LAB-2

DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

Divya Srivastava
Person number **– 50290383**
UBID **– divyasri**
And
Meghana Vasudeva
Person number **– 50290586**
UBID **- mvasudev**

Table of Contents

Abstract......3

Part-1......3

Data Collection......3

Twitter......3

New York Times......3

Common Crawl......4

Cleaning......4

Part-2......4

Virtual Machine Installation Steps......4

Screenshots of Execution on VM......5

Code Snippets for mapper.py & reducer.py (Word Count)......8

Top 10 Words For Word Count......8

Code Snippets for mapper.py & reducer.py (Word Co-occurrence)......9

Top-10 Words For Word Co-occurrence......11

Part-3......12

Visualization......12

Tableau Installation......12

Bubble/Textual Visualization......13

Twitter Word Count And Analysis......13

New York Times Word Count And Analysis......14

Common Crawl Word Count And Analysis......15

Common Crawl Word Co-occurrence And Analysis......16

New York Times Word Co-occurrence And Analysis......17

Twitter Word Co-occurrence And Analysis......18

Directory Structure......19

Demo Video Link......19

Conclusion......19

References......19

Abstract:

In this lab, we expanded our skills in data exploration developed in Lab1 and enhanced them by adding big data analytics and visualization skills. This document describes Lab2: Data Aggregation, Big Data Analysis and Visualization, involves (i) data aggregation from more than one source using the APIs (Application programming interface) exposed by data sources, (ii) Applying classical big data analytic method of MapReduce to the unstructured data collected, (iii) store the data collected on WORM infrastructure Hadoop and (iii) building a visualization data product.

We have leveraged the data collection and exploratory data analysis skills developed in Lab1 to accomplish the goals of Lab2.

Part – 1:

Data Collection

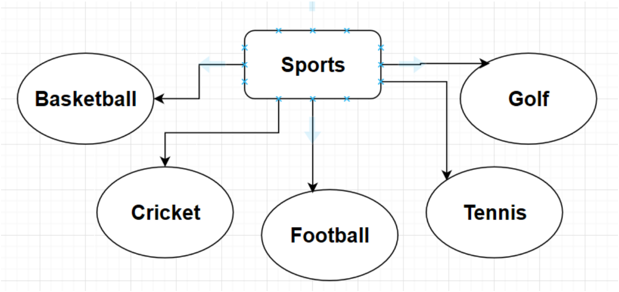
An important and critical phase of the data-science process is data collection. Several organizations including the federal government (data.gov) have their data available to the public for various purposes. Social network applications such as Twitter and Facebook collect enormous amount of data contributed by their numerous and prolific user. An API or application programming interface is a standard, secure and programmatic access to data by an organization that owns the data.

We collected the data from the following three sources-

1. Opinion-based social media in twitter
2. Research data in New York Times
3. Common crawl.

The above data was collected based on same set of keywords and topics. Then we processed the three data sets collected individually using classical big data methods and compared the outcomes using popular visualization methods.

We selected the following topic and sub topics for our analysis-



1. Twitter

   Twitter data is unique from data shared by most other social platforms because it reflects information that users choose to share publicly. We registered for twitter API in our lab1 and used it for our lab2 as well. These APIs are required to register an application. By default, applications can only access public information on Twitter.

2. New York Times

   We used the article search API from nytimesarticle to look up articles by keyword. We refined our search using various filters. We also used BeautifulSoup to extract the content. Beautiful Soup parses anything we give it, and does the tree traversal stuff for us. We can tell it "Find all the links", or "Find all the links of class externalLink", or "Find all the links whose urls match "foo.com", or "Find the table heading that's got bold text, then give me that text."

3. Common Crawl
   For extracting data from common crawl, we read the warc file that we downloaded from the common crawl website for March 2019 data. Crawling through it, we extracted the data from the links where language was English and matching keywords were found. We did this for all our keywords obtaining the data that we used for our lab.

Cleaning

The data obtained after data collection for twitter, New York times and common crawl through their respective APIs, was in the raw format. So our next step was cleaning for which we used a python script. We used natural language toolkit known as **NLTK** package to tokenize and tag the text and for identifying the named entities. We also used **regex** package for cleaning the data. This module provides regular expression matching operations. It is a sequence of character(s) mainly used to find and replace patterns in a string or file, so we used NLTK and regex both to filter the unwanted text from the data.
The text that we filtered out mainly contained the following unwanted elements-

1. Punctuations (‘!”#$%&\’()*+,-./:;⇔?@[\\]^_`{|}~’)
2. Stop words (I, me, my, myself, we, are etc)
3. Stemming words (running becomes run and so on)
4. Tokenize (
5. Digits (0-9)
6. URL
7. Tags
8. Unicode

Part – 2:

Virtual Machine Installation Steps

We followed the following steps for the installation of VM and the image to run the wordcount and co-occurrence with Hadoop Map Reduce on HDFS-

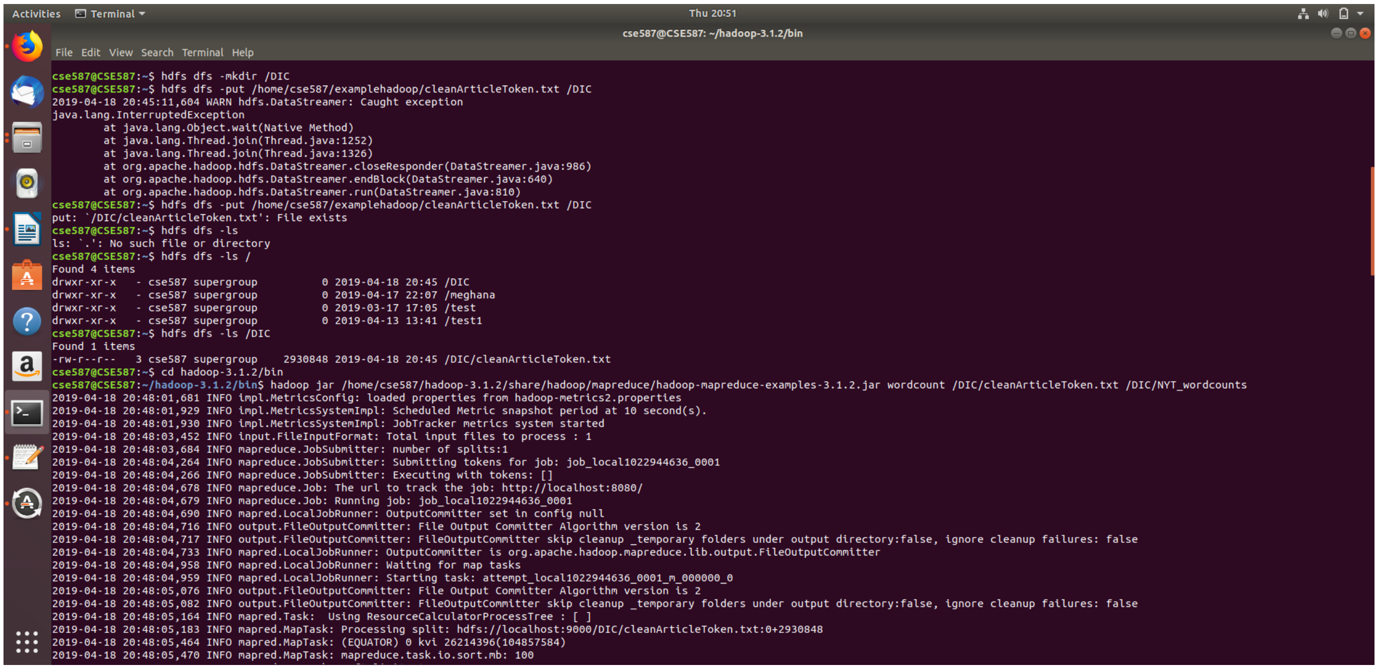
1. Installed virtualbox from https://www.virtualbox.org/
2. Download the virtual image provided by TA on piazza
3. For loading the image, opened virtualbox and clicked **New.**
4. Used the following configurations: Type **-** Linux, **Version** -64bit ,**Name -** CSE587
5. Selected RAM as 8192 MB
6. When selecting hard disk, used the one provided by TA on piazza
7. Installed ubuntu by running the image.

Opened the terminal and executed the following commands to execute the word count and co-occurrence using mapper and reducer on Hadoop-

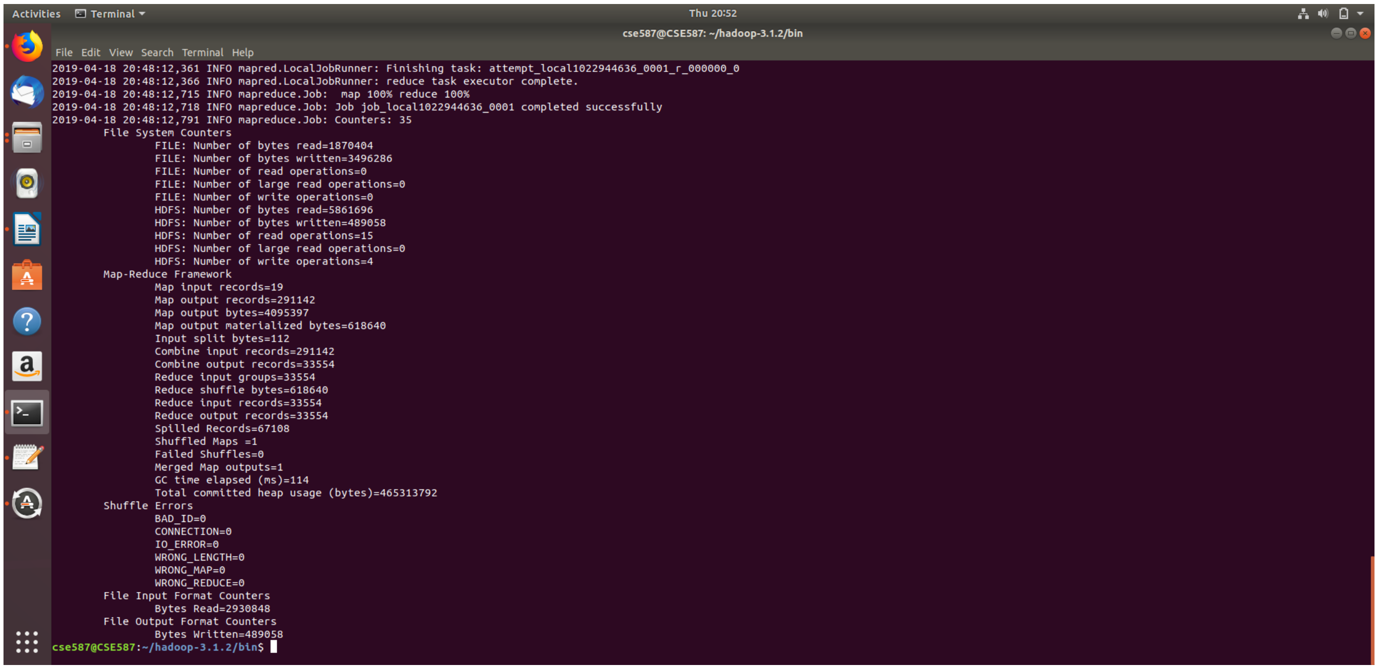
```
> start-dfs.sh                          //starting namenodes and datanodes. Setting up the single node
cluster.
> sudo rm -r /tmp/*                                     //cleaning everything in tmp directory
> hdfs namenode -format
> hdfs dfs -ls /                                        //check whether there is any directory in
hdfs
> hdfs dfs -mkdir /test                                 //if there is no directory create one named
test
```

```
> hdfs dfs -put /home/cse587/examplehadoop/exam.txt  /test   //putting the exam.txt from local directory into the test directory inside hdfs
> cd hadoop-3.1.2/bin
> hadoop jar /home/cse587/hadoop-3.1.2/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar wordcount /test/exam.txt /test/output
> hdfs dfs -ls /test/output                                        //You should see the output file as part-r0000
```

