The objective of this is to extract textual data articles from the given URL and perform text analysis to compute variables that are explained below. I want you to explain the complete process and write the code too. This code will be executed on google colab with the google drive mounted so save the output files into the drive through the code.

# **Data Extraction**

Input.xlsx: this excel file has ‘url-id’,’url’ as the columns

For each of the articles, given in the input.xlsx file, extract the article text and save the extracted article in a text file with URL\_ID as its file name.

While extracting text, please make sure your program extracts only the article title and the article text. It should not extract the website header, footer, or anything other than the article text.

**NOTE: YOU MUST USE PYTHON PROGRAMMING TO EXTRACT DATA FROM THE URLs. YOU CAN USE BEATIFULSOUP, SELENIUM OR SCRAPY, OR ANY OTHER PYTHON LIBRARIES THAT YOU PREFER FOR DATA CRAWLING.**

# **Data Analysis**

For each of the extracted texts from the article, perform textual analysis and compute variables, given in the output structure excel file. You need to save the output in the exact order as given in the output structure file, “Output Data Structure.xlsx”

**NOTE: YOU MUST USE PYTHON PROGRAMMING FOR THE DATA ANALYSIS**

# **Variables**

Definition of each of the variables given in the “Text Analysis” heading

Look for these variables in the analysis document “Text Analysis” heading.

1. POSITIVE SCORE
2. NEGATIVE SCORE
3. POLARITY SCORE
4. SUBJECTIVITY SCORE
5. AVG SENTENCE LENGTH
6. PERCENTAGE OF COMPLEX WORDS
7. FOG INDEX
8. AVG NUMBER OF WORDS PER SENTENCE
9. COMPLEX WORD COUNT
10. WORD COUNT
11. SYLLABLE PER WORD
12. PERSONAL PRONOUNS
13. AVG WORD LENGTH

# **Output Data Structure**

**Output Variables:**

1. All input variables in “Input.xlsx”
2. POSITIVE SCORE
3. NEGATIVE SCORE
4. POLARITY SCORE
5. SUBJECTIVITY SCORE
6. AVG SENTENCE LENGTH
7. PERCENTAGE OF COMPLEX WORDS
8. FOG INDEX
9. AVG NUMBER OF WORDS PER SENTENCE
10. COMPLEX WORD COUNT
11. WORD COUNT
12. SYLLABLE PER WORD
13. PERSONAL PRONOUNS
14. AVG WORD LENGTH

Checkout output data structure spreadsheet for the format of your output, i.e. “Output Data Structure.xlsx”.

Objective of this document is to explain methodology adopted to perform text analysis to drive sentimental opinion, sentiment scores, readability, passive words, personal pronouns and etc.

Table of Contents

1 Sentimental Analysis 2

1.1 Cleaning using Stop Words Lists 2

1.2 Creating dictionary of Positive and Negative words 2

1.3 Extracting Derived variables 2

2 Analysis of Readability 3

3 Average Number of Words Per Sentence 3

4 Complex Word Count 3

5 Word Count 3

6 Syllable Count Per Word 4

7 Personal Pronouns 4

8 Average Word Length 4

1 Sentimental Analysis

Sentimental analysis is the process of determining whether a piece of writing is positive, negative, or neutral. The below Algorithm is designed for use in Financial Texts. It consists of steps:

1.1 Cleaning using Stop Words Lists

The Stop Words Lists (found in the folder StopWords) are used to clean the text so that Sentiment Analysis can be performed by excluding the words found in Stop Words List.

1.2 Creating a dictionary of Positive and Negative words

The Master Dictionary (found in the folder MasterDictionary) is used for creating a dictionary of Positive and Negative words. We add only those words in the dictionary if they are not found in the Stop Words Lists.

1.3 Extracting Derived variables

We convert the text into a list of tokens using the nltk tokenize module and use these tokens to calculate the 4 variables described below:

Positive Score: This score is calculated by assigning the value of +1 for each word if found in the Positive Dictionary and then adding up all the values.

Negative Score: This score is calculated by assigning the value of -1 for each word if found in the Negative Dictionary and then adding up all the values. We multiply the score with -1 so that the score is a positive number.

