LAB 2

Data Aggregation, Big Data Analysis and Visualization

Group Members:
Yogesh Sawant
Amit Banerjee

Abstract

This project is based on the concepts of data aggregation, analysis of collected big data using the Map-Reduce algorithm and development of a data product using the visualization tool of Tableau. For the first part, we are expected to collect data from a variety of sources namely an opinion based social media Twitter, New York Times as a research resource and Common Crawl as an archive of random crawled data. A topic which is relevant to the widespread masses of USA is chosen as the subject for data collection and Big data analysis as part of this project. For the second part of the project, we are expected to process all the collected data using the big data analysis method of Map reduce. This method has been chosen as part of the data analysis method to calculate the word count and word co-occurrence for such a large volume of data gathered in part one. For the final part of our project, we are expected to visualize our analysis in form of a word cloud signifying the top words which are discussed among all the data sources which were primarily selected as part of the data collection process. We aim to further analyze the differences in word clouds generated by different data sources with one another to understand how data is collected and displayed in each of the aforementioned data sources

1. Prototype Data Collection

As part of the data collection process, we decided to choose Sports as our main topic of interest. For easing the process of data collection, we decided to search for an equal amount of data for multiple sub topics which are relevant in the United States of America. The sub topics which have been chosen for data collection in this project are as follows: -

- Soccer
- Basketball
- Baseball
- American Football
- Ice Hockey

Data collection process for each of the required data sources is explained briefly in subsequent portions of the report.

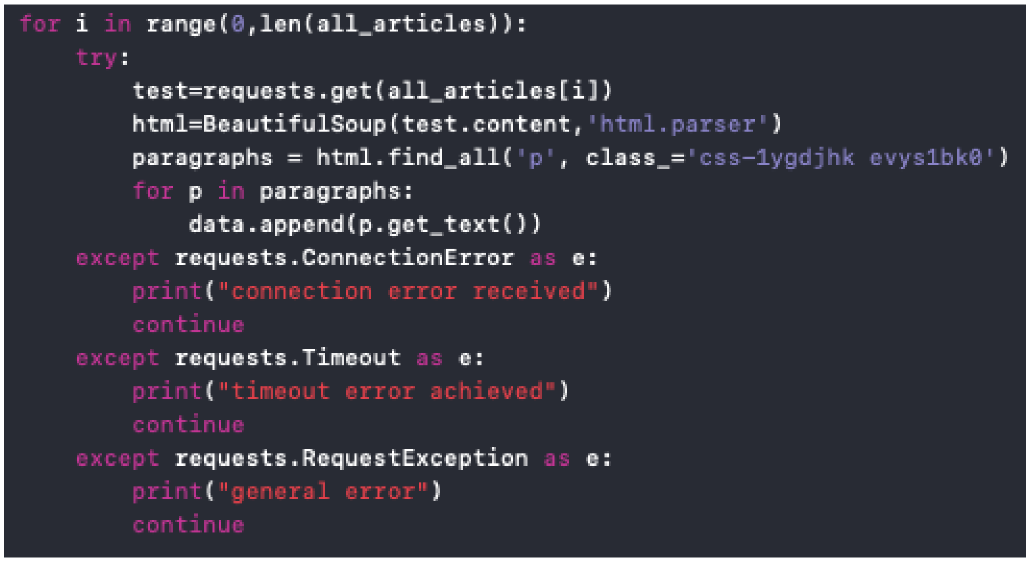
1.1 Twitter Data Collection:

For collection of tweet data using python as the programming language, we used the Tweepy wrapper library for accessing the Twitter API using our personal authorization keys. Since we are interested in data from only the current year, we set the begin date for searching tweets as first of January,2019. For keeping the data consistent, we mined for an equal amount of 5000 tweets for each subtopic. All the tweets were collected in such a manner that retweets were not present as part of our initial data dump.
After analyzing this initial dump of data, we realized that each tweet consisted of a hyperlink and various emoticons and hence we removed such abnormalities from our data, one tweet at a time. After this initial pre-processing on our data, we saved all tweets in a text formatted file. This file was read one tweet at a time and all forms of stop words were removed from the tweets using the

list of stop words provided by NLTK library. We also made use of the Snowball stemmer present inside NLTK for stemming all the words present inside a single tweet. This final collection of tweets was saved inside a new file namely processed_tweet.txt.