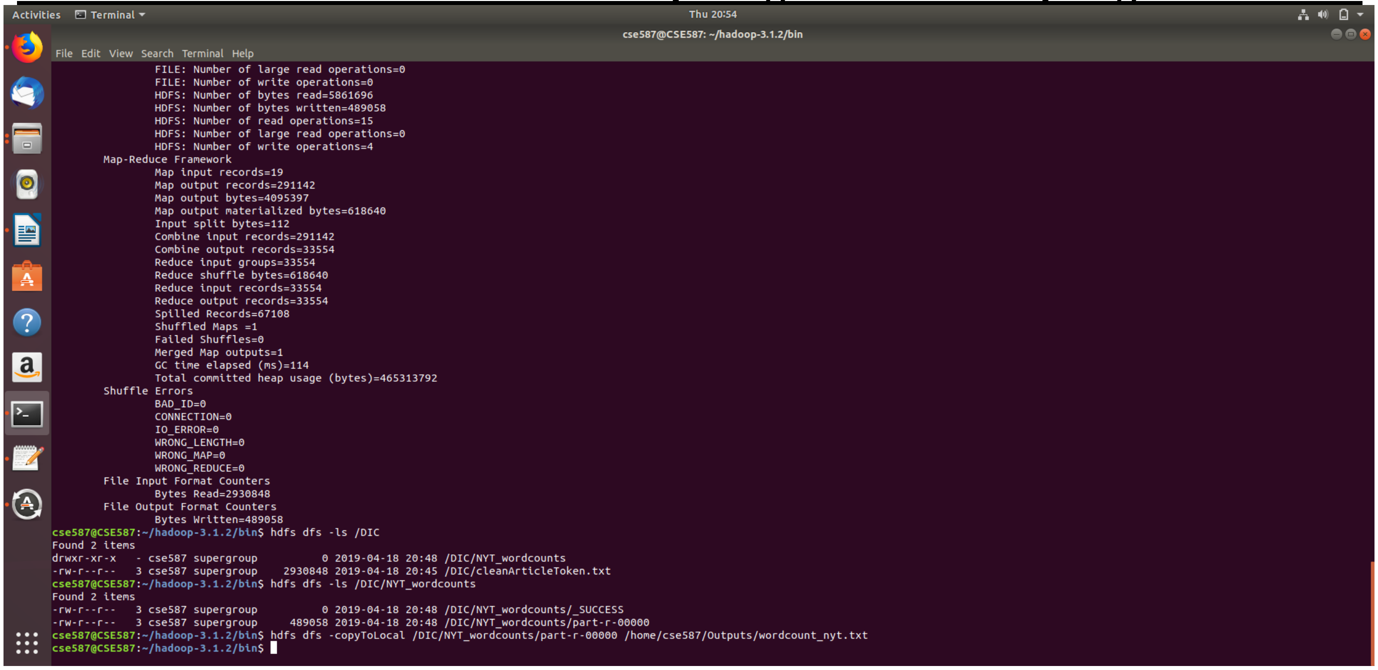
Screenshots of Execution on VM


Activities
Terminal
Thu 20:51
cse587@CSE587: ~/hadoop-3.1.2/bin
File Edit View Search Terminal Help

```
cse587@CSE587:~$ hdfs dfs -mkdir /DIC
cse587@CSE587:~$ hdfs dfs -put /home/cse587/examplehadoop/cleanArticleToken.txt /DIC
2019-04-18 20:45:11,604 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1252)
        at java.lang.Thread.join(Thread.java:1326)
        at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:986)
        at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:640)
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:810)
cse587@CSE587:~$ hdfs dfs -put /home/cse587/examplehadoop/cleanArticleToken.txt /DIC
put: `/DIC/cleanArticleToken.txt': File exists
cse587@CSE587:~$ hdfs dfs -ls
ls: `.': No such file or directory
cse587@CSE587:~$ hdfs dfs -ls /
Found 4 items
drwxr-xr-x   - cse587 supergroup          0 2019-04-18 20:45 /DIC
drwxr-xr-x   - cse587 supergroup          0 2019-04-17 22:07 /meghana
drwxr-xr-x   - cse587 supergroup          0 2019-03-17 17:05 /test
drwxr-xr-x   - cse587 supergroup          0 2019-04-13 13:41 /test1
cse587@CSE587:~$ hdfs dfs -ls /DIC
Found 1 items
-rw-r--r--   3 cse587 supergroup    2930848 2019-04-18 20:45 /DIC/cleanArticleToken.txt
cse587@CSE587:~$ cd hadoop-3.1.2/bin
cse587@CSE587:~/hadoop-3.1.2/bin$ hadoop jar /home/cse587/hadoop-3.1.2/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar wordcount /DIC/cleanArticleToken.txt /DIC/NYT_wordcounts
2019-04-18 20:48:01,681 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2019-04-18 20:48:01,929 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2019-04-18 20:48:01,930 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2019-04-18 20:48:03,452 INFO input.FileInputFormat: Total input files to process : 1
2019-04-18 20:48:03,684 INFO mapreduce.JobSubmitter: number of splits:1
2019-04-18 20:48:04,264 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1022944636_0001
2019-04-18 20:48:04,266 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-04-18 20:48:04,678 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2019-04-18 20:48:04,679 INFO mapreduce.Job: Running job: job_local1022944636_0001
2019-04-18 20:48:04,690 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2019-04-18 20:48:04,716 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-04-18 20:48:04,717 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2019-04-18 20:48:04,733 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2019-04-18 20:48:04,958 INFO mapred.LocalJobRunner: Waiting for map tasks
2019-04-18 20:48:04,959 INFO mapred.LocalJobRunner: Starting task: attempt_local1022944636_0001_m_000000_0
2019-04-18 20:48:05,076 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-04-18 20:48:05,082 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2019-04-18 20:48:05,164 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2019-04-18 20:48:05,183 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/DIC/cleanArticleToken.txt:0+2930848
2019-04-18 20:48:05,464 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2019-04-18 20:48:05,470 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
```

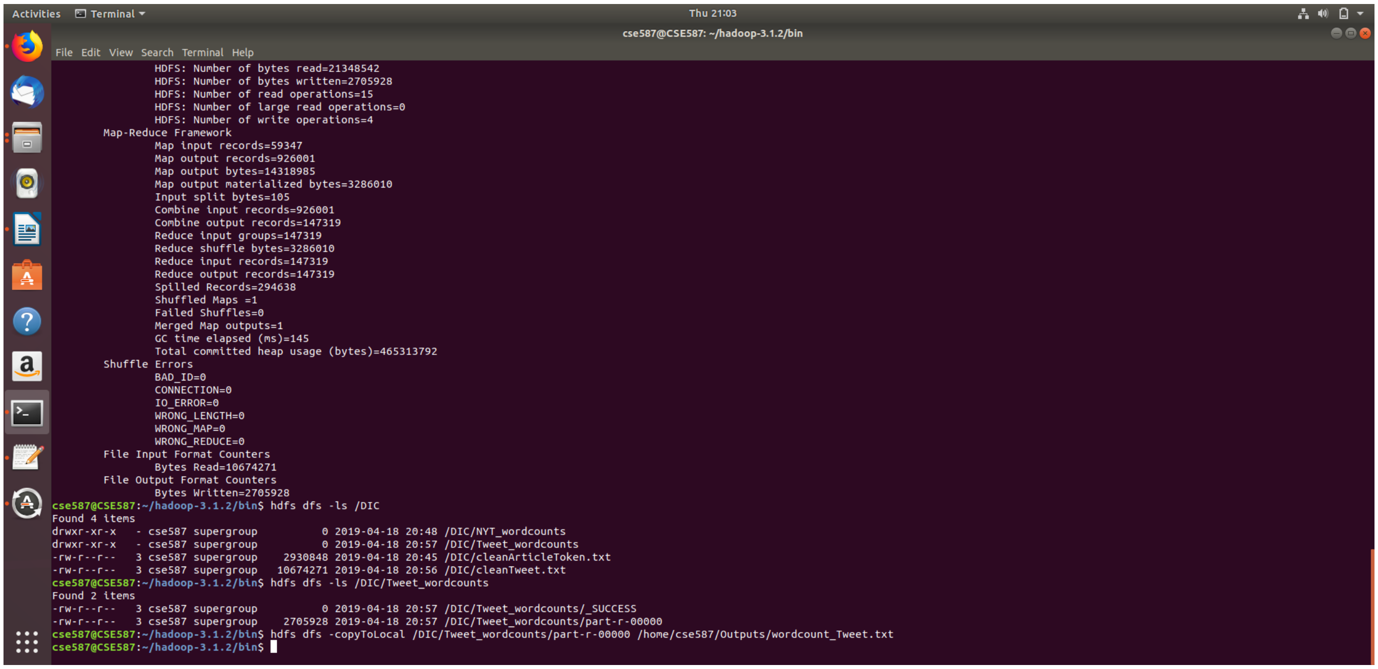

Activities
Terminal
Thu 20:52
cse587@CSE587: ~/hadoop-3.1.2/bin
File Edit View Search Terminal Help

```
2019-04-18 20:48:12,361 INFO mapred.LocalJobRunner: Finishing task: attempt_local1022944636_0001_r_000000_0
2019-04-18 20:48:12,366 INFO mapred.LocalJobRunner: reduce task executor complete.
2019-04-18 20:48:12,715 INFO mapreduce.Job:  map 100% reduce 100%
2019-04-18 20:48:12,718 INFO mapreduce.Job: Job job_local1022944636_0001 completed successfully
2019-04-18 20:48:12,791 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=1870404
                FILE: Number of bytes written=3496286
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5861696
                HDFS: Number of bytes written=489058
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=19
                Map output records=291142
                Map output bytes=4095397
                Map output materialized bytes=618640
                Input split bytes=112
                Combine input records=291142
                Combine output records=33554
                Reduce input groups=33554
                Reduce shuffle bytes=618640
                Reduce input records=33554
                Reduce output records=33554
                Spilled Records=67108
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=114
                Total committed heap usage (bytes)=465313792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2930848
        File Output Format Counters
                Bytes Written=489058
cse587@CSE587:~/hadoop-3.1.2/bin$ 
```



