

CSE 587 – DATA INTENSIVE COMPUTING – LAB 2

Submitted By: Esther Raja Kumari Katti, Keerthana Baskaran

UB# 50288205, 50288944

Contents:

1. *Introduction*
2. *Data collection*
 - *Twitter Data*
 - *New York times data*
 - *Common crawl data*
 - *Data preprocessing*
3. *Big Data Infrastructure*
 - *AWS*
4. *Analyze and visualize*
 - *Word count*
 - *Word co-occurrence*
 - *Visualize word count on all 3 sources data*
 - *Visualize word co-occurrence on all 3 sources data*
 - *How to run our lab?*
 - *Directory structure*
5. *Conclusion*

1. Introduction

In this lab, we have used our skills in data exploration developed in Lab1 and enhanced them by adding big data analytics and visualization skills. This document describes how we carried out our Lab2: Data Aggregation, Big Data Analysis and Visualization, which involves (i) data aggregation from more than one source using the APIs (ii) Applying classical big data analytic method of MapReduce (iii) building a visualization data product.

Our topic and subtopic that we used to collect data from all three sources are,

Topic: Sports

Subtopic: football, basketball, soccer, golf, cricket

2. Data Collection

a. Twitter data

We have created a twitter developer account from [www.developer.twitter.com](https://developer.twitter.com) and used the developer account's consumer key, consumer secret, access token, access secret keys. We used twitterR library and Searchtwitter function are used for collecting the tweets. Here we get 5000 tweets with English language. The data of 2019 are collected through this and removed retweets

using isretweet function. Duplicates are removed using unique function. At the end of this the data that we get is unclean unique data with punctuations and other expressions apart from character. We process this data using StemmingStopping.ipynb

A total of 12000 tweets are collected for main topic and a total of 27000 tweets are collected for the subtopics

b. New York Times data

For new York times data extraction, we have created a new York times developer account from www.developer.nytimes.com to use the access keys. The library used for data extraction is articleAPI from nytimesarticle. We extracted the links from nytimes by giving a query with our keyword to the article API along with the begin and end date. Here we extracted one week data from April 3 to April 11. Duplicate links are removed before we process the content. These links are then processed to get the content in html format which is sent to beautiful soup. By finding the <p> tags in the html content we extract the data. The data that we get here is unprocessed unclean data. We then process them using StemmingStopping.ipynb

We have extracted 500 links (articles) for the main topic and 100 links (articles) each for sub topic.

c. Common crawl data

For common crawl data, we used index.commoncrawl.org to filter our topic using www.espn.org. Common crawl is used to crawl the web pages and scrap the data from those espn pages and extract the information that we need in form of warc files. From the warc files we used warcio library to process and get the URL form the file. Then we hit each URL to get the content of the files. This is similar to what we did in NYTimes processing. We used beautiful soup to process and get the content of the files. We processed more than 500 links to extract the content that are relevant to our search keywords.

The content that we get here are unclean multilingual content which we processed using wordnet library to extract English characters alone. Finally, these texts are added to the file and sent to data preprocessing using StemmingStopping.ipynb

d. Data preprocessing

For processing the data. we have used the nltk library. The stemmer that we have used here is WordNetLemmatizer. For Stopwords we used stopwords from nltk library. For tokenizing the content we used TreebankWordDetokenizer from nltk.tokenizer.treebank. Using these packages, we have processed our data to get clean string only content. We have converted the text to lower characters and removed the numbers, punctuations, special characters from the unclean data. We have also applied stemming and stop words here in preprocessing.

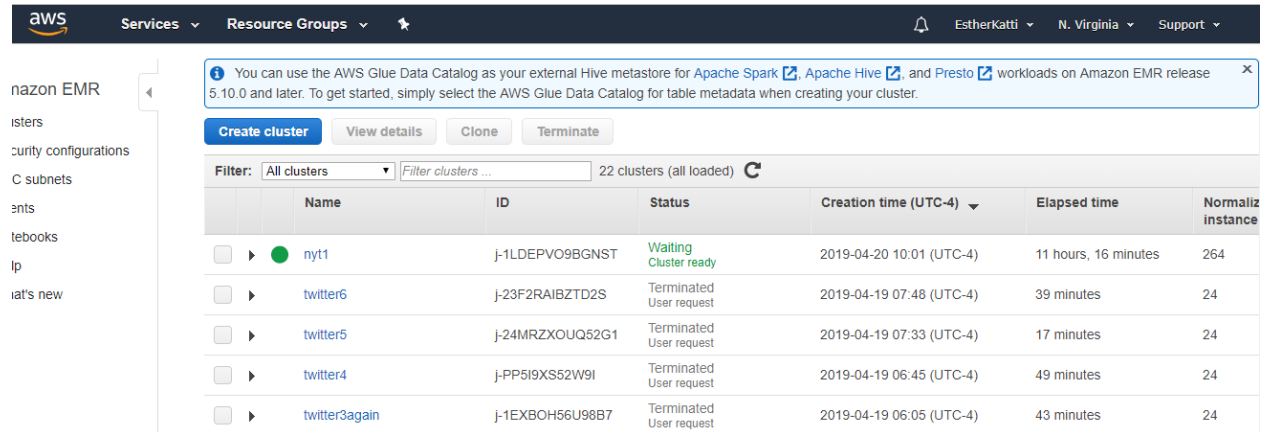
The output of this StemmingStopping.ipnyb is clean 26 alphabets lower cased English letters only.

