

CSE 535: Information Retrieval Project 4

Analyzing the impact of political rhetoric in traditional and social media

Aashana Mahajan (aashnama@buffalo.edu) - 50317416

Navpinder Singh (navpinde@buffalo.edu) - 50320145

Prithvisagar Rao (psrao@buffalo.edu) - 50320727

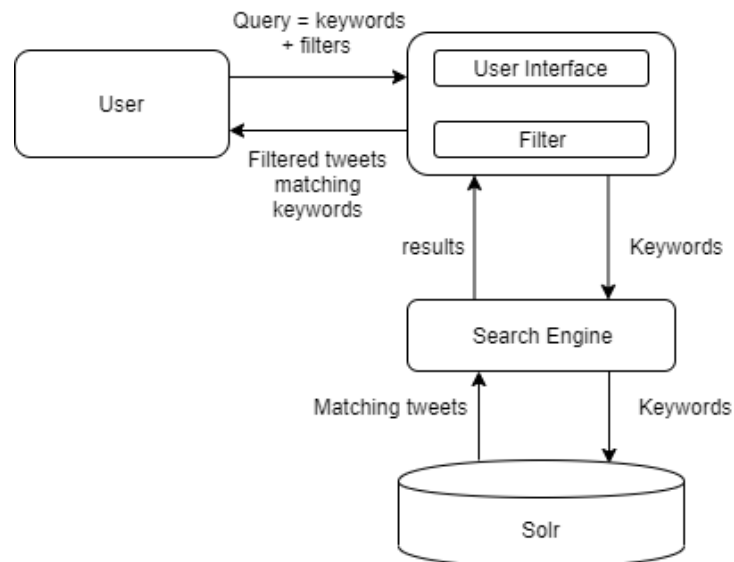
Priya Rao (prao4@buffalo.edu) - 50321961

Table of Contents

Analyzing the impact of political rhetoric in traditional and social media	1
Introduction	3
Implementation.....	3
Data Preprocessing.....	3
Language Translation.....	4
Sentiment Analysis	4
Topic Modeling.....	4
Solr.....	5
Data Analysis and Visualization.....	5
Tweet Analysis.....	5
Language Analysis.....	5
Topic Analysis	6
Sentiment Analysis	6
Hashtags Analysis	7
Global Distribution.....	8
Related Articles.....	8
UI Development	9
Video demonstration.....	10
Team contributions	11
References	11

Introduction

This project entails building an end-to-end IR solution involving content ingestion, search, topic categorization, data analytics and visualization. As part of project 1, at least 33,000 multilingual (English, Hindi and Portuguese) tweets were collected which included 15 POIs across 3 countries. This data was used during the process of content ingestion and was indexed using solr. As part of project 2, the algorithms behind query processing and scoring were explored by implementing Document As A Term (DAAT) AND and OR. As part of project 3, interpretation of the map scores using similarity models of DFR, LM and Okapi BM25. This project is a blend of all the above three mentioned projects along with designing a full-fledged website for searching, analyzing and visualizing the Twitter data collected. A general architecture of the system is depicted as:



Implementation

Data Preprocessing

As part of this project, data preprocessing was necessary in order to obtain the sentiments of all the tweets, including the POI's tweets and the replies per tweet. Also, due to the presence of multiple topics within the window of the tweets taken, a rigid topic list was necessary. But the limitation here was that due to the presence of a multilingual corpus, the tweets first has to be translated and then could be used for the aforementioned purposes