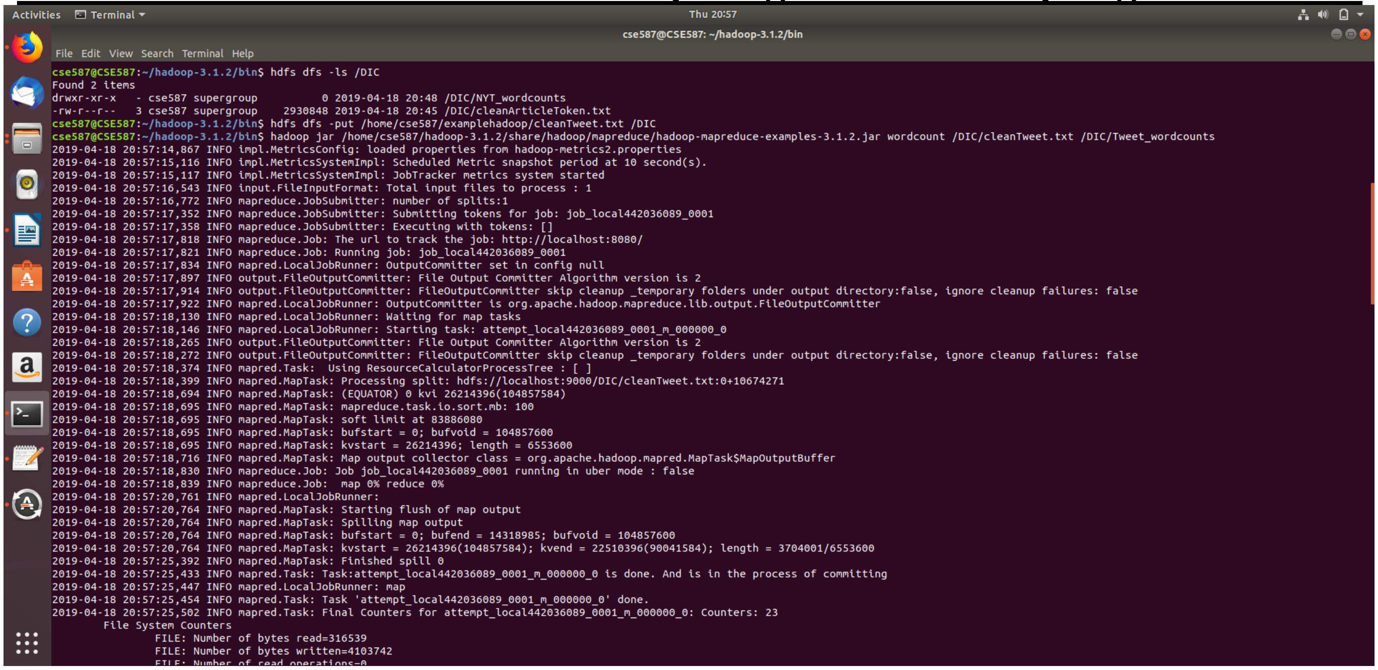
Activities
Terminal
Thu 20:54
cse587@CSE587: ~/hadoop-3.1.2/bin
File Edit View Search Terminal Help

```
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5861696
                HDFS: Number of bytes written=489058
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=19
                Map output records=291142
                Map output bytes=4095397
                Map output materialized bytes=618640
                Input split bytes=112
                Combine input records=291142
                Combine output records=33554
                Reduce input groups=33554
                Reduce shuffle bytes=618640
                Reduce input records=33554
                Reduce output records=33554
                Spilled Records=67108
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=114
                Total committed heap usage (bytes)=465313792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=2930848
        File Output Format Counters
                Bytes Written=489058
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -ls /DIC
Found 2 items
drwxr-xr-x   - cse587 supergroup          0 2019-04-18 20:48 /DIC/NYT_wordcounts
-rw-r--r--   3 cse587 supergroup    2930848 2019-04-18 20:45 /DIC/cleanArticleToken.txt
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -ls /DIC/NYT_wordcounts
Found 2 items
-rw-r--r--   3 cse587 supergroup          0 2019-04-18 20:48 /DIC/NYT_wordcounts/_SUCCESS
-rw-r--r--   3 cse587 supergroup     489058 2019-04-18 20:48 /DIC/NYT_wordcounts/part-r-00000
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -copyToLocal /DIC/NYT_wordcounts/part-r-00000 /home/cse587/Outputs/wordcount_nyt.txt
cse587@CSE587:~/hadoop-3.1.2/bin$ 
```

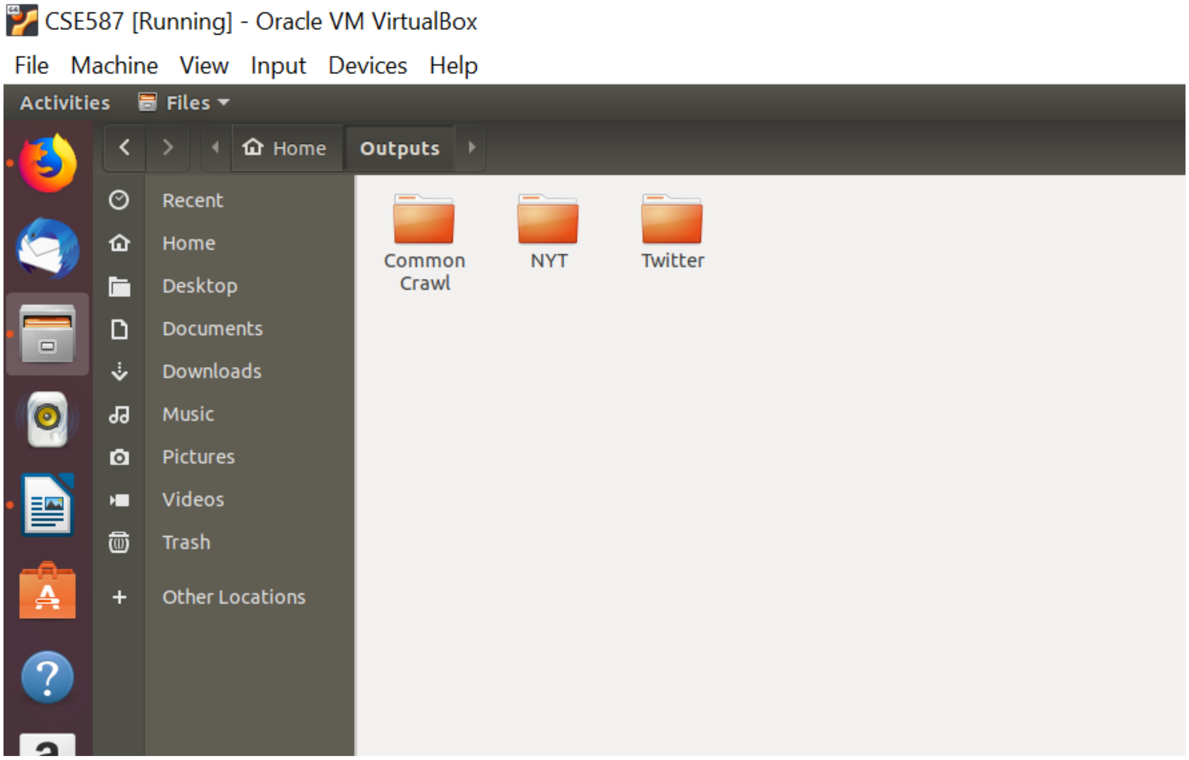


Activities
Terminal
Thu 21:03
cse587@CSE587: ~/hadoop-3.1.2/bin
File Edit View Search Terminal Help

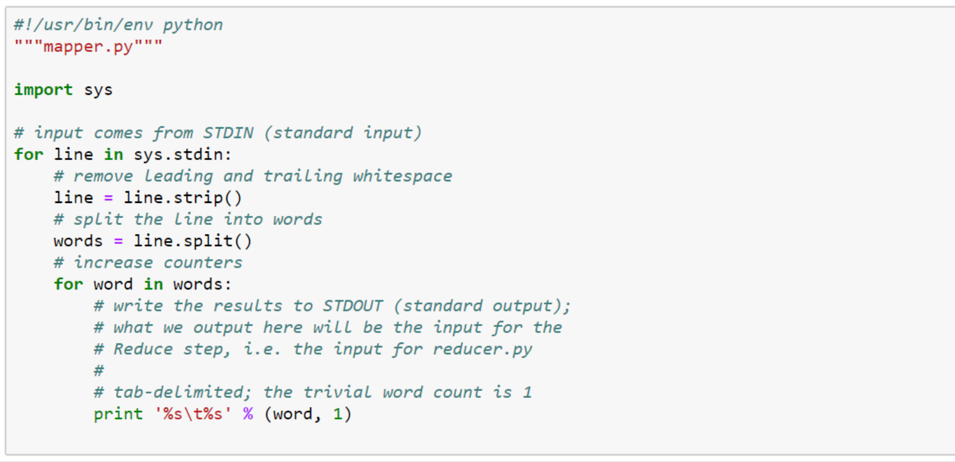
```
                HDFS: Number of bytes read=21348542
                HDFS: Number of bytes written=2705928
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
        Map-Reduce Framework
                Map input records=59347
                Map output records=926001
                Map output bytes=14318985
                Map output materialized bytes=3286010
                Input split bytes=105
                Combine input records=926001
                Combine output records=147319
                Reduce input groups=147319
                Reduce shuffle bytes=3286010
                Reduce input records=147319
                Reduce output records=147319
                Spilled Records=294638
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=145
                Total committed heap usage (bytes)=465313792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=10674271
        File Output Format Counters
                Bytes Written=2705928
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -ls /DIC
Found 4 items
drwxr-xr-x   - cse587 supergroup          0 2019-04-18 20:48 /DIC/NYT_wordcounts
drwxr-xr-x   - cse587 supergroup          0 2019-04-18 20:57 /DIC/Tweet_wordcounts
-rw-r--r--   3 cse587 supergroup    2930848 2019-04-18 20:45 /DIC/cleanArticleToken.txt
-rw-r--r--   3 cse587 supergroup   10674271 2019-04-18 20:56 /DIC/cleanTweet.txt
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -ls /DIC/Tweet_wordcounts
Found 2 items
-rw-r--r--   3 cse587 supergroup          0 2019-04-18 20:57 /DIC/Tweet_wordcounts/_SUCCESS
-rw-r--r--   3 cse587 supergroup    2705928 2019-04-18 20:57 /DIC/Tweet_wordcounts/part-r-00000
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -copyToLocal /DIC/Tweet_wordcounts/part-r-00000 /home/cse587/Outputs/wordcount_Tweet.txt
cse587@CSE587:~/hadoop-3.1.2/bin$ 
```


Activities
Terminal
Thu 20:57
cse587@CSE587: ~/hadoop-3.1.2/bin
File Edit View Search Terminal Help

