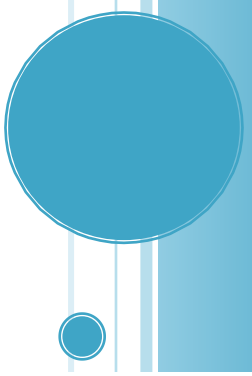
The Tweet Surf Engine

Browse through a wide Ocean of Tweets

Abhishek Gautam #50169657
Alizishaan Khatri #50169045
Armaan Goyal #50170093
Debika Dutt #50170009
Manasi Yerunkar #50169388
Rupesh Soni #50168098



THE TWEET SURF ENGINE

Browse through a wide Ocean of Tweets

INTRODUCTION

The goal of this project is to build a multilingual faceted search engine with an attractive user interface that enables the user to search and browse the multilingual data based on various criteria like topic, location, person, etc. We have implemented a search engine based on data collected from twitter using the twitter API and data related to the current events like Paris attacks, Syrian airstrikes, US elections etc. This data is processed for further querying and search by the user through front end.

DATASET

The dataset to implement this search engine was a corpus of twitter data that we crawled from Twitter spanning more than 20 days. We covered as wide as 10 different categories of tweets, like political, entertainment and sports, our focus being on current topics such as Paris attacks and the problem of Syrian refugees.

