CSE 435/535: INFORMATION RETRIEVAL

PROJECT REPORT

INSTRUCTOR

ROHINI SRIHARI

PROJECT 4 : DISSECTING TWITTER DATA TO ANALYZE GOVERNMENT AND PUBLIC ATTITUDE TOWARDS COVID AND VACCINES

Team Name : Deep Mind

Name	UB ID Number	Email
Tanush Tripathi	50411177	tanushtr@buffalo.edu
Sarveshwar Singhal	50418642	sarveshw@buffalo.edu
Oviyaa Balamurugan	50418472	oviyaaba@buffalo.edu
Aravind Balakrishnan	50412479	balakri2@buffalo.edu

INTRODUCTION:

With billions of contents online, there is a lot of information on the Internet and search engines make this information easier to find.It helps people find the information they are looking for online using keywords or phrases and return results quickly.

When a user enters a query into a search engine, a search engine results page (SERP) is returned, ranking the found pages in order of their relevance. It aims to understand how users search and give them the best answer to their query. This means giving priority to the highest quality and most relevant pages.

The information is indexed according to keywords. Once the information is indexed, users can search through that index by typing a query into the search bar. These searches use algorithms that sift through the massive index to find the most relevant results. When you submit a query, the algorithm simultaneously assesses the meaning of your query, the relevance of indexed webpages, the quality of the content on those webpages, the functionality of those webpages, and the context of your search.

You can search tweets posted by someone for whom you're searching, or tweets about a covid/vaccine related topics that interests you. For example, search for "Joe Biden", and the latest tweets from the president's Twitter account pop up in your search results.

IMPLEMENTATION:

Technologies used: python3.7, Tweepy, Flair, Latent Dirichlet Allocation(LDA), PySolr,

Data set: Twitter Data Comprising Political and Health Department POI tweets and News POI Tweets for each of the three countries (India, USA , Mexico), the reply tweets by the general public on Political and Health Department POI tweets.

Indexing: BM-25 Indexing onn Solr using the parameters b=0.5,k1=1.2

Code:

A sample of Political and Health agencies POI Tweets Indexed in Solr:-


```
{
        "id":"1467704133176028928",
        "poi_name":"narendramodi",
        "poi_id":18839785,
        "verified":true,
        "country":"India",
        "tweet_text":"India’s vaccination drive crosses another important
milestone. Important to keep this momentum to strengthen the fight against
COVID-19.   And yes, keep following all other COVID-19 related protocols
including masking up and social distancing. https://t.co/a26Cy65Jv2",
        "tweet_lang":"en",
        "text_en":"Indias vaccination drive crosses another important milestone.
Important to keep this momentum to strengthen the fight against COVID-19. And
yes, keep following all other COVID-19 related protocols including masking up
and social distancing.",
        "mentions":["NaN"],
        "tweet_date":"2021-12-06T04:00:00Z",
        "tweet_emoticons":["NaN"],
        "tweet_urls":["['https://t.co/a26Cy65Jv2']"],
        "text_hi":"NaN",
        "hashtags":["NaN"],
        "text_es":"NaN",
        "sentiment":"0.188451454043388",
        "max_sentiment":"0.999800026416779",
        "min_sentiment":"-1.0",
        "negative":"What use is it if the unvaccinated children are still being
made to go and give offline th and th board exams while the cases are still
rising as they are the ones who will be most affected. The entire progress will
be useless because, the lives of children are at a high risk",
        "positive":"our H'blePMNMJi who by initiating innovative reforms in
policy matters made the things move fast via automatic mode. Here, we must
acknowledge(3/4)",
        "total_replies_per_poi":"344.0",

"words":"https,india,important,milestone,crosses,largestvaccinedrive,vaccination
,making,strengthen,need",
        "type":"r",
        "replied_to_tweet_id":"0",
        "replied_to_user_id":"0",
        "_version_":1718895008293060608},
```

