

IR Project C - Report

Team - Find My Keys

By -

Ajay Partap Singh Chhokar

Avijeet Mishra

Priyanka Singh

Sahil Dureja

Suramrit Singh

Vaibhav Sinha

Contents.

1. Introduction.
2. Our Corpus.
3. Features Implemented.
 - Content Tagging (Monolingual).
 - Faceted Search.
 - Cross-Document Analytics.
 - Cross-Lingual Retrieval/Analysis.
 - Summarization.
 - Ranking tweets.
4. Conclusion.
5. Resources.

Introduction

The goal of this project was to build a multilingual faceted search system, including a front end that allows users to search and browse multilingual data based on various criteria: topic, location, person, etc.

Below is the Screenshot of our search engine. On top we have search bar, Along with that we animated top 10 hashtags from our corpus. On the left hand side we have option for faceted search. Center top we have the summary and below it we have the tweets.

Finally on the right hand side we have cross document analytics - Tweets distribution Heat Map , Hashtag distribution and Language distribution pie chart(not visible in this screenshot).

Tweet distribution Heatmap, hashtags distribution as well as language distribution chart change according to the retrieved results from the query.



Figure 1 - UI of our search Engine

Technologies used - PHP , Javascript , css , HTML.

Tested on browsers - google chrome , firefox.

Our Corpus

We collected data from twitter over a period of 10 days. We had collected over 11,000 tweets. We used the following hashtags for crawling the data. Our Data consisted of four languages i.e English, Russian , French and German.

#Syria	#messi	#prayforparis
#Refugee	#federer	#sachin
#crisis	#olympics	#helpparis
#paris	#modi	#isis
#attacks	#zuckerberg	#floods
#parisattacks	#facebook	#prayforparis
#PorteOuverte	#USA	#sachin
#paris	#BBC	#ManchesterUnited
#france	#africa	#obama
#terrorism	#india	#trump
#putin	#chennai	#potus
#syrian	#chennai	#tory
#terrorist	#floods	#merkel
#tennis	#disaster	#labourparty
#bombsyria	#desmond	#syriavote
#hollande	#ukflood	#sanders
#jesuisparis	#sports	#floods

On purpose we collected the topics with originated at different locations so that on our Heatmap the distribution is significant. For Example:

Chennai floods had a lot of tweets from India(South India).

Similarly **Paris attacks** had a major distribution of tweets over western Europe and also regarding **sports** example **Manchester United** we see a lot of distribution over UK

Content Tagging

We used Alchemy API for content tagging. AlchemyAPI is capable of automatically tagging your HTML, text, or web-based content. Alchemy employs sophisticated natural language processing technology to analyze the data, tagging the information in a manner similar to human-based tagging. AlchemyAPI's concept tagging API is capable of abstraction, understanding how concepts relate and tagging accordingly ("My favorite brands are BMW, Ferrari, and Porsche." = "Automotive Industry").

Some of the name entities that we extracted are listed below. These named entities were further used in the faceted search.

- City
- Company
- Continent
- Country
- Crime
- Drug
- Geographic
- Hashtag
- Holiday
- JobTitle
- NaturalDisaster
- Organization
- Person
- Quantity
- Region
- Technology
- TwitterHandle

Some of the entities were rarely used such as drug , crime , natural disaster etc.

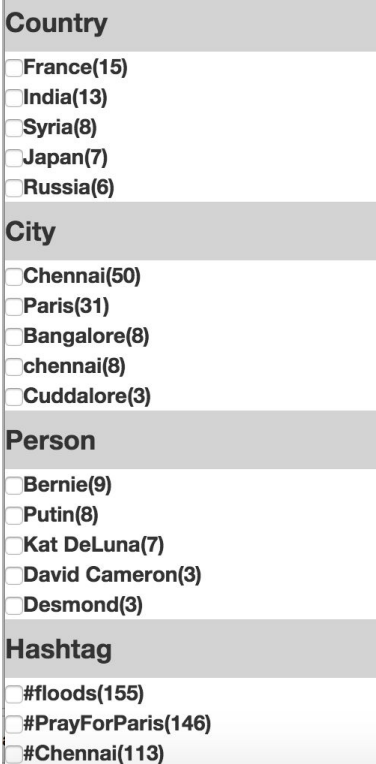
Faceted Search

Faceted search refers to a technique for accessing information organized according to a faceted classification system. A faceted classification system uses a set of semantically cohesive categories that are combined as needed to create an expression of a concept.

