Project 1:
Exploratory Data Analysis: Flesch-Kincaid Index

Natalie Tipton

Grand Valley State University

CIS 678

Dr. Greg Wolffe

January 21, 2020

Introduction

The Flesch-Kincaid Index is one that is commonly used to quantify the readability of a document. It does this based on the average number of words per sentence and syllables per word. The idea behind this is that if there are, on average, more words per sentence, the document is more difficult to comprehend because the reader must keep more information in mind as they read the entire sentence. Also, more syllables per word indicates that longer, more complex wording is being used, also making the document more difficult to comprehend. The higher the Flesch index, the easier the document is to read. This can also be converted into a relative grade-level for comprehension of the given reading. In this case, the higher the grade level, the more difficult the document is to comprehend. These calculations were done for three different documents including “Common Sense” by Thomas Paine, “The Adventures of Tom Sawyer” by Mark Twain, and a final project report written by myself.

Procedure

Since the Flesch-Kincaid Index relies on the number of sentences, words, and syllables in a document, the initial idea was to read in a text file and determine the number of those three features based upon the rules given in the project description. From this description words were to be determined by the presence of whitespace after a grouping of characters. Sentences were to be determined by the presence of a period, exclamation point, or question mark. Syllables had a few rules including a new syllable if a letter started with a vowel and when a vowel immediately followed a consonant. However, if an ‘e’ followed a constant by ending the word, a syllable was not to be counted. From there, the cumulative counts of the three features of interest could be used in the equations shown below to determine the resulting comprehension ratings. The Flesch-Kincaid Index could be calculated using Equation 1 and the corresponding grade level of comprehension could be calculated using Equation 2.

$$Flesch\ Index\ =\ 206.835 - 84.6\ (\frac{}{numWords}) - 1.015\ (\frac{}{numSentences}) \tag{1}$$

$$Grade\ level\ = 0.39\ (\frac{numWords}{numSentences})\ +\ 11.8(\frac{numSyllables}{numWords})\ -\ 15.59) \tag{2}$$

In order to implement these techniques, the text files were read in using the base python functions open and read. This read the document in all at once as opposed to line-by-line. In order to count the number of words in the document, the count function was used to sum all of occurrences of a space. After implementing that, however, it was clear that this technique missed counting the words that were followed by a new line instead of a space. Therefore, the summation of spaces was added to a count of new line characters. The only occasion that a new line character would exist in the document when a new word should not be counted is when there are two new line characters together at the end of a paragraph. Therefore, the total number of double new line characters was subtracted from the summation of spaces and new lines. This resulted in an accurate count of words when compared to the word count of the document in Microsoft Word.

Next, the number of sentences in the document was determined. To follow the rule explained in the project description, the first step was to use the count function to count how many periods, question marks, and exclamation points existed in the document. This on its own was a relatively accurate count, however, there was room for improvement. This method would inaccurately count new sentences whenever the document had an abbreviation followed by a period, an acronym separated by periods, an ellipsis, or any other time that multiple punctuation marks were used for emphasis. Therefore, a series of rules were included to ensure that a sentence would only be counted if the punctuation was preceded by an alphabetic character by using the function isalpha and if the punctuation was followed by a space, new line character, or quotation mark. This set of rules resulted in an accurate count of true sentences in the document.

Finally, the number of syllables were counted. This was done using the specified rules in the project description without the addition of any extra rules. A string was created including all vowels, not including the letter ‘y’. This allowed for comparisons of each letter to determine if it was a vowel or not so that the rules could be applied. In order to look at the document word by word, the split function was used to break it up by whitespace.

Once the various counts were determined, the Flesch-Kincaid Index and grade level were calculated. Matplotlib was used to create subplotted bar charts of those two calculations for easy visualization of the results for each document. In addition to that, the average words per sentence and syllables per word were also subplotted for each document to show where the main differences lay between the three documents.

Discussion

The program output can be seen in Appendix B. From here, the total number of sentences, words, and syllables for each document are seen as well as the Flesch-Kincaid Index and grade level. From the bar plots, it can be seen that “Common Sense” had the highest grade level for comprehension at 13, followed by my paper at 12, and finally “The Adventures of Tom Sawyer” at 7. There was a slight discrepancy with the raw Flesch-Kincaid Index. Here, my paper had the lowest index at 52.12. In this case, lower index values indicates more difficult comprehension. “Common Sense” had a slightly higher index at 58.94 and “The Adventures of Tom Sawyer” had a much higher index of 77.98. This is interesting since “Common Sense” had a higher grade level indicating it is more difficult to comprehend than my paper, however, the Flesch Index indicates it is easier to comprehend than my paper. From the bar chart for average words per sentence, it is clear that “Common Sense” is much higher than my paper. The chart for average syllables per word, however, shows that my paper is higher than the average for “Common Sense”.