A sample of News Agencies agencies POI Tweets Indexed in Solr:-


```
{
        "id":"1464274015497949210",
        "poi_name":"nytimes",
        "poi_id":807095,
        "verified":true,
        "country":"USA",
        "tweet_text":"RT @AliWatkins: Pre-COVID, NYPD was solving nearly 90% of
murders. But that rate plunged over COVID, even as violent crime rates
spiked.…",
        "tweet_lang":"en",
        "text_en":": Pre-COVID, NYPD was solving nearly % of murders. But that
rate plunged over COVID, even as violent crime rates spiked.",
        "mentions":["['AliWatkins']"],
        "tweet_date":"2021-11-26T17:00:00Z",
        "tweet_emoticons":["NaN"],
        "tweet_urls":["NaN"],
        "text_hi":"NaN",
        "hashtags":["NaN"],
        "text_es":"NaN",
        "sentiment":"-0.8224",
        "max_sentiment":"0.0",
        "min_sentiment":"0.0",
        "total_replies_per_poi":"0",
        "words":"https,covid,delhi,don,vaccine,know,country,govt,miss,trump",
        "type":"n",
        "replied_to_tweet_id":"0",
        "replied_to_user_id":"0",
        "_version_":1718897670223822848}
```


A sample of reply tweets to Political and health agency POI Tweets Indexed in Solr:-


```
{
        "id":"1467554371403588096",
        "country":"USA",
        "replied_to_tweet_id":"1467552606230822912",
        "replied_to_user_id":"146569971",
        "reply_text":["Covid shots do not protect against covid ."],
        "tweet_text":"Covid shots do not protect against covid .",
        "tweet_lang":"en",
        "text_en":"Covid shots do not protect against covid .",
        "mentions":["['CDCgov']"],
        "tweet_date":"2021-12-05T18:00:00Z",
```

```
"hashtags":["NaN"],
"tweet_emoticons":["NaN"],
"text_es":"NaN",
"tweet_urls":["NaN"],
"verified":false,
"text_hi":"NaN",
"sentiment":"-1.0",
"words":"covid,sir,new,best,son,government,come,president,joe,biden",
"max_sentiment":"0.0",
"min_sentiment":"0.0",
"total_replies_per_poi":"NaN",
"type":"re",
"poi_name":"Twitter User",
"poi_id":0,
"_version_":1718912049512185859},
```


(We have maintained uniformity while we are indexing the data in Solr , so there are some columns which are defaulted to zero , while some are only indexed but not used later(for analysis purposes only). These are not displayed in the UI for that particular type of tweet.)

METHODOLOGY:

1.Working with the Twitter Data:-

1.1 Data Scraping From Twitter API and Data Collection

Data for this project was collected from Twitter using the Twitter API.The data comprised of

1. POI Tweets , which are the political and health department entities, and the news agencies from all the three countries (India,USA and Mexico) and in all the three languages (English, Hindi , Spanish)

2.Non-POI Tweets in the form of replies to POI Tweets.

The scraper is a python script using the tweepy library and the Twitter search API. Post obtaining the raw data , we preprocessed the data using a preprocessor library, which is responsible for data cleaning activities such as removing hashtags, emoticons and urls from the cleaned text , as well as modifying the date to a suitable format

We have collected close to 60000 processed tweets and later saved them as a pickle file for further analysis.

1.2 Analyzing the Data

After we have obtained the tweets using the twitter API, the next task was to analyze the tweets and to gain valuable insights from it . Firstly, we linked the Political and health agencies related poi tweets with the reply tweets corresponding to that particular reply tweet. We also calculated the sentiment analysis on the reply tweets using the Python Library Flair (We experimented with three Python Libraries:- Flair, TextBlob and NLTK , and found that Flair gives the best sentiment score among these).We then calculated Average Sentiment, Maximum Sentiment Score among the replies on that POI Tweet , Minimum Sentiment Score among the replies on that POI Tweet and Replies with the maximum and minimum sentiment for a POI Tweet respectively.This analysis clearly showed what were the opinions and agreement of the general public towards a tweet by a POI. We also calculated the Sentiment Score on news tweets by a news provider POI showing the sentiment of the news . After Sentiment Analysis we also did Topic Modelling using Latent Dirichlet Allocation (LDA) in order to organize, understand and summarize large collections of information from the tweets

