SPARK - Project 4

Sai Siddhardha Reddy Thatiparthi

Amarnath Kothapalli

Yashmin Singla

Atmanand Citigori

Name: Amarnath Kothapalli

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

Dataset: Twitter

Platform: Python – pyspark

To get the above details, some information like username and his/her tweets are required. This is to find out the length of its tweets and to calculate the average length of all his/her tweets. Twitter user 'screen-name'and tweet 'text' are used. The following are the steps followed to get the required result.

- Using Json loads function, the twitter data is loaded.
- Read the twitter username and the tweet.
- Return the username as a key and tweet length value and 1.
- Using reduceByKey function, the user's tweets are added and the count along with sum of word lengths are obtained through username.
- By using the top function on count, the user who tweeted the most is acquired.
- Write the top 5 and the last 5 average tweet lengths using 'top' function.

Input Command: `spark-submit --master yarn-client avg_tweet.py hdfs://hadoop2-0-0/data/twitter/`

Output:

```
The top 5 users with most avg tweet length are
[(u'Huntersweat', (416.0, 1)), (u'RoyalEliteKiva', (350.0, 1)), (u'blackxhole',
(320.0, 1)), (u'KelleeMichele', (272.0, 1)), (u'pizzadellarry', (253.0, 1))]

The bottom 5 users of most avg tweet length are
[(u'ShakeIt4Rome', (1.0, 1)), (u'Im_Lil_Wanie', (1.0, 1)), (u'Abby_Palmiter', (1
.0, 1)), (u'HannahGarwood', (1.0, 1)), (u'Oliviacouss', (1.0, 1))]


 The User who tweeted the most is marilyn9743 making 3419 Tweets
```

Name: Yashmin Singla

Q) For each year available, plot the size of the set of words used. Year on the x-axis, number of words on y-axis. (Google onegram)

Dataset: Google Onegram

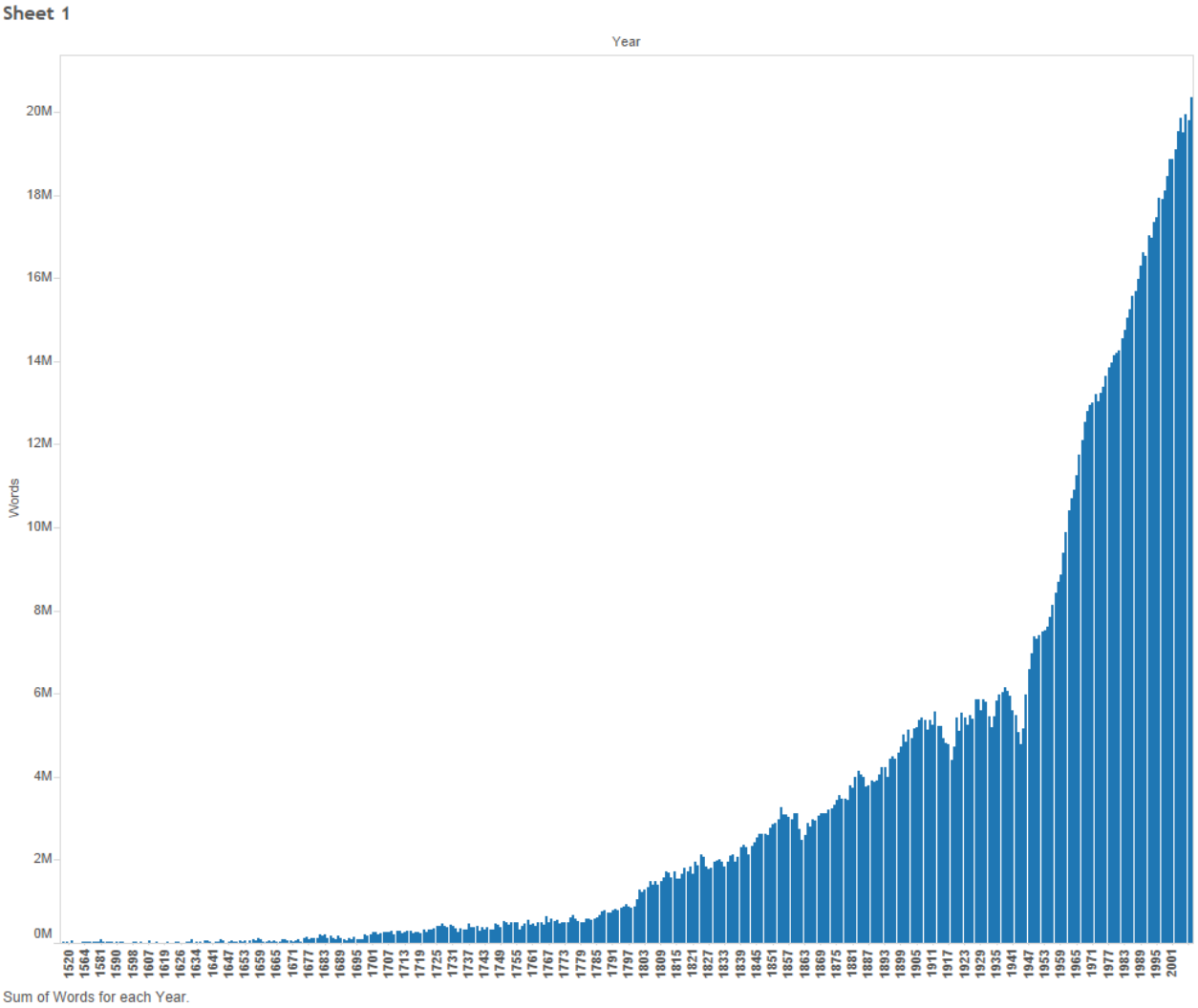
Platform: Python – pyspark

In this question, we need to find out the number of unique words that are used in each year. The following are the steps that are necessary:

- Read each line which has a format like “word year count unique count”.
- Divide them into components using the split function.
- Pass the year as the key and set a value of 1 as value.
- Get the total unique words in each year by using reduceByKey function.
- The plot is done using Tableau.
- Range of the graph: X-axis: 1505 -2008
  Y-axis: Number of unique words.
  The graph is plotted and the plot is dense as more than 500 years are in the plot.

Input command: `spark-submit --master yarn-client words.py hdfs://hadoop2-0-0/data/1gram/`

Output:



Name: Sai Siddhardh Reddy

Q) Plot the average word length for all unique words for all years available. Year on x-axis, average word-length on y-axis. (Google 1gram)

Dataset: Google 1 gram

Platform: Python – spark

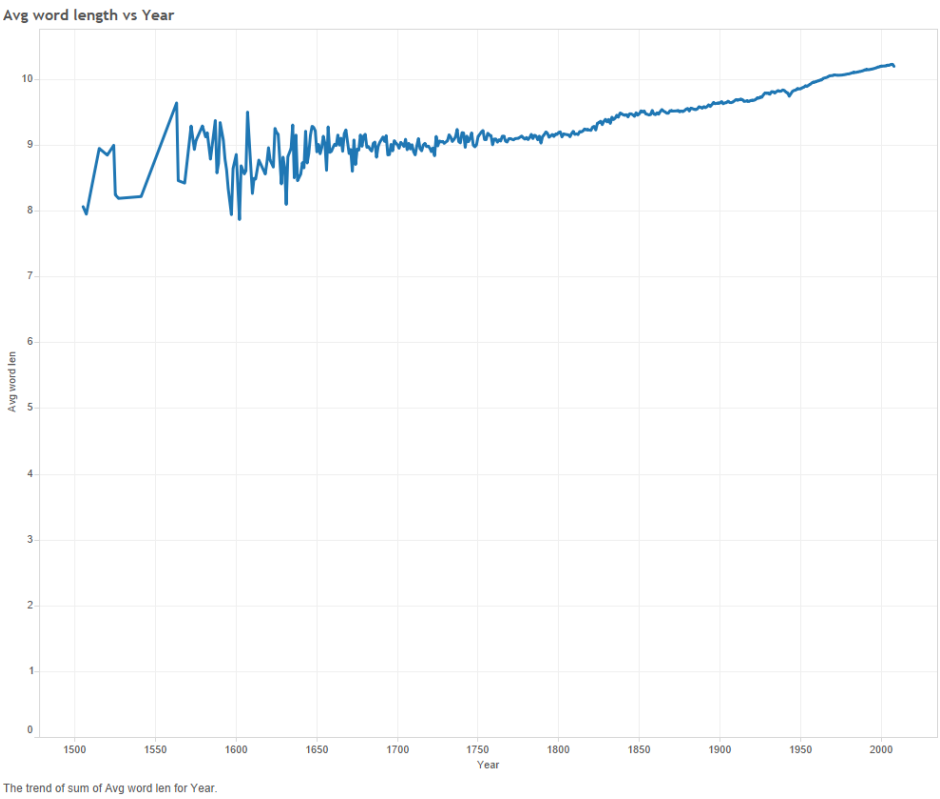
We need to plot the average word length for all the unique words for all the given years.

The following are the steps:

- The 1gram is split and it is mapped to extract the words and the times it is repeating.
- To obtain the year, the data is transformed
- Year being the key, ReducedByKey function is used on the data.
- The average word length and the data is acquired.
- The output is saved to a file
- The Plot is done using Tableau.
- Range of the graph: X-axis: 1505 -2008
  Y-axis: Avg_Length of unique words.

Input: `spark-submit --master yarn-client word_avg.py hdfs://hadoop2-0-0/data/1gram/`

Output:



Name: Atmanand Citigori

Q) For those tweets with location information, what lat/long (or city/state) is the centroid? What was the proportion of tweets with location to those without?

Dataset: Twitter

Platform: Python – spark

We need to find the location of most number of tweets coming from (centroid).

The following are the steps:

- Using Json loads function, the twitter data is loaded.
- Read the twitter location information from ‘coordinates’
- Return the ‘has_location’ or ‘No_location’ as a key and coordinates information along with 1.
- The tweet info with location and without information are separated using filter()
- The number of tweets with and without location information are calculated using count()
- The coordinates of the tweets with location are added using reduceBykey()

Input Command: `spark-submit --master yarn-client location.py hdfs://hadoop2-0-0/data/twitter/`

Output:

```
[Stage 1:=================================>                (2212 + 44) / 3213]
[Stage 1:==================================>               (2235 + 44) / 3213]
The proportion of tweets with location to those without location is :
0.443566476733
The centroid of the locations is
[-82.45294352  38.99033843]
```