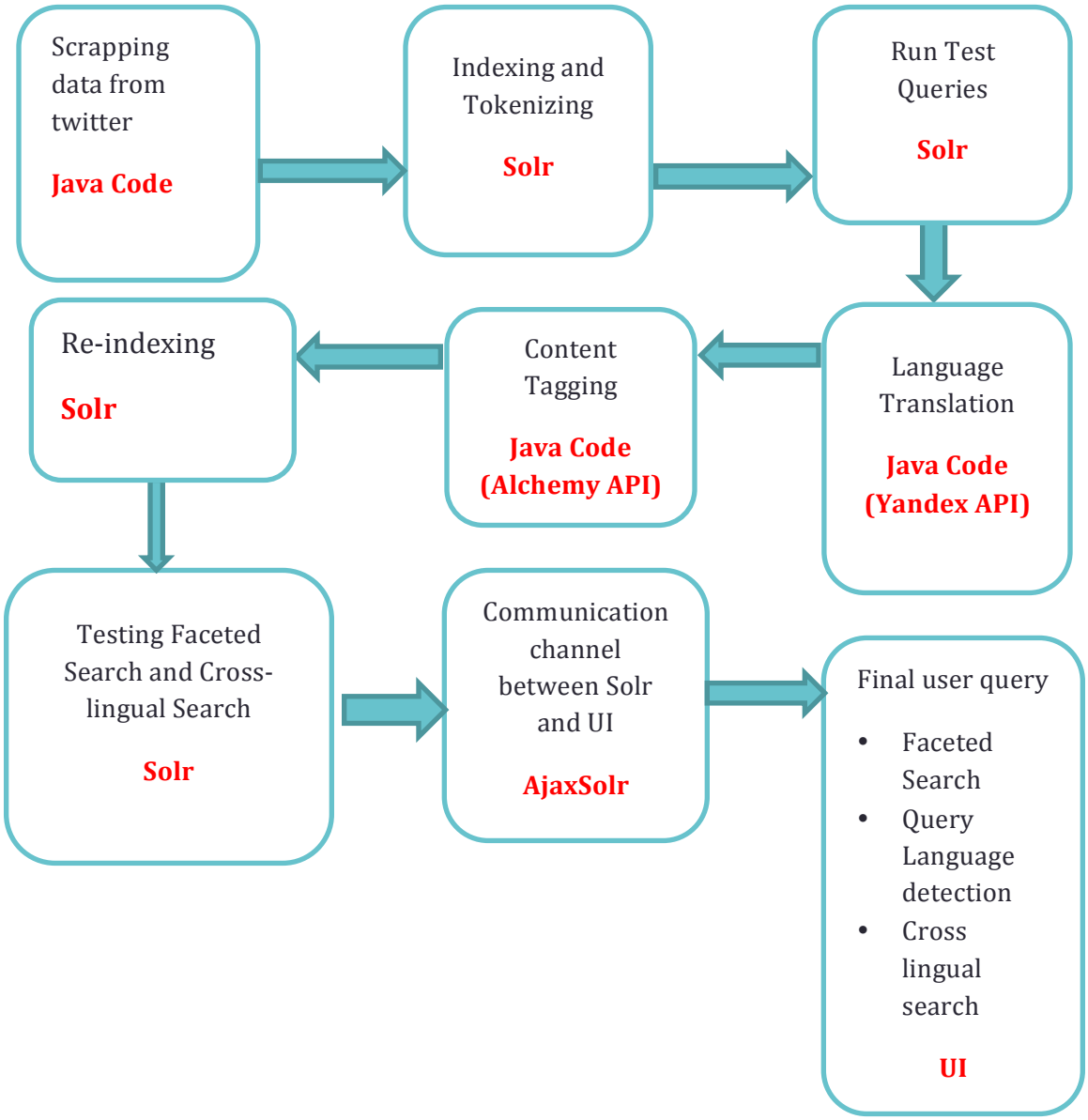
SOLUTION APPROACH

We have built our search engine following a simple approach:

1. The very first step was the collection of data. This was done using simple Java code that took date inputs for the days you wanted data from Twitter.
2. After collecting more than 10K tweets in four different languages, i.e., English, German, French and Russian, we indexed the data and optimized the results using Solr. This was the essence of Project 1.
3. The indexed data can now be queried from Solr, and it gave us the desired results for every query we ran.
4. Next, we implemented language translation, which enables the user to type in a language and to see results in the same language without worrying about seeing results in a language unknown to the user.
5. After the linguistic preprocessing on the data was performed, we applied tags to the data by using the concept of *Content Tagging*. Tagging, is fundamentally a means to classify content to make it structured, indexed and ultimately, useful.

6. Since the data was tagged now, our next aim was to implement *Faceted Search* that uses a hierarchy structure to enable users to browse information by choosing from a pre-defined set of categories. This allows user to type in their search options by navigating/ drilling down. We restricted ourselves from implementing this in a standard drop-down list and checkbox selection method and implemented the concept of *Tag Cloud* thus pushing far from conventional methods.
7. The data retrieved is displayed to the user using the Ajax Solr, which is a JavaScript library for creating user interfaces to Apache Solr.

The below flowchart explains our solution approach depicting the steps and the platform on which these steps are performed:



TECHNOLOGIES USED AND PROJECT DEVELOPMENT

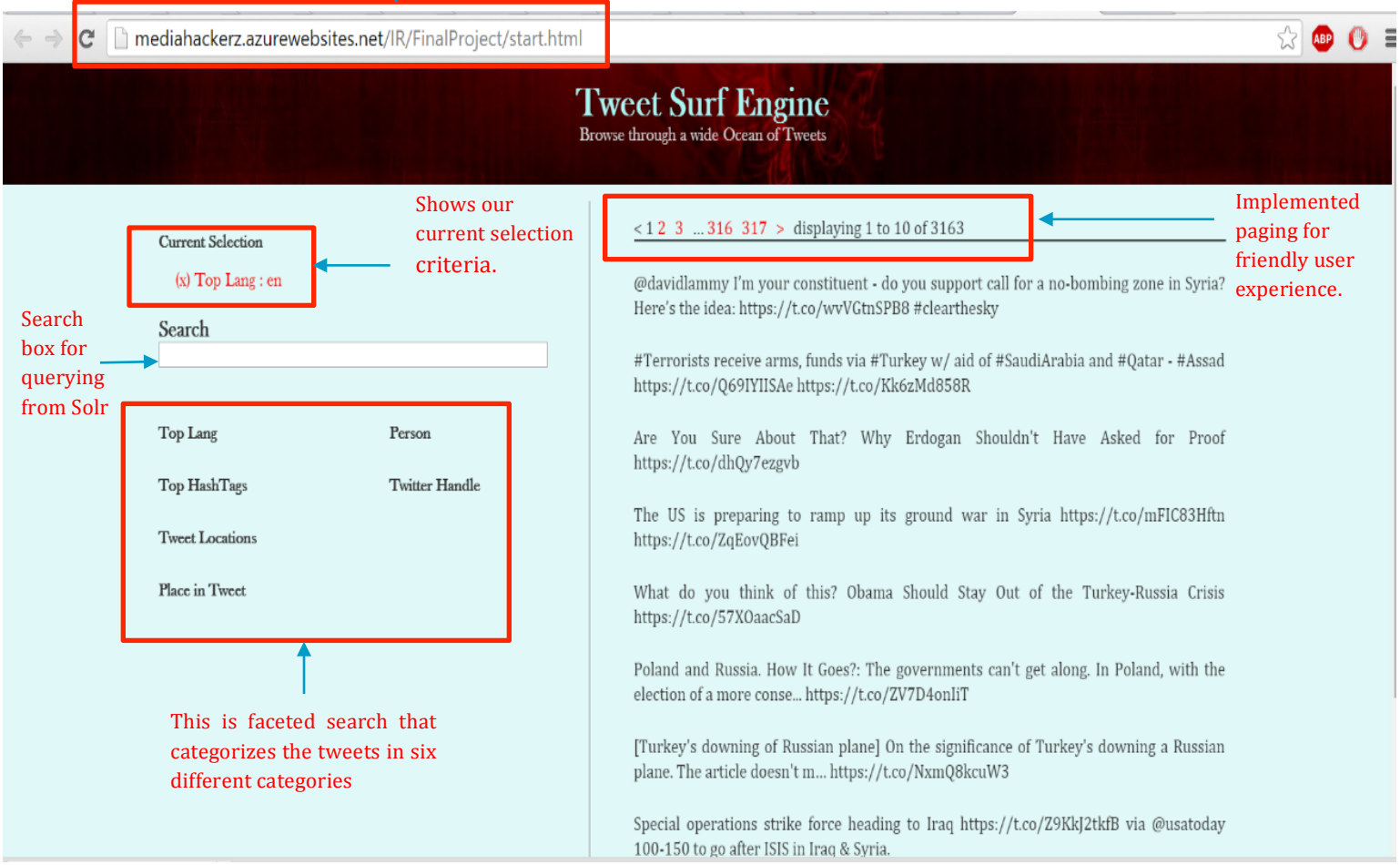
The technologies we have used are as follows-