Polarity Score: This is the score that determines if a given text is positive or negative in nature. It is calculated by using the formula:

Polarity Score = (Positive Score – Negative Score)/ ((Positive Score + Negative Score) + 0.000001)

Range is from -1 to +1

Subjectivity Score: This is the score that determines if a given text is objective or subjective. It is calculated by using the formula:

Subjectivity Score = (Positive Score + Negative Score)/ ((Total Words after cleaning) + 0.000001)

Range is from 0 to +1

2 Analysis of Readability

Analysis of Readability is calculated using the Gunning Fox index formula described below.

Average Sentence Length = the number of words / the number of sentences

Percentage of Complex words = the number of complex words / the number of words

Fog Index = 0.4 \* (Average Sentence Length + Percentage of Complex words)

3 Average Number of Words Per Sentence

The formula for calculating is:

Average Number of Words Per Sentence = the total number of words / the total number of sentences

4 Complex Word Count

Complex words are words in the text that contain more than two syllables.

5 Word Count

We count the total cleaned words present in the text by

1. removing the stop words (using stopwords class of nltk package).

2. removing any punctuations like ? ! , . from the word before counting.

6 Syllable Count Per Word

We count the number of Syllables in each word of the text by counting the vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

7 Personal Pronouns

To calculate Personal Pronouns mentioned in the text, we use regex to find the counts of the words - “I,” “we,” “my,” “ours,” and “us”. Special care is taken so that the country name US is not included in the list.

8 Average Word Length

Average Word Length is calculated by the formula:

Sum of the total number of characters in each word/Total number of words

Write the code to perform the following steps on the above generated file ‘URL\_ID’ Sentimental Analysis.

Sentimental analysis is the process of determining whether a piece of writing is positive, negative, or neutral. The below Algorithm is designed for use in Financial Texts. It consists of steps:

1.1 Cleaning using Stop Words Lists

The Stop Words Lists , whose folder link is provides are used to clean the text so that Sentiment Analysis can be performed by excluding the words found in Stop Words List.

1.2 Creating a dictionary of Positive and Negative words

The Master Dictionary (found in the folder MasterDictionary) is used for creating a dictionary of Positive and Negative words. We add only those words in the dictionary if they are not found in the Stop Words Lists.

1.3 Extracting Derived variables

We convert the text into a list of tokens using the nltk tokenize module and use these tokens to calculate the 4 variables described below:

Positive Score: This score is calculated by assigning the value of +1 for each word if found in the Positive Dictionary and then adding up all the values.

Negative Score: This score is calculated by assigning the value of -1 for each word if found in the Negative Dictionary and then adding up all the values. We multiply the score with -1 so that the score is a positive number.

Polarity Score: This is the score that determines if a given text is positive or negative in nature. It is calculated by using the formula:

Polarity Score = (Positive Score – Negative Score)/ ((Positive Score + Negative Score) + 0.000001)

Range is from -1 to +1

Subjectivity Score: This is the score that determines if a given text is objective or subjective. It is calculated by using the formula:

Subjectivity Score = (Positive Score + Negative Score)/ ((Total Words after cleaning) + 0.000001)

Range is from 0 to +1

In the Google Drive there is a folder ‘URL\_ID2’ having a number of .txt files for every file compute the below mentioned attributes and save them in a .csv file titled result2

Analysis of Readability

Analysis of Readability is calculated using the Gunning Fox index formula described below.

Average Sentence Length = the number of words / the number of sentences

Percentage of Complex words = the number of complex words / the number of words

Fog Index = 0.4 \* (Average Sentence Length + Percentage of Complex words)

Average Number of Words Per Sentence

The formula for calculating is:

Average Number of Words Per Sentence = the total number of words / the total number of sentences

Complex Word Count

Complex words are words in the text that contain more than two syllables.

Word Count

We count the total cleaned words present in the text by

1. removing the stop words (using stopwords class of nltk package).

2. removing any punctuations like ? ! , . from the word before counting.

Syllable Count Per Word

We count the number of Syllables in each word of the text by counting the vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

Personal Pronouns

To calculate Personal Pronouns mentioned in the text, we use regex to find the counts of the words - “I,” “we,” “my,” “ours,” and “us”. Special care is taken so that the country name US is not included in the list.