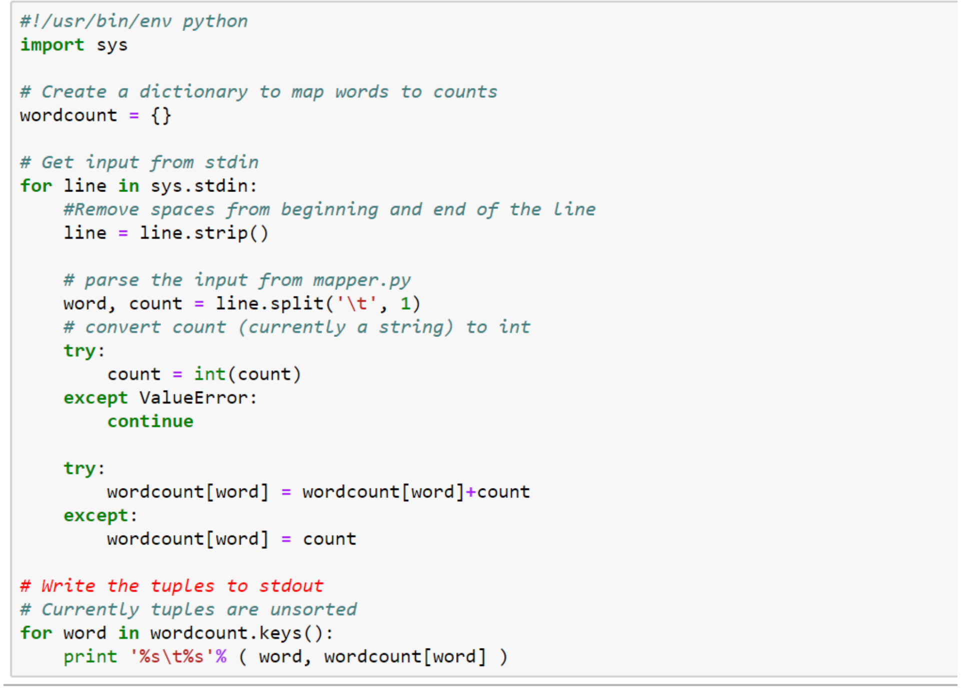
```
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -ls /DIC
Found 2 items
drwxr-xr-x   - cse587 supergroup          0 2019-04-18 20:48 /DIC/NYT_wordcounts
-rw-r--r--   3 cse587 supergroup    2930848 2019-04-18 20:45 /DIC/cleanArticleToken.txt
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -put /home/cse587/examplehadoop/cleanTweet.txt /DIC
cse587@CSE587:~/hadoop-3.1.2/bin$ hadoop jar /home/cse587/hadoop-3.1.2/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.2.jar wordcount /DIC/cleanTweet.txt /DIC/Tweet_wordcounts
2019-04-18 20:57:14,867 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2019-04-18 20:57:15,116 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2019-04-18 20:57:15,117 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2019-04-18 20:57:16,543 INFO input.FileInputFormat: Total input files to process : 1
2019-04-18 20:57:16,772 INFO mapreduce.JobSubmitter: number of splits:1
2019-04-18 20:57:17,352 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local442036089_0001
2019-04-18 20:57:17,358 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-04-18 20:57:17,818 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2019-04-18 20:57:17,821 INFO mapreduce.Job: Running job: job_local442036089_0001
2019-04-18 20:57:17,834 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2019-04-18 20:57:17,897 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-04-18 20:57:17,914 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2019-04-18 20:57:17,922 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2019-04-18 20:57:18,130 INFO mapred.LocalJobRunner: Waiting for map tasks
2019-04-18 20:57:18,146 INFO mapred.LocalJobRunner: Starting task: attempt_local442036089_0001_m_000000_0
2019-04-18 20:57:18,265 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-04-18 20:57:18,272 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2019-04-18 20:57:18,374 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2019-04-18 20:57:18,399 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/DIC/cleanTweet.txt:0+10674271
2019-04-18 20:57:18,694 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2019-04-18 20:57:18,695 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2019-04-18 20:57:18,695 INFO mapred.MapTask: soft limit at 83886080
2019-04-18 20:57:18,695 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2019-04-18 20:57:18,695 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2019-04-18 20:57:18,716 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2019-04-18 20:57:18,830 INFO mapreduce.Job: Job job_local442036089_0001 running in uber mode : false
2019-04-18 20:57:18,839 INFO mapreduce.Job:  map 0% reduce 0%
2019-04-18 20:57:20,761 INFO mapred.LocalJobRunner:
2019-04-18 20:57:20,764 INFO mapred.MapTask: Starting flush of map output
2019-04-18 20:57:20,764 INFO mapred.MapTask: Spilling map output
2019-04-18 20:57:20,764 INFO mapred.MapTask: bufstart = 0; bufend = 14318985; bufvoid = 104857600
2019-04-18 20:57:20,764 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 22510396(90041584); length = 3704001/6553600
2019-04-18 20:57:25,392 INFO mapred.MapTask: Finished spill 0
2019-04-18 20:57:25,433 INFO mapred.Task: Task:attempt_local442036089_0001_m_000000_0 is done. And is in the process of committing
2019-04-18 20:57:25,447 INFO mapred.LocalJobRunner: map
2019-04-18 20:57:25,454 INFO mapred.Task: Task 'attempt_local442036089_0001_m_000000_0' done.
2019-04-18 20:57:25,502 INFO mapred.Task: Final Counters for attempt_local442036089_0001_m_000000_0: Counters: 23
        File System Counters
                FILE: Number of bytes read=316539
                FILE: Number of bytes written=4103742
                FILE: Number of read operations=0
```



Code Snippets for mapper.py & reducer.py (Word Count)



```
#!/usr/bin/env python
"""mapper.py"""

import sys

# input comes from STDIN (standard input)
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()
    # split the line into words
    words = line.split()
    # increase counters
    for word in words:
        # write the results to STDOUT (standard output);
        # what we output here will be the input for the
        # Reduce step, i.e. the input for reducer.py
        #
        # tab-delimited; the trivial word count is 1
        print '%s\t%s' % (word, 1)
```



```
#!/usr/bin/env python
import sys

# Create a dictionary to map words to counts
wordcount = {}

# Get input from stdin
for line in sys.stdin:
    #Remove spaces from beginning and end of the line
    line = line.strip()

    # parse the input from mapper.py
    word, count = line.split('\t', 1)
    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        continue

    try:
        wordcount[word] = wordcount[word]+count
    except:
        wordcount[word] = count

# Write the tuples to stdout
# Currently tuples are unsorted
for word in wordcount.keys():
    print '%s\t%s'% ( word, wordcount[word] )
```

Top-10 Words Word count

Twitter

cricket	16145
golf	12281
basketball	4426
tiger	4051
ipl	3654
game	3271
play	2974
win	2966
baseball	2876
one	2826

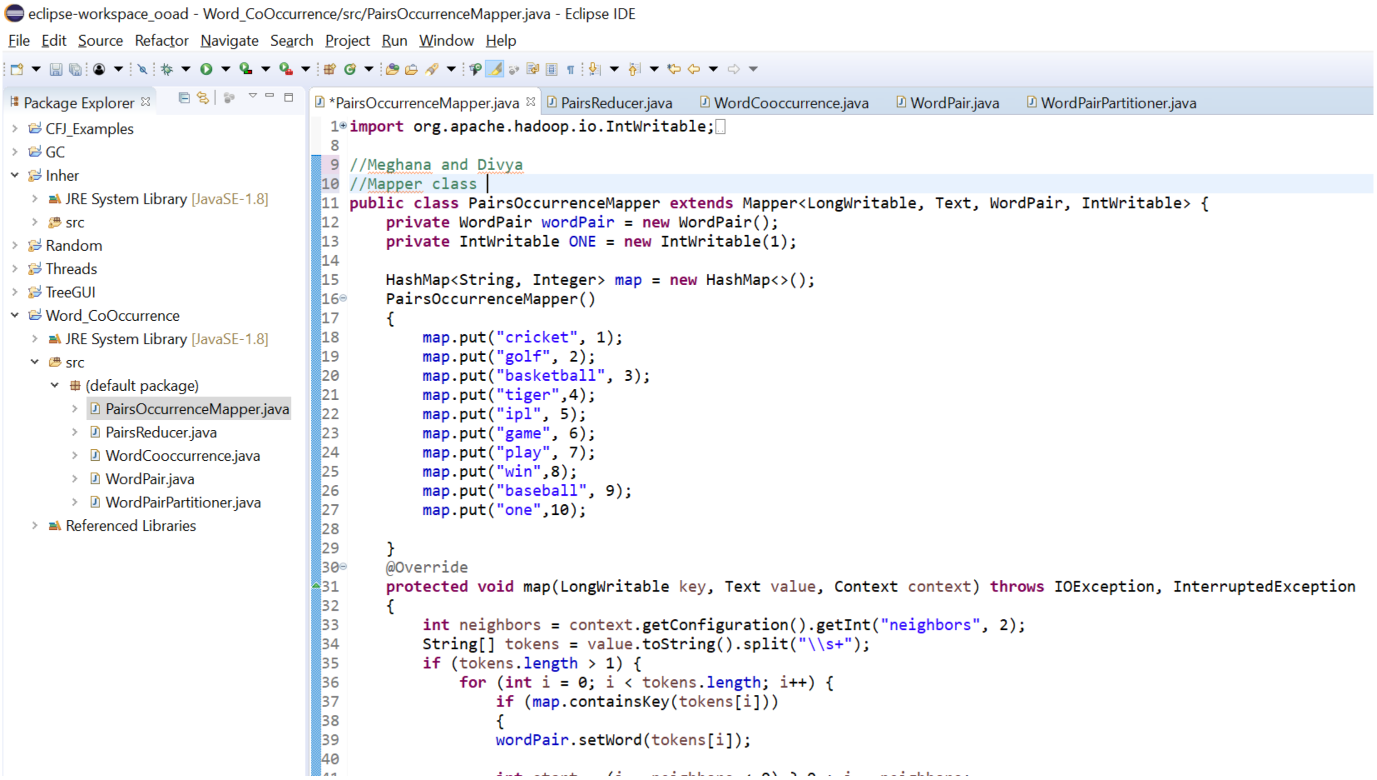
NYTimes

woods	1537
players	1501
team	1447
first	1367
game	1122
last	989
williams	957
year	902
hockey	880
play	854

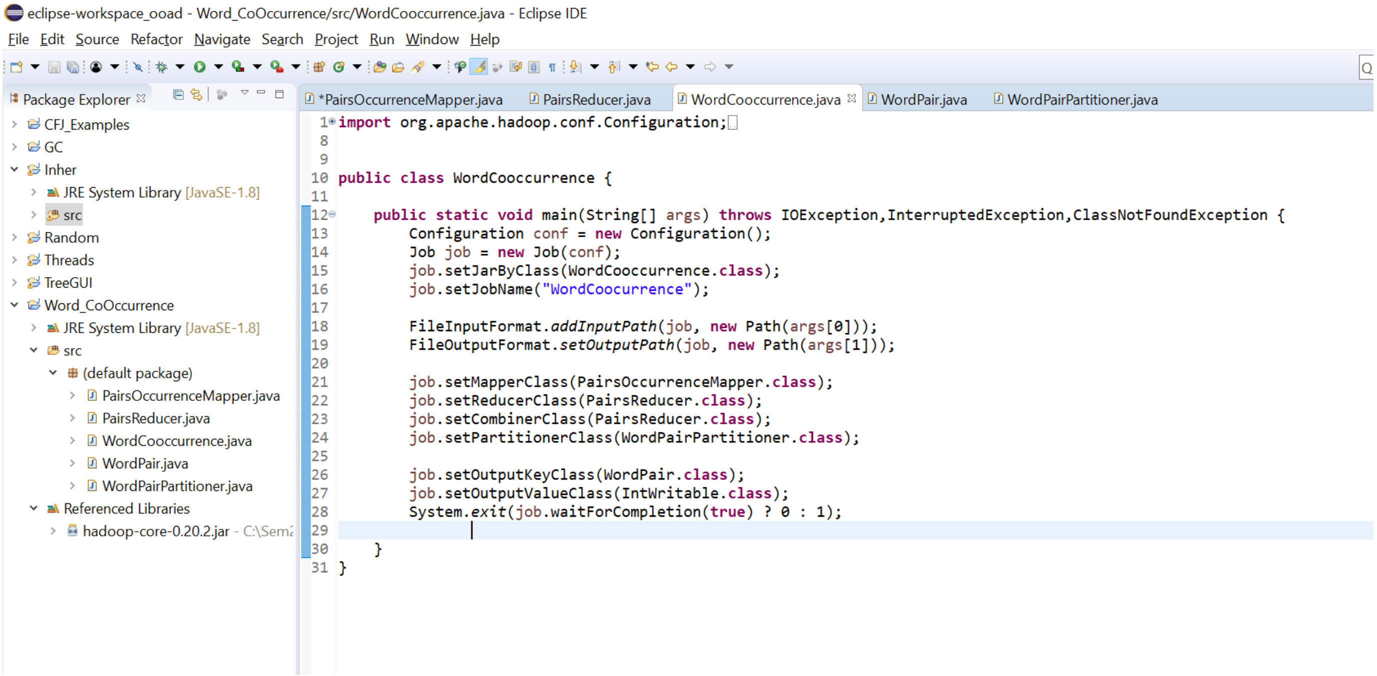
Common Crawl

game	7721
highlight	6366
sport	5645
virginia	5225
date	4721
time	4617
point	4589
win	4308
score	4204
team	4176

