Complete Search and Analytics solution based on dissecting Twitter data

Team Chwilio

Project Report

https://github.com/batistado/ChwilioWeb

https://github.com/batistado/ChwilioSearch

CSE535: Information Retrieval

Fall 2018

University at Buffalo

Highlights

A multilingual search

Can handle queries in 5 different languages

Based on twitter data corpus

With data of around 300,000+ tweets

Spanning across five cities

Consists of data for 5+ different topics

Sentiment Analysis

Trend analysis

1. Introduction

The goal of the project was to gain insight on various societal issues spanning 5 different cities from 3 different continents. It also dealt with creating an end-to-end search engine with capabilities such as multilingual index and queries, faceting, lemmatization, stemming, query boosting etc.

2. Data Collection

Data ingestion was performed with the help of the twitter API. The collection script made use of both the methods provided for fetching tweets i.e., collecting tweets using the Twitter REST API and also Streaming.

We have collected over 384,000 tweets here, over a period of four-six weeks in different languages, catering to topics like politics, social unrest, environment, crime, etc, spanning geographically over five major cities. The raw tweets were stored in order to help the search engine perform better and also for query refinement.

3. Steps to the solution

The following approach describes the flow of project:

1. The very first step was the collection of data. This step made use of a python script with the required constraints in date, location, language and topic.
2. After collecting more than 100K tweets, we indexed the data and optimized the results using Solr. The indexed data can now be queried from Solr, and it gave us the desired results for every query we ran.
3. After that, machine learning was performed on the fetched data using SKLearn for sentiment analysis to determining the sentiments of the people in any given city.
4. When the search was performed on the tweets, it was noticed that the results consisted of the tweets specific to only the language of the query, Hence, to deal with this, language translation was performed on the query using Microsoft Translator API and this resulted in tweets from all the languages being fetched in the result
5. Next, a user feedback system was implemented in the form of filters. The user could filter the search by omitting certain languages, cities and could also get the results from only the specified number of days in the past.
6. Further, to give users more ability an optional language translation feature was provided using which the user could translate the tweet into any language of their choice.
7. To display the trending hashtags based on the filters provided and to provide analysis of data, faceted search was performed.
8. After the back-end of the project was established, the front end was designed using VueJS 2 which would fetch the data from the back-end and place everything accordingly to display to the user.