Average Word Length

Average Word Length is calculated by the formula:

Sum of the total number of characters in each word/Total number of words

import os

import string

import csv

import nltk

from nltk.corpus import stopwords

import re

def calculate\_readability(text):

# Count number of sentences

sentences = nltk.sent\_tokenize(text)

num\_sentences = len(sentences)

# Count number of words

words = nltk.word\_tokenize(text)

words = [word.lower() for word in words if word.isalpha()]

num\_words = len(words)

# Count number of syllables

def count\_syllables(word):

vowels = 'aeiouy'

num\_vowels = 0

last\_char = None

for char in word:

if char in vowels and last\_char not in vowels:

num\_vowels += 1

last\_char = char

if word.endswith('es') or word.endswith('ed'):

num\_vowels -= 1

return num\_vowels

num\_syllables = sum([count\_syllables(word) for word in words])

# Calculate average sentence length

avg\_sentence\_len = num\_words / num\_sentences

# Calculate percentage of complex words

complex\_words = [word for word in words if count\_syllables(word) > 2]

pct\_complex\_words = len(complex\_words) / num\_words

# Calculate Fog Index

fog\_index = 0.4 \* (avg\_sentence\_len + pct\_complex\_words)

# Calculate average number of words per sentence

avg\_words\_per\_sentence = num\_words / num\_sentences

# Calculate complex word count

num\_complex\_words = len(complex\_words)

# Calculate personal pronoun count

personal\_pronouns = ['i', 'we', 'my', 'ours', 'us']

personal\_pronoun\_count = sum([len(re.findall(f'(?<!U.S )\\b{pronoun}\\b', text.lower())) for pronoun in personal\_pronouns])

# Calculate average word length

total\_word\_length = sum(len(word) for word in words)

avg\_word\_length = total\_word\_length / num\_words

# Create a dictionary with the calculated values

results = {'num\_sentences': num\_sentences,

'num\_words': num\_words,

'num\_syllables': num\_syllables,

'avg\_sentence\_len': avg\_sentence\_len,

'pct\_complex\_words': pct\_complex\_words,

'fog\_index': fog\_index,

'avg\_words\_per\_sentence': avg\_words\_per\_sentence,

'num\_complex\_words': num\_complex\_words,

'personal\_pronoun\_count': personal\_pronoun\_count,

'avg\_word\_length': avg\_word\_length}

return results

# Define paths

text\_folder = 'URL\_ID2'

stopwords\_path = 'StopWords.txt'

output\_file = 'result2.csv'

# Load stop words

stop\_words = set(stopwords.words('english'))

# Process each file in text\_folder

results = []

for filename in os.listdir(text\_folder):

if filename.endswith('.txt'):

# Load file

with open(os.path.join(text\_folder, filename), 'r') as file:

text = file.read()

# Remove stop words and punctuation

words = nltk.word\_tokenize(text)

words = [word.lower() for word in words if word.lower() not in stop\_words and word.isalpha()]

words = [word.translate(str.maketrans('', '', string.punctuation)) for word in words]

# Join words back into text

cleaned\_text = ' '.join(words)

# Calculate readability metrics

readability\_metrics = calculate\_readability(cleaned\_text)

# Add filename

continue the code from Define paths

text\_folder = 'URL\_ID2'

stopwords\_path = 'StopWords.txt'

output\_file = 'result2.csv'

# Process each file in text\_folder

results = []

for filename in os.listdir(text\_folder):

if filename.endswith('.txt'):

# Load file

with open(os.path.join(text\_folder, filename), 'r') as file:

text = file.read()

# Remove stop words and punctuation

words = nltk.word\_tokenize(text)

words = [word.lower() for word in words if word.lower() not in stop\_words and word.isalpha()]

words = [word.translate(str.maketrans('', '', string.punctuation)) for word in words]

# Join words back into text

cleaned\_text = ' '.join(words)

# Calculate readability metrics

readability\_metrics = calculate\_readability(cleaned\_text)

I have a folder named ‘URL\_ID’ in google drive containing some .txt files. There are some functions mentioned below which are to be computed for all the .txt files in the folder and then the results are to be made in a csv file results2.csv and saved in the google drive. In the stopwords removal for word count use nltk stopwords. Not using stopwords from a file in google but using the nltk stopwords

The csv should have the following columns:

1. AVG SENTENCE LENGTH
2. PERCENTAGE OF COMPLEX WORDS
3. FOG INDEX
4. AVG NUMBER OF WORDS PER SENTENCE
5. COMPLEX WORD COUNT
6. WORD COUNT
7. SYLLABLE PER WORD
8. PERSONAL PRONOUNS
9. AVG WORD LENGTH

Average Number of Words Per Sentence

The formula for calculating is:

Average Number of Words Per Sentence = the total number of words / the total number of sentences

Complex Word Count

Complex words are words in the text that contain more than two syllables.