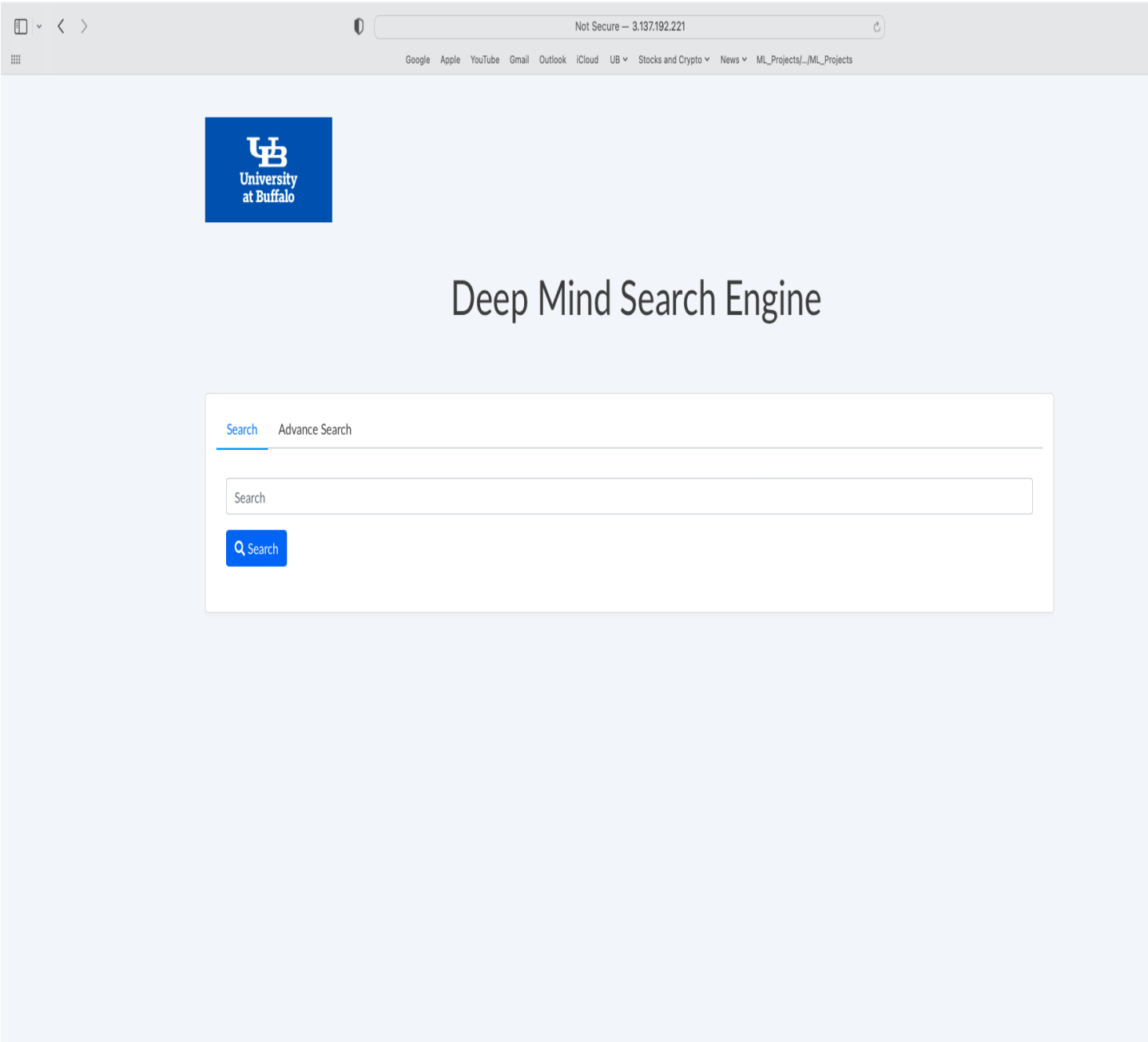
1.3 Indexing the Data

After we have completely analyzed the data, now was the time to index the data using Apache Solr. For indexing the data we have used a Python Library named PySolr. We took the Best-Matching25 (BM-25) Model for Indexing the data.We also tuned in the hyperparameters b and k1 and found the best hyperparameters that were b=0.5 and k1=1.2. During indexing the data we were indexing the document line by line since while indexing the large corpuses of data(>=30k), the Solr wasn’t accepting this huge data at one time , so we indexed it looping in tweet by tweet .