Language Translation

For language translation the cloud translation API provided as part of the Google Cloud Platform was integrated as part of a python script in order to skim through the tweets, translating only those which had the “*tweet_lang*” field other than that of English(en). This new translated field named “*translation*” as shown in the image below was then used for the purpose of Sentiment Analysis and Topic Modeling. If the language was already English, then the *text_en* field was copied as is to the translation field.

```
"tweet_text": "क्या भाजपा ऐसे राम राज्य की बात करती है? \n\nहम हिंदुओं के श्री राम चन्द्र जी के राज्य में तो महिलाओं का सम्मान होता था, कोई किसी महिला की तरफ अँख उठा कर \n\nभाजपा वालों, भगवान श्री राम को इस तरह बदनाम ना करो! https://t.co/70SMq5n9VD",
"tweet_lang": "hi",
"text_hi": "क्या भाजपा ऐसे राम राज्य की बात करती है हम हिंदुओं के श्री राम चन्द्र जी के राज्य में तो महिलाओं का सम्मान होता था कोई किसी महिला की तरफ अँख उठा कर नहीं देखता",
"text_en": null,
"text_pt": null,
"hashtags": null,
"mentions": null,
"tweet_urls": [
  "https://t.co/70SMq5n9VD"
],
"tweet_emoticons": null,
"tweet_date": "2018-07-08T11:00:00Z",
"tweet_loc": null,
"subjectivity": 0.75,
"polarity": 0.065,
"sentiment": "Positive",
"translation": "Does BJP talk about such a Ram Rajya? In the state of Shri Ram Chandra ji of us Hindus, there was respect for women, no one looked at any w
```

Sentiment Analysis

Using the translation field generated, sentiment analysis was performed using the TextBlob library in the python script. This generated a polarity which is essentially a float value in the range [-1, 1]. If the polarity was zero, the sentiment was classified as Neutral, if the polarity was greater than zero, it was classified as Positive and if it was lesser than zero then the polarity was classified as Negative. The screenshot in the above image shows a Positive sentiment tweet since the polarity was greater than zero. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective. Hence, the above tweet is a highly subjective tweet as it depicts a value of 0.75.

Topic Modeling

After obtaining the translated texts of all tweets, the tweets were split country wise. The translated texts of these country wise tweets were then grouped together. Then further data processing was performed using stemming, lemmatization and removing stop words specific to the corpus and the languages. Then a bag of words is obtained for each topic using Latent Dirichlet Allocation or LDA. The number of topics were decided based on how vast the corpus itself was. Upon repeatedly training and using the number of topics as the hyperparameter, we came up with a specific bag of words pertaining to a certain number of topics per country. The tweets were then mapped to the closest topic based on similarity and the reply thread of that tweet was mapped to the topic of the original tweet.

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 this corpus. We add the following snippet into the schema.xml file to further tweak the values of the model parameters b and k1: BM25 has two parameters, k1 and b, that can be tuned to improve the performance of the information retrieval system. After sufficient trial and error, we concluded that the default values of b = 0.75 and k1 = 1.2 were ideal for the setting of this corpus.

```
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.BM25SimilarityFactory">
    <float name="b">0.75</float>
    <float name="k1">1.2</float>
  </similarity>
</schema>
```

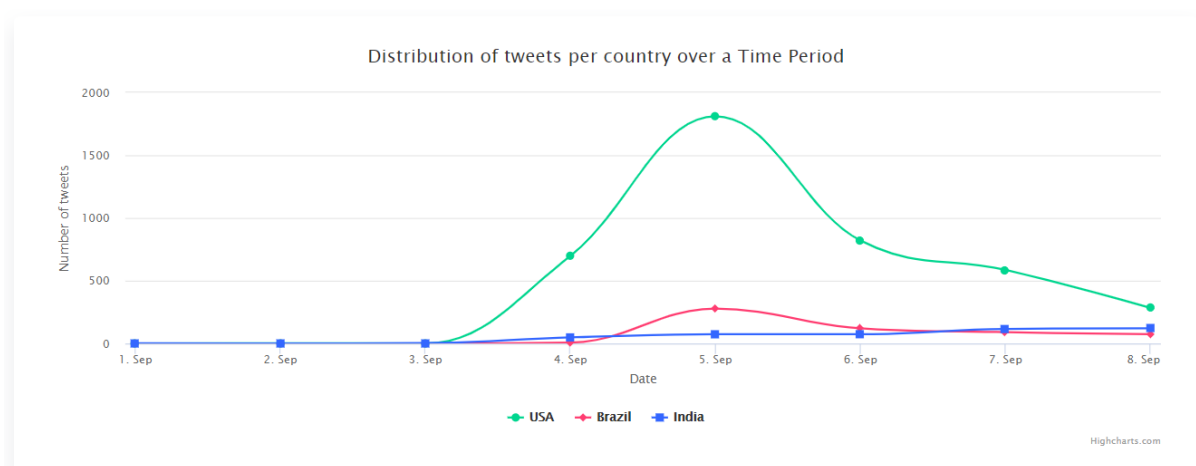
In Solr, in order to obtain the correct ranking of the results, the eDisMax parser was also used. Similar to DisMax, eDisMax not only reduced the number of errors in user-facing systems with limited management control that result from direct user queries but also provided a boost parameter which raised the score of a query in a similar fashion to DisMax's 'bf' parameter but functioned as a multiplicative parameter rather than an additive parameter.