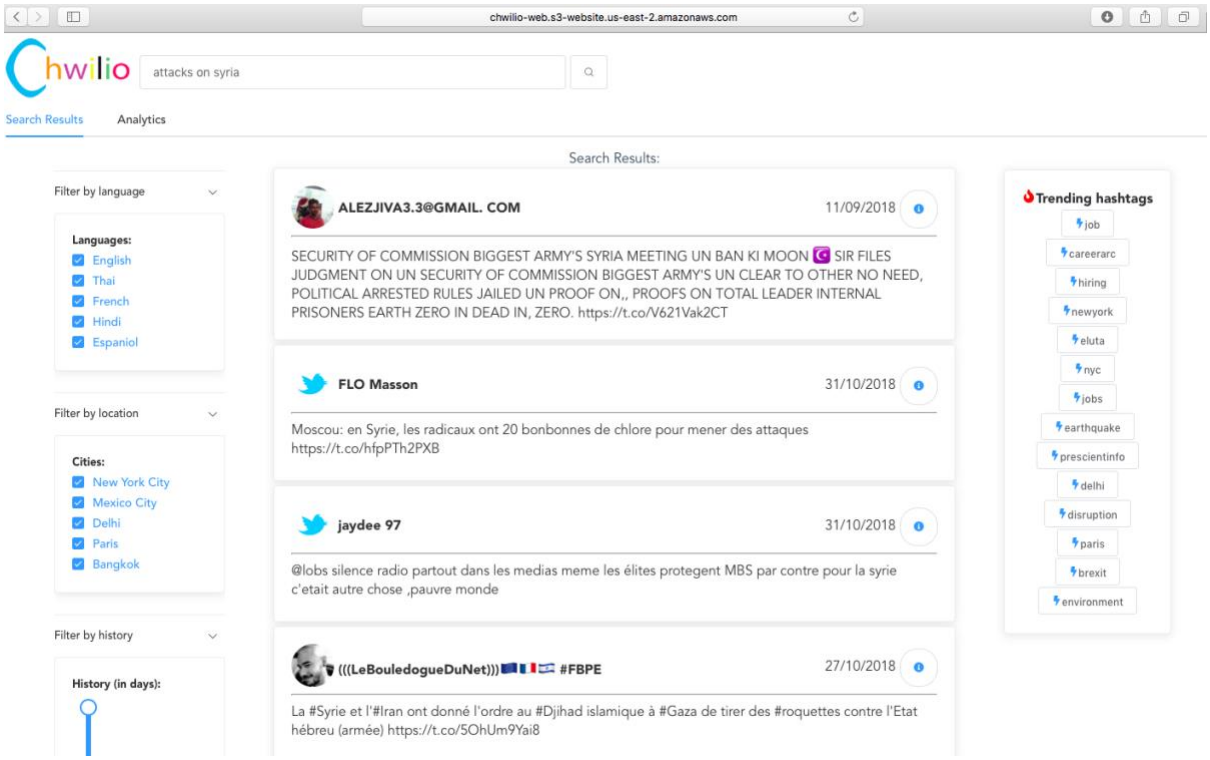
4. Technology Stack:

- **Front end –** VueJS 2, Element UI, Webpack
- **Back end –** Spring Boot, REST, Maven, SolrJ, Microsoft Translator API
- **Analytics –** Microsoft Azure Translator, NLTK, SKLearn

Below are few screenshots from the search engine to describe the results :

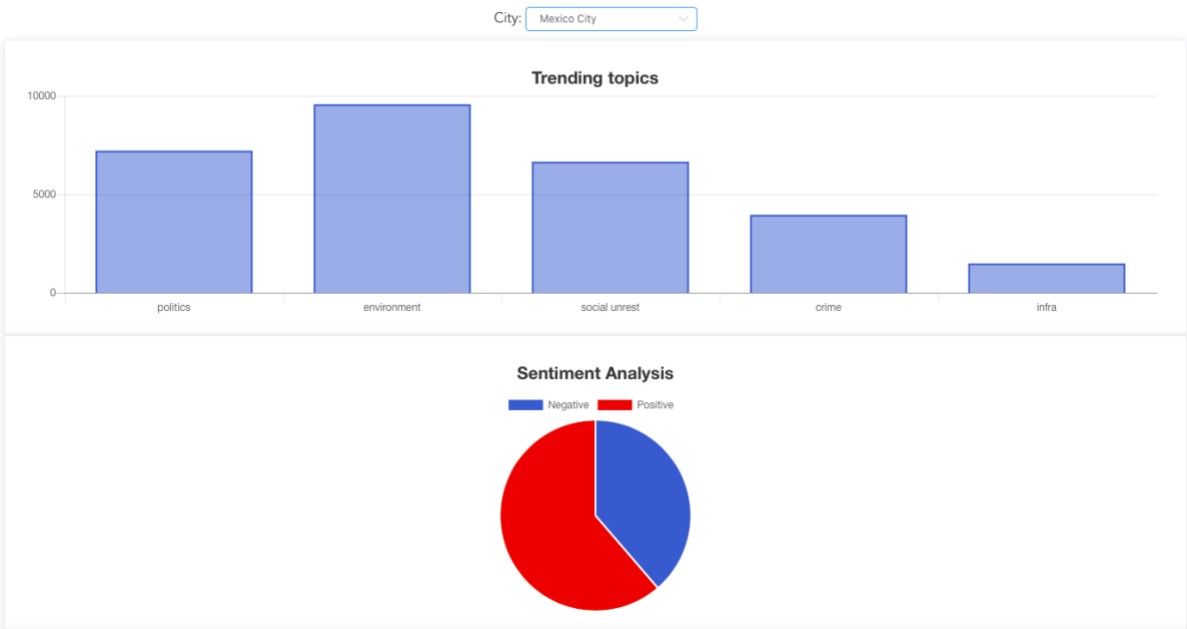
- This highlighted area shows the URL which is hosting our search engine on amazon AWS.