1.2 New York Times Data Collection:

For this phase of our project, we used the NYT Search API for collecting data for each of the same subtopics that were used in Twitter data collection process. All articles which were published in the current year were only desired as part of this step. A request to the NYT API was made with the desired subtopics as keywords, one word at a time. The response was filtered to obtain the relevant article URL's. We developed an algorithm for visiting each of the URL's and collecting all the elements which constituted the article web page. For scraping the article content in sections of paragraphs we used a python library called BeautifulSoup. This library provided us methods of scraping all the paragraph data effortlessly. The piece of code which performed the scraping functionality is shown below.



```
for i in range(0,len(all_articles)):
    try:
        test=requests.get(all_articles[i])
        html=BeautifulSoup(test.content,'html.parser')
        paragraphs = html.find_all('p', class_='css-1ygdjhk evys1bk0')
        for p in paragraphs:
            data.append(p.get_text())
    except requests.ConnectionError as e:
        print("connection error received")
        continue
    except requests.Timeout as e:
        print("timeout error achieved")
        continue
    except requests.RequestException as e:
        print("general error")
        continue
```

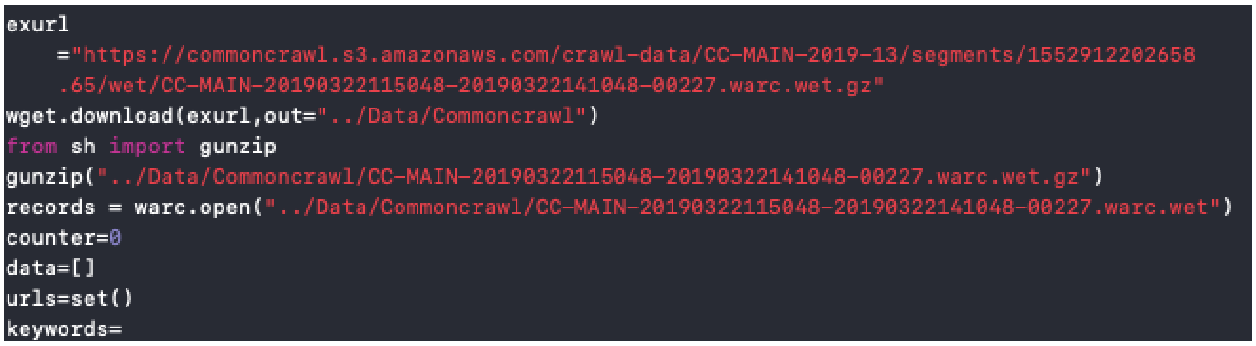
Fig 1: BeautifulSoup implementation

The next step was storing all the article data which had been separated by paragraphs into a text formatted file. This file was reopened for performing data pre-processing. Similar to the final step of Twitter data collection, stemming and removal of stop words was performed for the NYT data as well. The final NYT data was written to a file named processed_NytData.txt.

1.3 Common Crawl Data Collection:

Common crawl is a public repository of data collected from a wide range of websites. The data is stored in the form of multiple web archives which contain metadata about various articles. Some of the information present inside the metadata points to the domain of a website and their actual URL. The domain related websites are also scattered across various web archives(WARC) and it becomes a difficult task to access each WARC file and fetch data related to a particular main topic or sub topic. Hence data collected from common crawl consists of a lot of abnormalities. The size

of a single WARC file is in gigabytes and processing such volumes of data is inconvenient on local machines.
As a workaround to the problem, we downloaded the web text extracts(WET) March edition from the common crawl website. This file consists of all the URL's present in the WARC file along with the entire article data below the URL. Implementation of this step-in python is shown as a snippet in the figure below.



```
exurl
    ="https://commoncrawl.s3.amazonaws.com/crawl-data/CC-MAIN-2019-13/segments/1552912202658
    .65/wet/CC-MAIN-20190322115048-20190322141048-00227.warc.wet.gz"
wget.download(exurl,out="../Data/Commoncrawl")
from sh import gunzip
gunzip("../Data/Commoncrawl/CC-MAIN-20190322115048-20190322141048-00227.warc.wet.gz")
records = warc.open("../Data/Commoncrawl/CC-MAIN-20190322115048-20190322141048-00227.warc.wet")
counter=0
data=[]
urls=set()
keywords=
```