Code Snippets for mapper & reducer (Word Co-occurrence)

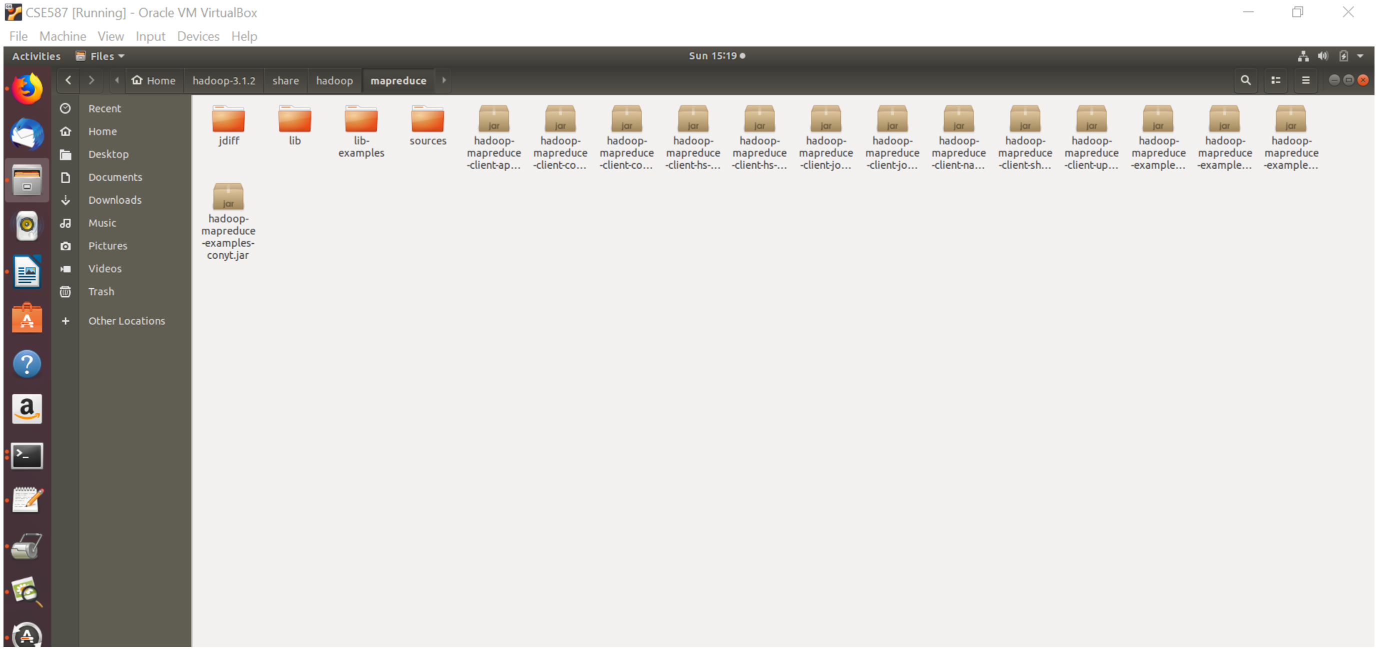

eclipse-workspace_ooad - Word_CoOccurrence/src/PairsOccurrenceMapper.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
CFJ_Examples
GC
Inher
JRE System Library [JavaSE-1.8]
src
Random
Threads
TreeGUI
Word_CoOccurrence
JRE System Library [JavaSE-1.8]
src
(default package)
PairsOccurrenceMapper.java
PairsReducer.java
WordCooccurrence.java
WordPair.java
WordPairPartitioner.java
Referenced Libraries
*PairsOccurrenceMapper.java
PairsReducer.java
WordCooccurrence.java
WordPair.java
WordPairPartitioner.java

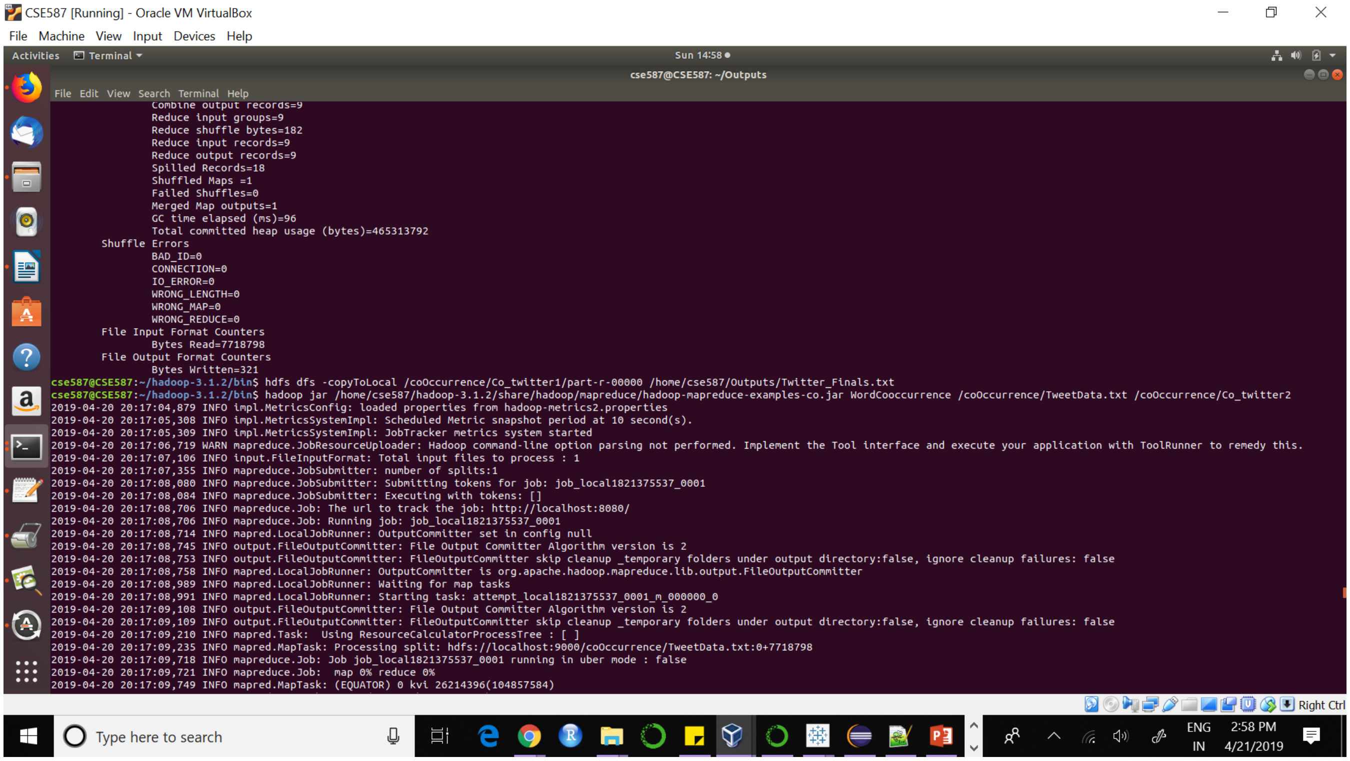
```
import org.apache.hadoop.io.IntWritable;□

//Meghana and Divya
//Mapper class
public class PairsOccurrenceMapper extends Mapper<LongWritable, Text, WordPair, IntWritable> {
    private WordPair wordPair = new WordPair();
    private IntWritable ONE = new IntWritable(1);

    HashMap<String, Integer> map = new HashMap<>();
    PairsOccurrenceMapper()
    {
        map.put("cricket", 1);
        map.put("golf", 2);
        map.put("basketball", 3);
        map.put("tiger",4);
        map.put("ipl", 5);
        map.put("game", 6);
        map.put("play", 7);
        map.put("win",8);
        map.put("baseball", 9);
        map.put("one",10);

    }
    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException
    {
        int neighbors = context.getConfiguration().getInt("neighbors", 2);
        String[] tokens = value.toString().split("\\s+");
        if (tokens.length > 1) {
            for (int i = 0; i < tokens.length; i++) {
                if (map.containsKey(tokens[i]))
                {
                wordPair.setWord(tokens[i]);

                int start = (i - neighbors < 0) ? 0 : i - neighbors;
```


eclipse-workspace_ooad - Word_CoOccurrence/src/WordCooccurrence.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
CFJ_Examples
GC
Inher
JRE System Library [JavaSE-1.8]
src
Random
Threads
TreeGUI
Word_CoOccurrence
JRE System Library [JavaSE-1.8]
src
(default package)
PairsOccurrenceMapper.java
PairsReducer.java
WordCooccurrence.java
WordPair.java
WordPairPartitioner.java
Referenced Libraries
hadoop-core-0.20.2.jar - C:\Sem2
*PairsOccurrenceMapper.java
PairsReducer.java
WordCooccurrence.java
WordPair.java
WordPairPartitioner.java

