Assignment 2

Data Management & Warehousing Analytics (CSCI 5408)

Shruthi Kalasapura Ramesh
B00822766

A. Cluster Setup:

- Created Amazon Web Services account as per the instruction provided in lab.
- Installed Apache Spark on AWS EC2.
- Installed MongoDB on EC2 instance to store data.

The above steps are performed as per the instructions provided in lab sessions.

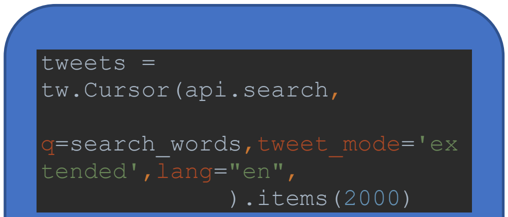
B. Data Extraction & Transformation:

Steps involved in extraction of twitter data and articles are:

i. Twitter Data Extraction & Transformation:

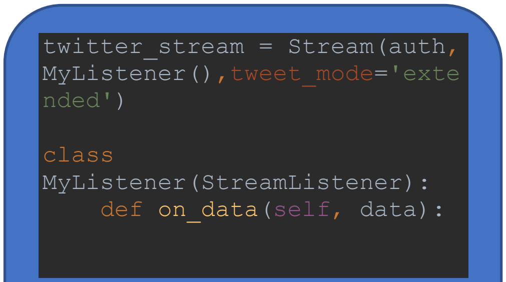
- A twitter developer account is created to access all development features of twitter.
- Twitter search and stream API (figure 1 & 2) are accessed using Tweepy (figure3) library in python [1].
- Search(2000 tweets) and stream(1500 tweets) tweets are extracted from API, each tweet is cleaned (figure 4) by removing url, special characters, emoticons, non-ASCII codes and metacharacters.
- And then the tweet text is stored in JSON file (figure 5) along with other information such as post id, metadata, location, date, text and location of retweets.
- Two JSON files with search and stream data are imported in MongoDB of EC2 instance using the following command [2].

```
mongoimport --db <database> --collection <collection>  --drop  --file <file path>
```



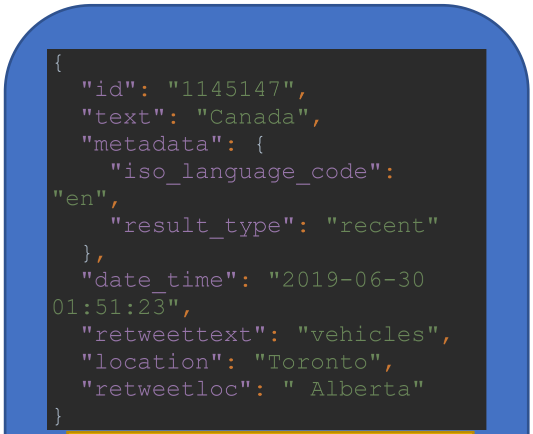
```
tweets =
tw.Cursor(api.search,

q=search_words,tweet_mode='ex
tended',lang="en",
                  ).items(2000)
```

Figure 1 : Search API



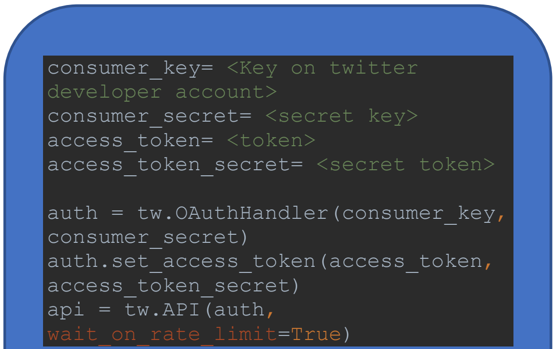
```
twitter_stream = Stream(auth,
MyListener(),tweet_mode='exte
nded')

class
MyListener(StreamListener):
    def on_data(self, data):
```

Figure 2 : Stream API



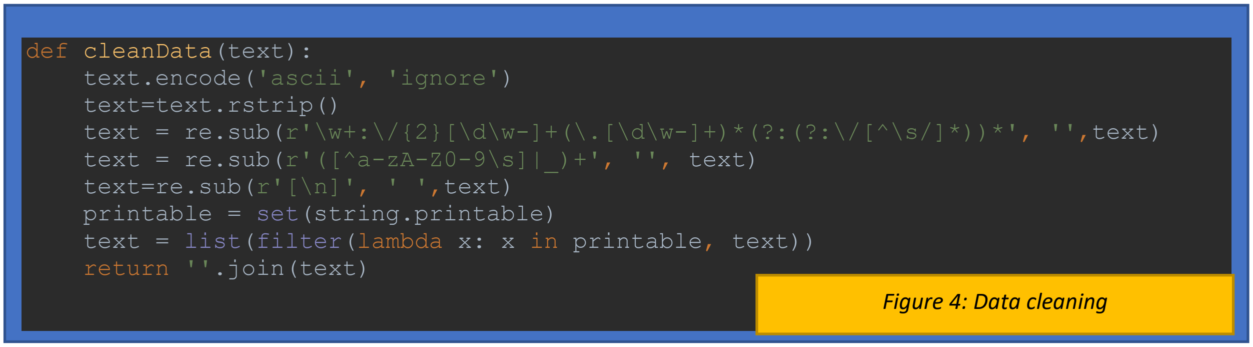
```
{
  "id": "1145147",
  "text": "Canada",
  "metadata": {
    "iso_language_code":
"en",
    "result_type": "recent"
  },
  "date_time": "2019-06-30
01:51:23",
  "retweettext": "vehicles",
  "location": "Toronto",
  "retweetloc": " Alberta"
}
```