1. *Virtual Machine with Solr*: It is an open source enterprise search platform, written in Java, from the Apache Lucene project. We used this for indexing and tokenizing the data crawled from twitter. Some of its features include full-text search, hit highlighting, faceted search, real-time indexing, dynamic clustering which we attempted to implement.

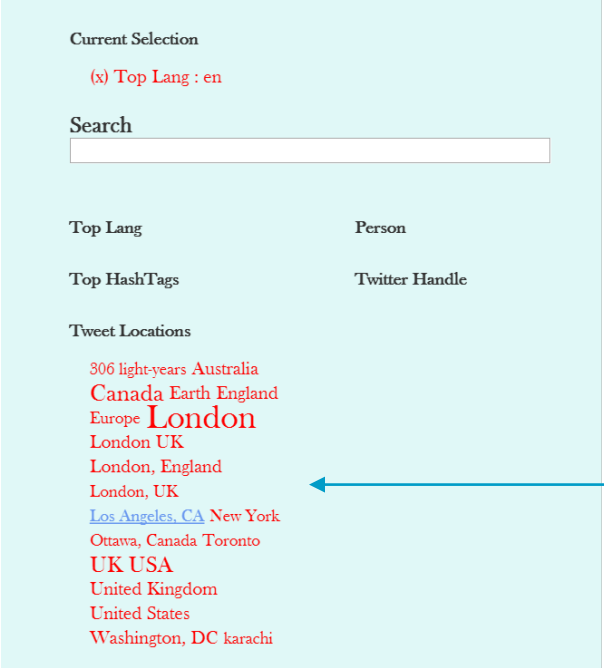
2. *AjaxSolr:* It is a JavaScript framework that uses AJAX implementation for achieving a communication channel between the Solr and our user interface. We have also used JQuery to find the language of the query text using an API so as to enable us to perform language translation and also get the parent query language for query purposes.

A closer view of our deliverable of given on the following pages using snapshots from the website itself.

This is our 24/7 online website featuring thousands of tweets in different languages



3. *Content Tagging:*



We have implemented Tag Cloud in our website design which expands on hover. The Tag Cloud is helps in website navigation as the terms are hyperlinked to items associated with the tag.

It highlights the text on hover and shows the importance of each tag with its font size.

