Google Books Ngrams

Zhengrong Gu(zg120), Shiying Sun(ss4296), Ruizhe Li(rl1025), Xiyu Chen(xc163)

ANLY502 Prof. Marck Vaisman

Spring 2020, Georgetown University

1. Introduction

Google books program, which scanned millions of books with 450 million words from university libraries, provides us with abundant information to study the rise and fall of words, the evolution of irregular verbs and the birth of new words over the decades. For example the word, ‘Awesome’, first appeared in the 1970s. People do not compliment others with the word ‘Awesome’ before the 1970s.

Words are created and shaped by the needs of a culture as it changes. Words and culture are fluid, shifting to reflect one another and the changing landscape of the world. After the revolution of gender equality in the 1990s, the usage of the Word, ‘Women’, went straight up. Thus with a more profound understanding of the shifting of words, we can have a more comprehensive scope of the culture we are living.

The goal of this study is to use state-of-the-art statistical methods with PySpark as a tool and AWS as a platform to study the trending of words in English Fictions from 1800 to 2009. For example, the questions we are interested in are:

- How does the frequency of n-grams change over time for certain words?
- Does the shifting culture we learnt in history books echo with the changing of frequencies in corresponding words?
- Does the frequency of the usage of words change over time?

2. DATA

The dataset we used in this project is Google Books Ngrams (https://aws.amazon.com/cn/datasets/google-books-ngrams/). The dataset is in sequence file format with block level LZO compression. The sequence file key is the row number of the dataset stored as a LongWritable and the value is the raw data stored as TextWritable.

2.1 Collection

Due to the enormous size of the whole dataset, we selected a corpus of Google Books Ngrams, English fiction with 228.9 GB in total. (s3://datasets.elasticmapreduce/ngrams/books/20090715/eng-fiction-all/[#]gram/data)

2.2 Variables

The value is a tab separated string containing the following fields:

n-gram - The actual n-gram

year - The year for this aggregation

occurrences - The number of times this n-gram appeared in this year

pages - The number of pages this n-gram appeared on in this year

books - The number of books this n-gram appeared in during this year

Table1. English Fiction

English Fiction		
1 gram	191,545,012	2.0 GB
2 gram	2,516,249,717	24.3 GB

3 gram	7,444,565,856	68.0 GB
4 gram	8,913,702,898	79.1 GB
5 gram	6,282,045,487	55.5 GB

2.3 Data Ingesting

We used data provided by google viewer which is uploaded on AWS S3 platform. Since the dataset was stored in sequenceFile form, we used a mapper to convert it into a spark dataframe for later analysis. With 2.5 billion rows in the data, the mapper process saves us a huge amount of time in later jobs.

Table2. Google Ngrams Table

ngram	occurrences	pages	books	year
! 165	1	1	1	1800
! 165	2	2	2	1804
! 165	1	1	1	1805
! 185	1	1	1	1806
! 185	4	4	4	1807
! 450	1	1	1	1804
! AH	2	2	2	1800

3. Methods

3.1 Data Preprocessing

After mapping the dataset into Spark dataframe, we aggregate the data by the ngram and decades after 1800. The table view of the aggregated data is below. Then we made a new matrix with the 21 data frames from 1800 to 2009 and scaled the occurrences by MinMax scaler. The new schema is provided below. We encountered the situation that there are tons of special characters, spaces, numbers in the NGRAM column. In order to roll out helpless information, we used regular expressions to filter out special characters. There are grams that appear to be identical to existing grams after applying regular expressions. Therefore, we merged those grams together to avoid redundancy and ensure analysis correctness.

Table3. Aggregated Table 1990-2000

Ngram	occurrences	Pages	Books
"Bulverhithe	4	4	2
Bur at	23	23	19
But Posy	2	2	2
Butler didn	60	60	50
COUNCIL HELD	14	13	9