- The below snapshot shows the user feedback system in the form of filters.
- On the right, trending hashtags are displayed and clicking on anyone of these would return results specific to that hashtag.



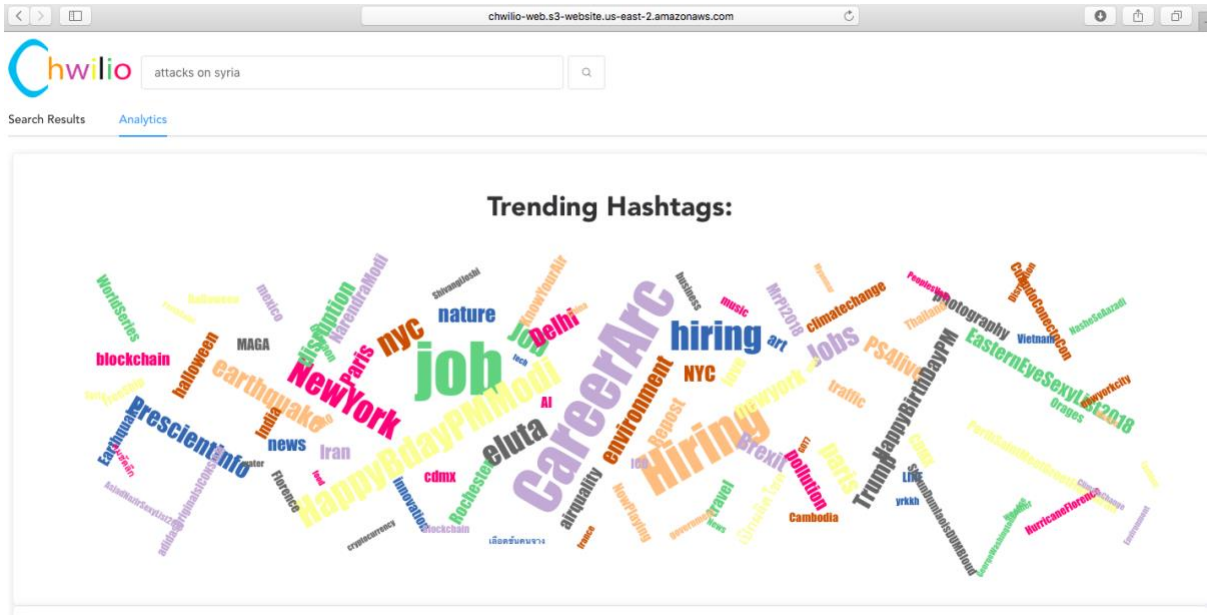
- ## Analysis

  This screenshot shows, that we can analyse the trending topics for a particular city, in bar graphs. Also, the sentiment analysis for a given city is shown in pie chart.



The below screenshot shows the timeseries i.e., based on a given city which topic was discussed the most on a particular date.

- The below snapshot shows the trending hashtags, spanning all the 5 cities in the form of a word cloud.



5. Working URLs

- AWS URL for search engine-

http://chwilio-web.s3-website.us-east-2.amazonaws.com

- Youtube URL with demo video-

https://www.youtube.com/watch?v=CG-ZiN-B9g0&feature=youtu.be

6. References

- https://scikit-learn.org/stable/
- https://lucene.apache.org/solr/6_6_0//solr-solrj/