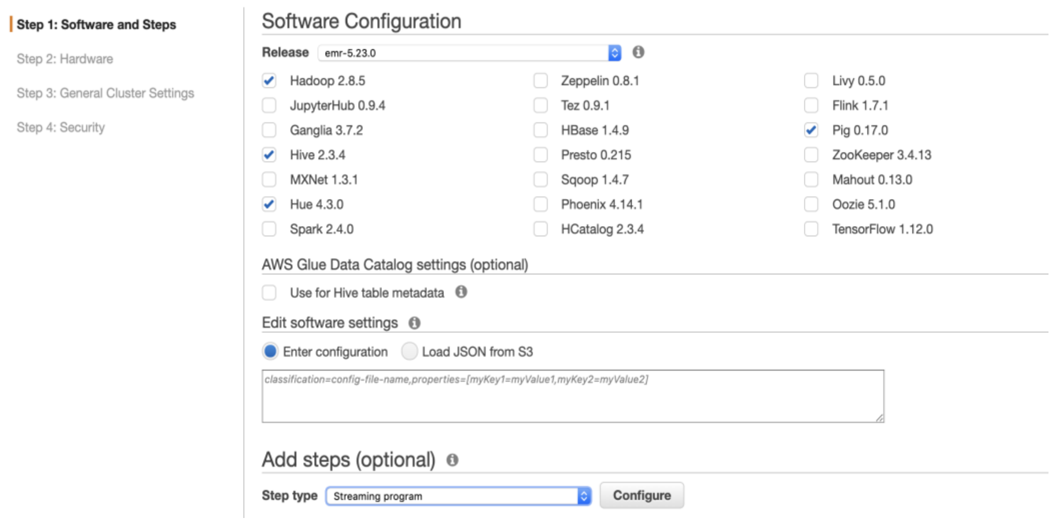
Fig 2: Downloading common crawl WET file

As the next step of data collection from common crawl, we check if content of each URL contains any of our subtopic words. We also check if the article data is in English. If both of the conditions are matched, we add the article data as part of our common crawl data. Once all the articles present in the WET file are traversed, we perform stemming and other pre-processing activities on the collected data. Since our topic keywords can be used in any context inside the common crawl article, we need to do some extra pre-processing for common crawl. For a particular paragraph, we also check if there is a word less than 3 characters. We omit such words and other special characters before sending the data as input to the mapper.

2. Big Data Infrastructure

The big data infrastructure is set up by making use of AWS. The infrastructure leverages Amazon S3 and Amazon Elastic MapReduce Services. The processed data, mapper and reducer files are stored in s3 bucket. The steps we performed in Amazon EMR are-

- Go to Amazon EMR, click on Create Cluster.
- Select Go to advanced options, present beside Quick Options at the start of the page.
- Select Software Configuration like Hadoop, Hive, Pig and Hue. Then in Add steps select "Streaming Program". Please refer figure 3 below for the same.


Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

Software Configuration

Release: emr-5.23.0

✓ Hadoop 2.8.5	Zeppelin 0.8.1	Livy 0.5.0
JupyterHub 0.9.4	Tez 0.9.1	Flink 1.7.1
Ganglia 3.7.2	HBase 1.4.9	✓ Pig 0.17.0
✓ Hive 2.3.4	Presto 0.215	ZooKeeper 3.4.13
MXNet 1.3.1	Sqoop 1.4.7	Mahout 0.13.0
✓ Hue 4.3.0	Phoenix 4.14.1	Oozie 5.1.0
Spark 2.4.0	HCatalog 2.3.4	TensorFlow 1.12.0

AWS Glue Data Catalog settings (optional)