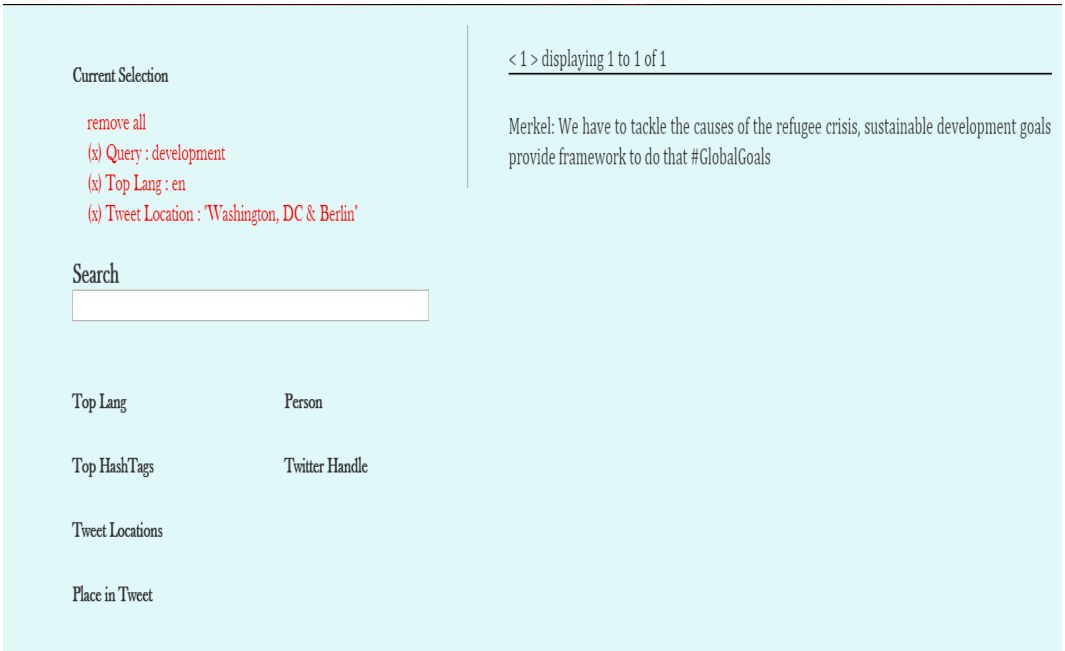
Tags are an important tool that we have used in helping to organize information on our portal and make it easier for our users to find content that they're looking for. Tags are words or phrases that you can attach to any content on the website. Tagging content will make our search results more accurate, and enable us to display content in an organized fashion on a web page.

4. *Faceted Search:*

We have made it easier for the users to quickly spot the most relevant search results by adding faceting functionality.

With faceting, search results are grouped under useful headings, like "Top Language", "Top Hashtags", "Tweet Locations", etc. using tags we apply ahead of

time to the documents in our index. Each time the user clicks a facet value, the set of results is reduced to only the items that have that value. Additional clicks continue to narrow down the search – the previous facet values are remembered and applied again.



In the above screen shot we can see that the user queried for the term "developments", then he further filtered his search to only English tweets, and then again filtered to a particular tweet location. At this point, it is very easy for the user to choose the right tweet result.

Faceted search results provide an easy-to-scan, browser friendly display that helps users quickly narrow down each search.

5. *Cross Lingual Retrieval:*

One of the most interesting functionality of this project was the cross lingual tweet retrieval using language translation. A search performed for a particular term or phrase was translated using APIs so that the search took place simultaneously in multiple languages, four in our case.
The following code snippet shows how the language detection has been done:

```
//Sending POST request to Yandex API to detect language
var test=$.ajax({
    type: "POST",
    url: "https://translate.yandex.net/api/v1.5/tr.json/detect?key=trnsl.1.1.20151211T053517Z.01be4c2cc9c95a6
    data: "&text="+value,
    success: function(data){
        var language=data['lang'];
        //alert(language)


        if(!(language=="en"||language=="de"||language=="ru"||language=="fr")){
            language="en";
        }

        //Build query request
        var value1="text_"+language+":"+value;

        //Send modified request to Solr
        if (value1 && self.set(value1)) {
            self.doRequest();
        }

    },
    dataType: "json"
    });

}
```

This is how it works. A User feeds a query in French language to the engine to present his information need. Strikingly, the engine detects the language by sending a call to the Yandex API through JavaScript POST operation, in which the user typed the query and if you check the results returned closely, the results are not limited to French, but also contain English, German, and Russian tweets. Another interesting thing to note here, is that the tag cloud also implements cross lingual facets. Let's verify this, by taking a look at the hashtags. Indeed! The hashtags contain all the 4 languages as you can see.

6. *User Interface using HTML/CSS:*

   This is a simple user interface created using the HTML and CSS tags to make it dynamic and interactive. This displays the tag clouds created dynamically based on the various facetted search criteria implemented like topic, person, location, etc.

CONCLUSIONS AND FINDINGS

The project was successfully implemented and all the components thought of by the team over regular meetings were completed without fail. Each module was a challenge in itself and achieving them, aided in great learning.

To get a real feel of the product, please visit - http://mediahackerz.azurewebsites.net/IR/FinalProject/start.html

To view a demo, please visit – https://youtu.be/PTdd4fWmj1w

REFERENCES:

https://cwiki.apache.org

https://tech.yandex.com

http://www.alchemyapi.com