Data Analysis and Visualization

For the purpose of data analysis and visualization, charts as well as related articles were also drawn from the corpus. The charts were constructed using Highcharts since it is a multi-platform charting library and a subset of Highcharts – Highmaps which was used in the construction of the maps. The detailed chart analysis are as follows:

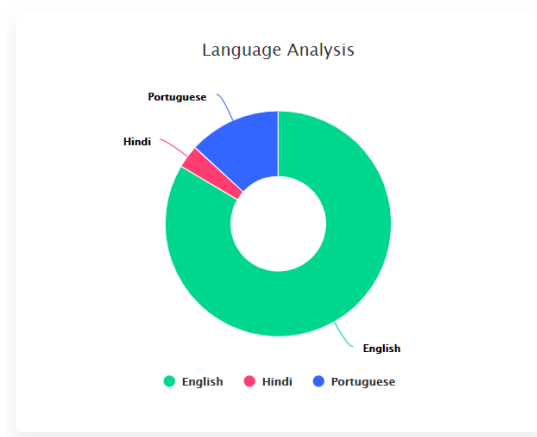
Tweet Analysis

The tweets distribution over a period of time is given by the below time series graph:



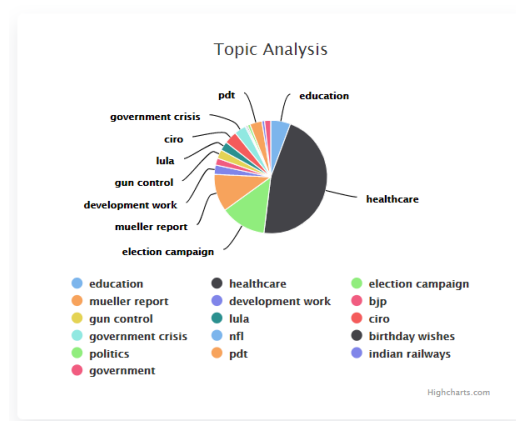
Language Analysis

The language analysis was done using a pie chart which depicted the distribution of the languages of tweets particular to the query.



