**Spark Project -4**

**Google 1 Gram Question 4**

Plot the average number of syllables on the y-axis, and time (year) on the x-axis

**Required data to answer the questions:**

1. Number of years in data.
2. Words and Number of words per each year

**Problem Solution approach:**

1. Solved the above question by using Hadoop streaming.

**Steps to solve Problem**

1. Read the data from file.

2. Transform the data to map and apply syllable function on the list in a map

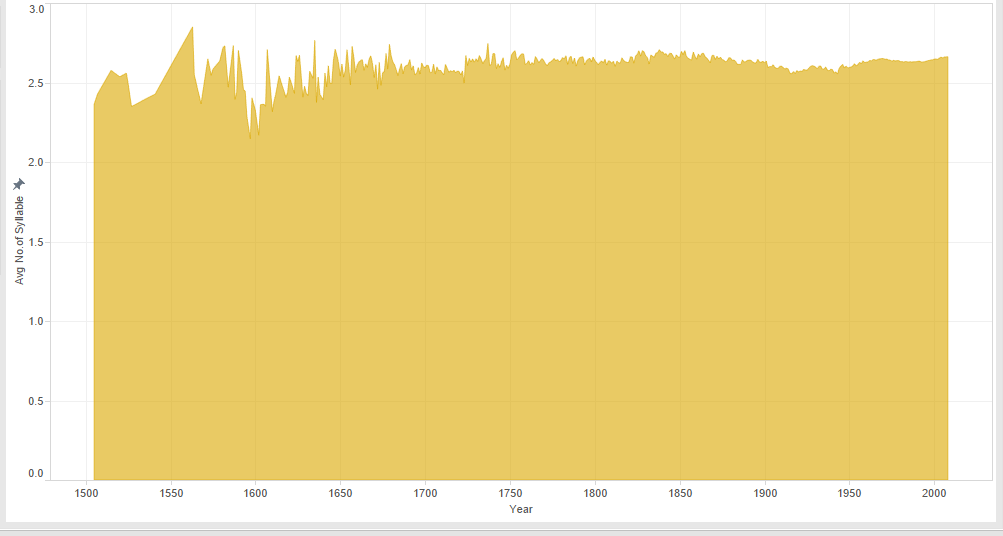
3. The syllable function returns number of words in a syllable and year and 1 which is used to represent the word occurs in a particular year. Here year acts as a key and value acts as (Syllable Count,1).

4. Now apply the reducebykey on the data returned by the syllable procedure which adds the length of syllable and number of words occurred in that year

5. Apply another mapper on the data produced by reducer and find the average of length of syllables in a year

6. Get the average data and plot the graph.

**Output**



**Twitter 5th Question**

What twitter user tweeted the most? What is the top 5 longest tweeters over each’s average tweet length? Bottom 5?

**Solution and Approach:**

Required fields from input data: hour of the day of tweet, tweet text message & userName

• Import print\_function, SparkContext, SQLContext.

• A Mapper function is created which parses each input line using json library and extracts the Screen Name from user data and Twitter text.

def getInfo(line):

js = json.loads(line)

text = js['text'].encode('ascii', 'ignore')

user = js['user']['screen\_name']

return [(user,text)]

• Dataset is then loaded into a RDD using the sc.textFile(sys.argv[1]) when the file name is sent from the command line parameters.

• Using “flatMap” the RDD is then transformed by using the function getInfo which was initially written to parse each input line.

• Using “sqlContext” and “createDataFrame”, a map is implemented to create a new transformed RDD as shown below

df = sqlContext.createDataFrame(texts.map(lambda (u,t): Row(text = t, length = len(t), user = u)))

• A temporary table is created using the function registerTempTable(“Name\_Of\_Table”).

• 3 SQL queries are written to find out the user who tweeted the most and the top/bottom users from the temporary table created in the above step.

sqlContext.sql("SELECT user, avg(length) as length FROM tweets group by user order by length")

sqlContext.sql("SELECT user, avg(length) as length FROM tweets group by user order by length desc")

sqlContext.sql("SELECT user, count(\*) as count FROM tweets group by user order by count desc")

**command to execute : spark-submit --master yarn-client SQL.py hdfs://hadoop2-0-0/data/twitter/\***

**Output:**

**========**

**a) User who tweeted most?**

username / count => marilyn9743 / 3419

**b) Top 5 users with highest avg tweet length**

username || avg length

==============================

Huntersweat 416

RoyalEliteKiva 350

blackxhole 320

KelleeMichele 272

pizzadellarry 253

**c) Bottom 5 user with highest avg tweet length**

username || avg length

==============================

Laila\_Lafrai 0

Fun\_Size20 0

Trevorsturgill5 0

2013Afi9 0

lm\_Lil\_Wanie 0

**(Google 1 gram Question 3)**

**Q3 - Plot the average word length for all unique words for all years available.  Year on x-axis, average word-length on y-axis**

**soln)**

The input 1 gram file destination is taken through sys.argv[1] from the command prompt to *input* variable.

A configuration for spark job is created using SparkConf() and an AppName is set in my code written.

The input files are read into the RDD as (key,value) pairs by using the sc.textFile() and by applying the map function and the data is split using "\t" and the input is transformed into a list of arrays . where each array contains the elements in the following order [word, year, frequency, number of documents].

calculated the Year wise word lengths by transforming it into another RDD

***yearlyAllwordsLen***  by performing a map on the input\_data (arrays of input data)

and generating a subset which is of the following format

[year, (wordlength,1)]

And then we calculate the **total word length**  and **Wordcount**  by performing a redcuceByKey on the ***yearlyAllwordsLen***  RDD and we save the result back in a new RDD (**reduceyearlylen**).

This RDD contains the data as (year , ( Totalwordlength, totalwordcount))

From this RDD we can calculate the Year wise average word length and save the result in a new RDD ***YearlywordAvg***  which contains the final key value pair in the following format.

**( Year, Averagewordlength )**

Results are saved to a file using **saveAsTexFile().**

Output: We run the script file containing the command.

**#spark-submit --master yarn-client /root/Desktop/spark/3gram.py/1Gram3.py /input\_data/1gram/ /data/1gram/\***

**Plot:**

Plot is created by having the **Year** on the (x-coordinate) and **average word length** on the (y-coordinate).