Use for Hive table metadata

Edit software settings

[✓ Enter configuration] Load JSON from S3

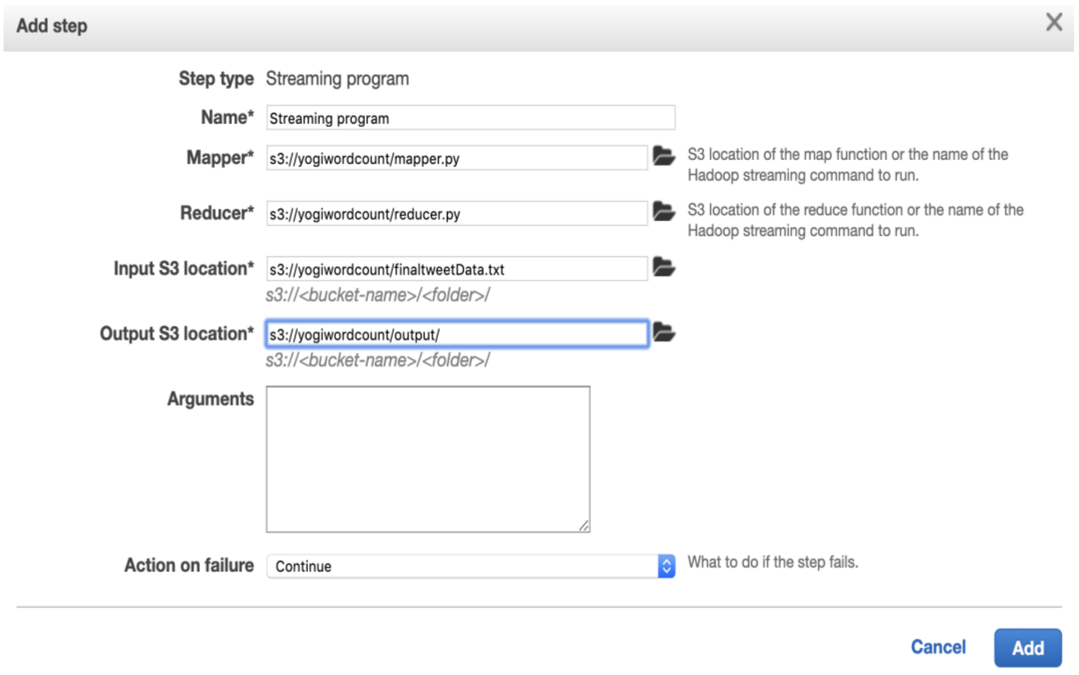
classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]

Add steps (optional)

Step type: Streaming program

Fig 3: AWS Cluster Software Configuration

- Select the location from S3 of mapper, reducer, input file and destination for output in S3 and click on “Add” and “Next”. Please refer figure 4 for reference.


Add step

Step type: Streaming program

Name*: Streaming program

Mapper*: s3://yogiwordcount/mapper.py — S3 location of the map function or the name of the Hadoop streaming command to run.

Reducer*: s3://yogiwordcount/reducer.py — S3 location of the reduce function or the name of the Hadoop streaming command to run.

Input S3 location*: s3://yogiwordcount/finaltweetData.txt — s3://<bucket-name>/<folder>/

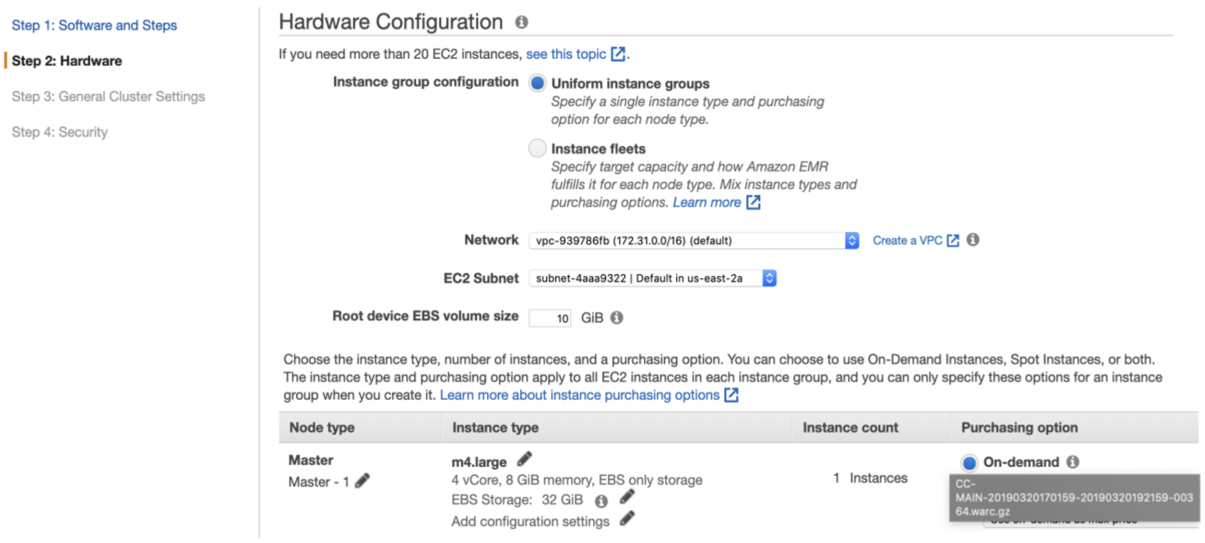
Output S3 location*: s3://yogiwordcount/output/ — s3://<bucket-name>/<folder>/

