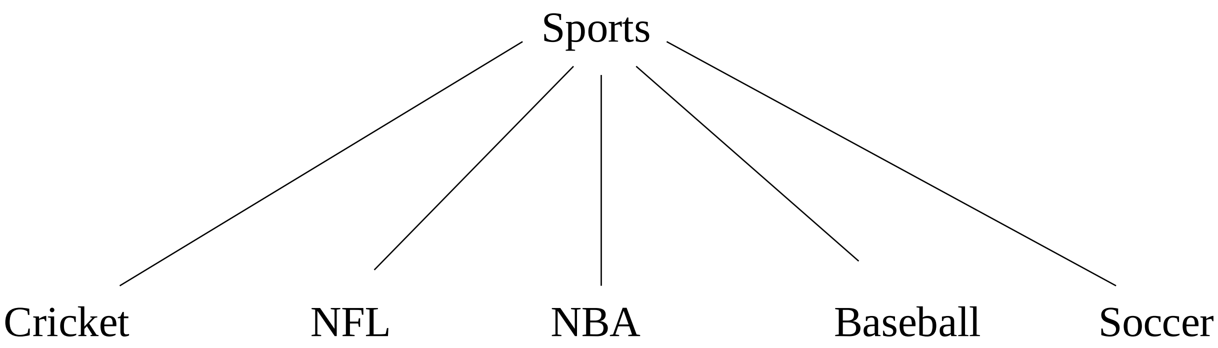
Data Aggregation, Big Data Analysis and Visualization
Project 2 – Report
CSE 587

Submitted by – Yash Chandra and Mehul Awasthi

The Topic

The main topic we chose for this project is ‘sports’. Our five sub-topics are ‘soccer’, ‘cricket’, ‘NFL’, ‘NBA’ and ‘baseball’.



Part 1 – Prototype Data Collection

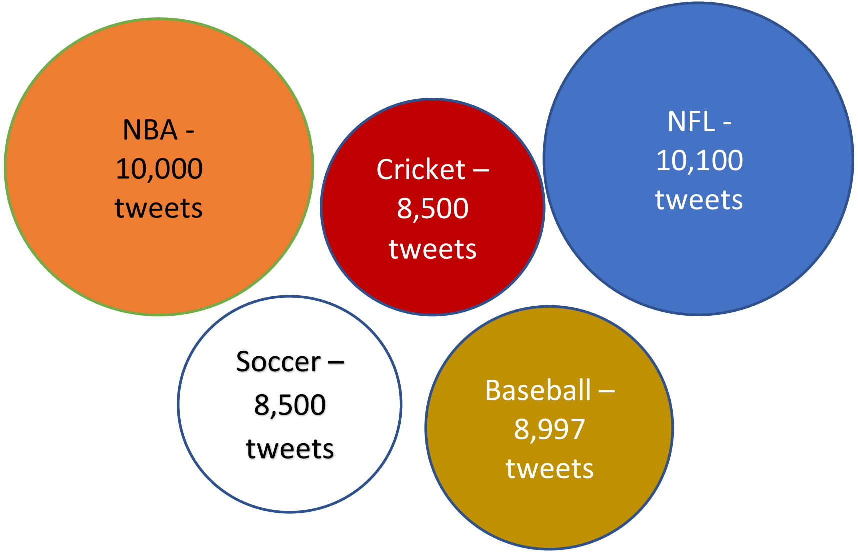
Twitter Data Collection

The python library Tweepy was used to collect tweets relevant to our topics. The python code was really simple and straightforward for carrying out this process.

- First the OAuthHandler is set up by providing the different keys that are necessary to communicate with the Twitter API.

- Next the relevant tweets are parsed using tweepy.Cursor() and stored in a Pandas data-frame. The query is passed to tweepy.Cursor() (which could be any one of the sub-topics such as ‘cricket’ or ‘NFL’). The data-frames are then saved in the form of .CSV files. A reasonable number of tweets are collected each time varying between 500-1000. Retweets are NOT collected in this process.
- Once a reasonable number of tweets per subtopic has been collected, the multiple .CSV files are merged by first reading the files into data-frames and concatenating the multiple data-frames into one main data-frame. The data-frame is then rendered free of any duplicate tweets. This merged data-frame is then written to a .CSV file. This process is carried out for each different sub-topic.