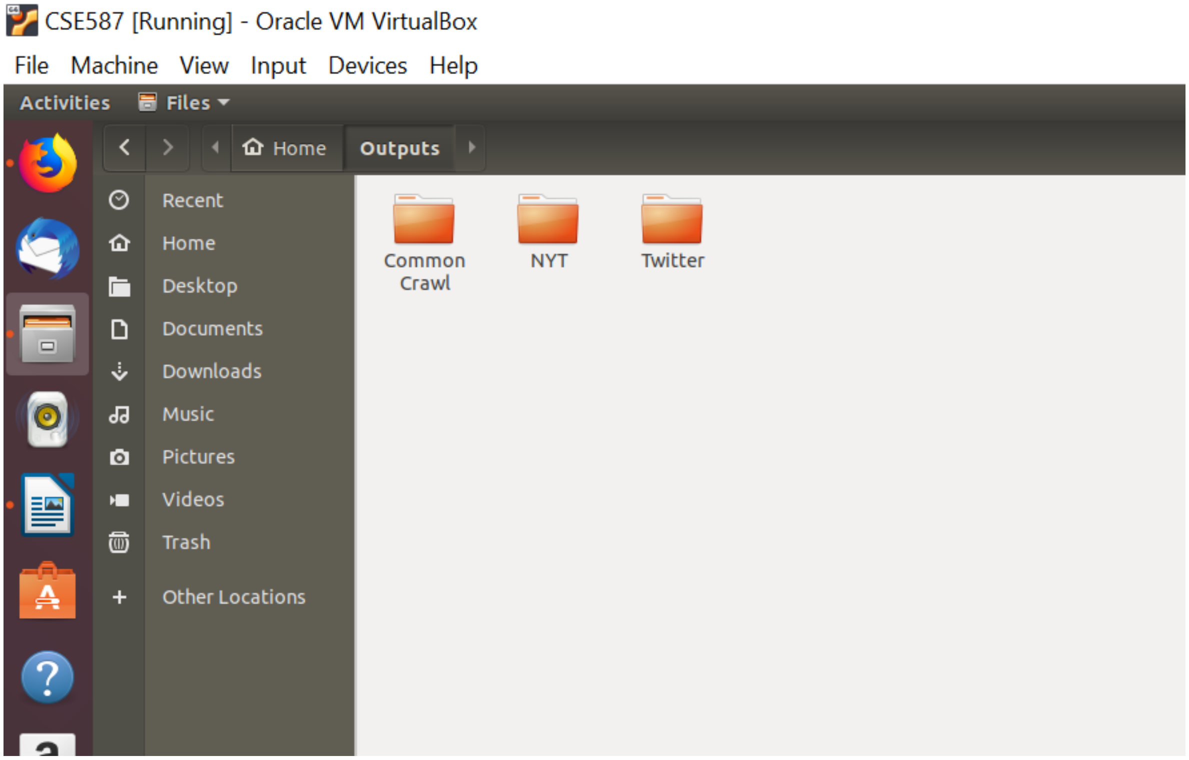
```
import org.apache.hadoop.conf.Configuration;□


public class WordCooccurrence {

    public static void main(String[] args) throws IOException,InterruptedException,ClassNotFoundException {
        Configuration conf = new Configuration();
        Job job = new Job(conf);
        job.setJarByClass(WordCooccurrence.class);
        job.setJobName("WordCoocurrence");

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.setMapperClass(PairsOccurrenceMapper.class);
        job.setReducerClass(PairsReducer.class);
        job.setCombinerClass(PairsReducer.class);
        job.setPartitionerClass(WordPairPartitioner.class);

        job.setOutputKeyClass(WordPair.class);
        job.setOutputValueClass(IntWritable.class);
        System.exit(job.waitForCompletion(true) ? 0 : 1);

    }
}
```

Converted the following code into jar files for each data source and ran on Virtual Machine. The following screenshots to run the code is as follows.


CSE587 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Files
Sun 15:19
Home
hadoop-3.1.2
share
hadoop
mapreduce
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
jdiff
lib
lib-examples
sources
hadoop-mapreduce-client-ap...
hadoop-mapreduce-client-co...
hadoop-mapreduce-client-co...
hadoop-mapreduce-client-hs-...
hadoop-mapreduce-client-hs-...
hadoop-mapreduce-client-jo...
hadoop-mapreduce-client-jo...
hadoop-mapreduce-client-na...
hadoop-mapreduce-client-sh...
hadoop-mapreduce-client-up...
hadoop-mapreduce-example...
hadoop-mapreduce-example...
hadoop-mapreduce-example...
hadoop-mapreduce-examples-conyt.jar


CSE587 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Sun 14:58
cse587@CSE587: ~/Outputs
File Edit View Search Terminal Help
Right Ctrl
Type here to search
ENG
IN
2:58 PM
4/21/2019

```
                Combine output records=9
                Reduce input groups=9
                Reduce shuffle bytes=182
                Reduce input records=9
                Reduce output records=9
                Spilled Records=18
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=96
                Total committed heap usage (bytes)=465313792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=7718798
        File Output Format Counters
                Bytes Written=321
cse587@CSE587:~/hadoop-3.1.2/bin$ hdfs dfs -copyToLocal /coOccurrence/Co_twitter1/part-r-00000 /home/cse587/Outputs/Twitter_Finals.txt
cse587@CSE587:~/hadoop-3.1.2/bin$ hadoop jar /home/cse587/hadoop-3.1.2/share/hadoop/mapreduce/hadoop-mapreduce-examples-co.jar WordCooccurrence /coOccurrence/TweetData.txt /coOccurrence/Co_twitter2
2019-04-20 20:17:04,879 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2019-04-20 20:17:05,308 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2019-04-20 20:17:05,309 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2019-04-20 20:17:06,719 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2019-04-20 20:17:07,106 INFO input.FileInputFormat: Total input files to process : 1
2019-04-20 20:17:07,355 INFO mapreduce.JobSubmitter: number of splits:1
2019-04-20 20:17:08,080 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1821375537_0001
2019-04-20 20:17:08,084 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-04-20 20:17:08,706 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2019-04-20 20:17:08,706 INFO mapreduce.Job: Running job: job_local1821375537_0001
2019-04-20 20:17:08,714 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2019-04-20 20:17:08,745 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-04-20 20:17:08,753 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2019-04-20 20:17:08,758 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2019-04-20 20:17:08,989 INFO mapred.LocalJobRunner: Waiting for map tasks
2019-04-20 20:17:08,991 INFO mapred.LocalJobRunner: Starting task: attempt_local1821375537_0001_m_000000_0
2019-04-20 20:17:09,108 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-04-20 20:17:09,109 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2019-04-20 20:17:09,210 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2019-04-20 20:17:09,235 INFO mapred.MapTask: Processing split: hdfs://localhost:9000/coOccurrence/TweetData.txt:0+7718798
2019-04-20 20:17:09,718 INFO mapreduce.Job: Job job_local1821375537_0001 running in uber mode : false
2019-04-20 20:17:09,721 INFO mapreduce.Job:  map 0% reduce 0%
2019-04-20 20:17:09,749 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
```



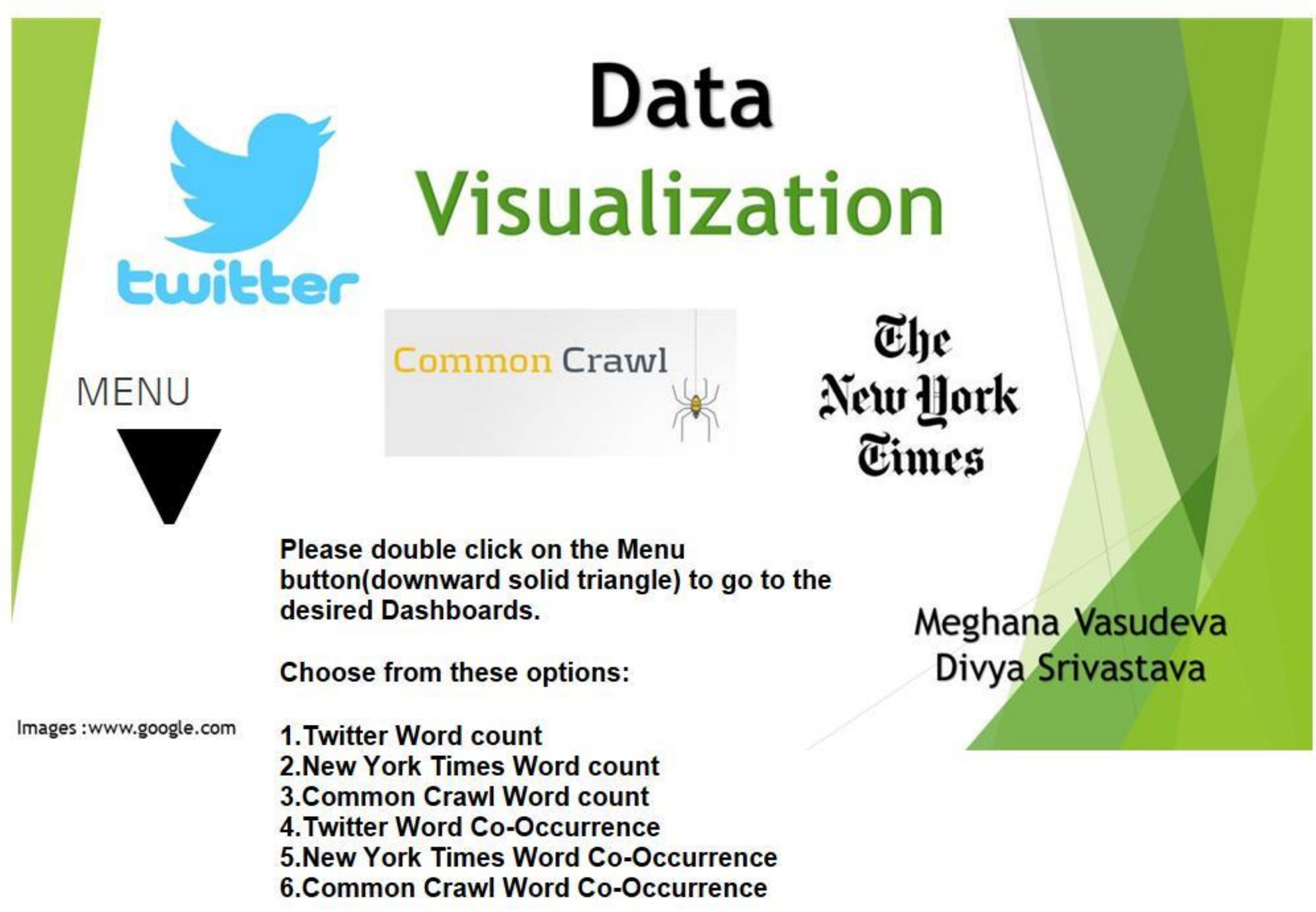
Top 10 words Co-Occurrence

New York Times		Twitter		Common Crawl	
Data	Count	Data	Count	Data	Count
team, year	25	basketball ,game	238	year, game	6
hockey, play	26	game, baseball	261	hockey , play	8
team, players	26	play ,basketball	265	play , game	9
team, first	31	game ,cricket	340	team, last	11
play,game	34	play ,cricket	409	play , first	14
woods, first	43	play,golf	532	last , game	39
players, hockey	48	golf ,game	607	first , year	55
first, game	55	tiger,golf	875	last ,year	55
team, hockey	68	ipl,cricket	980	first ,team	58
year, last	208	ipl,ipl	1092	game ,first	94

Part – 3:

Visualization
Tableau Installation:

1. Downloaded the setup for student version from https://www.tableau.com/academic/students
2. Got the product key and registered for it.
3. Ran the server setup
4. Configured the essential Tableau server settings
5. Set the authentication type
6. Set run as service account
7. Set the port and continue configuration
8. Created a tableau server administrator user



Data Visualization

Please double click on the Menu button(downward solid triangle) to go to the desired Dashboards.

