CSE 587

virinchi urivinti
ubit: 50292192

srujan sheshank reddy
ubit:50291486

Introduction:

The objective of the project is to collect the data from different sources using their api and explain the relatability of different data sources. We have applied different Mapreduce algorithmm like word count and word cooccurence have been implemented. The outputs generated have been visualized using Tableau visualisation tool and results have been hosted on online website.

Data collection and prepocessing:

We have collected data from the three sources twitter, common crawl, Nytimes. From the twitter we have collected 20000 tweets and from twitter and 500 articles from NY Times and 500 articles from common crawl. ArtcileApi of nyt is used to extract data from nyt.The warc files from commoncrawl are used to extract data.we remove stop words to make the data meaningful. Stop words are natural language words which have very little meaning, such as "and", "the", "a", "an" etc. Wordnet lemmatizer is also used to make the data meaningful.

Twitter data :
To gather the tweets on related topic, we used twitteR API of Twitter. we used R Script to achieve this. The Script makes use of API Key and return tweets for given ‘search query’ and ‘date range’. we have used following hashtags/search strings to collected tweets from united states .#baseball , #basketball, #football, #soccer, #golf.The script uses the tweets screen nane to retrieve the user location and stores all the information about ttweets into a data frame. Then this data frame is used to extract text data of the tweet and this data is stored

NY-times data:
To gather the articles , we used the API provided NY times .we used python for achieving this. Using the API key,search keywords and date. We gathered the URL of the articles and store it . The response of the API call is in JSON format. We parsed the response to the information. The urls are then scraped to extract useful information. The extraction of data from these urls is doing using the beautiful soup library present in python. Beautiful soup is used to extract data from the json files of the paragraph tags.

Word Count using the MapReduce Framework

MapReduce is a framework using which I can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner. I use the MapReduce framework to count the number of times a word occurred in the tweets or articles. I

do this to find the most frequent words which capture the essence of the topic. Mapper and Reducer are implemented as follows:

Mapper:

The Mapper them emits (outputs) a key value pair for each word in the article or tweet. The key in this case is the word and value is 1. we will later, in Reducer, aggregate these 1 for every key (unique word) to find the count of that word.

Reducer: Here we find the actual counts. Output of the mapper is fed as an input to the Reducer. The reducer collects <key,value> pairs which have the same key. The<key,value> pair are of the type <word,1>.It then sums over all the values received for this key.This generates the count for that key. The reducer then emits this value

Co-occurrence using the Map Reduce Framework

we find out the co-occurring words in each article/ tweet. The aim is to find the pair of words which occur together most frequently. For this we use the top ten most frequently occurring words captured after running the Word Count on tweets and articles. The context for co-occurrence is chosen as a tweet or an article in a paragraph. The approach followed is similar to the one used in word count using Map Reduce. The Mapper and Reducer are implemented as follows. □

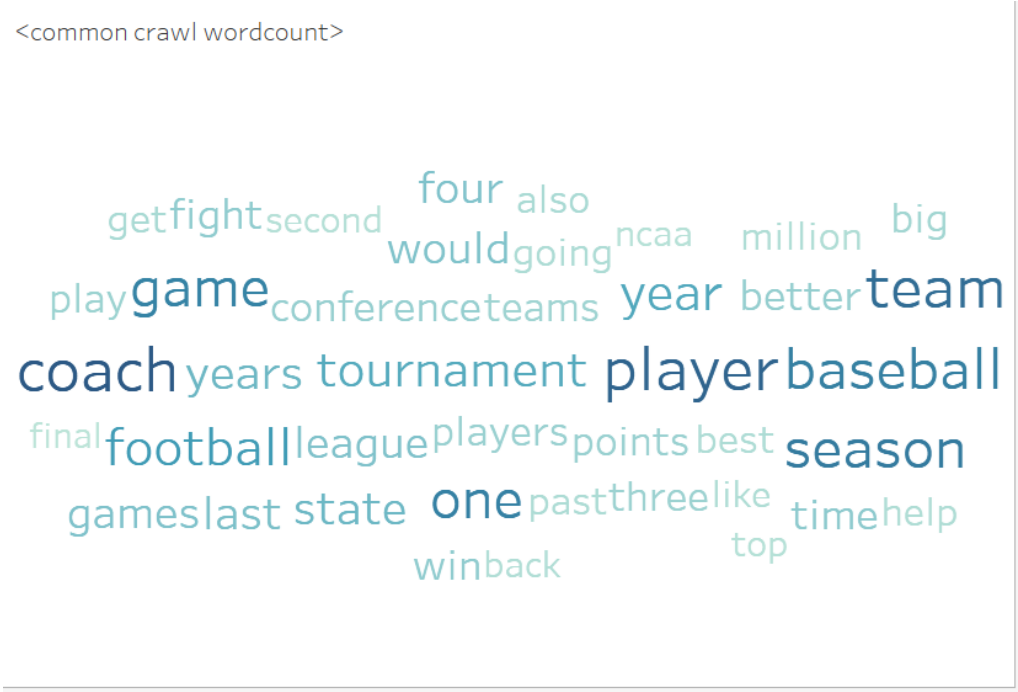
Mapper: The first step of the mapper is to clean the data (just like in Word Count). We did this by removing the stop words and other common words which do not reflect the data. We then selected pairs of words from the tweet/article, such that at least one of those words lies in the top ten words which we have identified. This word-pair, consisting of two words, is emitted by the Mapper in <key,value> format as <word-pair,1>