Unique tweets collected per subtopic –



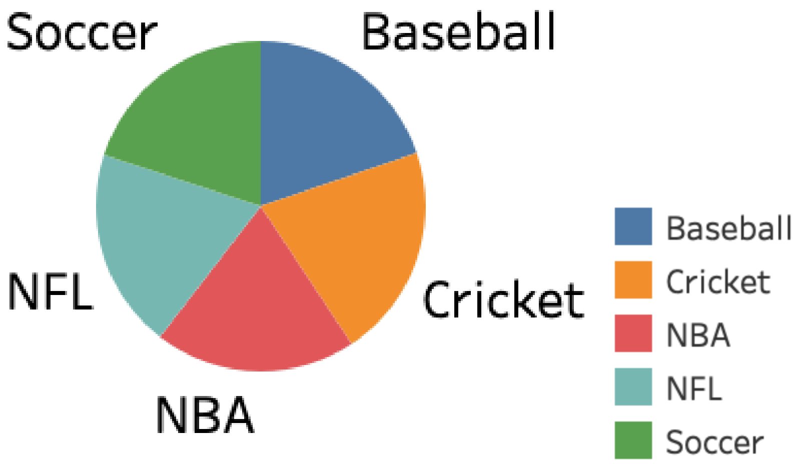
Total unique tweets collected – 46,097

New York Times Data Collection

- The NY Times Article Search API is used to gather urls for each subtopic, namely NBA, Soccer, NFL, Baseball, and Cricket. The articles are filtered to include only 2019 articles using the publish_date field on the article and unique ones. All the urls are then written to a file, one file for each sub topic.
- Python library ‘requests’ is used to fetch each of the articles from the urls. Python Library BeautifulSoup is used to obtain only articles by doing a search on all p tags. The article is filtered to remove advertisement texts.

Number of articles per subtopic –

Sub topic	Number of Unique Articles
Soccer	102
NBA	101
NFL	100
Baseball	101
Cricket	107



Common Crawl Data Collection

Common Crawl

- The latest indices are obtained from the Common Crawl website for 2019 data.
- Common Crawl Index API is used to find crawl data for user defined URLs using the url field option in the Index API. For this purpose, multiple URLs are used namely: espn.com, nba.com, cricbuzz.com, nfl.com, cbssports.com, skysports.com
- The JSON from above links provides details about each crawl data. It has the URL field which was used to filter out relevant data.

- Relevant article links were found by identifying url pattern for articles for each of the domain/sub-topic. This made sure only article text was included in the output data. The links were filtered to include unique ones.
- For each of url found in the above step, the anchor tags from the html file corresponding to the url is obtained to include more articles finding the relevant article pattern.
- The articles are fetched using python library requests. Python library BeautifulSoup is used to obtain only articles by doing a search on all p tags.

Sub topic	Number of Unique Articles
Soccer	205
NBA	149
NFL	149
Baseball	129
Cricket	139

