CSE-587: DATA INTENSIVE COMPUTING
LAB-2
ANUNAY RAO – 50291493
APOORVA BISERIA -50291145

In this Lab, we needed to choose a topic and its 5 sub-topics. We chose the main topic as "SPORTS" and sub topics as FOOTBALL, BASKETBALL, GOLF, TENNIS AND BASEBALL.

DESCRIPTION AND METHODOLOGY:

a) The first step we needed to do was to collect 20K tweets from twitter, and 500 articles from New York Times and Common Crawl for the year 2019.

1. For collecting data from twitter (TwitterFetchData.ipynb), we fetched the API keys from the twitter developer account, wrote the code in python such that it eliminates the duplicate tweets (Retweets) and collect tweets that are of this year only. We collected 4000 unique tweets from each sub topic.
2. For collecting articles from New York Times (NYTFetchData.ipynb), we created an account in https://developer.nytimes.com and generated API for article search. Using the API we collected 100 articles on each subtopic.
3. For collecting articles from Common Crawl (CommonCrawl.ipynb) we did index search on Common Crawl Index of 2019 namely [2019-04, 2019-09 and 2019-13] for the domains restricted through news API and other sources. We applied filter to the URL so that the URL must contain our subtopic keyword to have more relevant data. For each record returned, we could extract the data in two ways one is to get the offset from the record and get the data from AWS Public Dataset of Common Crawl and the other is to get the URL and then use request method to get the data. One problem with the former method is you might find lot of segments which do not return the data or they are not reachable, plus the process is bit more time consuming as compared to the latter.

b) After collecting data to continue the process, we created the set up for Hadoop on the Docker by following the Cloudera Docker Hadoop Document provided by the Professor Dr. Ramamurthy.

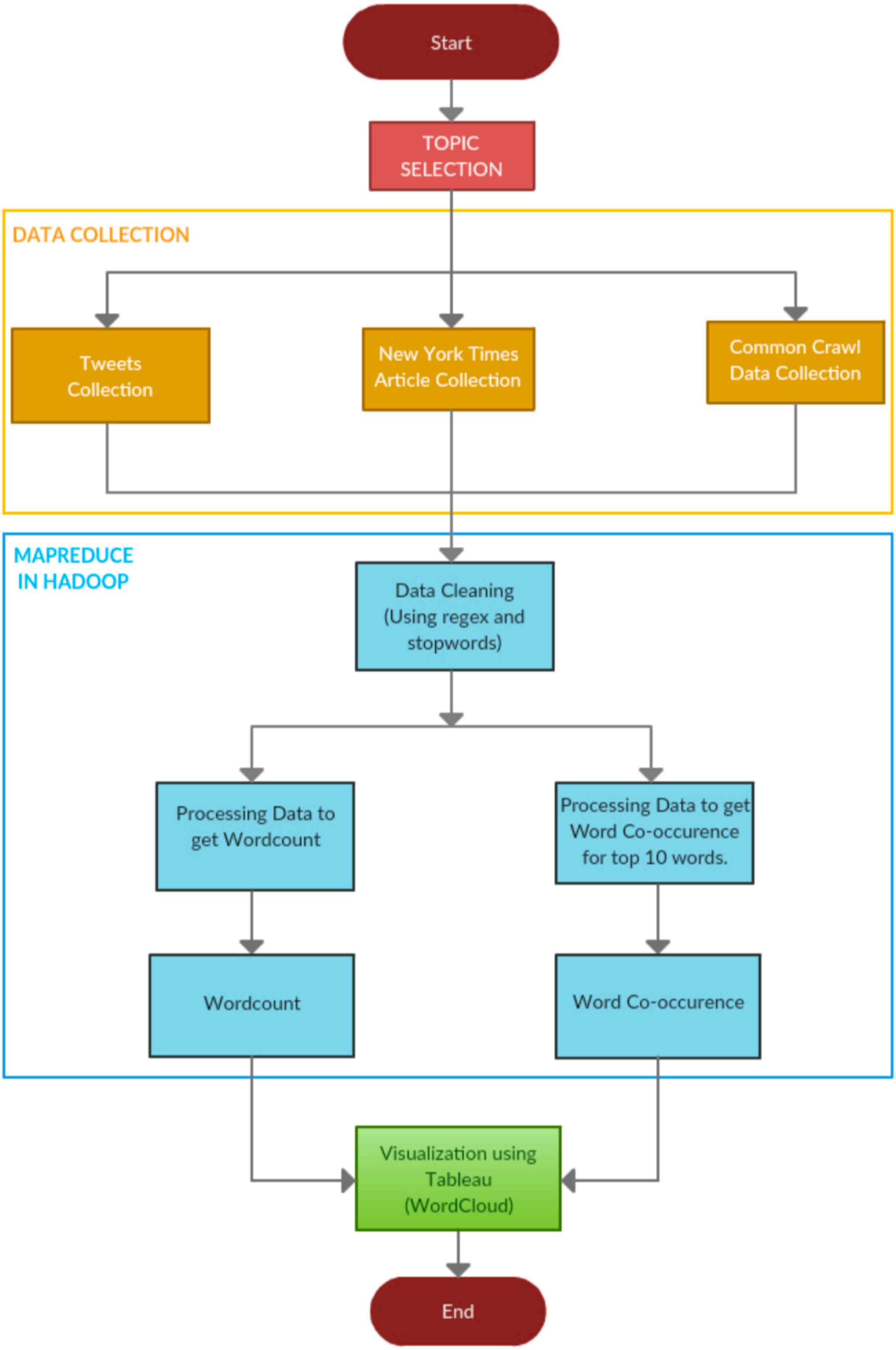
c) For further steps, we designed and developed the code for Mapper (mapper.py) as well as Reducer(reducer.py). In the Mapper phase we removed all the stop words and did all the necessary processing of data using regex in order to get better results.

