Summary

We are creating a program that will transform the way books are published. A high-powered professional grammar program is the goal. It will revolutionize the way people read. The product will be user friendly. The goal of the program is to analyze individual books or a set of documents. This program will analyze word frequencies and phrase frequencies. Temporarily the program will just determine if a certain set of known words exist in combination with a given phrase. Later it may analyze only grammar significant phrases. We may add a grammar analysis later. This program will have an optional graphic interface to view the data as it processes with red, yellow and green indicators and an optional sound. The sound will be much the same sound as the old modems make. You may type in a phrase and the program will analyze the book being reviewed. The relevant data will be in a CSV file and the proposed methods will be described below.

**Methods**

Word Frequency- the file will be parsed.

Dictionary of Word Frequency- a CSV file will be created that lists all the words in the book ranked by most frequent to least frequent.

Letter- letter counts will be considered.

Color (Optional)- Each alphabet will have a visual representation. The colors will be chosen with preference given to my color scheme.

Sound (Optional)- The sound will indicate a process has been completed.

Step Completion (Optional)- As a step is completed all the rectangles at the top will flash and revert back to the original alphabets message. Green will indicate successful completion of a step. The step must be a step that leads to another step. Yellow will flash if the process is busy computing. Red will flash if a step is unable to complete.

You translate it into sentences. First break each sentence into letters. Create a word count for each letter. Compare value of phrases and vocabulary letter counts with the value of letter counts in each sentence. For now use Typoglycemic document. If there is a match letter counts will be equal.

Create letter counts for sections in Typoglycemic.

Number of words in a sentence may be approximated by counting vowels a,e,i,o,u,y.

Limit the number of time i appears as an individual word. For now make it 2.

Create delays so the visuals slow down and is adjustable.

The letter counts must be equal to be a solution.

We may limit certain words to prevent too many solutions.

Calculate letter counts, word length, sentence length, number of words in a sentence and more.

**User Actions**

Type a phrase up to 20 characters long.

**Output**

All known word combinations will be given as a phrase in length of up to 80 characters.

**Layout**

Near the top of the screen there will be a row of 80 rectangles. Each rectangle will have in the center 3 pixels in a row. Each group of 3 pixels will represent a letter. So there will be 80 alphabets represented in this 80 rectangle set up.

User Input- Below the 80 rectangles there will be a similar sized rectangle where you may type in a phrase up to 20 characters to check and see what other words may be used to compile a sentence with the phrase included. The data files generated will be very large. A process is being created to manage the large data sets. The alphabets typed in this rectangle will instantly appear in the rectangle above.

Box below the user input- There will be a box below the user input which will contain the data in visual form as it is processed. The data will be the analysis in process. The data as it fills the bottom rectangle in the box will scroll up. Since we cannot see the data all at the same time, this scrolling will allow us to see only the latest analysis.

Go Button- Below the data rectangle will be a Go Button. This Go Button will begin the program.

**Limit Data**

To narrow solution just look for solutions with phrase included. To reduce further, only find solutions with minimum to maximum frequency of phrases. Some solutions might have multiple phrases though. You also may adjust minimum and maximum frequency for vocabulary. You may look for phrase with low frequency words when searching and creating alternative sentences.

**Things I need to do**

Try to find a way to incorporate all the data collected and add to the program. Help develop a method to help run the program. Provide specifications which are manageable. For example we would not want the program to run too long or we would not want to fry the graphic card. Create a Typoglycemic version of the file to test the program’s functioning. Review files such as the output for the Word Frequency and Phrase Frequency lists. Eliminate words from the document that do not contain a vowel a,e,i,o,u y. Decide how I may incorporate the star data into the program. Change date format to 2021/08/20. Create a CSV file for entries and time data, word count etc. Label each word in my book with the part(s) of speech.

**Things you need to do**

Allow for modifications. Program the program. Keep in contact by email. Choose a method I may understand. Such as how big the data is. How my suggested method may work. How my suggested method may not work. Reply to all emails as needed or reply as time allows you to in your schedule. (Optional) Create a GitHub account. Explain to me how to use the GitHub account. Consider using Photoshop. Create a list of letters used in the document. Then create a CSV file with number of times a letter is used. This will be the Letter Frequency chart. Calculate the percentage to 9 significant digits. Let me know how the computer program works in ordinary terms.

**Some Questions**

How does your program work? The definition of phrase is? How come you do not answer my email about phrasing? Should we be considering phrases that appear only once? If you read 80 characters at a time say 1-80 then 81-160 then 161-240… Is that an effective method? Should we consider changing the definition of phrase? Should you consider taking advantage of the ability to distinguish between lower and upper case letters? Would you run the phrasing program with the latest book? How may we incorporate the star data? Too much data?