The difference in results comes from the fact that there is a larger weight on the average syllables per word in the Flesch Index equation than the grade level equation. In the Flesch Index equation, syllables per word is weighted 83.3 times greater than the words per sentence. However, in the grade level equation, syllables per word is only weighted 30.25 times greater. Therefore, the higher syllables per word average in my paper than “Common Sense” must have been enough to sway the Flesch Index toward revealing that my paper was more complex whereas in the grade level, the higher words per sentence average for “Common Sense” swayed that calculation in the opposite way. This example shows that the two equations are not perfect, however, they still provide a roundabout idea of the complexity of a document.

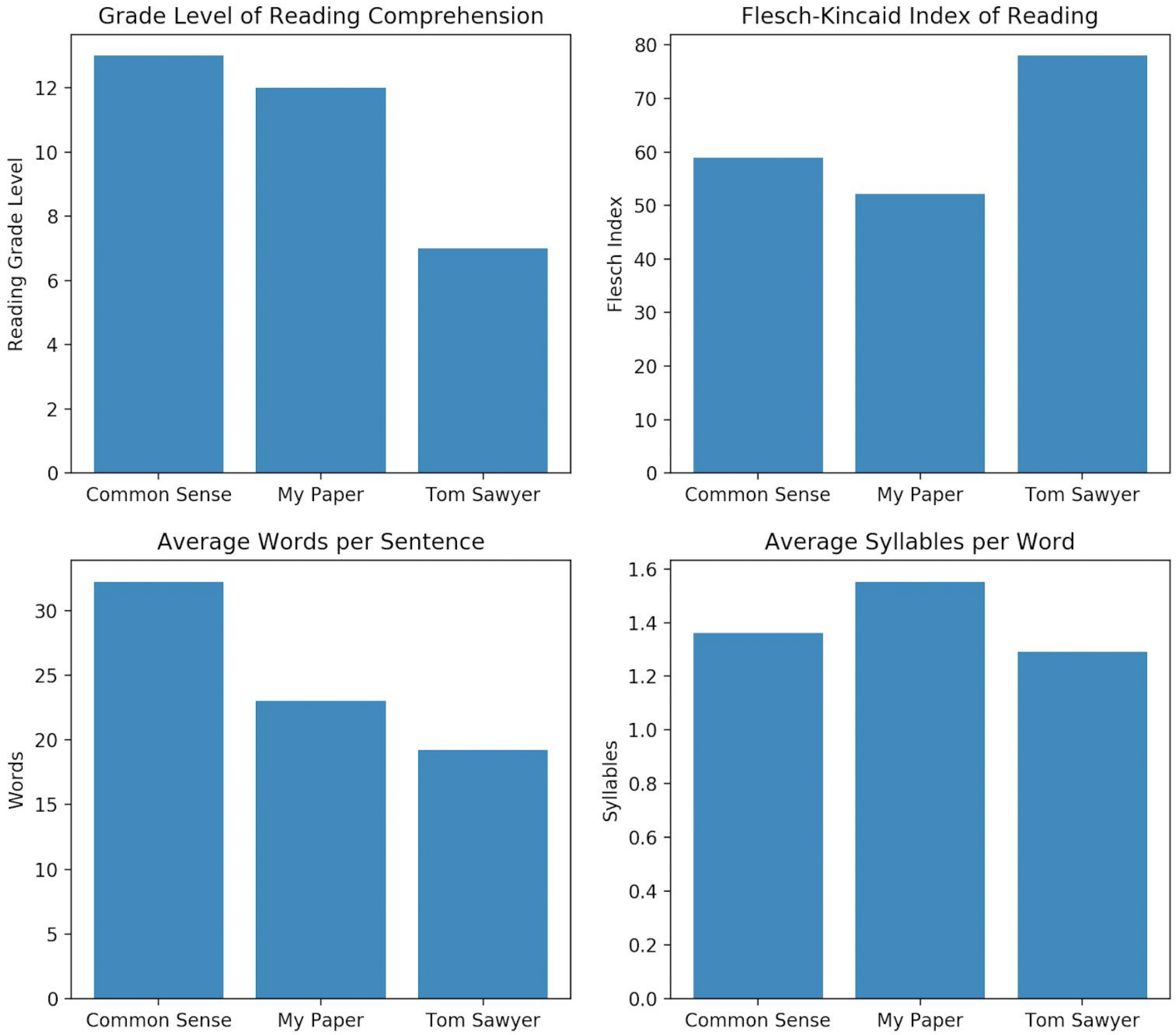
Appendix A - Source Code

```
#######################################################
# Title: Project 1 - Flesch Kincaid Index
# Class: CIS 678 - Machine Learning
# Professor: Dr. Wolffe
# Date: January 16, 2020
# Description: This program opens text files containing
#   and calculates the Flesch Kincaid Index and
#######################################################

import numpy as np
import matplotlib.pyplot as plt

##########################################################################

# determine the number of sentences, words, and syllables in a string
# and calculate the flesch index and resulting reading grade level
def flesch_index(text):

    # new word every time there is a space or a new line
    words = text.count(" ") + text.count("\n") - text.count("\n\n")

    sentences = 0

    # new sentence every time there is a . ! or ? followed by a space, new line,
    # or quotation mark with some letter in front of the punctuation
    for letter in range(0, len(text) - 1):
        if text[letter] == "." or text[letter] == "!" or text[letter] == "?":
            if text[letter - 1].isalpha():
                if (
                    text[letter + 1] == " "
                    or text[letter + 1] == "\n"
                    or text[letter + 1] == '"'
                ):
                    sentences += 1

    # add one sentence for the last sentence of the file
    sentences += 1
    vowels = "aeiouy"

    syllables = 0

    # new syllable every time a letter starts with a vowel or a vowel follows a
consonant
    for word in text.split():
        word = word.lower()
        if word[0] in vowels:
            syllables += 1
        for letter in range(1, len(word)):
            if word[letter] in vowels and word[letter - 1] not in vowels:
                syllables += 1
        if word.endswith("e"):
            syllables -= 1

    # calculate flesch-kincaid index and corresponding grade level
    flesch = 206.835 - 84.6 * (syllables / words) - 1.015 * (words / sentences)
    grade = round(
        0.39 * (words / sentences) + 11.8 * (syllables / words) - 15.59
    )

    return words, sentences, syllables, flesch, grade


