Project 3
CS6065
Frazier Baker
Jeremiah Greer

**Twitter Problem 5:**

This problem is really a 3-part problem, namely: to determine which user tweeted the most, to determine the top 5 longest tweeters by average tweet length, and to determine the bottom 5 longest tweeters (top 5 shortest) by average tweet length. Having such statistics can serve to characterize users, indicate outliers, and show general trends in the extremes of the data.

To analyze this data, we first calculated the sum of the lengths of the tweets from each twitter user while simultaneously counting the number of tweets from each user. To do this in parallel on the Hadoop cluster, we used the numerical summarization design pattern. After the average tweet length and number of tweets was calculated for each user, the output was put through a second map-reduce job, which followed the top-n filter design pattern, with n=5 and n=1. Because a user's tweets may be spread far throughout the dataset, and therefore cannot be guaranteed to go to the same mapper the first time through a map-reduce algorithm, these two design patterns cannot be combined for this application.

Table 1. Results

| **Top 5 Tweeters by Average Tweet Length** | **Bottom 5 Tweeters by Average Tweet Length** | **Most Frequent Tweeter** |
|---|---|---|
| Huntersweat *(416.0)*<br>tayyathomas *(362.0)*<br>RoyalEliteKiva *(350.0)*<br>blackxhole *(320.0)*<br>uemucchan *(302.0)* | _theycallmecat_ *(1.0)*<br>Malik_Gberry *(1.0)*<br>_yung0z *(1.0)*<br>harrison23 *(1.0)*<br>T_iStone *(1.0)* | EmanuelCanil *(26 tweets)* |

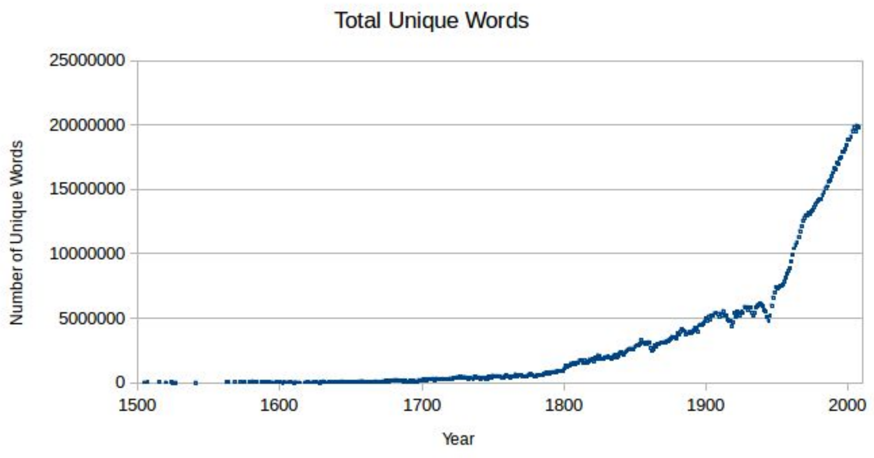
The results of this analysis can be seen in Table 1. It is important to note that the character counts for the Top 5 Tweeters by Average Tweet Length have Tweets that exceed 140 characters. This can occur when a user uses an "Emoji" character, which is a unicode character that encodes as multiple bytes. When coerced to ASCII plaintext, these characters can become as many as 4 characters, not including when emoji are used in conjunction with other emoji-related characters (e.g. customized skin color). It is also surprising that the averages are whole numbers, which likely means that these users only had one tweet in the dataset. Additionally, there were probably many ties for the lowest average tweet length, but only 5 were shown in the table. One might argue that all users with average tweet length of 1.0 should be reported, but for the sake of brevity and of answering the question given, "Bottom 5" was taken to mean exactly 5, no more, no less.

**Google 1gram Problem 1:**

This problem asks, "what is the total number of unique words used each year?" This would ideally give an indication of the total human vocabulary and how it has changed throughout history, given the Google Books 1gram dataset.