Stemming and cleaning the data

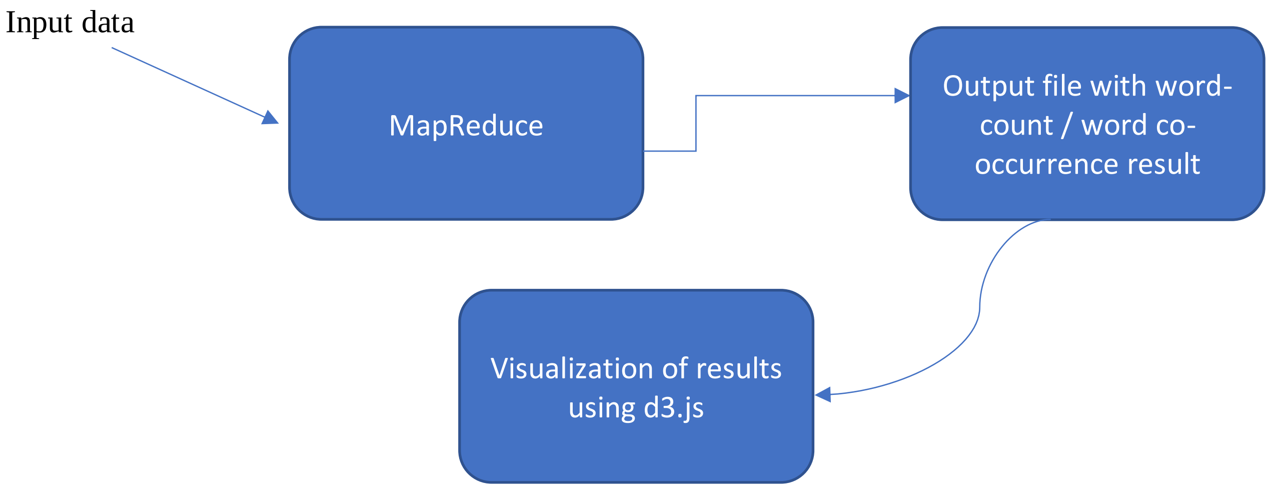
- The Python library NLTK is used to perform data cleaning.
- The articles are tokenized into words. The words are converted to lowercase to avoid redundancy.
- The special characters from the words are removed. This makes sure the words are always alpha-numeric.
- The words are then stemmed and lemmatized using the NLTK library.
- The words are further cleaned to remove links and stop words. For this purpose, NLTK provides a list of stop words. Apart from that, the list was appended with additional 1527 stop words.

Part 2 – Setting up the Big Data infrastructure

The Hadoop infrastructure was set up for this project with the help of ‘Docker for desktop’ designed for Macintosh. The Hadoop infrastructure was used for storing out ‘big data’ files namely the data from Twitter, Common Crawl and New York Times. The mapper and reducer were run on Docker as well, once the infrastructure was set up.

- After configuring the docker directory and mapping to the local workspace the input locations in the HDFS workspace were created using simple commands.
- The commands were as follows –
    - `$ hadoop fs -mkdir /user/mehul`
    - `$ hadoop fs -mkdir /user/mehul/MR`
    - `$ hadoop fs -mkdir /user/mehul/MR/input`
- Next the text files of the data from each of the three sources were copied from the shared folder /src/data to the HDFS shared folder. The three files were data_all_cc.txt, data_all_nyt.txt, data_all_tweets_new.txt.

- The MapReduce program was then run on each file separately and the output of each file was then stored separately too. These output files contained the wordcount of all the words present in the separate files.


Input data
MapReduce
Output file with word-count / word co-occurrence result
Visualization of results using d3.js

Various processes being carried out

Part 3 – Analyzing the data, data visualization and Webapp deployment

Data Visualization and Webapp :

- Angular was used to build the webapp with D3.js.
- The output from reducer is further processed with a python script to provide the appropriate format for d3.js input.
- The appropriate format or input to D3.js :

```
 [
 { ‘text’ : ‘game’, ‘size’ : 20 },
{ ‘text’ : ‘score, ‘size’ : 15 },
{ ‘text’ : ‘points, ‘size’ : 12 }
]
```

- D3 js word cloud builds the word cloud using the above data and frequency as the size.
- Further customizations were made to the D3.js api to show larger font for more occurring words and smaller font for less occurring words.
- Customizations were made to display two different kind of word clouds for Word Count and Word Co-occurrence.
- The angular web app is bundled in a single folder and deployed to HEROKU at https://dicpro-ychandra-mehulawa.herokuapp.com/. This is to make sure the application runs on the Instructors’ end. If the Instructor would like to run the code on the local machine, Node.js must be installed. Navigate into the ‘Webpage/dic_angular_project/’

directory and inside terminal, run ‘npm install @angular/cli –g’ and then run command npm install. Next, run command ‘ng serve’ and the web app will run on localhost:4200.