Figure 5: Search and Stream data JSON format



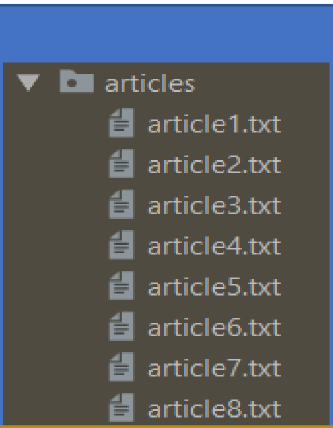
```
consumer_key= <Key on twitter
developer account>
consumer_secret= <secret key>
access_token= <token>
access_token_secret= <secret token>

auth = tw.OAuthHandler(consumer_key,
consumer_secret)
auth.set_access_token(access_token,
access_token_secret)
api = tw.API(auth,
wait_on_rate_limit=True)
```

Figure 3: Authorization to twitter API using tweepy



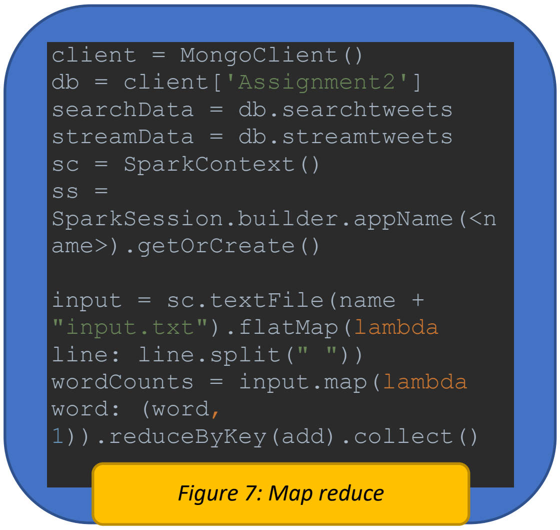
```
def cleanData(text):
    text.encode('ascii', 'ignore')
    text=text.rstrip()
    text = re.sub(r'\w+:\/{2}[\d\w-]+(\.[\d\w-]+)*(?:(?:\/[^\s/]*))*', '',text)
    text = re.sub(r'([^a-zA-Z0-9\s]|_)+', '', text)
    text=re.sub(r'[\n]', ' ',text)
    printable = set(string.printable)
    text = list(filter(lambda x: x in printable, text))
    return ''.join(text)
```

Figure 4: Data cleaning



```
▼ articles
      article1.txt
      article2.txt
      article3.txt
      article4.txt
      article5.txt
      article6.txt
      article7.txt
      article8.txt
```

Figure 6: Article files from REUTERS files



```
client = MongoClient()
db = client['Assignment2']
searchData = db.searchtweets
streamData = db.streamtweets
sc = SparkContext()
ss =
SparkSession.builder.appName(<n
ame>).getOrCreate()

input = sc.textFile(name +
"input.txt").flatMap(lambda
line: line.split(" "))
wordCounts = input.map(lambda
word: (word,
1)).reduceByKey(add).collect()
```

Figure 7: Map reduce

ii. News Article Data Extraction & Transformation:

- Two reuters files were broken down into separate text files based on <TEXT></TEXT> tags using python with the help of regular expression[4].
- The content in each <TEXT> tag is cleaned by removed url, , special characters, emoticons, non-ASCII codes and metacharacters(figure 1) and stored in a new text file (figure 6).
- These newly created files (refer reutExtraction.py) are stored in EC2 instance.

C. Data Processing:

- I have used python to write a map reduce script that processes the data from stored tweets and articles (figure 7)
- Pymongo library is used to connect to mongo client to fetch tweets, pyspark is used to perform map reducing on data.
- Frequencies of words are calculated(refer mapReduce.py) based on mapping, filtering and reducing techniques (figure 7)

References

[1]"Getting started — tweepy 3.5.0 documentation", *Docs.tweepy.org*, 2019. [Online]. Available: http://docs.tweepy.org/en/v3.5.0/getting_started.html. [Accessed: 02- Jul- 2019].

[2]"MongoDB Import and Export JSON Data Example", *Examples Java Code Geeks*, 2019. [Online]. Available: https://examples.javacodegeeks.com/software-development/mongodb/mongodb-import-export-json-data-example/ . [Accessed: 02- Jul- 2019].

[3]"Introduction to big-data using PySpark: Map-filter-Reduce in python", *Annefou.github.io*, 2019. [Online]. Available: https://annefou.github.io/pyspark/02-mapreduce/. [Accessed: 02- Jul- 2019].

[4]"7.2. re — Regular expression operations — Python 2.7.16 documentation", *Docs.python.org*, 2019. [Online]. Available: https://docs.python.org/2/library/re.html. [Accessed: 02- Jul- 2019].