3. Big Data infrastructure

a. AWS EMR

We have used Amazon AWS EMR as a big data infrastructure. Amazon Elastic MapReduce (EMR) is an Amazon web services (AWS) tool for big data processing and analysis.

We have uploaded our data and mapper reducer python code to amazon S3 bucket. Then we configured and launched our cluster using AWS management console. Once the cluster is done our output was stored in the S3 bucket path that we provided while adding steps to the cluster.



Name	ID	Status	Creation time (UTC-4)	Elapsed time	Normalized instance
ny1	j-1LDEPVO9BGNST	Waiting Cluster ready	2019-04-20 10:01 (UTC-4)	11 hours, 16 minutes	264
twitter6	j-23F2RAiBZTD2S	Terminated User request	2019-04-19 07:48 (UTC-4)	39 minutes	24
twitter5	j-24MRZXOUQ52G1	Terminated User request	2019-04-19 07:33 (UTC-4)	17 minutes	24
twitter4	j-PP5I9XS52W9I	Terminated User request	2019-04-19 06:45 (UTC-4)	49 minutes	24
twitter3again	j-1EXBOH56U98B7	Terminated User request	2019-04-19 06:05 (UTC-4)	43 minutes	24

4. Analyze and Visualize

a. Word count

We used the mapper and reducer python code provided by professor for running our word count algorithm on our collected data. Below are the top 10 words that we got for each topic and subtopic by different sources

Topic: Sports		
NYTimes data	Twitter data	Common crawl
Season	Sports	season
Game	Game	team
team	Baseball	year
Points	Team	play
League	Win	golf
Win	Football	football
Scored	Nfl	home
Play	Schedule	sport
Years	Play	game
coach	season	information

Subtopic: FootBall	
NYTimes data	Twitter data
Football	Football
Team	League
Years	News
Play	National
Game	Game
League	nfl
Yards	Team
Reuters	College
Work	Season
season	coach

Subtopic: BasketBall	
NYTimes data	Twitter data
Basketball	Basketball
Game	Team
Team	Coach
Season	Game
Points	Men
Coach	Play
Final	School
Tournament	Season
Win	Player
national	national

Subtopic: Soccer	
NYTimes data	Twitter data
Team	Soccer
Soccer	Team
Game	Girls
Reuters	Game
Even	Boys
League	Diver
United	League
Country	Play
years	Sports
national	rescued

Subtopic: Golf

NYTimes data	Twitter data
Golf	Golf
National	Course
President	Club
Years	Play
Women	Trump
round	Boys
Tour	Team
Club	Tournament
Week	Game
trump	check

Subtopic: Cricket	
NYTimes data	Twitter data
Pakistan	Cricket
India	World
Cricket	cup
team	Twenty
Cup	Squad
Australia	Run
Match	Match
Reuters	Team
Ipl	Game
ground	australia

b. Word Co-occurrence

The mapper and reducer of word co-occurrence will find the top 10 frequent words that is occurring in the text and find the frequent co-occurring word along with that. Below are the top 10 co-occurring words that we got for each topic and subtopic by different sources.