The method of analysis for this question is to gather all of the unique words within the 1gram dataset and then count how many words are contained within this unique non-duplicated set. To do this in parallel, we use the numerical summarization design pattern, as well as some restructuring of data, to fulfill the necessary requirements for this solution method. First, we reorganize the data within our map function so that for every line, we increment the value in our dictionary, using that line's year as the key. The Google 1grams dataset does not include duplicates for year for a given word, meaning that "dog" and 1984 will both be paired together only once. This allows us to simply count the number of pairs for a year, giving us the resulting total for a given year.

The results of this analysis are shown below. The chart indicates an exponential increase in the total number of unique words, with a small fluctuation from 1920-1950. This potentially hints at an increase in vocabulary in the past century, given the Google Books dataset; however, there is inherent bias in these records, as there are fewer records of older written works.

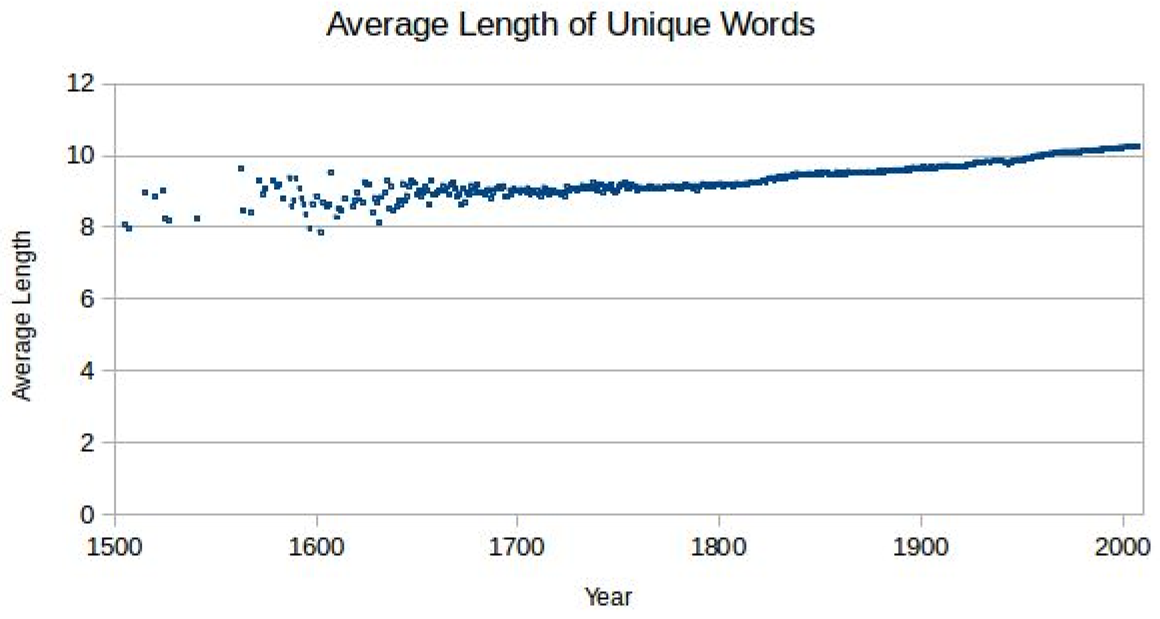


**Google 1gram Problem 3:**

This question asks "What is the average length of unique words used each year?' This would potentially show an increase in the complexity of human vocabulary given the average length of words used.

The method of analysis for this question is to sum the lengths of all unique words within the dataset and divide that result by the total number of words, thus giving an average length. To do this in parallel, we once again follow the numerical summarization design pattern to find the average word length. Our map function takes each line in the 1gram dataset, uses a dictionary with the year as the key, and increments two values. The total number of words for that year is incremented by 1, and the total lengths of all those words is incremented by the length of the new word. The map function then loops through that dictionary and outputs these keys and values. The reduce function takes these sums each one together by its year.

The results of this analysis are shown in the graph below. The graph plots the average length of unique words for each year. While sporadic at first given the sparsity of the dataset at that time period, this graph indicates that the average length of unique words has increased over time in a slow, linear fashion. This reveals that, over time, the english language has grown more and more complex, though at a slow pace.


Average Length of Unique Words
Average Length
Year