Word Count

We count the total cleaned words present in the text by

1. removing the stop words (using stopwords class of nltk package).

2. removing any punctuations like ? ! , . from the word before counting.

Syllable Count Per Word

We count the number of Syllables in each word of the text by counting the vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

Personal Pronouns

To calculate Personal Pronouns mentioned in the text, we use regex to find the counts of the words - “I,” “we,” “my,” “ours,” and “us”. Special care is taken so that the country name US is not included in the list.

Average Word Length

Average Word Length is calculated by the formula:

Sum of the total number of characters in each word/Total number of words

Analysis of Readability

Analysis of Readability is calculated using the Gunning Fox index formula described below.

Average Sentence Length = the number of words / the number of sentences

Percentage of Complex words = the number of complex words / the number of words

Fog Index = 0.4 \* (Average Sentence Length + Percentage of Complex words)

I have a folder named ‘URL\_ID2’ containing some .txt files. There are some functions mentioned below which are to be computed for all the .txt files in the folder and then the results are to be made in a csv file results2.csv . In the stopwords removal for word count use nltk stopwords. Not using stopwords from a file in google but using the nltk stopwords

The csv should have the following columns:

1. AVG SENTENCE LENGTH

2. PERCENTAGE OF COMPLEX WORDS

3. FOG INDEX

4. AVG NUMBER OF WORDS PER SENTENCE

5. COMPLEX WORD COUNT

6. WORD COUNT

7. SYLLABLE PER WORD

8. PERSONAL PRONOUNS

9. AVG WORD LENGTH

Average Number of Words Per Sentence

The formula for calculating is:

Average Number of Words Per Sentence = the total number of words / the total number of sentences

Complex Word Count

Complex words are words in the text that contain more than two syllables.

Word Count

We count the total cleaned words present in the text by

1. removing the stop words (using stopwords class of nltk package).

2. removing any punctuations like ? ! , . from the word before counting.

Syllable Count Per Word

We count the number of Syllables in each word of the text by counting the vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

Personal Pronouns

To calculate Personal Pronouns mentioned in the text, we use regex to find the counts of the words - “I,” “we,” “my,” “ours,” and “us”. Special care is taken so that the country name US is not included in the list.

Average Word Length

Average Word Length is calculated by the formula:

Sum of the total number of characters in each word/Total number of words

Analysis of Readability

Analysis of Readability is calculated using the Gunning Fox index formula described below.

Average Sentence Length = the number of words / the number of sentences

Percentage of Complex words = the number of complex words / the number of words

Fog Index = 0.4 \* (Average Sentence Length + Percentage of Complex words)

files = glob.glob('/content/drive/MyDrive/URL\_ID2')

for filename in files:

with open(filename, 'r') as f:

I want to access all files from a folder present in a url

with open('results2.csv', mode='w', newline='') as file:

writer = csv.writer(file)

writer.writerow(['Filename', 'Avg Sentence Length', '% of Complex Words', 'FOG Index', 'Avg Number of Words per Sentence', 'Complex Word Count', 'Word Count', 'Syllables per Word', 'Personal Pronouns', 'Avg Word Length'])

files = glob.glob('./URL\_ID2/\*.txt')

for filename in files:

with open(filename, 'r') as f:

text = f.read()

sentences = nltk.sent\_tokenize(text)

total\_words = len(nltk.word\_tokenize(text))

total\_sentences = len(sentences)

I want the program to access the text files from inside a folder titles ‘URL\_ID2’

In python

Add the below columns from Input.xlsx

url\_id

url

Add the below columns from Results.csv

Positive Score

Negative Score

Polarity Score

Subjectivity Score

Add the below columns from results2.csv

Avg Sentence Length

% of Complex Words

FOG Index

Avg Number of Words per Sentence

Complex Word Count

Word Count

Syllables per Word,

Personal Pronouns

Avg Word Length.

Into a csv titled finalresult.csv