Reducer: In the reducer, we have collected all the <key,value> pairs that are emmited by the Mapper. All the values assosiated with the same key (word-pair) are aggregated to get the count of the word-pair. This is emitted by the Reducer as the output. This procedure is applied to every unique word-pair sent as the key. The resulting output gives us the frequency of co-occurrence of the word-pairs so that We can identify the co-occurring words with the highest frequencies.

Visualization: We have used tableau software to generate the word cloud and after that the generated word cloud is then published in the website using the tableau online .

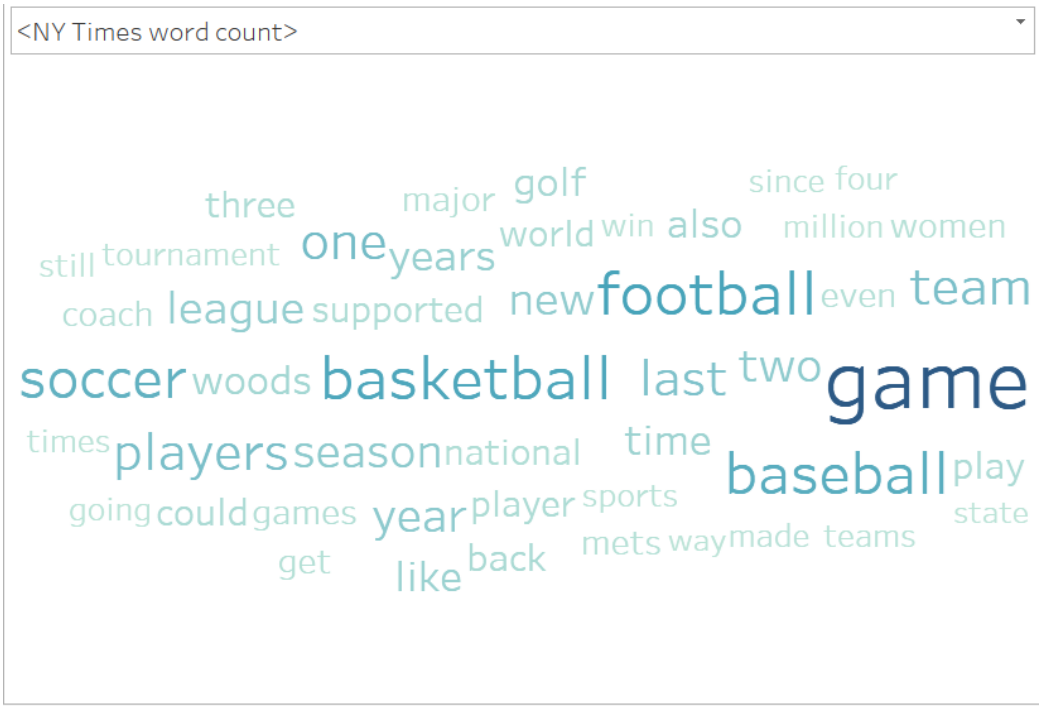
Common crawl cooccurance

Common crawl wordcount



NY times cooccurance

NY Times word count



Twitter coccurence

Twitter word count