Arguments:

Action on failure: Continue — What to do if the step fails.

Fig 4: Stream Programming step

- Enter the hardware configurations as per the requirements and click on “Next”. Example: We are using instance of type m4.large with only one Master instance. Figure 5 provides a reference for the aforementioned step.



Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

Hardware Configuration

If you need more than 20 EC2 instances, see this topic.

Instance group configuration (●) **Uniform instance groups** *Specify a single instance type and purchasing option for each node type.*
( ) **Instance fleets** *Specify target capacity and how Amazon EMR fulfills it for each node type. Mix instance types and purchasing options. Learn more*

Network vpc-939786fb (172.31.0.0/16) (default) Create a VPC

EC2 Subnet subnet-4aaa9322 | Default in us-east-2a

Root device EBS volume size 10 GiB

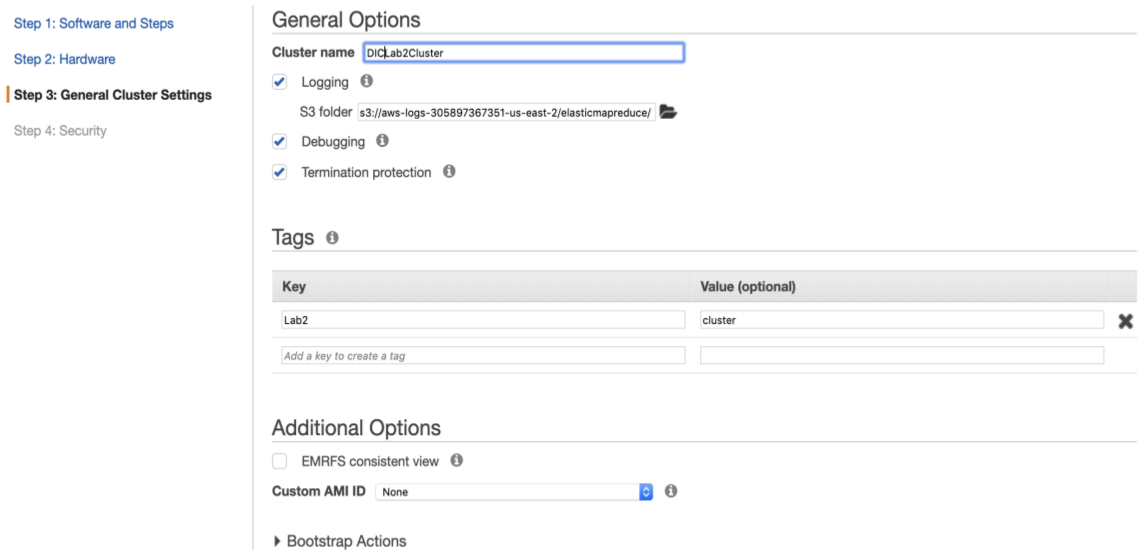
Choose the instance type, number of instances, and a purchasing option. You can choose to use On-Demand Instances, Spot Instances, or both. The instance type and purchasing option apply to all EC2 instances in each instance group, and you can only specify these options for an instance group when you create it. Learn more about instance purchasing options

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	**m4.large** 4 vCore, 8 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	(●) **On-demand**

CC-MAIN-20190320170159-20190320192159-00364.warc.gz

Fig 5: Hardware Configuration

- Then in General Cluster Settings give name and key value pair for the Cluster and click on “Next”. Please refer figure 6 for the same.



Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

General Options

Cluster name DICLab2Cluster

[✓] Logging

S3 folder s3://aws-logs-305897367351-us-east-2/elasticmapreduce/

[✓] Debugging

[✓] Termination protection

Tags

Key	Value (optional)
Lab2	cluster
Add a key to create a tag	

Additional Options

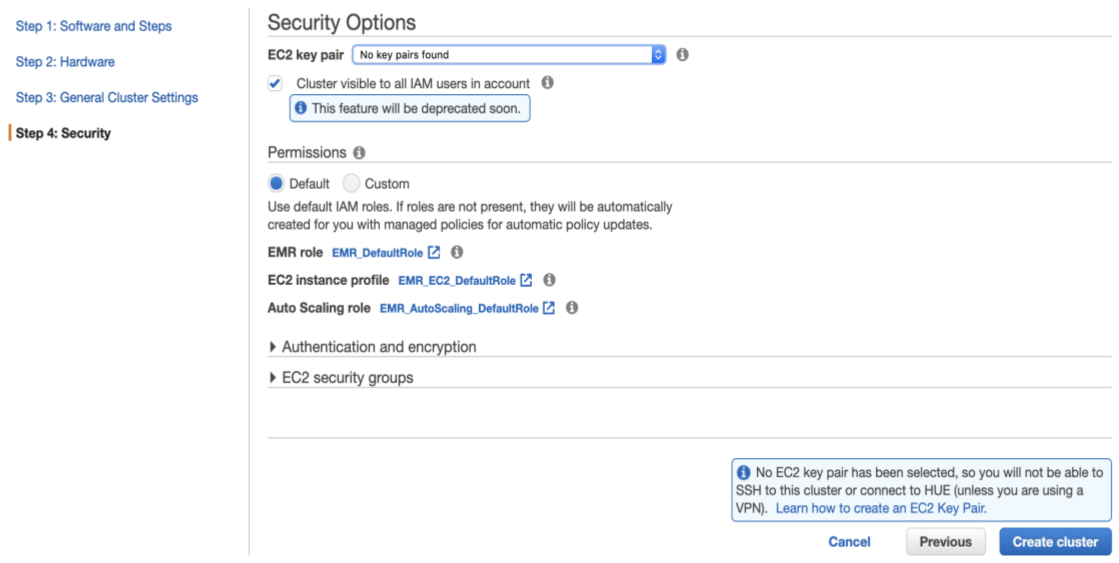
[ ] EMRFS consistent view

Custom AMI ID None

Bootstrap Actions

Fig 6: General Cluster Settings

- Select the Security options like the EC2 Key pair, IAM role and Security group settings which will to securely access the cluster instance remotely. Then click on current cluster. Please refer figure 7 for the same.


Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

Security Options

EC2 key pair No key pairs found

✓ Cluster visible to all IAM users in account

This feature will be deprecated soon.

Permissions

● Default ○ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole

EC2 instance profile EMR_EC2_DefaultRole

Auto Scaling role EMR_AutoScaling_DefaultRole

Authentication and encryption

EC2 security groups

No EC2 key pair has been selected, so you will not be able to SSH to this cluster or connect to HUE (unless you are using a VPN). Learn how to create an EC2 Key Pair.

Fig 7: Security settings

This creates the Hadoop big data infrastructure.

2.1 Mapper and Reducer for Word Count

All the steps written in this section are followed for each data source which has been used during the data collection phase.

2.1.1. Mapper:

In this task, we are process the Input data stored in S3 Bucket, by sending it paragraph-wise to the mapper which in turn generate intermediate key value pair(Word,1) for each word. In case of Twitter it is one tweet at a time and for New York Times and Common Crawl it is one paragraph at a time of an article.

2.1.2 Reducer

In this task the obtained intermediate key value pairs (Word, count) from Mapper which is processed to generate the output in alphabetically order of the words present in the input data, which is stored in a folder mentioned in Stream Program Step of Amazon EMR.