d) After running the Map Reduce Job we got the output. Then for the top 10 words from word count, we processed the word co-occurrence Mapper(mapper1.py) and Reducer(reducer.py).

e) We created an external script for converting output file to CSV (texttocsv.py) and then sorting the csv file (sort.py) for word count and word co-occurrence so that the top 10 can be used further for visualization.

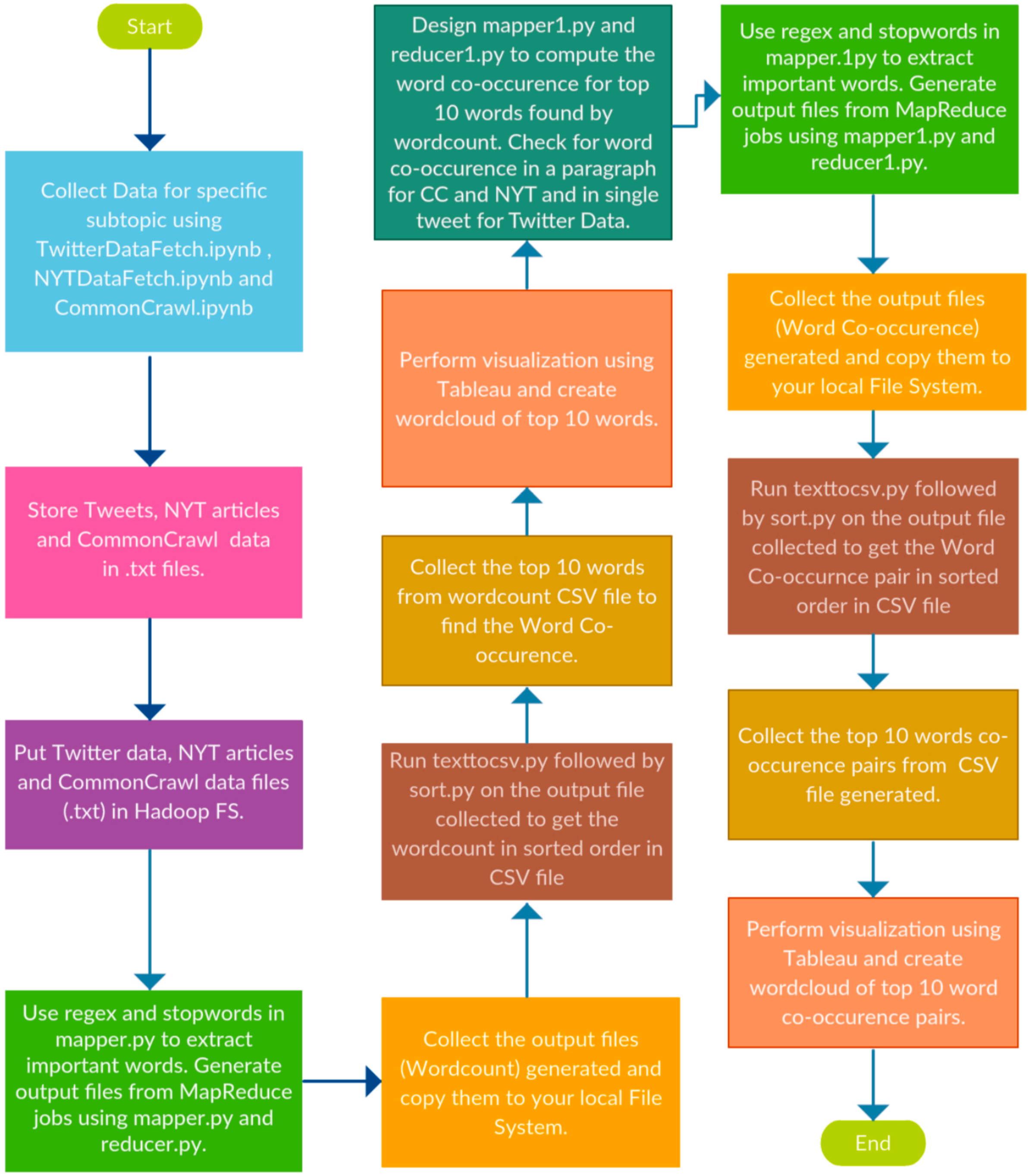
f) For visualization of the top 10 words and pairs we needed to create word clouds for each type of data collected. So, we used Tableau, you can create an account on tableau for a trial period and can perform visualization using drag and drop. After creating the workbook of all 6 word clouds, we published our workbook on tableau public, which can also be done using creating an account at Tableau public. Navigating through the entire web page, all six word clouds can be seen by clicking on 6 tabs.

BASIC PROCESS FLOW:



Start
TOPIC SELECTION
DATA COLLECTION
Tweets Collection
New York Times Article Collection
Common Crawl Data Collection
MAPREDUCE IN HADOOP
Data Cleaning (Using regex and stopwords)
Processing Data to get Wordcount
Processing Data to get Word Co-occurence for top 10 words.
Wordcount
Word Co-occurence
Visualization using Tableau (WordCloud)
End

DETAILED PROCESS FLOW:



Start

Collect Data for specific subtopic using TwitterDataFetch.ipynb , NYTDataFetch.ipynb and CommonCrawl.ipynb

Store Tweets, NYT articles and CommonCrawl data in .txt files.

Put Twitter data, NYT articles and CommonCrawl data files (.txt) in Hadoop FS.

Use regex and stopwords in mapper.py to extract important words. Generate output files from MapReduce jobs using mapper.py and reducer.py.

Collect the output files (Wordcount) generated and copy them to your local File System.

Run texttocsv.py followed by sort.py on the output file collected to get the wordcount in sorted order in CSV file

Collect the top 10 words from wordcount CSV file to find the Word Co-occurence.

Perform visualization using Tableau and create wordcloud of top 10 words.

Design mapper1.py and reducer1.py to compute the word co-occurence for top 10 words found by wordcount. Check for word co-occurence in a paragraph for CC and NYT and in single tweet for Twitter Data.