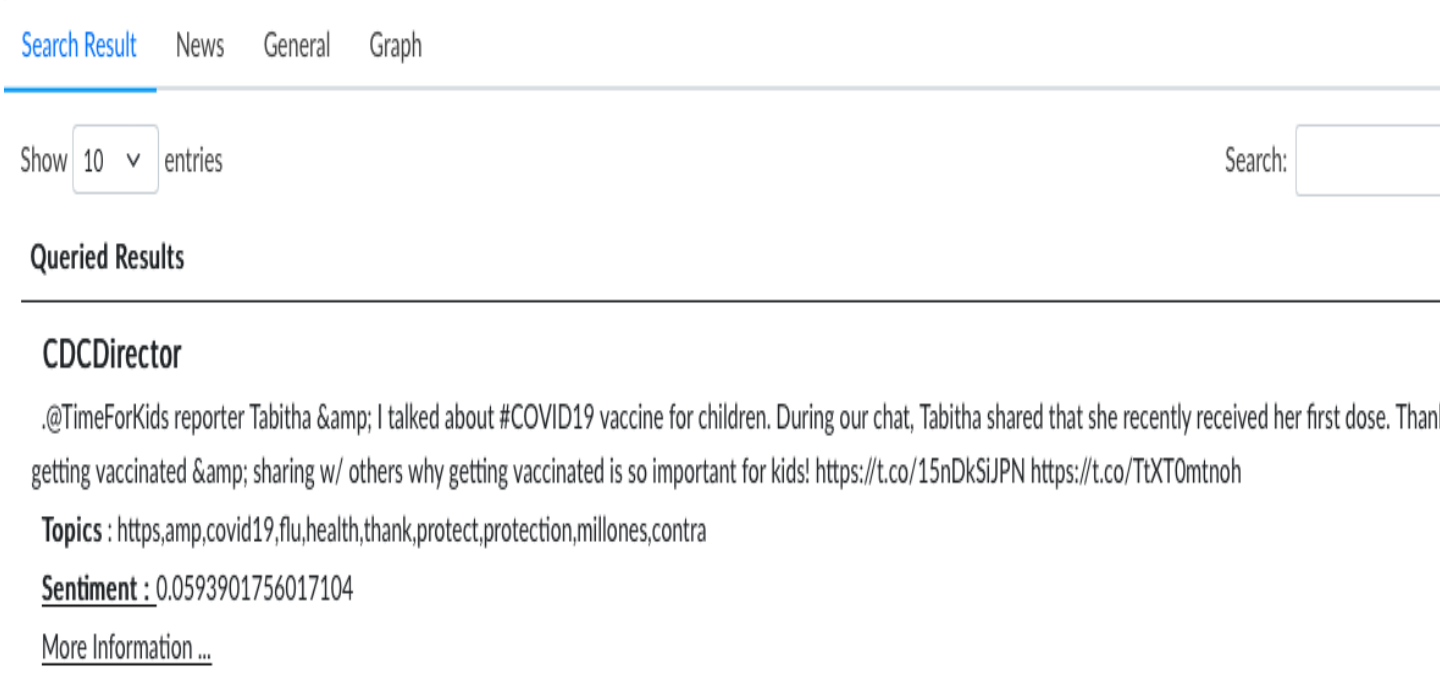
SAMPLE SCREENSHOTS:

UI:

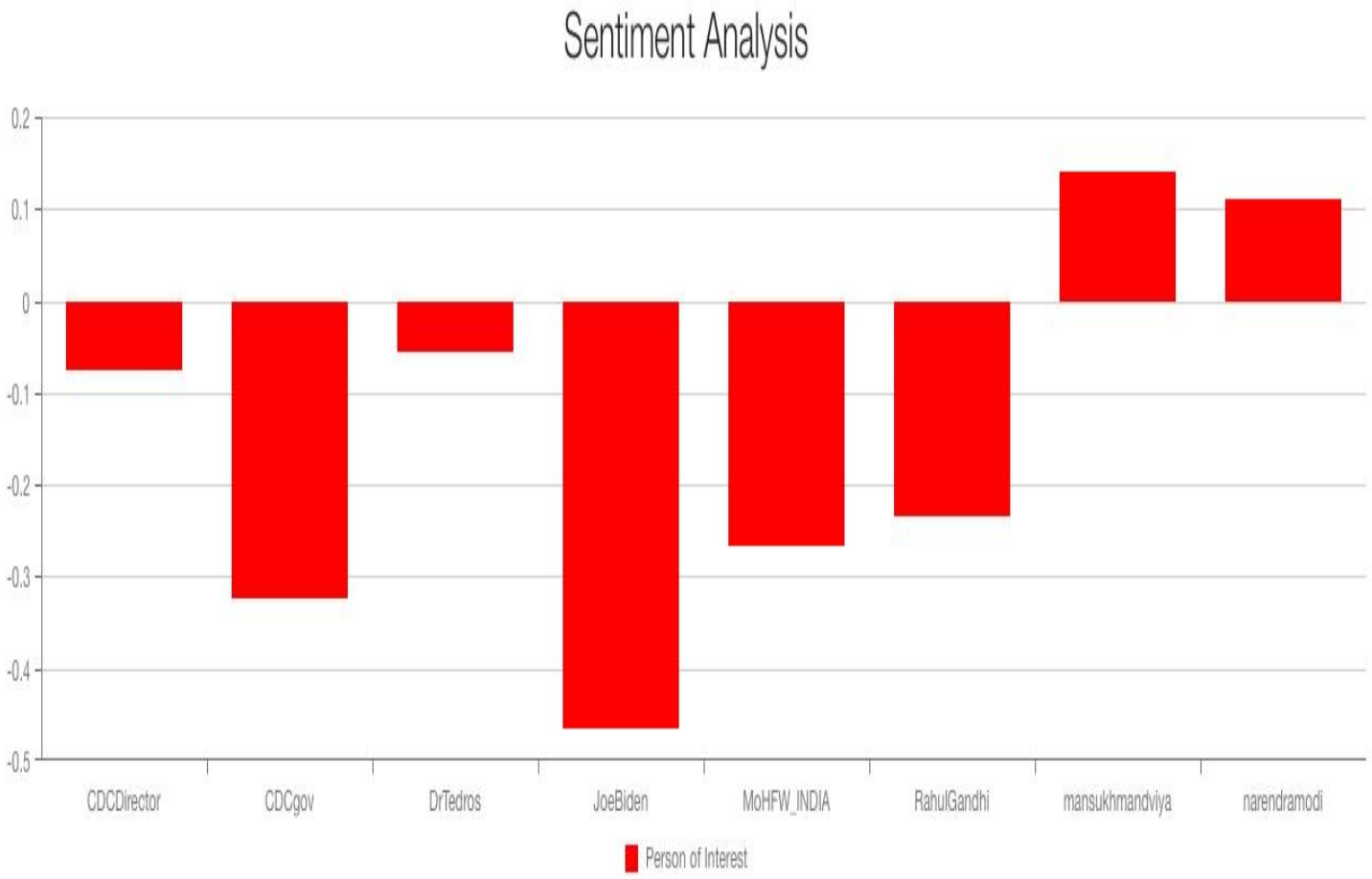
1. Homepage



2. A tweet div after Searching for the keyword “covid”



3. Sentiment Analysis graph for the keyword “covid”:


Sentiment Analysis
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
CDCDirector
CDCgov
DrTedros
JoeBiden
MoHFW_INDIA
RahulGandhi
mansukhmandviya
narendramodi
Person of Interest