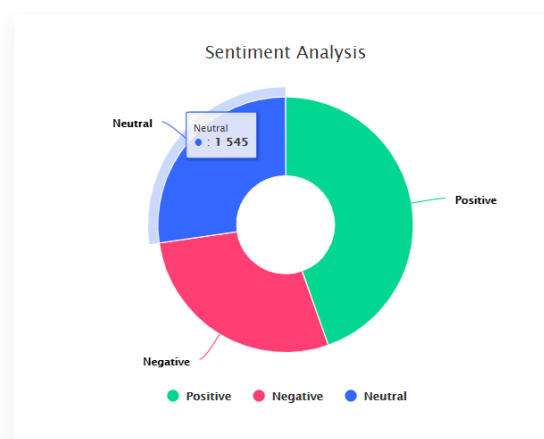
Topic Analysis

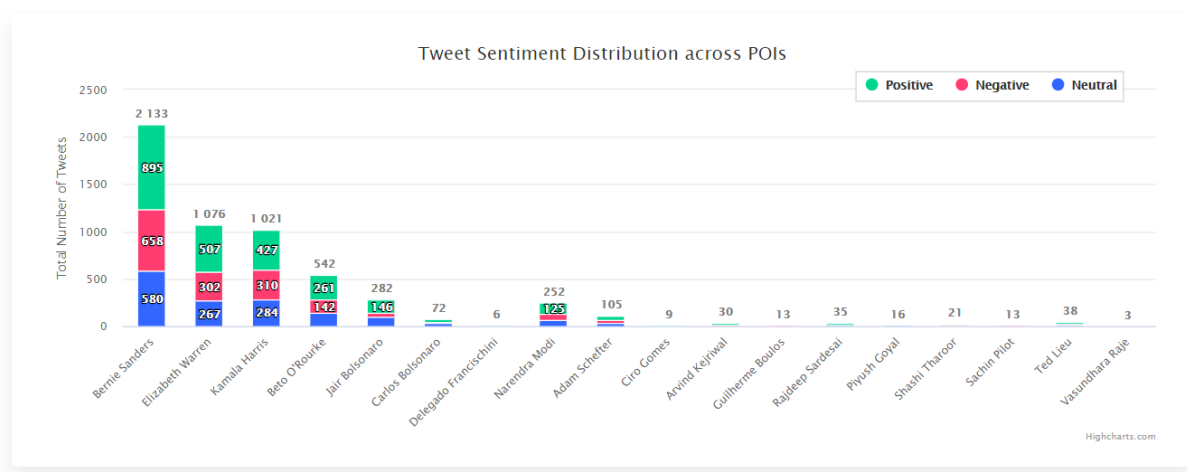
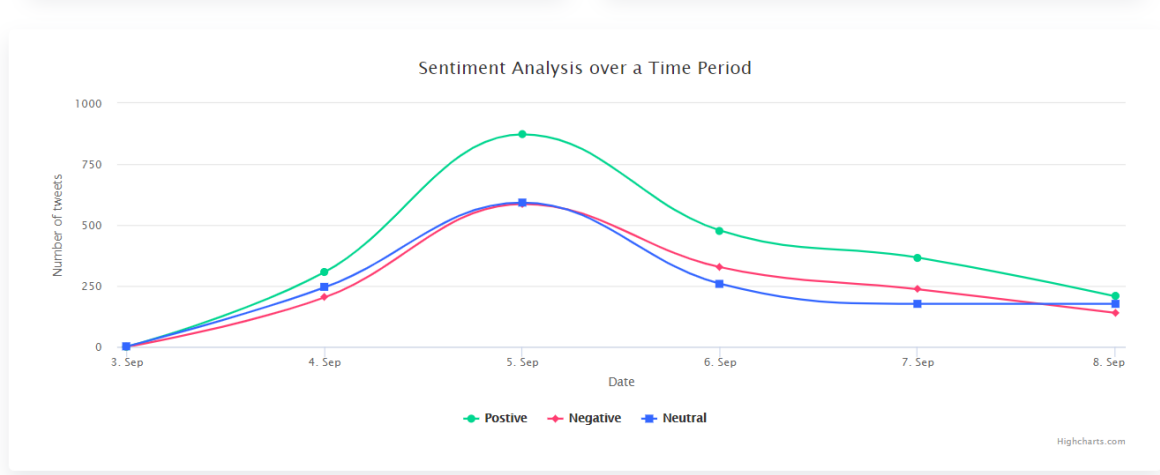
The topics distribution across the entire dataset can be analyzed in the below pie chart.



Sentiment Analysis

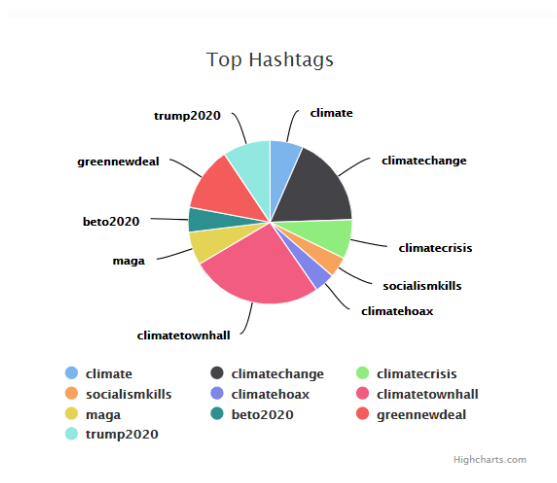
Sentiment analysis is depicted by three charts – one of which is a donut chart showing the distribution of Negative, Neutral and Positive pieces, another which is a time series chart which shows the spread of the sentiments across a given window of days and the last one which is a bar graph of the count of sentiments across POI's. All these charts are drawn from the resulting tweets of the query that the user has typed in.