Use regex and stopwords in mapper.1py to extract important words. Generate output files from MapReduce jobs using mapper1.py and reducer1.py.

Collect the output files (Word Co-occurence) generated and copy them to your local File System.

Run texttocsv.py followed by sort.py on the output file collected to get the Word Co-occurnce pair in sorted order in CSV file

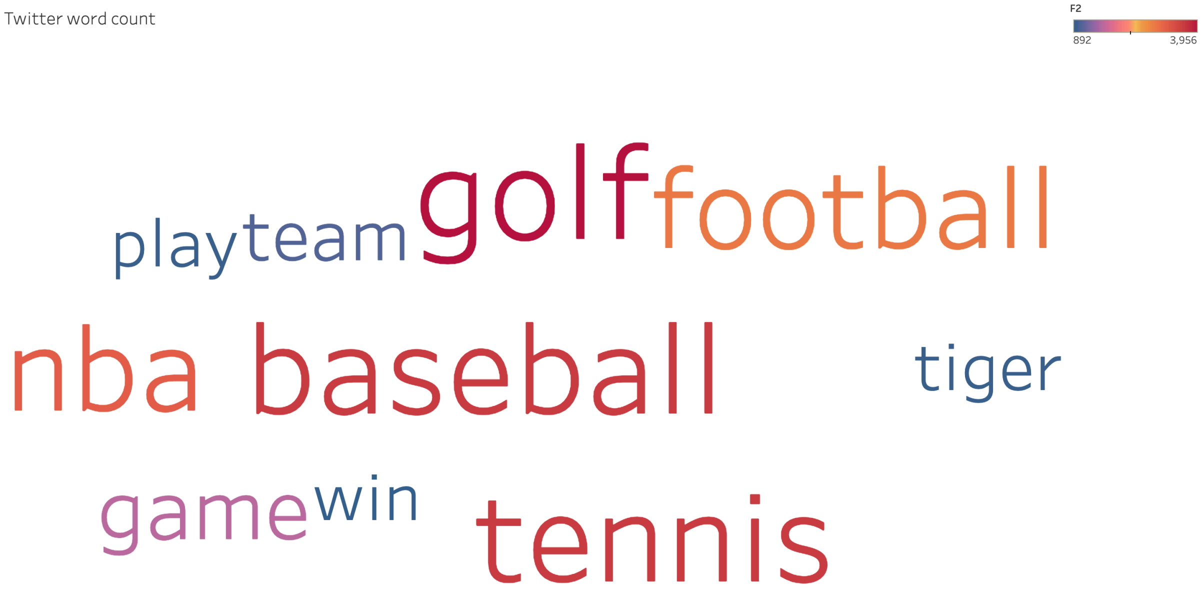
Collect the top 10 words co-occurence pairs from CSV file generated.

Perform visualization using Tableau and create wordcloud of top 10 word co-occurence pairs.

