DATA 34200 Final Project

Soham Mall

Aim: To examine how the complexity of words used in English-language fiction has evolved over time

Data Sources:

- English fiction n-grams dataset from Google Books (BigQuery public dataset)
    - Total logical bytes: 6.93GB
    - Format: BigQuery internal table
- EFLLex: A dataset of ~15,000 English words and their lexical complexity, as determined by the Common European Framework of Reference for languages (CEFR), with difficulty levels ranging from A1 (easiest) to C1 (most difficult)
    - Total logical bytes: 7.2MB
    - Format: TSV file

Data Cleaning

The first cleaning step I took was to make sure that the n-grams data was clean: I wanted to turn all the words to lowercase and remove any duplicates. I ran a SQL query within BigQuery to do this, and then examined the results for a few different words. What I found was that there were multiple rows with the same word. For example, there were multiple rows that had the value “waiter” in the ‘term’ column. In the original n-grams dataset, there was only one such value. These duplicates were examples of terms that were originally tagged in a different way, such as misspellings/different spellings, or specific part-of-speech taggings. The dataset had a ‘tag’ column that made this more clear: one of the “waiter” entries was tagged “Zwaiter”, another “Awaiter”, another “waiter_NOUN.”

I decided that the best strategy for this project’s purposes would be to ignore the different variations of the word. The ‘main’ entry for each word always had a much higher frequency than any of the variations (e.g. the main “waiter” had a frequency of ~2 million while the variations had frequencies in the hundreds or the tens). Luckily, rather than having to process each word and find a way to filter out all the variations, it turned out that by using the original n-grams data without “cleaning” the words, removing punctuation and making them lowercase, the only “waiter” term would be that ‘main’ term with frequency of 2 million. This meant that as long as the EFLLex data was formatted in a similar way, with the words in lowercase, an inner join between the two datasets would inherently filter out all the troublesome variants of the words.

The EFLLex data was stored in a TSV file with 113 columns. Along with the word, there were 5 columns of interest which were quantitative measures of the frequency of the word at a certain level (A1, A2, B1, B2, C1). This complicated my analysis, because I originally thought that each word would be tagged to one of the five levels. I did some research on the EFLLex website and found that one of the ways words are “tagged” is by the level of writing in which they first

appear. I decided to follow this tagging methodology myself and tag each word to the lowest level at which its frequency is > 0. I also retained only the 5 columns of interest along with the word itself, removing the other columns from the dataset.

I uploaded this cleaned EFLLex data to BigQuery and ran

```
SELECT *
FROM `bigquery-public-data.google_books_ngrams_2020.eng_fiction_1` AS bq
JOIN `de-hw1-447917.final_proj_dataset.efllex_table` AS efl
ON bq.word = efl.word;
```

to join the two datasets - I then saved the results as a BQ table.

Analysis

My next task was to produce some kind of difficulty score by year so that I could plot the trend over time. First, I decided to create a numerical score to map to each of the five difficulty levels, with 1 corresponding to A1 and 5 corresponding to C1. Also, for each word, the yearly data is stored in a nested manner.

Field name
term
term_frequency
document_frequency
tokens
has_tag
▼ years
year
term_frequency
document_frequency

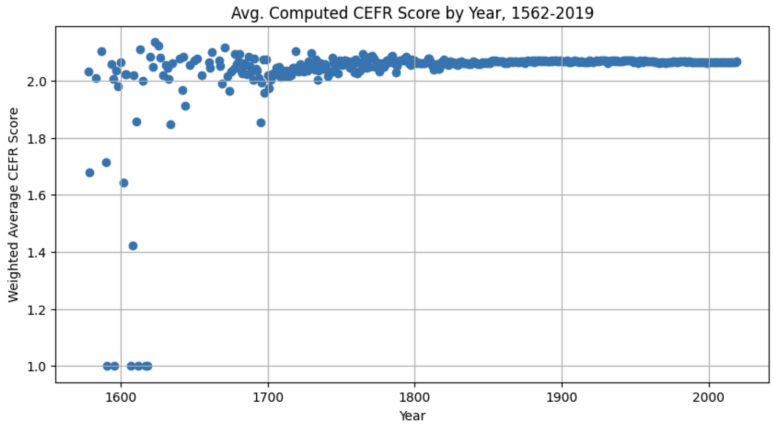
In order to ensure each year had its own complete row of data, I had to use the UNNEST feature of SQL. I wanted to design a query that would solve the following problems:

- Give each word a quantitative difficulty score, and then compute an average difficulty score for each year, weighted by the frequency of the words in that year
- Un-nest the n-grams data so that each word-year would be its own row

I designed and ran the following query in BigQuery:

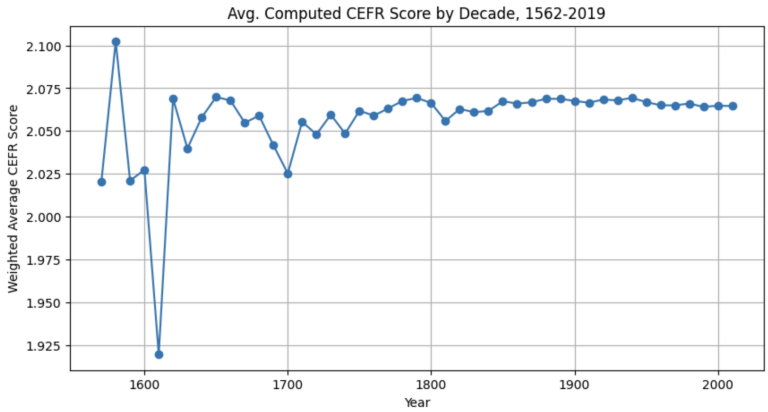
```
WITH unnested_data AS (
 SELECT CASE
     WHEN cefr_label = 'A1' THEN 1
     WHEN cefr_label = 'A2' THEN 2
     WHEN cefr_label = 'B1' THEN 3
```

```
    WHEN cefr_label = 'B2' THEN 4
    WHEN cefr_label = 'C1' THEN 5
    ELSE 0
  END AS cefr_score, y.year AS year,y.term_frequency AS term_frequency
 FROM `de-hw1-447917.final_proj_dataset.ngrams_cefr_labeled` AS t
 CROSS JOIN UNNEST(t.years) AS y
)
SELECT year, SUM(cefr_score * term_frequency) / SUM(term_frequency) AS avg_cefr_score
FROM unnested_data GROUP BY year ORDER BY year;
```



I also wanted to chart a similar trend by decade. With the same CTE for unnested_data as in the above query, I ran this query:

```
SELECT FLOOR(year / 10) * 10 AS decade,
 SUM(cefr_score * term_frequency) / SUM(term_frequency) AS avg_cefr_score
FROM unnested_data GROUP BY decade ORDER BY decade;
```



Across both the annual and decade-wise views, I saw that other than significant fluctuations around the 1600s and 1700s, the weighted average CEFR score has historically hovered

between 2.05 and 2.075. It also seems to have remained relatively flat. To further explore the distribution of levels over time, I ran queries to generate the proportion of vocabulary in each year that was of a particular CEFR level, such as the following (same unnested_data as above):

```
SELECT FLOOR(year / 10) * 10 AS decade,
 SUM(c1_term_frequency) / SUM(term_frequency) AS c1_proportion
FROM unnested_data GROUP BY decade ORDER BY decade;
```

I charted each CEFR level's data in a separate plot and found that

- A1-level vocab has made up >50% of the words used in these English fiction texts, and has been increasing over the last 4 decades covered
- The proportion of A2-level vocab has dropped steadily over time, from >17% in the 1590s to <14% in 2019
- The proportion of B1 and B2 vocab has remained relatively constant over time; furthermore, B2 vocab is significantly more common than A2 and B1 vocab
- The proportion of C1 level vocab, while still low, has been increasing steadily from ~3.5% in 1700 to ~4.2% in the 2010s