In order to give faceted search capability to the user, the system uses some of the entities generated by the alchemyAPI by analysing the corpus of tweets. The system then uses the entities and lists the content within those entities as checkbox items

The user can then select content by checking values within each entity field. The system then re queries the corpus based on the content selected and generates a new entity-content list.

In the example shown below, Under country category we have 15 tweets with france entity , 13 tweets of india , 8 of syria , 7 of japan and 6 of russia. Similarly can be deduced for City , Person and Hashtag Categories.



Country
<input type="checkbox"/> France(15)
<input type="checkbox"/> India(13)
<input type="checkbox"/> Syria(8)
<input type="checkbox"/> Japan(7)
<input type="checkbox"/> Russia(6)

City
<input type="checkbox"/> Chennai(50)
<input type="checkbox"/> Paris(31)
<input type="checkbox"/> Bangalore(8)
<input type="checkbox"/> chennai(8)
<input type="checkbox"/> Cuddalore(3)

Person
<input type="checkbox"/> Bernie(9)
<input type="checkbox"/> Putin(8)
<input type="checkbox"/> Kat DeLuna(7)
<input type="checkbox"/> David Cameron(3)
<input type="checkbox"/> Desmond(3)

Hashtag
<input type="checkbox"/> #floods(155)
<input type="checkbox"/> #PrayForParis(146)
<input type="checkbox"/> #Chennai(113)

figure 2 - example of faceted search

Cross-Document Analytics.

We did 3 things for cross document analytics. All are listed below.

1. Heat Map of user tweeting location.
2. Hashtag distribution.
3. Language distribution pie chart.

For example - For the query “**paris attacks**” we get the following heat map as shown below. We used the google geocoding API to get the location latitude and longitude. we used the user location from tweet to get the location coordinates.

- One of the drawback of this approach was that sometime user entered incorrect location information such “**Everywhere**” or “**Asia**”, so for that case user location was not accurate/wrong.
- Also google geocoding API had a limit of 1000 calls a day after which the API gives no response.

we see a lot of people talking about paris attacks in the europe region and north america region as shown in the heat map below.



figure 3 - Heat Map

For the same query “**paris attacks**” get following hashtag distribution and language distribution. We used google charts to show the next two graph.

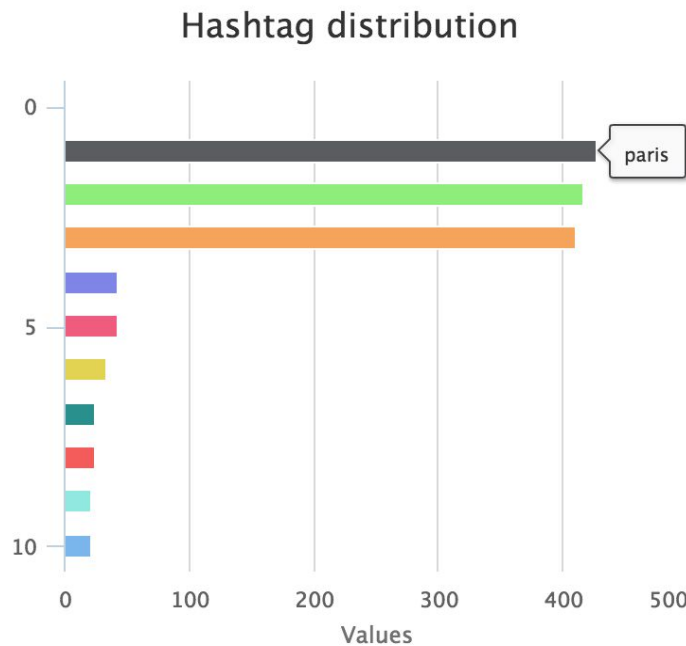


Figure 4 - Hashtag Distribution

we can see that even if our query was in english, still significant amount of tweets returned were in french language(**41.5%**). This also shows our cross lingual search result. This language pie chart is helpful in studying the result responses from different languages for different queries.