2.1.3 Generating Top 10 words

In this step, we run a python script on reducer output of previous step which generates top 10 words as an output of the word count exercise. The output is written to a text formatted file and used as an input for generating word cloud inside Tableau.

2.2. Mapper and Reducer for Word Co-occurrence

All the steps written in this section are followed for each data source which has been used during the data collection phase.

2.2.1. Mapper

In this task, we are explicitly mentioning the list of top 10 words in Mapper code. We process the input data present in S3 bucket, by sending it paragraph-wise to the mapper which in turn generates intermediate key value pairs (Word1-Word2, 1) of the words which are co-occurring with top10 words in a particular paragraph.

2.2.2. Reducer

In this task the obtained intermediate key value pairs (Word1-Word2, 1) from Mapper which is processed to generate the output in alphabetically order of the word pair present in the input data, which is stored in a folder mentioned in Stream Program Step of Amazon EMR.

2.3. Generating Top 10-word pair

In this step, we run a python script on reducer output of previous step which generates top 10 words as an output of the word co-occurrence exercise. The output is written to a text formatted file and used as an input for generating word cloud inside Tableau.

3. Visualization

As part of the visualization process, we use the top 10-word count and word co-occurrence text files generated by our python script during our big data analysis phase. Visualization is generated for each text file using Tableau desktop software. On inputting the text file into tableau, it automatically creates a table of counts and words as the data source to be used for generating a word cloud. A blank dashboard is generated with the name "Sheet 1" on the bottom of the screen. For generating the final word cloud, we drag the words dimension into "Text" tab located inside Marks. We also need to drag the count measures indicated by F2 into "Size" tab inside Marks. Finally, we drag the words dimension indicated by F1 into "Color" tab inside Marks.

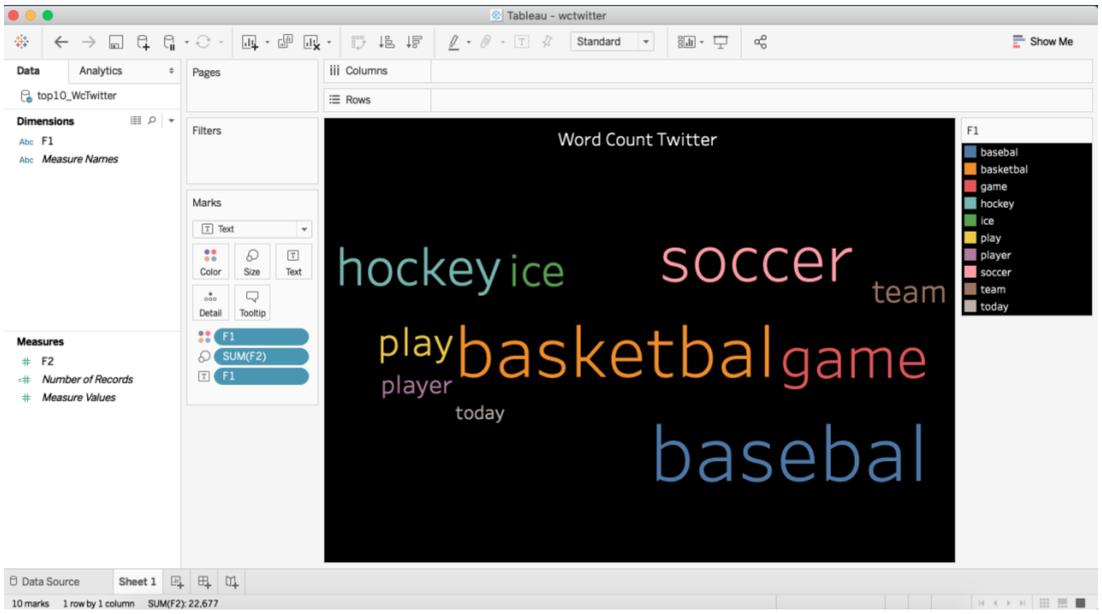


Fig 8: Creating a Word Cloud inside Tableau Desktop Software

4. Data Product

We created a static website using HTML, CSS, JavaScript and jQuery, which displays all the word clouds created with the Tableau desktop software. This website acts as a functional data product for our project.

