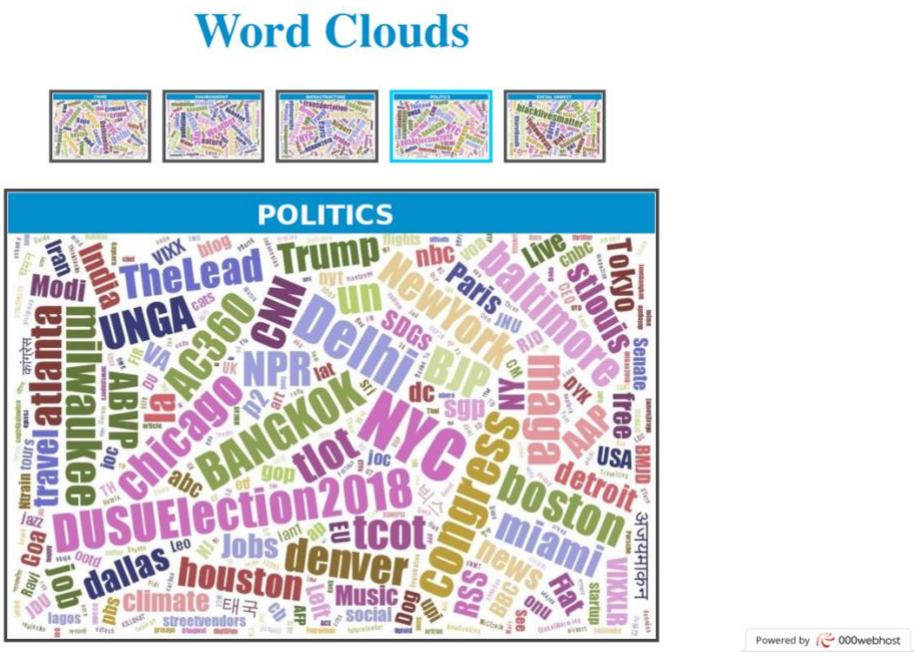
i. Tweet Map and location-wise analysis

Along with the search functionality, we also have a button for displaying the data analytics of the corpus in the form of a map distribution. The google maps provides an API to work with maps and customize the code according to the project requirements. Here we see the cities marked with markers which have sizes proportional to the number of tweets collected from the respective cities. Markers are interactive. Clicking on the markers displays the distribution of tweets according to topics as a 3D pie-chart. Each slice of the chart represents the percentage of tweets of that particular topic. Hovering over any slice displays the actual tweet count. The map is scalable.



ii. Word cloud

Word Clouds are graphical representations of Word frequency that give greater prominence to Words that appear more frequently in a Source Text. The larger the Word in the visual the more common the Word was in the document. We ran all the retrieved tweets through a script that generated word clouds according to their topics.



iii. Sentiment Analysis

For sentiment analysis, we created a happy.txt with all the words that can be classified as happy and a sad.txt file with all the words that can be classified as sad. After running through the entire tweet, the overall sentiment of each tweet was decided by comparing each word in the tweet to these text files. The sentiment of maximum number of words was decided to be the final sentiment of the tweet. We added a '`tweet_polarity`' tag to the json files of each tweet. If `tweet_polarity=-1`, the sentiment of that tweet is sad and if `tweet_polarity=1`, the sentiment of that tweet is happy. The sentiment of majority of the tweets was observed to be sad because of the topics selected. The sentiment for each tweet was represented as an emoticon in the reaction field.

WGRZ

VIDEO: Devastating destruction from Florence in Wilmington https://t.co/D8UeZSar2i

Topic: Environment City: Nyc

Reaction:

Video demonstration

We've demonstrated the functionality of our Information Retrieval system in the video below:

https://youtu.be/u1SpF9V1lBY

Team contributions:

Member	Contribution
Anish Gadekar	UI Design & overall website functionality
Parth Shah	Data Analytics and word cloud
Saiyam Shah	Data Analytics and geocoding
Shruti Bendale	Crawling tweets, and preprocessing of tweets
Utkarsh Bansal	Sentiment analysis and preprocessing

References:

1. https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/
2. https://developers.google.com/chart/interactive/docs/gallery
3. https://www.census.gov/dataviz/
4. https://www.jasondavies.com/wordcloud/