Table4. Schema of the Aggregated Data

Root

|-- ngram: string (nullable = true)

|-- occurrences1800_Scaled: double (nullable = true)

|-- occurrences1810_Scaled: double (nullable = true)

|-- occurrences1820_Scaled: double (nullable = true)

|-- occurrences1830_Scaled: double (nullable = true)

|-- occurrences1840_Scaled: double (nullable = true)

|-- occurrences1850_Scaled: double (nullable = true)

|-- occurrences1860_Scaled: double (nullable = true)

|-- occurrences1870_Scaled: double (nullable = true)

|-- occurrences1880_Scaled: double (nullable = true)

|-- occurrences1890_Scaled: double (nullable = true)

|-- occurrences1900_Scaled: double (nullable = true)

|-- occurrences1910_Scaled: double (nullable = true)

|-- occurrences1920_Scaled: double (nullable = true)

|-- occurrences1930_Scaled: double (nullable = true)

|-- occurrences1940_Scaled: double (nullable = true)

|-- occurrences1950_Scaled: double (nullable = true)

|-- occurrences1960_Scaled: double (nullable = true)

|-- occurrences1970_Scaled: double (nullable = true)

|-- occurrences1980_Scaled: double (nullable = true)

|-- occurrences1990_Scaled: double (nullable = true)

|-- occurrences2000_Scaled: double (nullable = true)

After cleaning and aggregating the data, we create a bar plot to see the distinct word changes according to the decade. We can easily see the linear increase of the words used each decade with more and more books published.

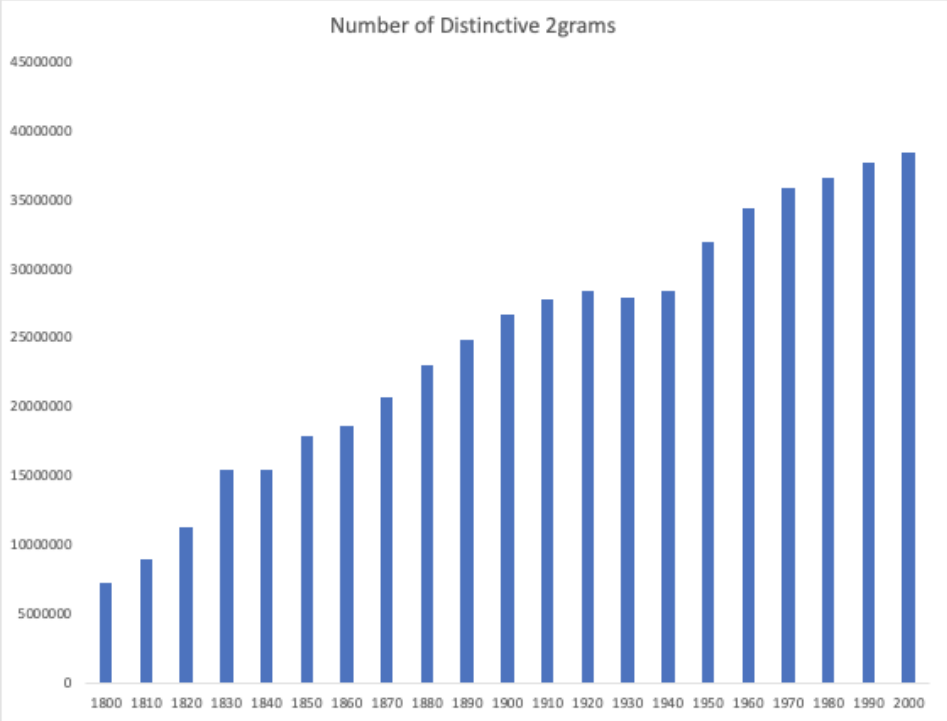


Fig1. Word Counts vs. Decades

3.2 Modeling

The main question we want to find out is whether the frequency of the occurences of words change during 1800 to 2009. In order to answer this question, we used K-means clustering on the 21 features which are standardized occurrences of words between 1800 to 2009. By using a silhouette with squared euclidean distance, we find out the most efficient K for our K Means model.

3.3 Visualization

Lastly, we visualized our data based on year and frequency via Matplotlib. From the visualization, we want to check if there are some significant changes in the appearance of words in English fictions.

