University at Buffalo
The State University of New York

INTRODUCTION TO INFORMATION RETRIEVAL
CSE 535

FALL-2016

PROJECT-4 REPORT
CROSS LINGUAL IR - TWOOGLE

SUBMITTED BY:
MANISHKUMARREDDY JARUGU (50206843)
HANEESH REDDY PODDUTURI(50208280)
HEMA MADHAV JAKKAMPUDI(50206563)
ARUN CHANDRA PENDYALA(50207136)

CROSS LINGUAL INFORMATION RETRIEVAL SYSTEM

OVERVIEW:

The data was crawled from Twitter for the given topic – Trump using the twitter search API. The crawled data was then processed using python script to extract the required information such as text, hashtags, user names , language etc. The geographical information such as locality, state and country was also added to the processed tweet JSON files using Google Maps Geoencoding API. The processed tweet collection is then indexed in Solr. We developed a UI which we named as Twoogle(derived from the names of Twitter and Google) such that Solr acts as the backend of it. We also integrated the analysis of search results to make it more interesting for the user. We also added the advanced search feature to allow the user to filter the tweet results based on the language. A query parser was also developed as part of this project to ensure that our IR system caters to the information need of the user. The homepage of our search engine is as shown below:

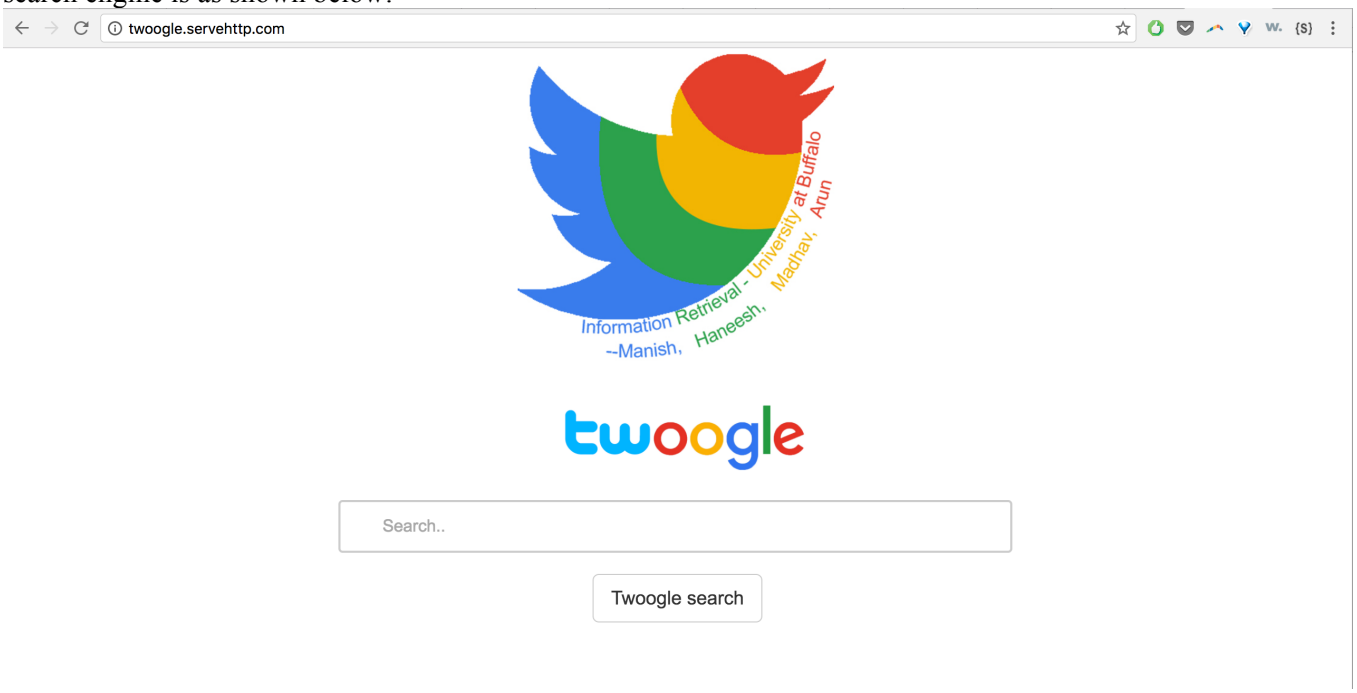

fig- Homepage of Twoogle

FEATURES:

- We crawled tweets in five different languages- English, Spanish, French, Russian and German.
- UI displays 10 results in each page.
- We implemented analysis of returned tweets corresponding to languages, locations, sentiment analysis, top hashtags.
- The results can be filtered based on the language according to user requirements.
- User can query in a language and expect to obtain results in other languages.

IMPLEMENTATION DETAILS:

Preprocessing Data :

We implemented the preprocessing of data similar to the implementation we used for project 1 by using a python script. We gathered information for the fields such as user information fields- followers count, list count, user-name, screen-name, profile-image, location and necessary fields such as date, text, emoticons, hashtags,etc. The data is processed in order to index the data effectively corresponding to the required fields. The preprocessing of data also included the addition of location fields – locality, state, country based on the geographical coordinates information present in the raw tweet files.

Google Maps Geocoding API :

We used the Geocoder python package to extract geographical information from the location attribute in the raw JSON file. The Google Maps geoencoding API provides a direct method to access these services using HTTP requests which is included in the python script used for preprocessing of raw tweet files. The locality, state and country of the user are encoded into the processed JSON file. This information of users can be used to analyse the localities of the users who tweet the most about the search query.

Microsoft Translator API:

We used the Microsoft Translator API which is a cloud based machine translation service. We used the API to detect the language of the given query and translate it to other languages such that the returned results are relevant to the user

Twoogle custom query parser:

We integrated this concept of implementation of detection and translation of queries by building our own query parser which is extended from the extended dismax query parser which is known as edismax query parser, an improved version of dismax query parser. The standard query parser, which is the default query parser, is intolerant of syntax errors as it expects the query to be well-formed. On the other hand, DisMax query parser is known to be a more forgiving parser as it is useful for directly passing in a user-supplied query string. The edismax query parser is an improved version of the dismax query parser with additional features. We boosted the tweets which are in the same language as the language of the query so that this language is favoured over others. In addition to this, if the query contains hashtag, the tweets with hashtags are boosted.

Faceted Search:

Faceting is the arrangement of search results into categories based on indexed terms. This technique of using faceted fields can be exploited in order to perform the analysis of the returned search results in terms of language count, location of the user. Various charts have been created for the visual representation of the analysed data.
The concept of faceted searching is also used to narrow down the search results by setting parameters such as language in order to get much more relevant results for the user.
The top trending hashtags corresponding to the query can also be determined by the concept of faceted search.

Twoogle UI:

We developed a UI for our application by using HTML , PHP and Javascript. Bootstrap was used to enhance the user experience and make the user interface more aesthetically appealing. We used Google charts tool for making the donut charts , pie chart and bar graph for the visualisation of analytical data displayed on the webpage.Our UI and Logo design was inspired from Google and Twitter. The results page displays 10 results at a time in various pages similar to the design observed in Google Search Engine. The advanced search option allows the language based filtering of the tweets. In individual tweets, various information has been displayed such as user name , screen name, retweets , follower count using the data from processed tweet files.

