# Text Analysis Script Documentation

**Author- Shubham Soni |Contact- shubhamsoni616.22@gmail.com**

## Description

This script is designed to analyse the text content of a given article URL and generates a text file containing various metrics and analysis results. I have test the script on different websites including ‘Inverse’, ‘Hindustan Times’, ‘NDTV’, ‘Times of India’, and ‘Indian Express’.

The script is capable of extracting relevant text content from the webpage and performing operations such as sentiment analysis, keyword extraction, and summarization using Python libraries such as ‘BeautifulSoup’, ‘nltk’, ‘regex’, ‘requests’, and ‘textstat’.

Once the analysis is complete, the script writes the results to a text file. This script can be useful for researchers, journalists, or anyone else who needs to analyse web articles' text content in a systematic and automated way.

Note: I have used some sample links given in the ‘Sample Article Links.txt’ file to test the script and the generated analysis results are in the ‘Generated Result files’ folder.

## Requirements

This script requires the following dependencies to be installed:

- ‘beautifulsoup4’
- ‘nltk’
- ‘textstat’

In addition, it also needs files which is given in the directory:

- ‘Stopwords.txt’: A list of stop words that will be excluded from the extracted text to perform sentiment analysis.
- ‘positive-words.txt’: A list of positive words that will be used for sentiment analysis.
- ‘negative-words.txt’: A list of negative words that will be used for sentiment analysis.
- ‘Generated Result files’ folder: to store the generated analysis result files.

# How to Use

To use the script, follow these steps:

- Ensure that all the dependencies are installed and the required text files are available in the same directory as the script.
- Run the script in a Python environment.
- When prompted, enter the URL of the article you want to analyse.
- When prompted, enter a name for the generated text file. Note: Be careful while giving a file name in the input prompt as it can overwrite an existing text file in the directory with the same name.
- Wait for the script to complete the analysis.
- Find the generated text file in the 'Generated Result files' directory.

# Analysis Results

The script generates a text file that contains the following metrics and analysis results including the URL of the article that was analysed and the article content:

1. **The word count**
   It calculates word count using regular expression.

2. **The average word length**
   The average word length is calculated by the formula-
   $$Average\_Word\_Length = total\_characters/number\_of\_words$$

3. **The average number of syllables per word**
   To calculate average number of syllables per word, the script counts the number of vowels present in every word without counting the 'e' of the words ending with "es" or "ed" and then divide it by the total number of words.

4. **The average number of words per sentence**
   It calculates number of sentences and number of words in the text using regular expression. The average sentence length is calculated using this formula-
   $$Avg\_Sent\_Length = number\_of\_words/number\_of\_sents$$

5. **The count of complex words**
   The complex words count is calculated by counting the words in the text that contain more than 2 syllables excluding the stop-words. I have used 'textstat' library to count the syllables in a word.

6. **The complex words percentage**
   The percentage of complex words is calculated using this formula-
   $$Per\_Complex\_Words = Complex\_Words\_Count \,/\, number\_of\_words * 100$$

7. **The count of personal pronouns**
   It counts the number of personal pronouns ('I', 'we', 'my', 'ours', and 'us') present in the text excluding the country name 'US' using regular expression.

8. **The Fog Index**
   It calculates the gunning Fox index using this formula-
   $$Fog\_Index = 0.4 * (Avg\_Sent\_Length + Per\_Complex\_Words)$$

9. **The positive score**
   After cleaning the text by removing the stop words present in the 'Stopwords.txt' file, the positive score is calculated by counting the words of the text that are present in the 'positive-words.txt' file.

10. **The negative score**
    After cleaning the text by removing the stop words present in the 'Stopwords.txt' file, the negative score is calculated by counting the words of the text that are present in the 'negative-words.txt' file.

11. **The polarity score**
    It calculates the polarity score with this formula-
    $$Polarity\_Score = (Positive\_Score - Negative\_Score)/ ((Positive\_Score + Negative\_Score) + 0.000001)$$

12. **The subjectivity score**
    It calculates the subjectivity score with this formula
    $$Subjectivity\_Score = (Positive\_Score + Negative\_Score)/ ((len(clean\_text)) + 0.000001)$$