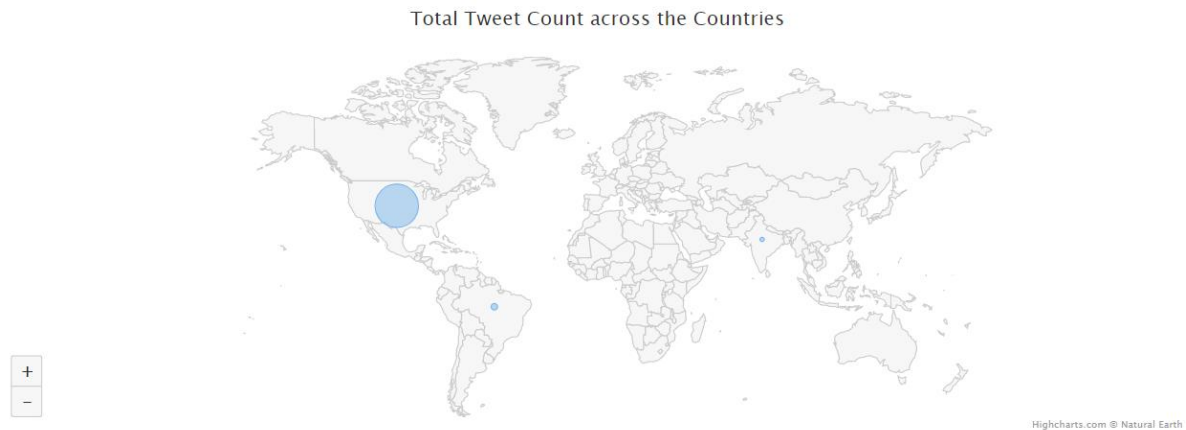
Hashtags Analysis

Hashtags encountered from the resulting tweets pertaining to a particular query are depicted in the pie chart as shown below:




Global Distribution

The global distribution (world map) of tweets with respect to the query entered is depicted as a map bubble as shown below. This gives the proportion of tweets that were relevant to the query which were mentioned by users across the three countries.



Related Articles

The related articles tab in the website corresponds to all news articles which were related to the topic being addressed by the POI's tweets. The POIs were obtained by crawling through the results of the query entered by the user and using the GNews API which would then fetch the relevant articles that could've been the cause of influence for the tweets within a specific time window. For the time window we have taken into consideration all the dates when the POI has tweeted and fetched all news articles between a few days after the POI's most recently tweeted date and the oldest tweeted date. This way for the API, the query is generated using the POI name and query time date calculation as above to see the societal impact of the POI's tweet. GNews API was used for its feature-rich API library and the structured object which returns the news articles.

Q

Country ▾

Language ▾

Person of Interest ▾

Only POI Tweets ▾

Verified ▾

Topics ▾

Sentiments ▾

Hashtags ▾

Showing Top 1000 tweets for climate change

RESULTS

ANALYTICS

RELATED ARTICLES

The thirteenth hour at Jaipur Literature Festival

a discussion featuring Boria Majumdar, TV journalist Rajdeep Sardesai and MP Shashi Tharoor ... migration, colonisation, climate change, desire, love, mythology, art and Bollywood. It was also a day ...

Published by The New Indian Express on Feb 3, 2019

Bolton visits Bolsonaro: a US meeting of the minds with Brazil's fascistic president-elect