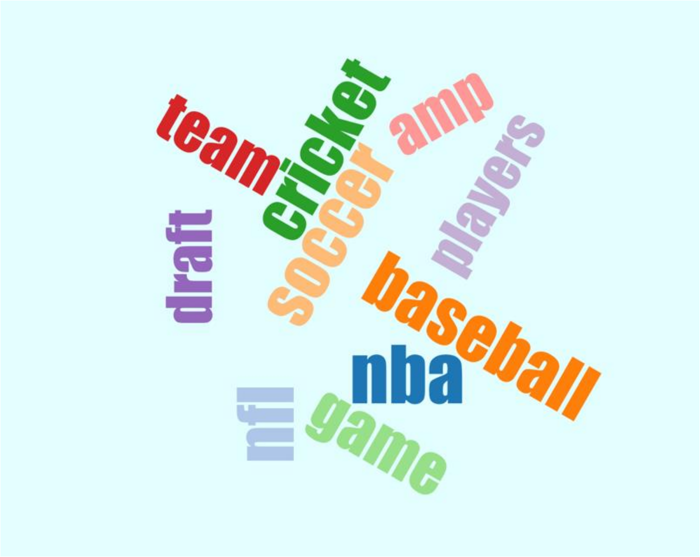
- All the D3.js code is bundled into a single angular folder. In order to look at the angular code specific to the visualization, navigate to dashboard folder within the ‘src/app/views’. The component code for the word cloud visualization is found in ‘src/app/views/dashboard/word-cloud/word-cloud.component.ts
- For Angular web app theme and styles, CoreUI template is used.

Word Cloud Output

The word cloud output images for each dataset are as follows -

Twitter

Word Count- Twitter



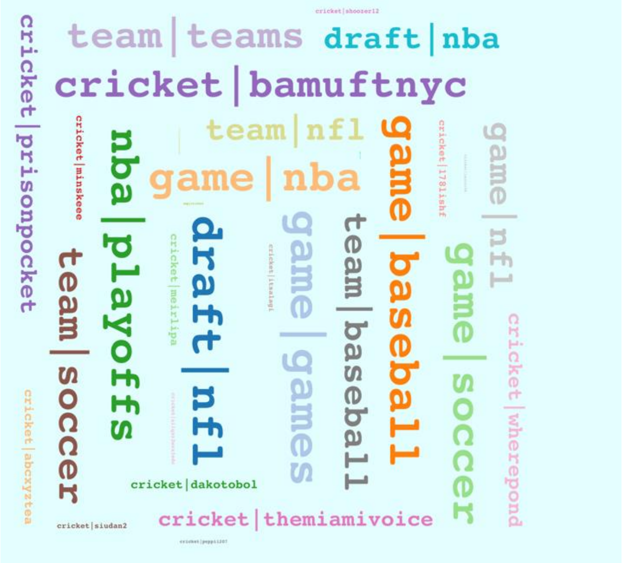
The word count word cloud for the data from twitter is interesting. The query words used to actually parse the tweets and find them are present in the word cloud. It’s of interest to note that since the NFL draft is going to take place in the near future, people on Twitter have already started talking about. This is evident from our word cloud visualization!

The top ten occurring words in our Twitter data were-

1. Draft
2. NBA
3. Game
4. Cricket
5. Amp
6. Soccer

7. Team
8. Baseball
9. NFL
10. Players

Word Co-occurrence – Twitter



The word co-occurrence for Twitter provides some wonderful insights. The NBA playoffs are currently going on and as you can see from the word cloud, the two words NBA and playoffs are two words that occur the most often.

New York Times

Word Count- Twitter

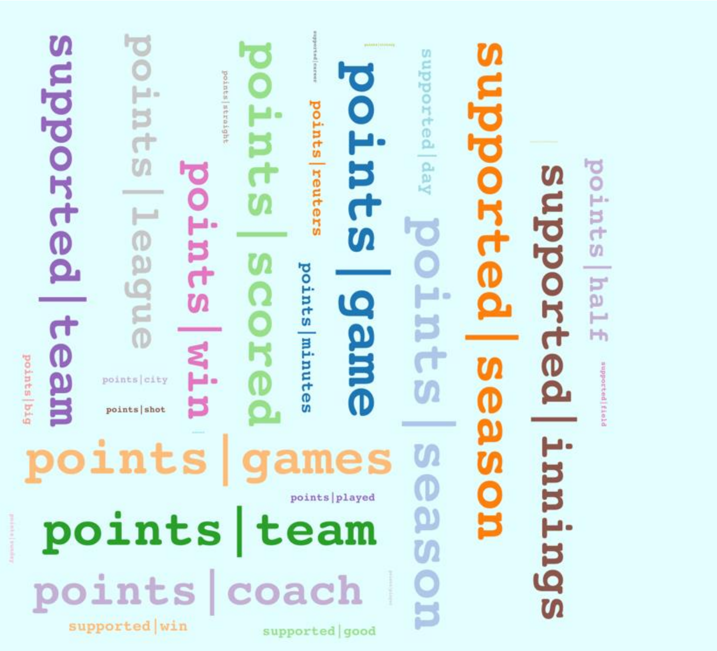
supported
team
game
points
season
time
games
league
win
scored