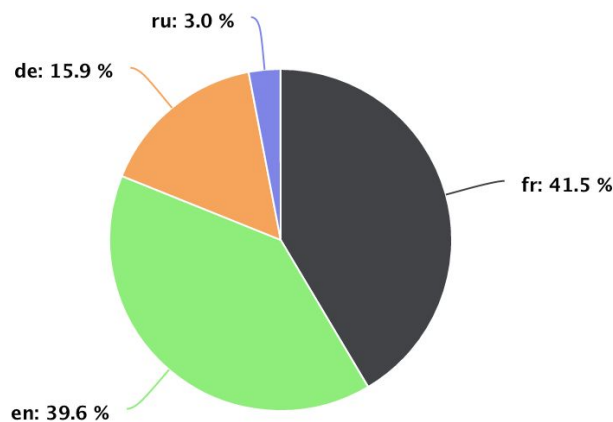


Figure 5 - Language Distribution

Cross-Lingual Retrieval/Analysis

The system supports cross-lingual retrieval (required translations are done at the time of query). The same was implemented in the following manner.

Upon submission of a query, irrelevant of the query language (this makes language detection moot), the system translates the given query into n-languages (n=number of languages present in the corpus. Our corpus contains data in four languages i.e. n=4) . The query is then expanded to include the translated versions as well. An example is given below:

Query: US intervention in Syria

Query sent to Solr:

```
text_en:"US intervention in Syria"~2^4 OR text_de:"US-Intervention in Syrien"~1^4 OR
text_ru:"Вмешательство США в Сирии"~2^4 OR text_fr:"Intervention des États-Unis en Syrie"~2^4 OR
tweet_hashtags:"US intervention in Syria"~2^4 OR tweet_hashtags:"US-Intervention in Syrien"~1^4 OR
tweet_hashtags:"Вмешательство США в Сирии"~2^4 OR tweet_hashtags:"Intervention des États-Unis en
Syrie"~2^4 OR tweet_urls:"US intervention in Syria"~2^4 OR tweet_urls:"US-Intervention in Syrien"~1^4
OR tweet_urls:"Вмешательство США в Сирии"~2^4 OR tweet_urls:"Intervention des États-Unis en
Syrie"~2^4 OR AlcEnt:"US intervention in Syria"~2^4 OR AlcEnt:"US-Intervention in Syrien"~1^4 OR
AlcEnt:"Вмешательство США в Сирии"~2^4 OR AlcEnt:"Intervention des États-Unis en Syrie"~2^4text_en:
(US intervention in Syria) OR text_de:(US-Intervention in Syrien)^3 OR text_ru:(Вмешательство США в
Сирии)^3 OR text_fr:(Intervention des États-Unis en Syrie)^3 OR tweet_hashtags:(US intervention in
Syria)^3 OR tweet_hashtags:(US-Intervention in Syrien)^3 OR tweet_hashtags:(Вмешательство США в
Сирии)^3 OR tweet_hashtags:(Intervention des États-Unis en Syrie)^3 OR tweet_urls:(US intervention in
Syria)^3 OR tweet_urls:(US-Intervention in Syrien)^3 OR tweet_urls:(Вмешательство США в Сирии)^3 OR
tweet_urls:(Intervention des États-Unis en Syrie)^3 OR AlcEnt:(US intervention in Syria)^3 OR AlcEnt:
(US-Intervention in Syrien)^3 OR AlcEnt:(Вмешательство США в Сирии)^3 OR AlcEnt:(Intervention des
États-Unis en Syrie)^3
```

Please note that field 'AlcEnt' is a common field to which all other entities which were identified by Alchemy are copied into. This was done to limit the query length and ease of readability. It is also worth noting that the order in which fields occur in the long OR clause is maintained across languages. Also, sloppy phrase queries are given a higher boosting score than entity fields identified by solr.

We used the Microsoft Translator API for translation. The query expansion helps us to retrieve documents from the corpus which are relevant to the user's query, irrespective of the language user queried in. However, due to limitations on the number of times the translation API could be used, we decided not to translate the retrieved documents into the language used by the user to form his query.

Summarization.

We used NY times API for summarization. We didn't use Wikipedia because we thought that wikipedia gave more generalized results rather than latest information. The link to NY times API is given below.

http://developer.nytimes.com/docs/read/article_search_api_v2

Our Approach was to take the top five hashtags and fetch the response from the NY times API. But many times the response we got was not relevant to the our query. So we decide to do OR between Query and hashtags to get more relevant result.

Query OR (top 5 hashtags)

For example - for query “**support for chennai flood**” the result response is shown below.

Summary - “**Residents in Chennai, India, said that drainage water was mixing with flood waters and that people were afraid to stay inside after torrential rain caused heavy flooding the area.**”

If you read the top tweets you will get to know that the summary is relevant to the query and tweets.