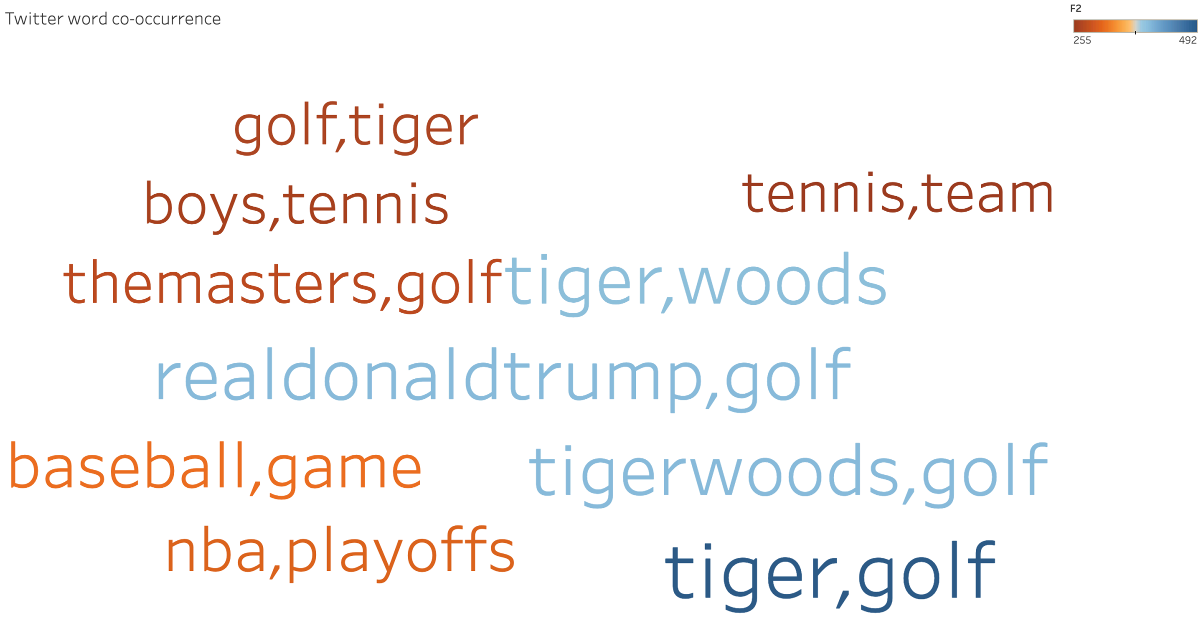
End

VISUALISATION RESULT:

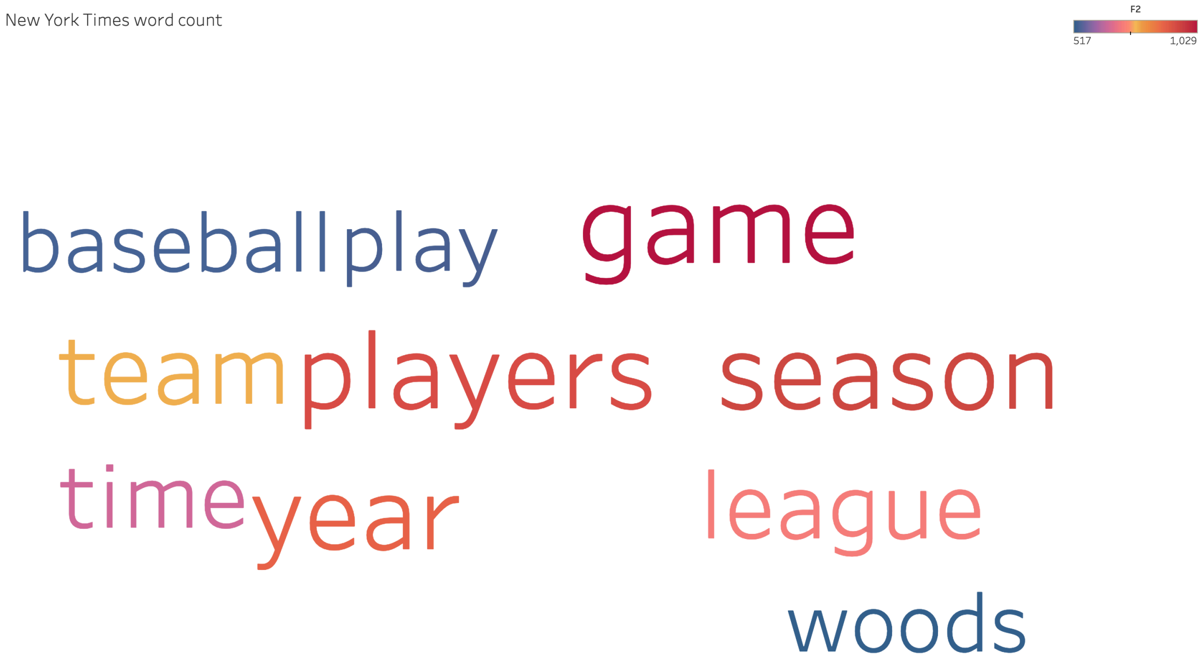
1. Word Cloud for Word Count Twitter Data:



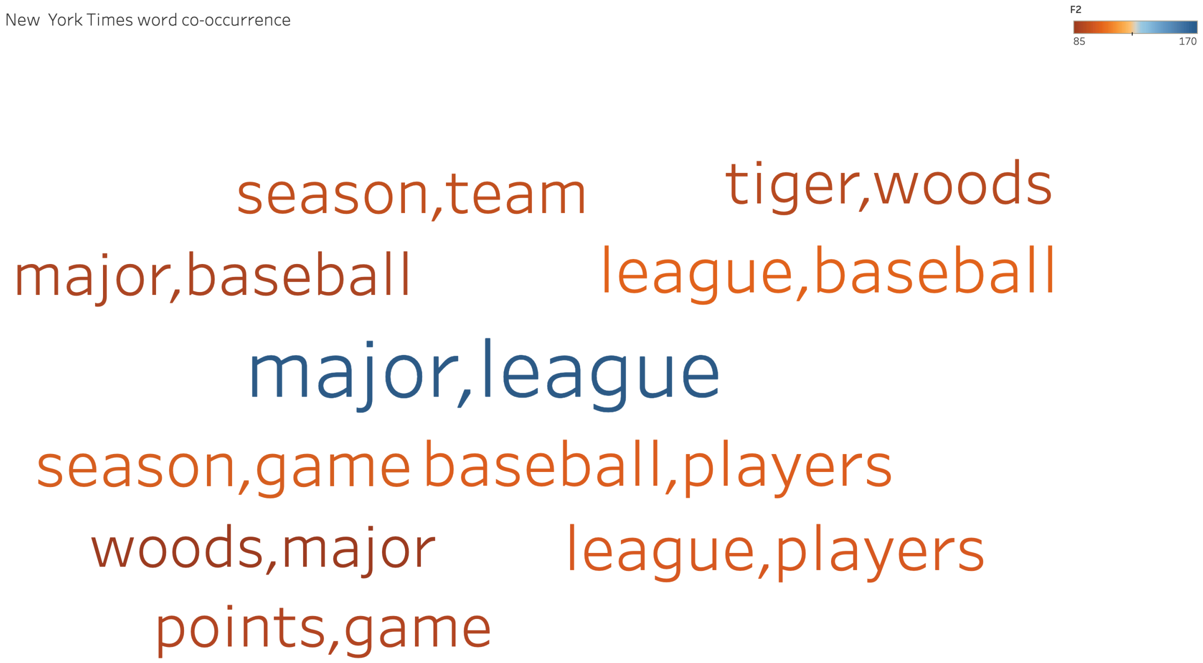
2. Word Cloud for Word Co-occurrence Twitter Data:



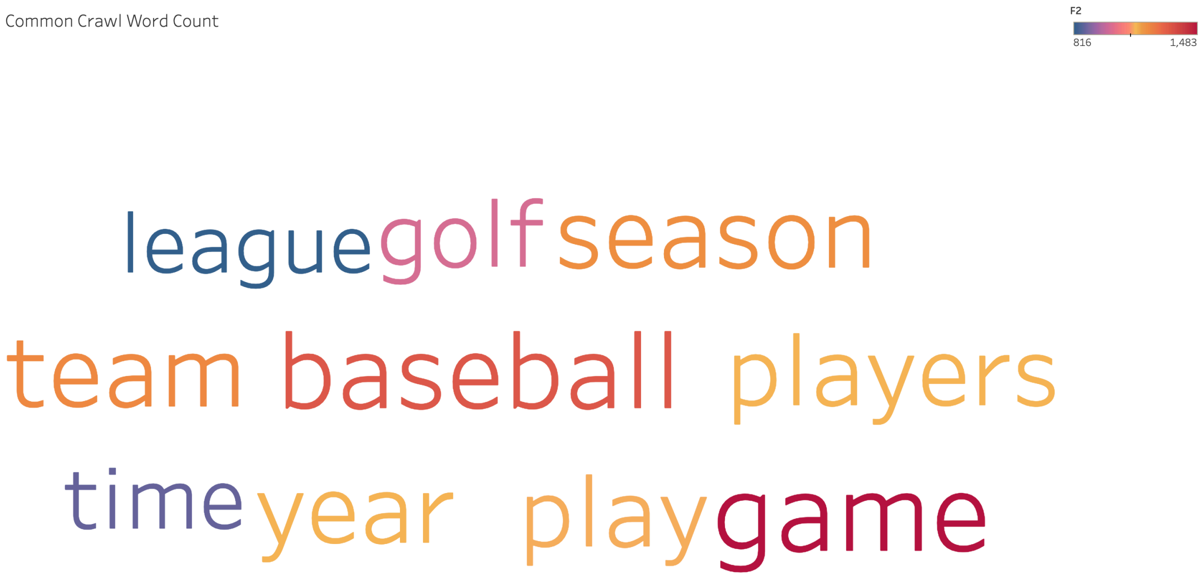
3. Word Cloud for Word Count NYT Data:



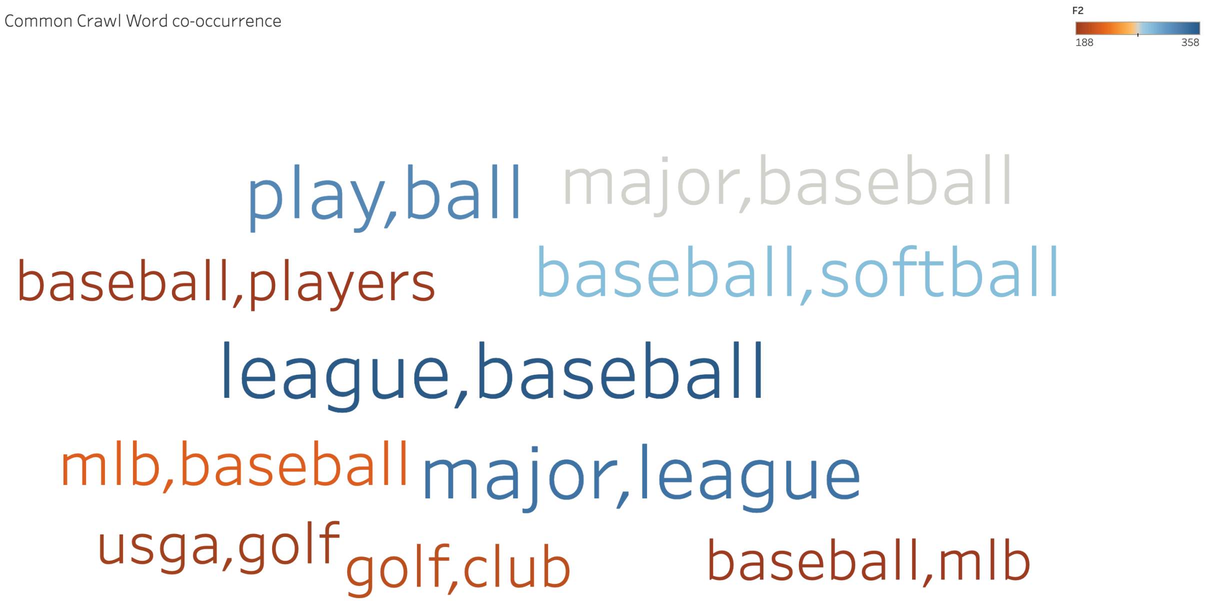
4. Word Cloud for Word Co-occurrence NYT Data:



5. Word Cloud for Word Count Common Crawl Data:



6. Word Cloud for Word Co-occurrence Common Crawl Data:



All the visualization results are available on Tableau Public at:
https://public.tableau.com/profile/apoorva.biseria#!/vizhome/wordcloudsLab2DIC/CommonCrawlWordCount?publish=yes