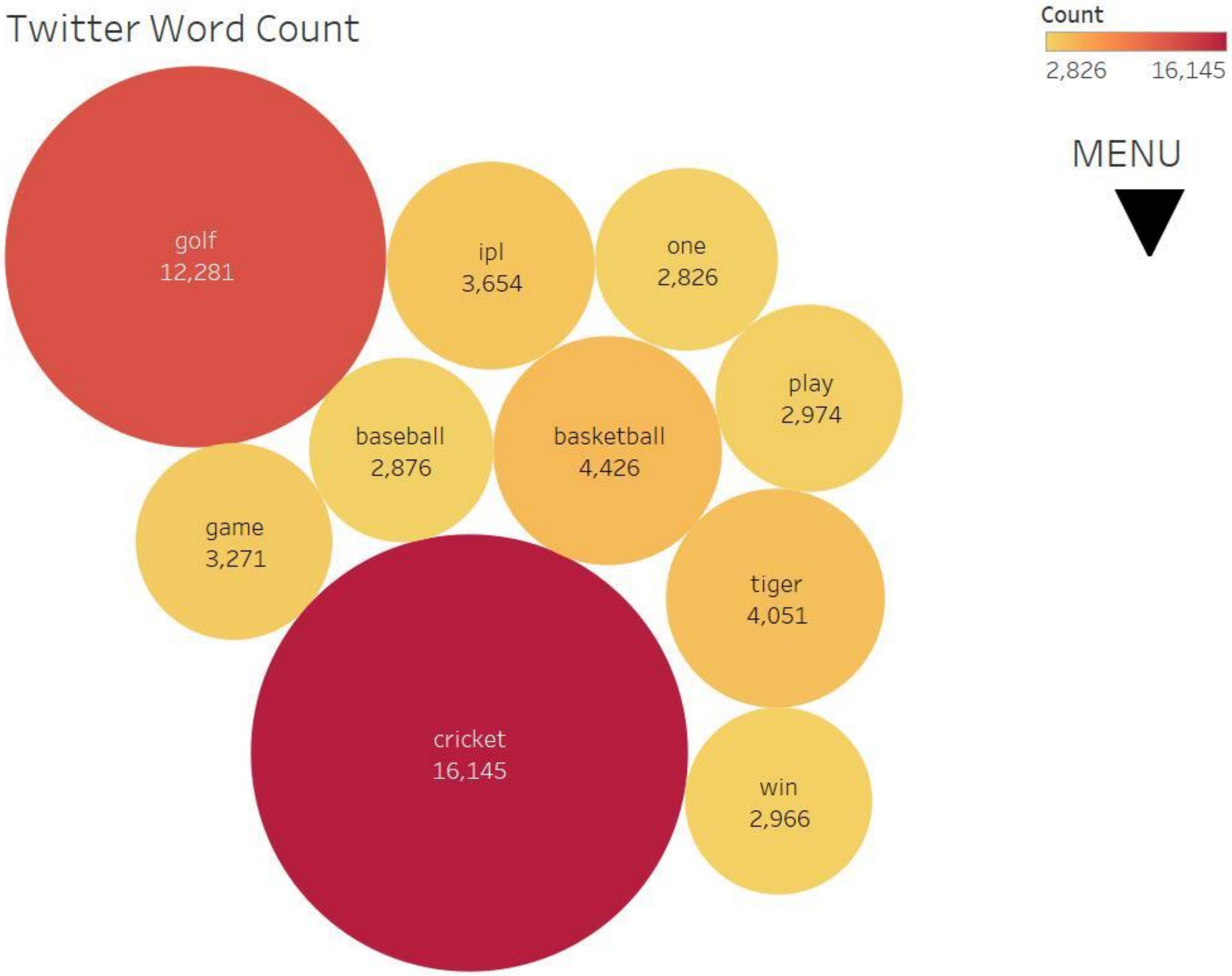
Choose from these options:

1.Twitter Word count
2.New York Times Word count
3.Common Crawl Word count
4.Twitter Word Co-Occurrence
5.New York Times Word Co-Occurrence
6.Common Crawl Word Co-Occurrence

Meghana Vasudeva
Divya Srivastava

Images :www.google.com

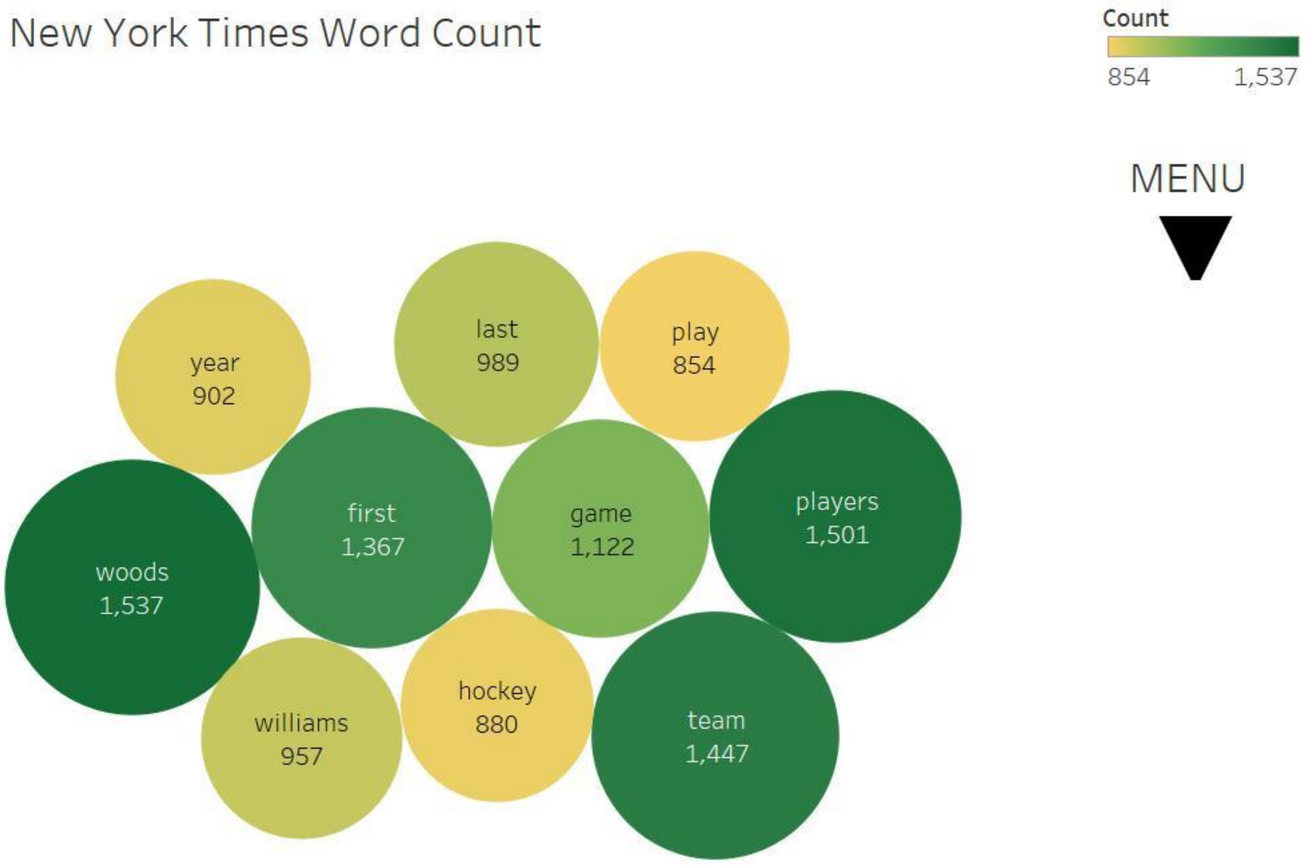
This is our menu page where we can choose different visualization using the dropdown button.



Analysis-

This is the bubble visualization for top 10 words for Twitter. The color here shows the count strength shown in the count section on the right side. We have Menu button in each dashboard to navigate to all other dashboards.

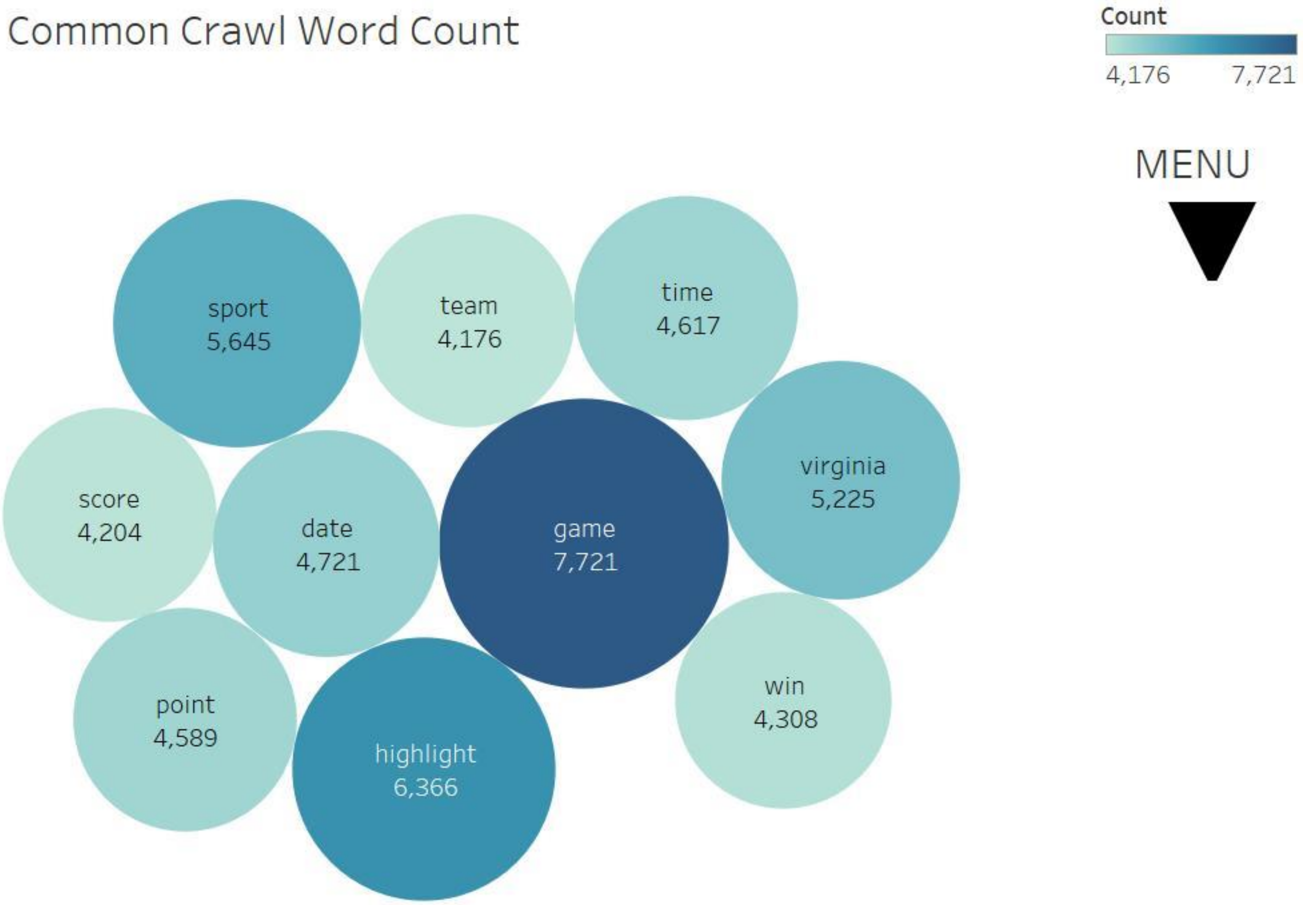
Here the word with the highest count is cricket and the least is one.



Analysis-

This is the bubble visualization for top 10 words for New York Times. The color here shows the count strength shown in the count section on the right side. We have Menu button in each dashboard to navigate to all other dashboards.