Search Bar: support for chennai floods

Filters:

- TwitterHandle**
 - ☐ @twitter(36)
 - ☐ @ABIDEPress(25)
 - ☐ @geanny83(23)
 - ☐ @CumbriaCrack(22)
 - ☐ @al3xlewis(17)
- Country**
 - ☐ France(15)
 - ☐ India(13)
 - ☐ Syria(8)
 - ☐ Japan(7)
 - ☐ Russia(6)
- City**
 - ☐ Chennai(50)
 - ☐ Paris(31)

Results:

- 1** RT @IMD_Weather: Live Rain: #Chennai is raining heavily, Hindustan university has received 66mm rainfall till 3.30am today #chennairains #C...
Followers: 8 Date:2015-12-05T06:46:33Z
- 2** RT @arvi_NDN: This was in my aunt's bathroom in #chennai, Cobras need a dry Place to stay
to I guess #ChennaiFloods https://t.co/hzTEIELq...
Followers: 1 Date:2015-12-05T06:45:19Z
- 3** Live Rain: #Chennai is raining heavily, Hindustan university has received 66mm rainfall till 3.30am today #chennairains #ChennaiFloods
Followers: 1 Date:2015-12-05T06:39:30Z

Tweet Ranking Algorithm

When we retrieve result from improved IR model from proj B then before displaying the results to the user. We take several factors into consideration for our tweet ranking algorithm such Retweet count , user followers , favorite count and whether user is verified or not.

Pseudo Code Steps:

- 1) Retrieve the results from solr for a simple/ multilingual query with scores and 4 fields **Favourite_Count, Retweet_count, User_Followers, Is_User_Verified**.
- 2) Divide the retrieved results into chunks of 10 10 each .
- 3) Within Each Chunk
apply the below ranking formula
Ranking Formula=
(Favourite_Count+Retweet_count+User_Followers) X Is_User_Verified?**1.5:1**
- 4) Reorder the results according to the new rank within the chunks.
- 5) Join all the chunks into big array.
- 6) Json_encode back and return the result.

```
<?php
function new_rank($json) {
    $decoded = json_decode($json);
    $all = $decoded->response->docs;
    $size = 10;
    $chunks = array_chunk($all, $size, $preserve_keys = false); $new_array = array();
    foreach ($chunks as $key => $value) {
        $TR = array(); $PR = array(); $VI = array(); $STR2 = array(); $PR2 = array();
        for ($twt = 0; $twt < count($value); $twt++) {
            if (($value[$twt]->is_user_verified) == NULL) {
                $value[$twt]->is_user_verified = 0;
            }
            $pv = ($value[$twt]->favorite_count + $value[$twt]->retweet_count + $value[$twt]->user_followers);
            if ($value[$twt]->is_user_verified == 1) {
                $pv = $pv * 1.25;
            }
            array_push($PR, $pv);
            array_push($TR, $value[$twt]->created_at);
        }
        array_unshift($TR, null);
        unset($TR[0]);
        array_unshift($PR, null);
        unset($PR[0]);
        arsort($TR);
        arsort($PR);
        for ($twt = 1; $twt <= count($value); $twt++) {
            $strank = array_search($twt, array_keys($TR));
            $strank = $strank + 1;
            array_push($STR2, $strank);
            $sprank = array_search($twt, array_keys($PR));
            $sprank = $sprank + 1;
            array_push($PR2, $sprank);
            $viralityIndex = $sprank / $strank;
            array_push($VI, $viralityIndex);
        }
        arsort($VI);
        $ranking = array_keys($VI);
        $reordered = array();
        for ($twt = 0; $twt < count($value); $twt++) {
            $r = $ranking[$twt];
            $reordered[$twt] = $value[$r];
            $new_array[] = $reordered[$twt];
        }
        $value = $reordered;
    }
    $decoded->response->docs = $new_array;
    $json = json_encode($decoded);
    return $json;
}
```

Figure - Tweet ranking algorithm

Conclusion

Our Designed System is portable to other other kind of corpus.

As faceted tagging are not restricted to only limited fields on the backend side.

Ranking algorithm has separate functionality i.e can be added or removed in a single line.

Multilingual functionality is also made with modular approach; can be removed from the system by just one line comment.

In future to further expand our project and make it portable for any person to get it working we will provide a configuration page so as to turn on off of the global flags if any of the implemented features such as custom ranking, faceted fields, multilingual, type of charts, etc is required by the user.