4. Results and Conclusions

4.1 Clustering Results

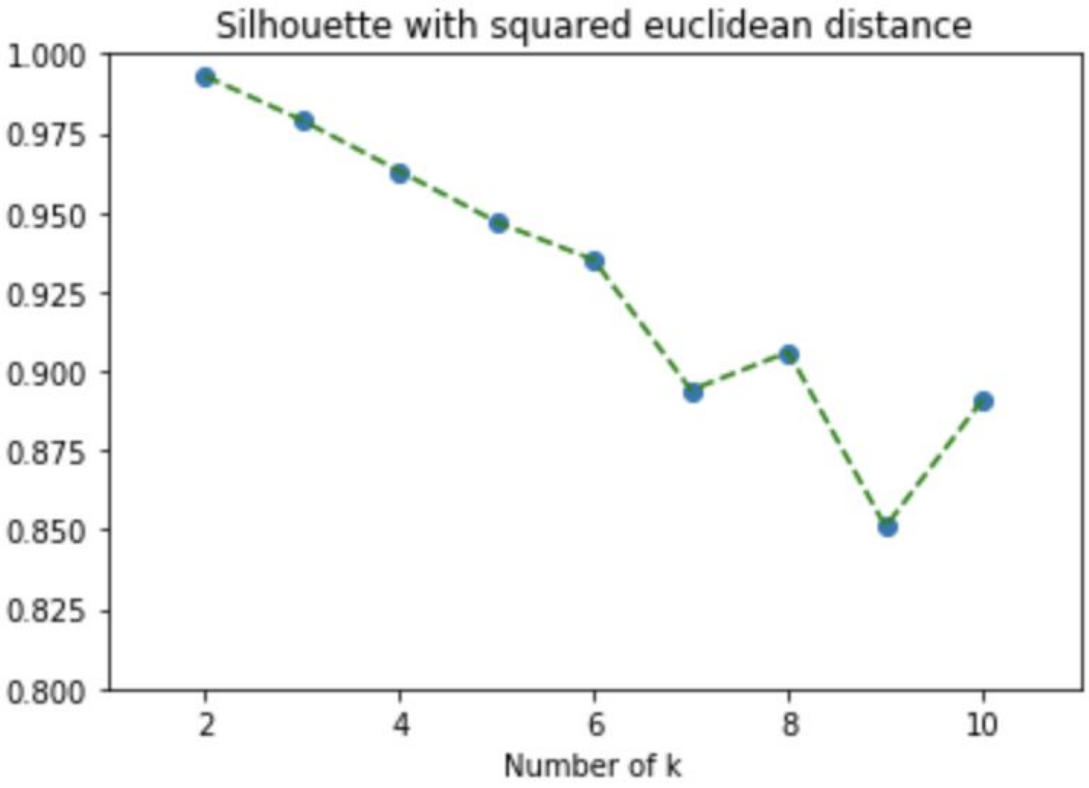


Figure 2. Clustering results

Based on the clustering results shown above, we can see that when k = 2, the silhouette with squared euclidean distance has the maximum, almost equal to 1. This means the frequency of occurrences of words doesn't change over time. And when k becomes larger, the silhouette with squared euclidean distance generally decreases with a few exceptions, e.g. k = 8.