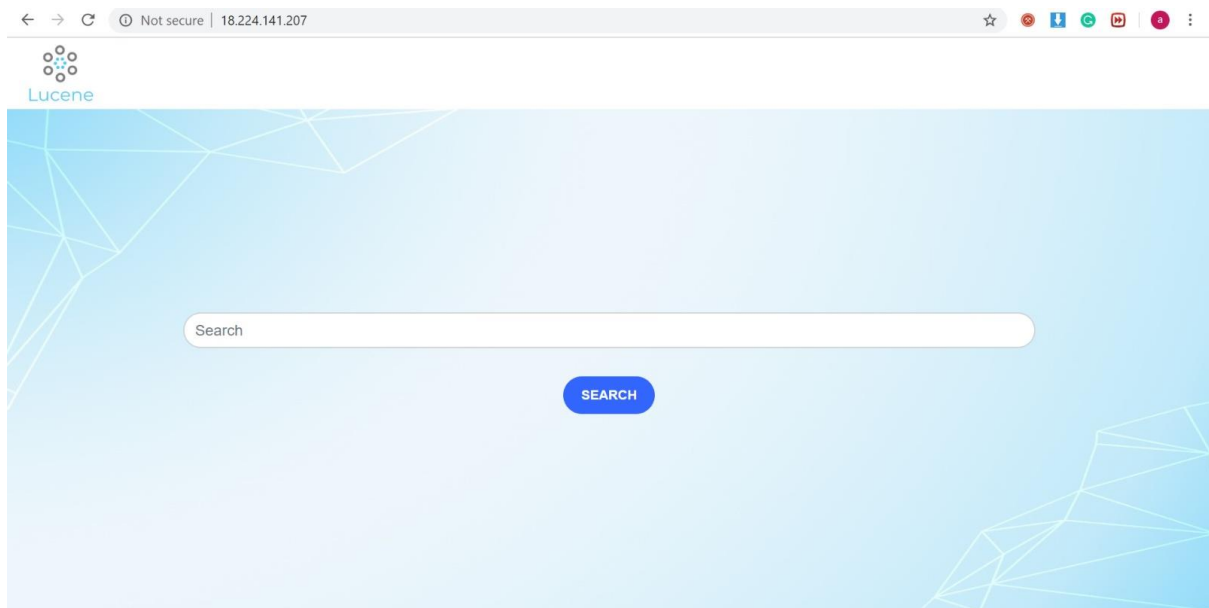
Also present was his foreign minister, Ernesto Araujo, who has described climate change as a plot by "cultural ... Pseudo-left leader Guilherme Boulos of the Homeless Workers Movement (MTST) described ...

Published by wsws on Nov 29, 2018

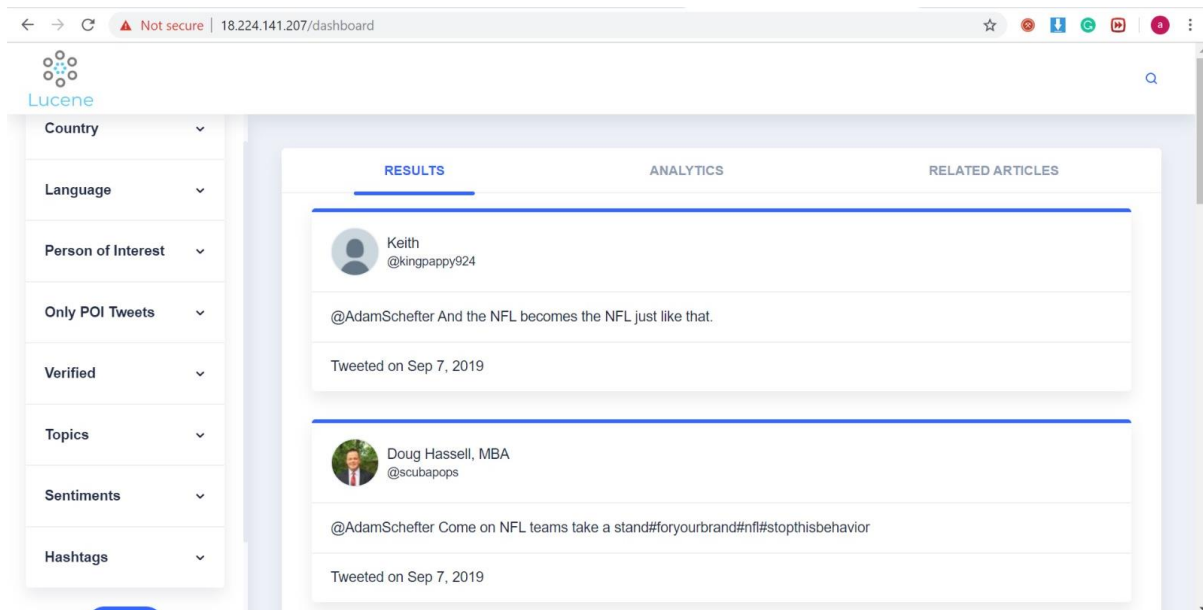
UI Development

For the purpose of UI development, Angular 8 was used along with Nebular wrapper. Angular 8 was considered for the purpose of UI creation since it is an open-source, client-side TypeScript based JavaScript framework and it is very similar to its preceding versions except for a few features newly integrated. The Nebular wrapper was used since it is a highly customizable Angular UI Library based on Eva Design System specifications.