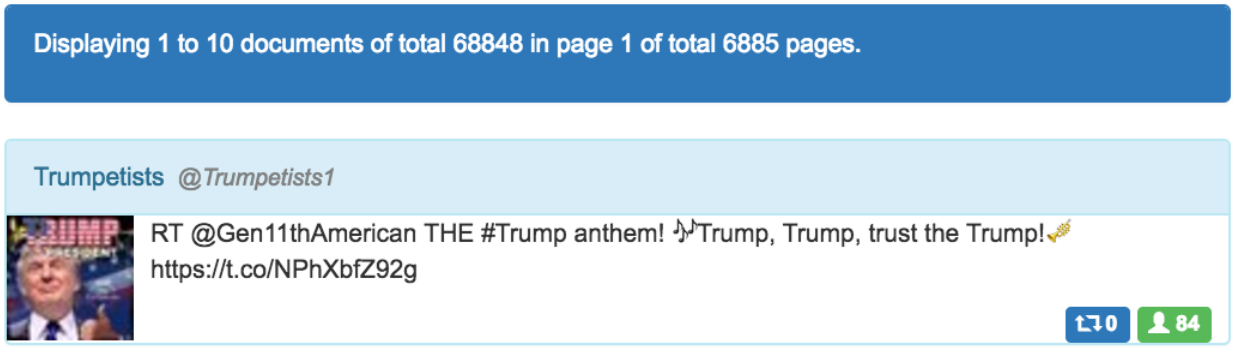

fig- Individual tweet

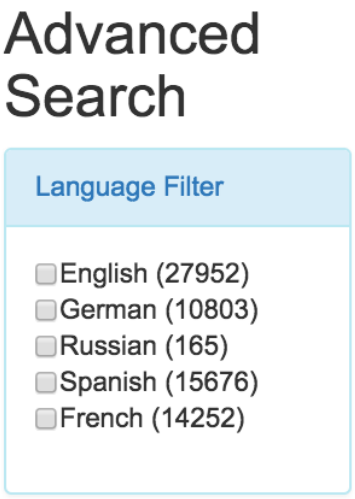

fig – Advanced Search options

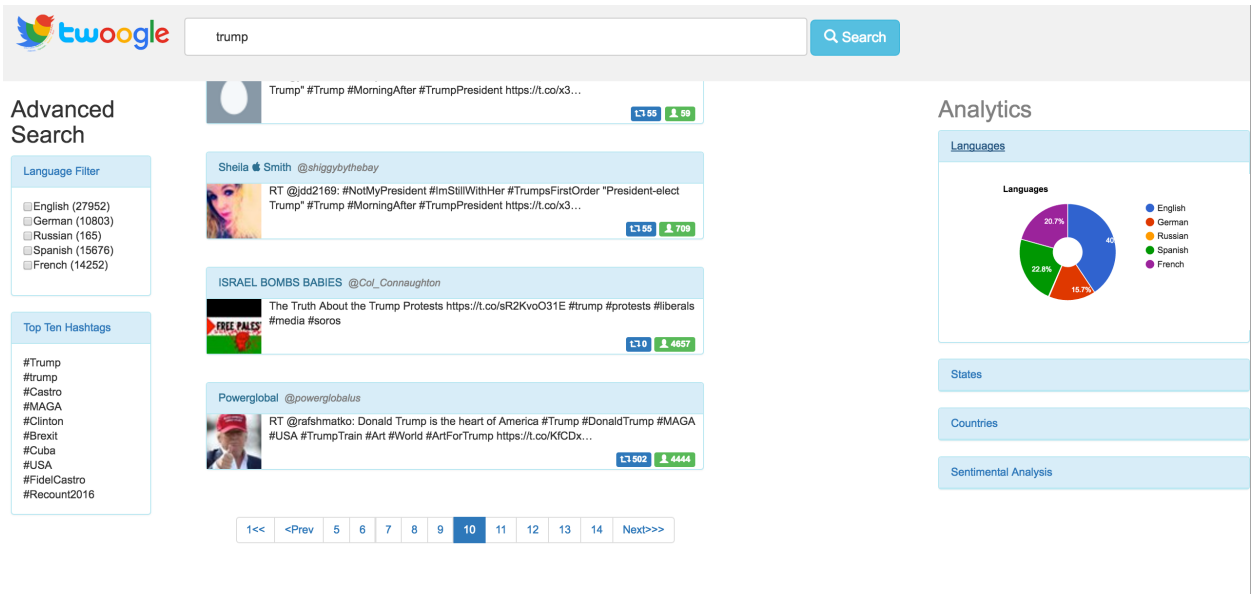

fig – Pages Panel

Pages panel , which is at the bottom of the search results page, has been implemented to organize the tweets into several parts containing 10 results in each page. The pages panel also has the option of directly navigating to the first page.