##########################################################################

# open each file and run calculations on it
f = open("CommonSense.txt", "r")
text = f.read()
words1, sentences1, syllables1, flesch1, grade1 = flesch_index(text)
word_per_sent1 = words1 / sentences1
syl_per_word1 = syllables1 / words1
print(
    "\nCommon Sense has",
    sentences1,
    "sentences,",
    words1,
    "words, and",
   syllables1,
   "syllables.",
)
print("Its Flesch Index is", flesch1, "and its reading grade level is", grade1)

f = open("finalpaper.txt", "r")
text = f.read()
words2, sentences2, syllables2, flesch2, grade2 = flesch_index(text)
word_per_sent2 = words2 / sentences2
syl_per_word2 = syllables2 / words2
print(
   "\nMy paper has",
   sentences2,
   "sentences,",
   words2,
   "words, and",
   syllables2,
   "syllables.",
)
print("Its Flesch Index is", flesch2, "and its reading grade level is", grade2)

f = open("TomSawyer.txt", "r")
text = f.read()
words3, sentences3, syllables3, flesch3, grade3 = flesch_index(text)
word_per_sent3 = words3 / sentences3
syl_per_word3 = syllables3 / words3
print(
   "\nThe Adventures of Tom Sawyer has",
   sentences3,
   "sentences,",
   words3,
   "words, and",
   syllables3,
   "syllables.",
)
print("Its Flesch Index is", flesch3, "and its reading grade level is", grade3)

# for simple plotting, combine flesch indexes, grades, and titles into lists
flesch = [flesch1, flesch2, flesch3]
```

```
grades = [grade1, grade2, grade3]
word_per_sent = [word_per_sent1, word_per_sent2, word_per_sent3]
syl_per_word = [syl_per_word1, syl_per_word2, syl_per_word3]
titles = ("Common Sense", "My Paper", "Tom Sawyer")

yaxis = np.arange(len(titles))

# subplots of flesch index and grade levels of each reading
plt.figure(1)

plt.subplot(221)
plt.bar(yaxis, grades)
plt.title("Grade Level of Reading Comprehension")
plt.xticks(yaxis, titles)
plt.ylabel("Reading Grade Level")

plt.subplot(222)
plt.bar(yaxis, flesch)
plt.title("Flesch-Kincaid Index of Reading")
plt.xticks(yaxis, titles)
plt.ylabel("Flesch Index")

plt.subplot(223)
plt.bar(yaxis, word_per_sent)
plt.title("Average Words per Sentence")
plt.xticks(yaxis, titles)
plt.ylabel("Words")

plt.subplot(224)
plt.bar(yaxis, syl_per_word)
plt.title("Average Syllables per Word")
plt.xticks(yaxis, titles)
plt.ylabel("Syllables")
plt.show()
```

Appendix B - Output

```
Common Sense has 706 sentences, 22761 words, and 30986 syllables.
Its Flesch Index is 58.940660494076106 and its reading grade level is 13

My paper has 158 sentences, 3635 words, and 5644 syllables.
Its Flesch Index is 52.126657104800394 and its reading grade level is 12

The Adventures of Tom Sawyer has 3645 sentences, 70065 words, and 90557 syllables.
Its Flesch Index is 77.98137443802186 and its reading grade level is 7
```



Grade Level of Reading Comprehension

Flesch-Kincaid Index of Reading

Average Words per Sentence

Average Syllables per Word

Figure 1. Bar charts for visual output of grade level, Flesch Index, average words per sentence, and average syllables per word for each of the three documents.