BACK END :

1. Search Function ()

```
50
51  @app.route('/search', methods=["POST", "GET"])
52  def search():
53      pp = Preprocessor()
54      global QUERY_RESULT
55      global SEARCH_TYPE
56      global SELECTED_LANGUAGE
57      global POI
58
59      QUERY_RESULT = pd.DataFrame()
60      SEARCH_TYPE = 0
61      SELECTED_LANGUAGE = ''
62      POI = ''
63      input_data = json.loads(flreq.data.decode())
64      SEARCH_TYPE = input_data['val']
65      if input_data['val'] == 1:
66          user_query = input_data['basic']
67      else:
68          user_query = input_data['adsearch']
69      solr_query = pp.get_query(user_query)
70      print(solr_query)
71      solr_search(solr_query)
72      result = QUERY_RESULT
73      result = result[result['type'] == 'r']
74      if input_data['val'] == 2:
75          if input_data['languageselected']['name'] != 'Choose..':
76              result = result[result['tweet_lang'] == LANG[input_data['languageselected']['name']]]
77              SELECTED_LANGUAGE = LANG[input_data['languageselected']['name']]
78              print(SELECTED_LANGUAGE)
79          if input_data['poiselected']['name'] != 'Choose..':
80              result = result[result['poi_name'] == input_data['poiselected']['name']]
81              POI = input_data['poiselected']['name']
82              print(POI)
83      return_json = {}
84      temp = []
85      for i in range(len(result)):
86          x = pd.DataFrame.to_json(result.iloc[i])
87          temp.append(json.loads(x))
88      return_json['queryresult'] = temp
89      return jsonify(return_json)
90
```

2. getchartResult()

```
def solr_search(query):
    global QUERY_RESULT
    inurl = 'http://' + EC2_IP + ':8983/solr/' + CORE_NAME + '/select?q='
    inurl += 'tweet_text:*' + query + '*'
    field = '*'
    inurl += '&fl=' + field + '&wt=json&indent=true&rows=1000'
    print(inurl)
    data = urllib.request.urlopen(inurl)
    docs = json.load(data)['response']['docs']
    QUERY_RESULT = pd.json_normalize(docs)
    QUERY_RESULT.drop(['id','text_en','text_es','text_hi','poi_id','_version_'],axis=1, inplace=True)
```

3. Hitting solr

```
125
126 @app.route('/api/get-chart-result')
127 def chart1():
128
129     global QUERY_RESULT
130     global SEARCH_TYPE
131     global POI
132     global SELECTED_LANGUAGE
133
134     result = QUERY_RESULT
135     poi = result[result['type'] == 'r']
136     print(poi)
137     if POI != '':
138         poi = poi[poi['poi_name'] == POI]
139     if SELECTED_LANGUAGE != '':
140         poi = poi[poi['tweet_lang'] == SELECTED_LANGUAGE]
141     li = []
142     for i in poi['sentiment']:
143         li.append(round(float(i), 5))
144     poi = poi.assign(sentiment=li)
145     ind = poi.groupby('poi_name').describe().index
146     print(ind)
147     graph_data = []
148     for i in ind:
149         data = {}
150         temp = poi[poi['poi_name'] == i]
151         data['poi_name'] = i
152         data['count'] = temp['sentiment'].mean()
153         graph_data.append(data)
154     print(graph_data)
155     return jsonify(graph_data)
156
```

WORK BREAKDOWN BY TEAMMATES:

Tanush: Performed Data Collection, indexing, doing sentiment analysis and topic modeling.

Sarveshwar: Developed the Controller part in python where we're getting data from solr in realtime. Developed multiple APIs for search tab, news tab, general tab and multiple graphs. Worked on connecting the frontend with Controller..

Oviyaa: Used angular js to make the entire website dynamic.

Aravind: Developed the website using html and css and hosted the website using flask on aws.

CONCLUSION:

We realized how difficult it is to Create & Maintain Google. The Internet is an ocean of information and how fastly Google brings up millions of results for millions of users across the globe.