Sentiment Analysis:

We implemented the simplest form of Sentiment analysis on the tweet results set using a word list called AFINN-111.The file AFINN-111 is the latest version which consists of 2477 words and phrases. The approach we used compares the terms in tweet_text with the list of words in AFINN-111 file which contains pairs of word and precomputed sentiment score associated with the word(ranging from -5 to +5). The sentiment of the tweet is determined by adding up the sentiment components of the individual words in the tweet. For the results returned for the user-given query, the number of positive, neutral and negative tweets present in the search results is displayed.

AFINN-111 file was available only in English language. This file was parsed and translated into the other languages in order to create sentiment based word-lists for other languages. By doing this, we can now derive the sentiment analysis for tweet results of all languages.

RESULTS:

The resulting user interface is as follows:

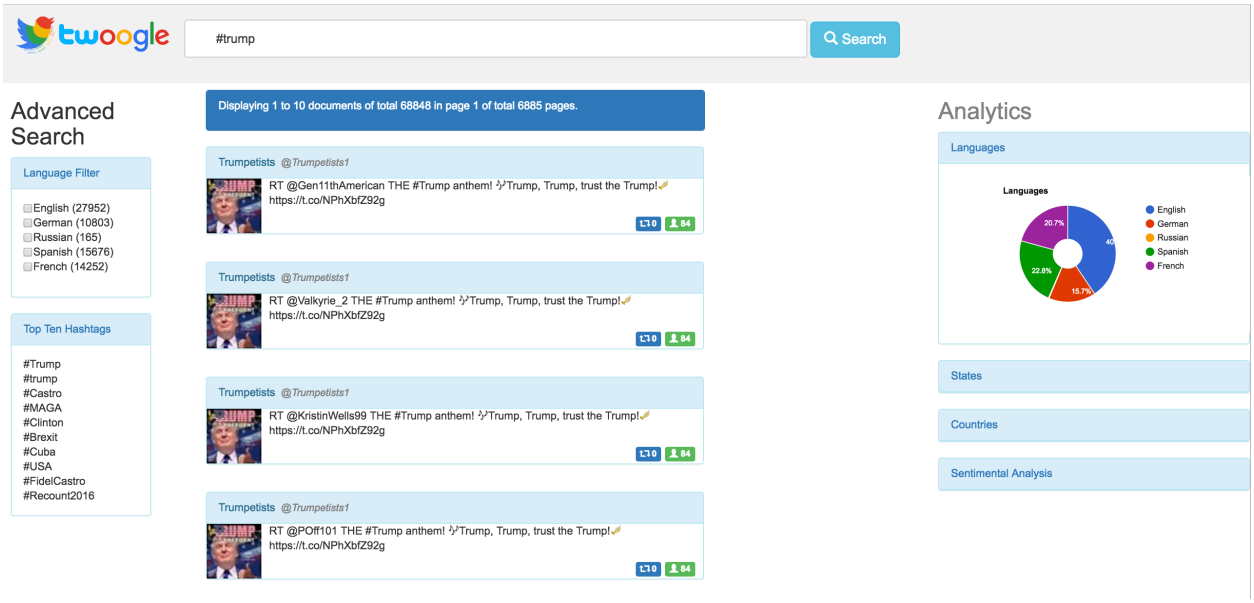

fig – Results page

Analytics:

fig – Top 5 countries tweeting about #trump in our corpus

States

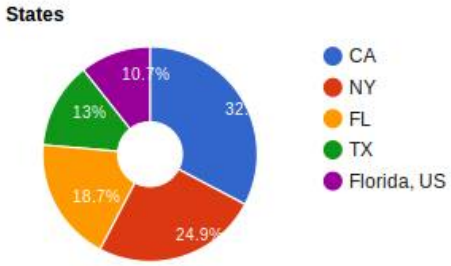

fig – Top 5 states

Languages

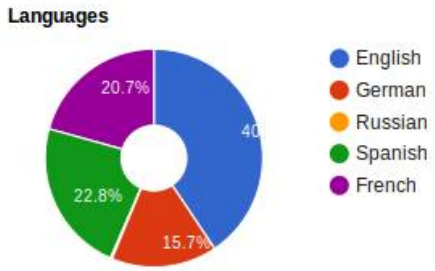

fig – Top 5 languages of tweets

Sentimental Analysis

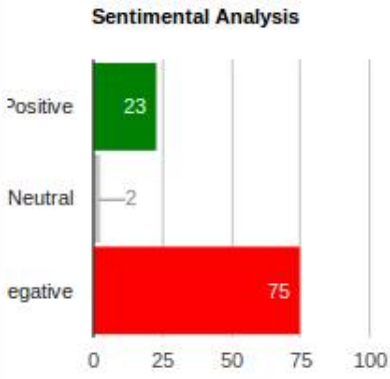

fig – Sentiment analysis of returned tweets

fig – Top ten hashtags

ADDITIONAL IMPLEMENTATION:

We tried out an additional implementation to enhance the innovativeness of the project but it was not used in the final system as the results produced were not favourable.

The crawling system was built in such a way that every request crawls a maximum of 100 tweets. Hashtags from these tweets are extracted and used as feedback to build the search terms for next request to the API. This helped in retrieving more tweets relevant to the topic apart from the tweets retrieved using the initial query. But, in long run this was having a negative effect as there were hashtags in the crawled tweets which were off the topic leading to irrelevant results. Later a strategy was developed to avoid the irrelavent hashtags as feedback by limiting the number of search terms which were appended to the original search term list given to API. The new hashtags are appended to the initial terms and if the number of search terms reach the limit, an appended hashtag is replaced with the incoming hashtag by retaining the original search terms used initially for crawling of tweets. But even this didn't improve the performance of the system.
After investigation of tweets, we realised that performance of this mechanism was not good as the tweets had a lot of irrelevant hashtags along with the hashtags relevant to the topic – Trump. So it was decided to drop this idea in final implementation.

The proposed implementation can be better explained by the following diagram:

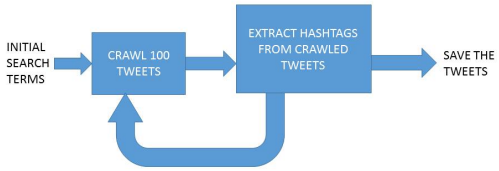

fig – Proposed Crawling Mechanism

VIDEO DEMONSTRATION:

We demonstrated the functionality of our IR system in a video :

https://youtu.be/awYW5DhT9zE

In this, we queried for - Machen Amerika wieder groß (***Make America great again*** in German Language). It was demonstrated in the video that the language of the query was detected and relevant tweets in other languages were returned. The analysis of the results was highlighted in the demonstration.

TEAM CONTRIBUTIONS:

TEAM MEMBER	**UBIT NUMBER**	**UBID**	**TASKS**
ManishKumarReddy Jarugu	MJARUGU	50206843	Complete UI Design
Haneesh Reddy Podduturi	HANEESHR	50208280	Crawling tweets, Geocoding and Preprocessing of tweets
Hema Madhav Jakkampudi	HEMAMADH	50206563	Query parser, Language detection and translation
Arun Chandra Pendyala	APENDYAL	50207136	Report , Sentiment Analysis

REFERENCES:

1) Geocoding API - https://pypi.python.org/pypi/geocoder
2) Microsoft Translator API - https://www.microsoft.com/en-us/translator/translatorapi.aspx , https://tika.apache.org/1.8/examples.html
3) Faceted searching – https://examples.javacodegeeks.com/enterprise-java/apache-solr/solr-faceted-search-example/
4) Sentiment Analysis - http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010 , http://www.slideshare.net/faigg/tutotial-of-sentiment-analysis
5) Google Chart tools - https://developers.google.com/chart/interactive/docs/