4.2 Visualization Results

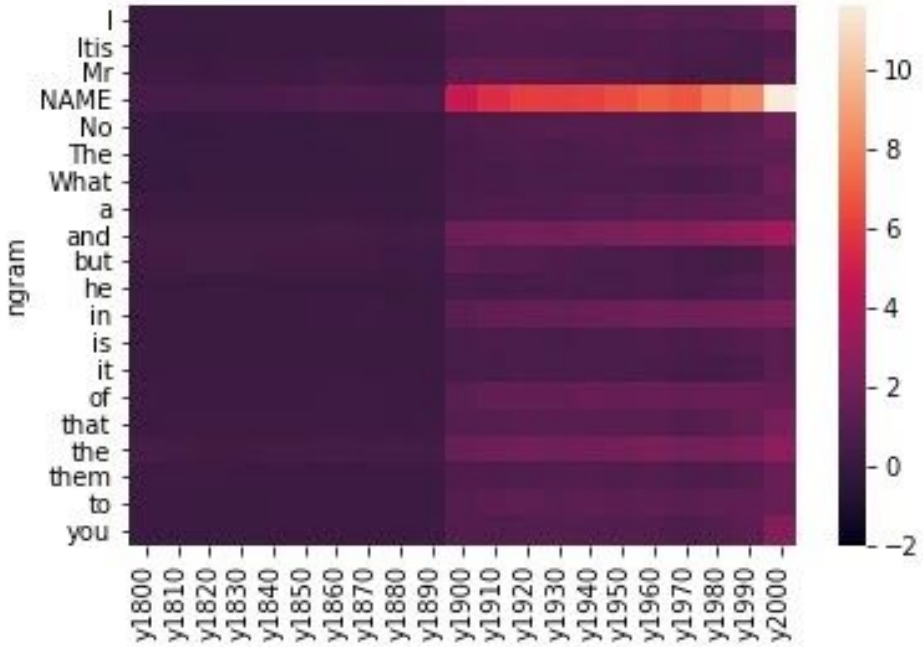


Fig3. Heatmap

In the above heatmap, the y axis is the 20 words that occured most frequently in the corresponding year shown on the x axis. An evident pattern is that the frequencies of most of the words increase smoothly over the years. However, the word ‘Name’ had a steep rise in the thear 1890. After checking the population and growth rate of 1880, 1890 and 1900, we found out that the growth rate of 1980 boosted. The pattern of the frequency of the word ‘Name’ coincides with the pattern of growth rate which demonstrates the relationship between the development of words and the trends in our culture. Next page contains the word clouds for 20 decades.

NAME the and but to them of Mr which ofthis that
NAME and the them but for of which that Mr
and NAME the but that for them of you
NAME and the but that them Mr for
NAME but the and that them you of Mr
NAME the but and that in of Mr
NAME the and but that of you Mr
NAME and but the that of Mr them
NAME the and that but of Mr
NAME and the but that you Mr in
NAME the and but of you Mr that
NAME the and but of in
NAME the and of you that Mr in
NAME the and of you that Mr in
NAME the and of that in Mr you
NAME the and of them you that in
NAME and in of the that Mr you
NAME and the of in that them
NAME and the that in of them
NAME and the them of in you

4.3 Conclusion and Discussion

The frequency of words does not significantly change over the past 200 years. This is mainly because the characteristics of English, e.g. grammar, usage of words, are not different from today's. Similar to Google Books which provide a clear visualization for the appearance frequency of words for Ngrams by year, we calculate the frequency by decade. The clustering results also support this conclusion.

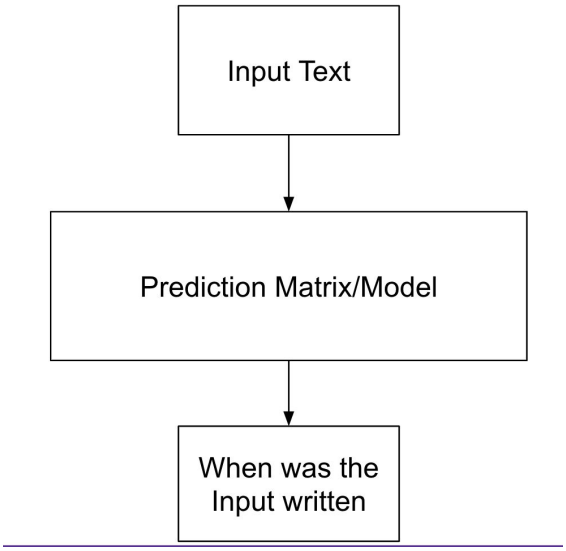
2-grams has some limitations, however. For example, some symbols such as "~, *, -" are far more frequently appearing than other words which influence the final result. 2-grams dataset has already been 24GB. Averagely speaking, aggregating every decade's data here costs PySpark more than 30 minutes to run. So we did not work on 3-grams, 4-grams, etc., because as the N grows, the size of datasets becomes larger and larger.

Aggregated data has been well processed during this project. It could be applied in our future work easily. Below is the link for those data:

https://s3.console.aws.amazon.com/s3/buckets/chenxiyuanly502/project/

5. Future Work

We plan to turn our Spark dataframe into a prediction matrix that takes grams as input and return the decade that the gram most likely belonged to as output. This is a flow chart of our idea:



However, we do not have enough background knowledge and datasets to really build a model for the calculation. With only simply adding the frequency of words in a sentence or article, the results will always be 2000 or 1990. Some manipulation and scaler should be applied to the matrix to build the model.

6. References

https://aws.amazon.com/datasets/google-books-ngrams/

https://registry.opendata.aws/google-ngrams/

https://stackoverflow.com/

7. Codes

https://github.com/CXYSean/anly502-project

Datapreprocessing.ipynb: code used to preprocessing the sequence file

Make_matrix.ipynb: code used to make the frequency matrix

Clustering.ipynb: code used to do the clustering

Project_vis.ipynb: code for visualization

Sample_future_idea.ipynb: code for future work idea

8. Division of Labor

Zhengrong Gu: Data aggregation, clustering
Shiying Sun: Visualization
Xiyu Chen: Data cleaning, future work
Ruizhe Li: Data cleaning, data aggregation