Topic: Sports		
NYTimes data	Twitter data	Common crawl
scored,goals	sports,fan	golf,digest
scored,points	sports,fans	information,violation
points,assists	sports,talk	football,games
season,finale	sports,radio	golf,course
points,rebounds	sports,teams	golf,swing
team,mate	sports,betting	home,games
league,baseball	sports,illustrated	information,provided
league,history	nfl,draft	play,game
game,series	sports,car	football,game

years,ago	sports,team	golf,ball
-----------	-------------	-----------

Subtopic: FootBall	
NYTimes data	Twitter data
football,tiki	college,prospect
football,player	football,league
play,well	football,player
football,hall	football,team
team,filed	college,football
football,confederation	football,season
football,stadium	league,news
game,model	nfl,draft
work,force	football,game
yards,touchdowns	national,football

Subtopic: BasketBall	
NYTimes data	Twitter data
national,championship	basketball,association
coach,chris	basketball,coach
national,player	basketball,program
points,assists	national,basketball
points,per	basketball,game
basketball,team	men,basketball
points,rebounds	play,basketball
basketball,hall	basketball,camp
final,game	basketball,team
national,semifinal	basketball,player

Subtopic: Soccer	
NYTimes data	Twitter data
national,team	boys,soccer
soccer,game	diver,save
soccer,stadium	soccer,game
league,title	play,soccer
national,soccer	soccer,player
united,states	soccer,ball
even,though	soccer,team
years,ago	girls,soccer
game,turns	rescued,tennessee

soccer,federation	team,rescued
-------------------	--------------

Subtopic: Golf	
NYTimes data	Twitter data
women,amateur	golf,outing
golf,courses	golf,courses
trump,administration	golf,club
trump,florida	golf,course
golf,club	golf,tournament
golf,course	golf,cart
president,trump	play,golf
trump,organization	boys,golf
national,women	golf,ball
years,ago	golf,team

Subtopic: Cricket	
NYTimes data	Twitter data
india,captain	cricket,score
cricket,ground	cricket,wireless
australia,union	world,cup
ground,scg	australia,world
match,fc	cricket,match
pakistan,india	cricket,squad
india,prime	cup,squad
match,allowed	cricket,season
cricket,wicket	cricket,team
cup,england	cricket,world

Visualization of word count on all three data sources

For visualization we used Tableau.

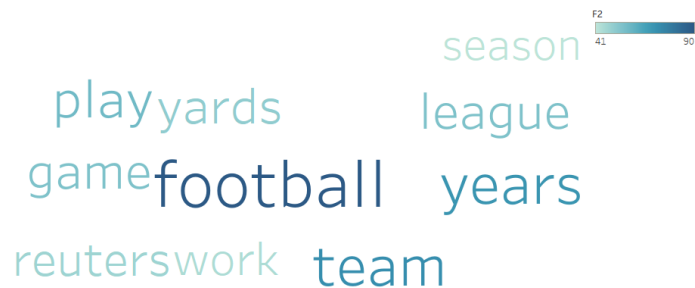
The image given below is the output of common crawl data for word count. The topic here is sports



The image given below is the output of new York times data for word count. The topic here is sports



The image given below is the output of new York times data for word count. The topic here is football



The image given below is the output of new York times data for word count. The topic here is basketball



The image given below is the output of new York times data for word count. The topic here is soccer



The image given below is the output of new York times data for word count. The topic here is golf



The image given below is the output of new York times data for word count. The topic here is cricket



The image given below is the output of twitter data for word count. The topic here is sports



The image given below is the output of twitter data for word count. The topic here is basketball



The image given below is the output of twitter data for word count. The topic here is cricket



The image given below is the output of twitter data for word count. The topic here is football



The image given below is the output of twitter data for word count. The topic here is golf



The image given below is the output of twitter data for word count. The topic here is soccer



Visualization of word cooccurrence on all three data sources

The image given below is the output of common crawl data for word count. The topic here is sports



The image given below is the output of new York times data for word count. The topic here is sports



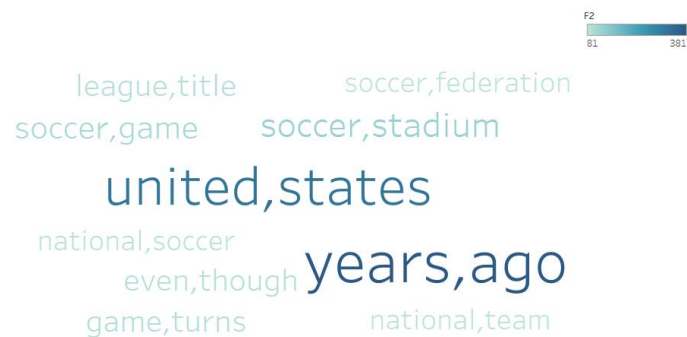
The image given below is the output of new York times data for word count. The topic here is football



The image given below is the output of new York times data for word count. The topic here is basketball