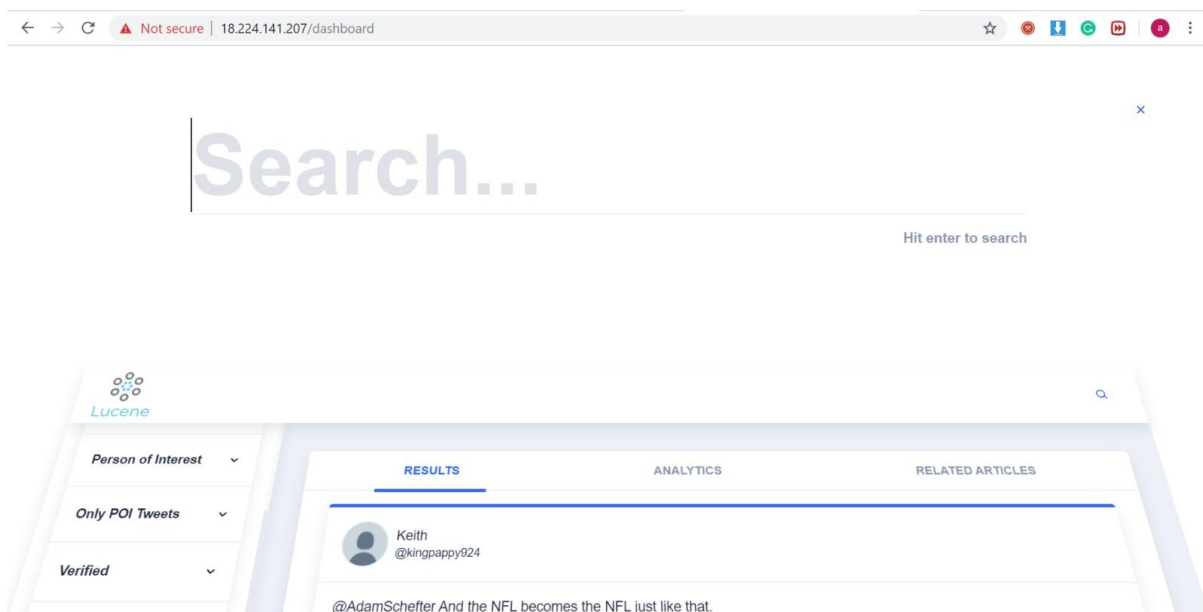
The homepage of our project displays a search bar along with the team logo as seen in the below screenshot:



The tweets resulting from the query entered are displayed as cards in the UI as shown below. The tweets can be filtered based on the Country (Brazil, India and USA), Languages (English, Hindi and Portuguese), Person of Interest (POI), Topics, Sentiments and Hashtags (faceted search). There are also boolean filters of obtaining only POI tweets or only tweets from verified users. These filters when applied are applicable to all three tabs of Results, Analytics and Related Articles.



The user can also carry out a new search on this same page as shown in the screenshot below:



Video demonstration

The video demonstration for this project can be found at the link below:

<https://youtu.be/OPqHPOODSM4>

Team contributions

Team Member	Contribution
Aashna Mahajan	UI Design and overall website functionality
Navpinder Singh	Data analytics and visualization
Prithvisagar Rao	Solr data integration and related articles tab
Priya Rao	Language translation, sentiment analysis and topic modeling

References

<https://cloud.google.com/docs/>

<https://angular.io/docs>

<https://www.nltk.org/>

<https://gnews.io/>

<http://flask.palletsprojects.com/en/1.1.x/>

<https://lucene.apache.org/solr/resources.html#documentation>

<https://stackoverflow.com/>

<https://textblob.readthedocs.io/en/dev/>

<https://www.kaggle.com/ktattan/lda-and-document-similarity>

<http://youtube.com/>

<https://www.highcharts.com/demo>

<https://www.kaggle.com/younad/data-exploration-and-topic-modeling-lsa-vs-lda>

<https://towardsdatascience.com/scraping-web-articles-using-newsapi-in-python-a0e97fbab8ed>

<https://akveo.github.io/nebular/docs/getting-started/what-is-nebular#what-is-nebular>