The data from New York Times is from professionally written articles. Therefore it's reasonable to assume that some top words will be different from the top words of twitter. And that does seem to be the case as you can see from the word cloud. The word game is a common top word in both and that does make sense.

The top ten words in the New York Times data are –

1. League
2. Season
3. Games
4. Team
5. Scored
6. Points
7. Game
8. Win
9. Time
10. Supported

Word Co-occurrence – New York Times



The co-occurrence word cloud of the NYT data makes so much sense. For example, look at ‘points’ and ‘league’. This pair is one of the highest occurring pairs in the data. It only makes sense because a league’s is based on points! Supported and team also occur together frequently. That is again a very sensible occurrence. This word cloud can therefore be branded an extremely intuitive word cloud.

Common Crawl

Word Count- Common Crawl

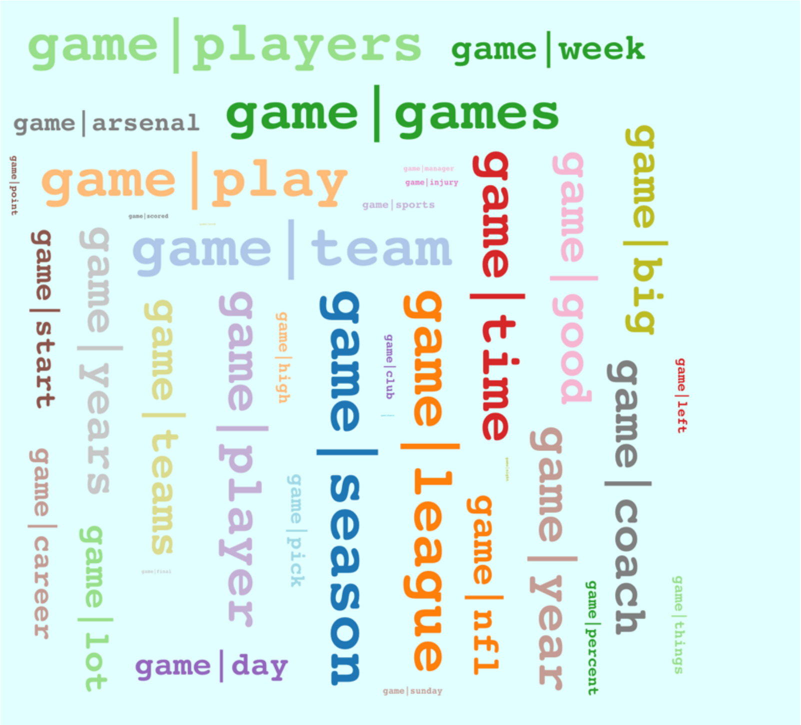
play
league
players
team
time
game
games
season
points
baseball

The Common Crawl top words are very similar to the NYT top words. That proves to be a curious proposition. The word game occurs here as well as a top word and it goes to show that this is one of the most commonly associated word with any sport. While that may only be obvious, it also means that the world cloud being visualized as a result of the data we gave the MapReducer, is an acute representation of the content present on our topic and various sub-topics.

The top ten words in the Common Crawl data are as follows –

1. Play
2. League
3. Time
4. Games
5. Season
6. Game
7. Team
8. Points
9. Baseball
10. Players

Word Co-occurrence – Common Crawl



The word ‘game’ appears to be everywhere for the Common Crawl data’s co-occurrence. All the pairs of co-occurrence are extremely relevant and reflect the occurrence of the word in content on the web.

References –

[1] https://www.bellingcat.com/resources/2015/08/13/using-python-to-mine-common-crawl/
[2] https://gist.github.com/sebleier/554280
[3] https://github.com/coreui/coreui-free-angular-admin-template