We hosted this website by making use of Static Hosting feature of the S3 Bucket. Please find the below mentioned link of our website.

Link: http://websitelab2.s3-website-us-east-1.amazonaws.com/

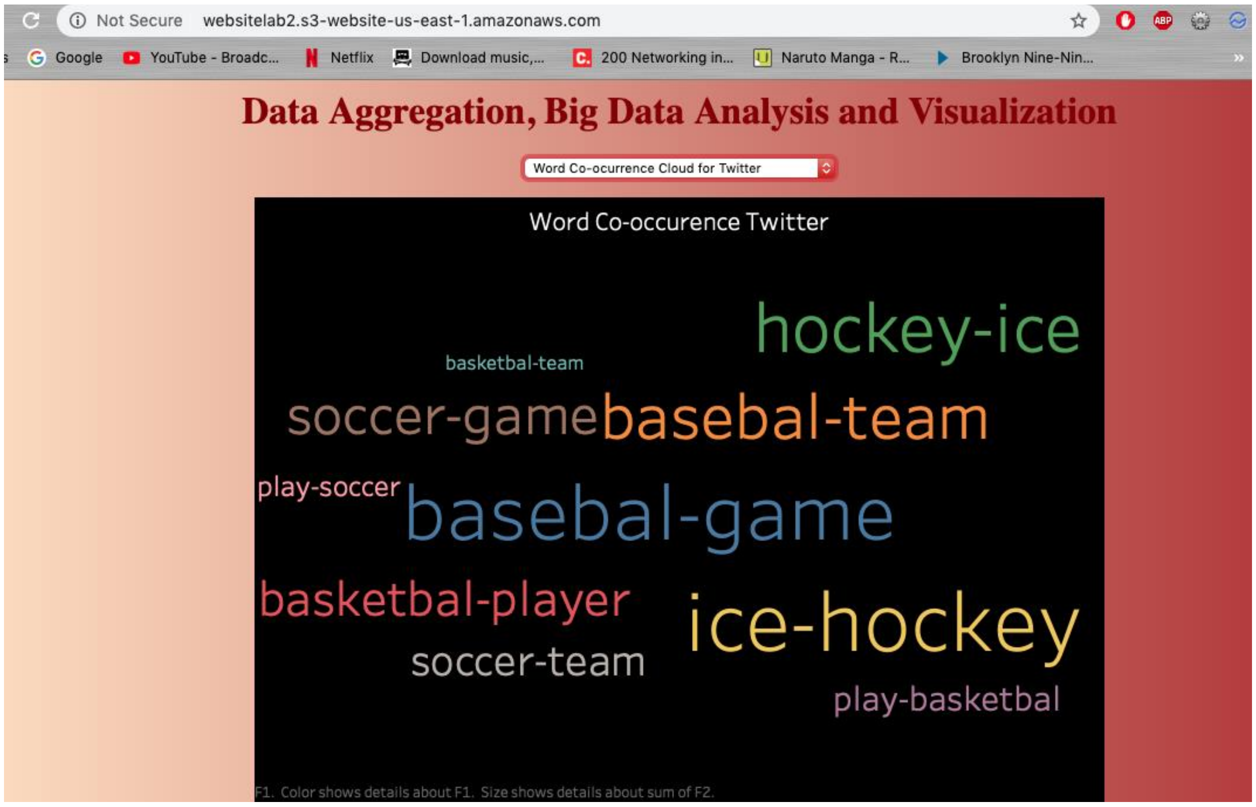

Not Secure websitelab2.s3-website-us-east-1.amazonaws.com
Google
YouTube - Broadc...
Netflix
Download music,...
200 Networking in...
Naruto Manga - R...
Brooklyn Nine-Nin...
Data Aggregation, Big Data Analysis and Visualization
Word Co-ocurrence Cloud for Twitter
Word Co-occurence Twitter
hockey-ice
basketbal-team
soccer-game
basebal-team
play-soccer
basebal-game
basketbal-player
ice-hockey
soccer-team
play-basketbal
F1. Color shows details about F1. Size shows details about sum of F2.

Fig 9: Screenshot of Hosted Data Product