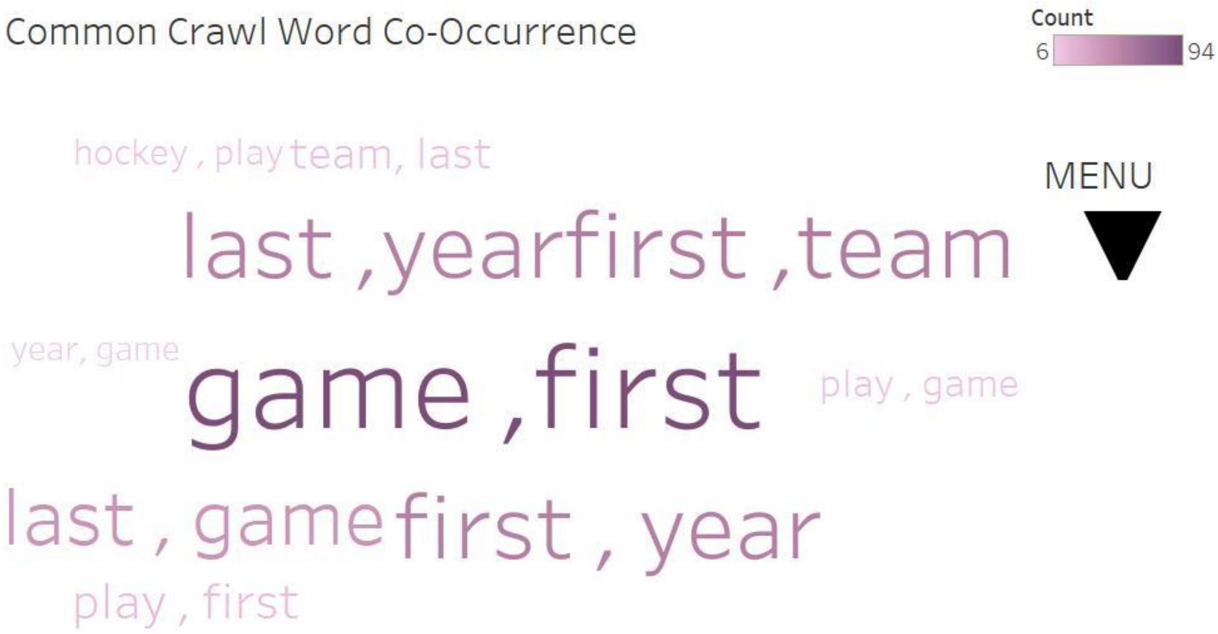
Here the word with the highest count is woods and the least is play.



Analysis-

This is the bubble visualization for top 10 words for Common Crawl. The color here shows the count strength shown in the count section on the right side. We have Menu button in each dashboard to navigate to all other dashboards.

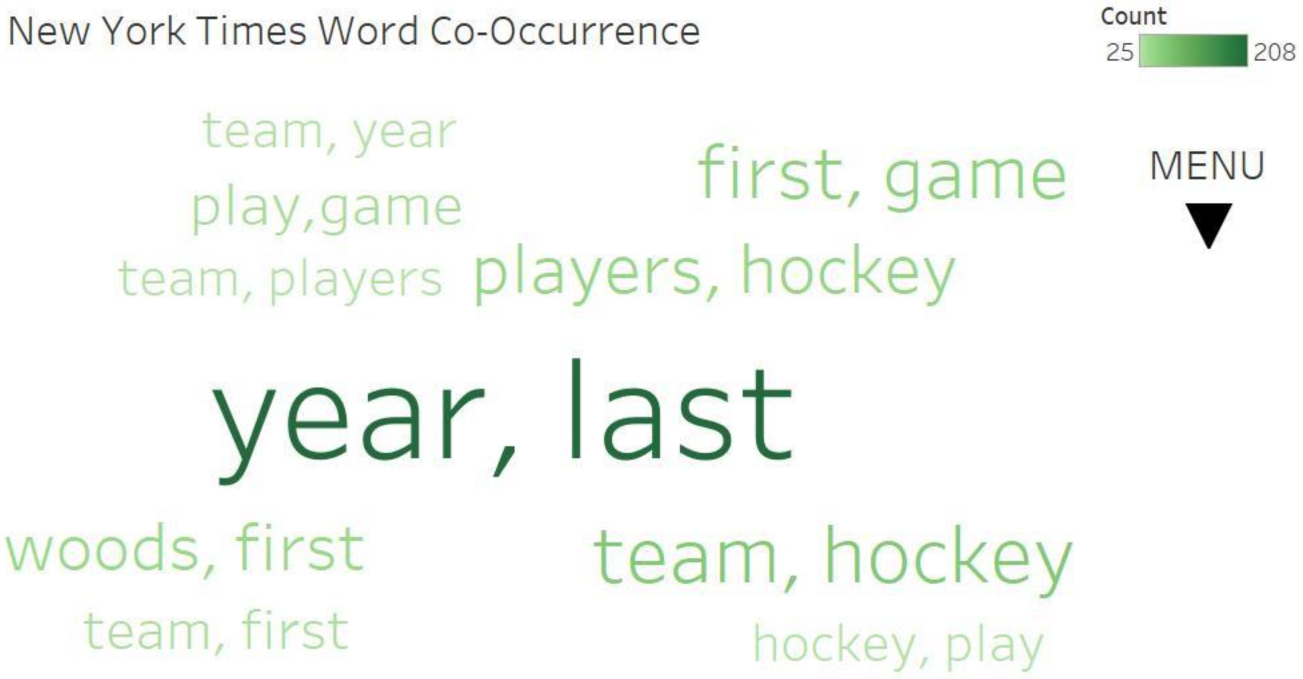
Here the word with the highest count is game and the least is team.



Analysis-

This is the Textual visualization for top 10 co-occurrence words for Common Crawl. The color here shows the count strength shown in the count section on the right side. We have Menu button in each dashboard to navigate to all other dashboards.

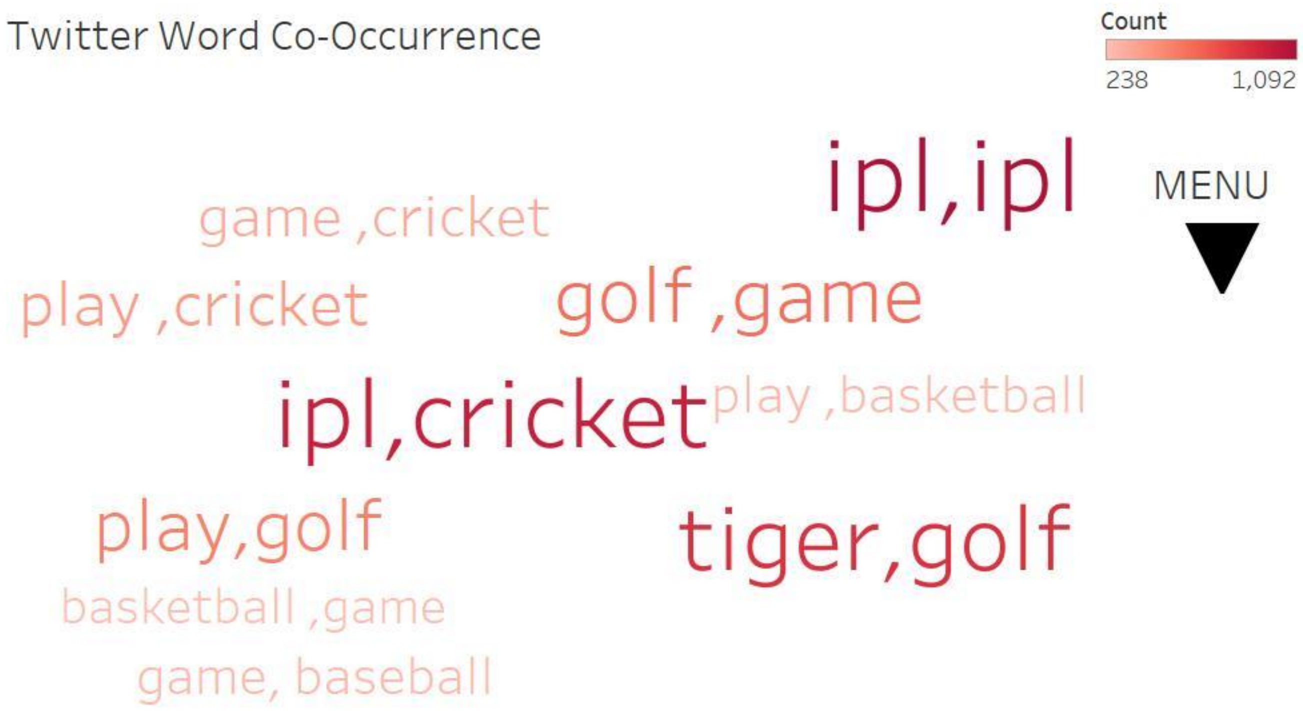
Here the co-occurrence words with the highest count are game and first and the co-occurrence words with the least count are year and game.



Analysis-

This is the Textual visualization for top 10 co-occurrence words for New York Times. The color here shows the count strength shown in the count section on the right side. We have Menu button in each dashboard to navigate to all other dashboards.

Here the co-occurrence words with the highest count are year and last and the co-occurrence words with the least count are team and year.



Analysis-

This is the Textual visualization for top 10 co-occurrence words for Twitter. The color here shows the count strength shown in the count section on the right side. We have Menu button in each dashboard to navigate to all other dashboards.

Here the co-occurrence words with the highest count are ipl and cricket and the co-occurrence words with the least count are basketball and game.

Link to published tableau: https://us-east-1.online.tableau.com/#/site/divyasrivastava/workbooks

We have published the whole online and the above is the link… Please login to view this..

Directory Structure:

- **Part1**
    - **Code**
        - **Common Crawl**
        - **NYT**
        - **Twitter**
    - **Data**
        - **Common Crawl**
        - **NYT**
        - **Twitter**
- **Part2**
    - **Code**
        - **Co-occurrence**
        - **Word Count**
    - **Data**
        - **Common Crawl**
        - **NYT**
        - **Twitter**
    - **Screenshots**
- **Part3**
        - **Images**
        - **Workbook**
        - **Data folder**
- **Report**
- **readme**

Demo Video link: The video is made by an online software and hence has a watermark. Please ignore watermark.
Meghana Vasudeva's UB BOX:**https://buffalo.box.com/s/x6jc9nrkv4rs5gtp7dkaw68tz9gkkjcq**
Divya Srivastava's UB BOX: **https://buffalo.box.com/s/t72kq1b6zogahmzfd175kqr57028985r**

Conclusion:

- Automated data collection from multiple sources using the APIs offered by the businesses.
- Explained the importance of evaluating the reliability of data. Applied classical big data analytical methods: MapReduce for word count word occurrence.
- Worked on Hadoop and HDFS and processed the data using big data algorithms.
- Learnt a high level language-based data analysis by exploring Python as data processing language.
- Applied modern visualization methods and disseminate results using the web/mobile interface.

References:

1. Installation steps for VM: https://www.virtualbox.org/
2. Install steps for Tableau: https://www.tableau.com/academic/students
3. Steps for generating the twitter API: https://developer.twitter.com/
4. Steps for generating the New York Times API: https://developer.nytimes.com/docs/articlesearch-product/1/overview