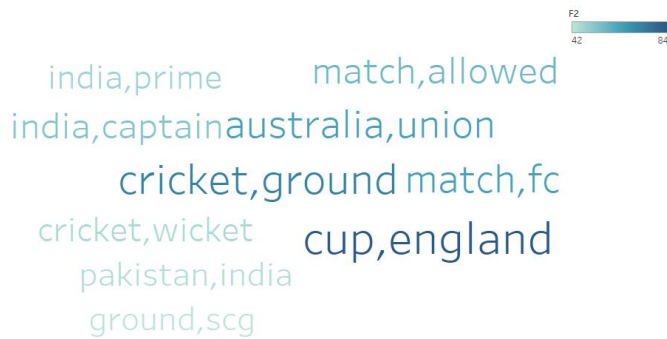
The image given below is the output of new York times data for word count. The topic here is soccer



The image given below is the output of new York times data for word count. The topic here is golf



The image given below is the output of new York times data for word count. The topic here is cricket



The image given below is the output of twitter data for word count. The topic here is sports



The image given below is the output of twitter data for word count. The topic here is basketball



The image given below is the output of twitter data for word count. The topic here is cricket



The image given below is the output of twitter data for word count. The topic here is football



The image given below is the output of twitter data for word count. The topic here is golf



The image given below is the output of twitter data for word count. The topic here is soccer



How to run our lab?

Step 1: Collect the data for NYTtimes using the NYTimes_data.ipynb file (placed inside part1 -> code -> nytimes data collection code -> NYTimes_data.ipynb), for twitter use DIC_LAB2_TWITTERDATA.ipynb (placed inside part1 -> code -> twitter data collection code -> DIC_LAB2_TWITTERDATA.ipynb) and for common crawl use CommonCrawlCode.ipynb file (placed inside part1 -> code -> common crawl data collection code -> CommonCrawlCode.ipynb). The output of these files are text files which are unclean data. For common crawl only, you need to mention the warc file as input to extract the data from CommonCrawlCode.ipynb. The warc file is placed in part1 -> data -> common crawl data -> source files from common crawl – warc files

(OR)

you can also use the extracted data files to proceed to step 2 (use files from part1 -> data -> “(different sources)” -> unprocessed data.

Step 2: After getting the text files, run the StemmingStopping.ipynb file to pre-process the data for twitter and nytimes. For commoncrawl data run StemmingStopping_commoncrawl.ipynb file. These files are placed in the same directory where the original data collection code is placed in step 1. The output of this step is cleaned data.

(OR)

You can also use the preprocessed data files placed in part1 -> data -> “different sources” -> processed data -> “files” to proceed to step 3

Step 3: Set up the AWS EMR cluster and run the mapper.py and reducer.py inside count folder which is placed inside part2 -> word count -> mapper.py/reducer.py. This gives the output of word count for all three data sources. For word co-occurrence, run the reducer.py and corresponding mapper.py inside the cooccurrence folder which is inside part2 -> word cooccurrence -> mapper.py/reducer.py

For word count use same mapper.py and reducer.py for all files. For co-occurrence change the mapper file accordingly as specified in the folder (part2 -> word cooccurrence -> mapper.py/reducer.py) for each sub topic. The reducer file is same for all topics.

(OR)

You can use the output file from reducer placed in part2 -> output of wordcooccurrence/output of word count -> “different sources” -> all files.txt to proceed to step 4

Step 4: After getting the output for word count and cooccurrence, run the code Top_WordsExtraction.ipynb to extract the top 10 frequent words. This code is placed inside part3 -> “different sources” -> code -> “count/cooccurrence” -> extraction of top 10 words -> Top_wordExtraction.ipynb

(OR)

You can use the top 10 extracted files placed in part3 -> “different sources” -> code -> count/cooccurrence -> extraction of top 10 words -> “the topic needed” -> tableau.csv to proceed to step 5

Step 5: Now you have top frequent words in both word count and cooccurrence. Pass these files to tableau to generate a word cloud.

(OR)

You can view the generated word count outputs placed in part3 -> “different sources” -> images
-> count/cooccurrence -> Visualization -> png image

Step 6: Now you have all the output word cloud images. Publish them on the tableau server

(OR)

View the already published dashboard using the link provided in Webpage -> link -> count/cooccurrence.txt

Directory Structure

kbaskara.zip

- Report.pdf
- Video.mp4
- part1
 - Code
 - Common crawl data collection code -> *all scripts and codes for collecting data and processing*
 - Nytimes data collection code -> *all scripts and codes for collecting data and processing*
 - Twitter data collection code -> *all scripts and codes for collecting data and processing*
 - Data
 - Twitter data
 - Processed data -> *all output files after processing the data*
 - Unprocessed data from twitter -> *all output files containing unprocessed data*
 - NYT data
 - Processed data -> *all output files after processing the data*
 - Unprocessed data extracted from nytimes -> *all output files containing unprocessed data*
 - Commoncrawl data
 - Processed data -> *all output files after processing the data*
 - Source file from common crawl – warc files -> *all warc files from common crawl to input commoncrawlcode.ipynb*
 - Unprocessed data extracted from warc files -> *all output files containing unprocessed data*
- part2
 - Count -> *all mapper reducer scripts and codes for running word count*
 - Cooccurrence -> *all mapper reducer scripts and codes for running word cooccurrence*
 - Output of word cooccurrence
 - Common crawl -> *all output folders of word cooccurrence for topic sports*
 - NYT
 - NYTmain -> *all output folders of word cooccurrence for topic sports*
 - NYTsubtopic1 -> *all output folders of word cooccurrence for subtopic football*
 - NYTsubtopic2 -> *all output folders of word cooccurrence for subtopic basketball*

- NYTsubtopic3 -> *all output folders of word cooccurrence for subtopic soccer*
 - NYTsubtopic4 -> *all output folders of word cooccurrence for subtopic golf*
 - NYTsubtopic5 -> *all output folders of word cooccurrence for subtopic cricket*
 - Twitter
 - Tmain -> *all output folders of word cooccurrence for topic sports*
 - Tsubtopic1 -> *all output folders of word cooccurrence for subtopic basketball*
 - Tsubtopic2 -> *all output folders of word cooccurrence for subtopic cricket*
 - Tsubtopic3 -> *all output folders of word cooccurrence for subtopic football*
 - Tsubtopic4 -> *all output folders of word cooccurrence for subtopic golf*
 - Tsubtopic5 -> *all output folders of word cooccurrence for subtopic soccer*
- Output of word count
 - Common crawl -> *all output folders of word count for topic sports*
 - NYT
 - NYTmain -> *all output folders of word count for topic sports*
 - NYTsubtopic1 -> *all output folders of word count for subtopic football*
 - NYTsubtopic2 -> *all output folders of word count for subtopic basketball*
 - NYTsubtopic3 -> *all output folders of word count for subtopic soccer*
 - NYTsubtopic4 -> *all output folders of word count for subtopic golf*
 - NYTsubtopic5 -> *all output folders of word count for subtopic cricket*
 - Twitter
 - Tmain -> *all output folders of word count for topic sports*
 - Tsubtopic1 -> *all output folders of word count for subtopic basketball*
 - Tsubtopic2 -> *all output folders of word count for subtopic cricket*
 - Tsubtopic3 -> *all output folders of word count for subtopic football*
 - Tsubtopic4 -> *all output folders of word count for subtopic golf*
 - Tsubtopic5 -> *all output folders of word count for subtopic soccer*
- part3
 - Twitter
 - Code
 - Cooccurrence
 - Extraction of top 10 words -> *code and output to extract top 10 words from reducer output file*
 - Count
 - Extraction of top 10 words -> *code and output to extract top 10 words from reducer output file*
 - Images
 - Cooccurrence
 - Visualization -> *output word cooccurrence visualization*
 - Count
 - Visualization -> *output word count visualization*
 - NYT (folder)
 - Code (folder)
 - Cooccurrence

- Extraction of top 10 words -> *code and output to extract top 10 words from reducer output file*
 - Count
 - Extraction of top 10 words -> *code and output to extract top 10 words from reducer output file*
- Images (folder)
 - Cooccurrence
 - Visualization -> *output word cooccurrence visualization*
 - Count
 - Visualization -> *output word count visualization*
- Commoncrawl
 - Code
 - Cooccurrence
 - Extraction of top 10 words -> *code and output to extract top 10 words from reducer output file*
 - Count
 - Extraction of top 10 words -> *code and output to extract top 10 words from reducer output file*
 - Images
 - Cooccurrence
 - Visualization -> *output word cooccurrence visualization*
 - Count
 - Visualization -> *output word count visualization*
- Webpage
 - Link -> *link to tableau server*
 - Output -> *output dashboards that are published in tableau server*

5. Conclusion

In this lab, we have expanded our skills on data exploration that we developed in Lab1 and enhanced them by adding big data analytics and visualization skills. We have applied big data analytic method of MapReduce to unstructured data collected and built a visualization on the data using Tableau. We also applied the same steps to small data collected per day and visualized the outputs were similar to what we got for big data.