Text Analysis Blackcoffer Assignment

Documentation

Made by-Jitesh Dadlani

This is a mini-project which does text analysis of financial documents available on the sec website and returns the polarity, complex words, positive score, negative score and several other attributes in an excel sheet  
Make sure to download the list of stop words, positive words , negative words

**Objective**

Objective of this assignment is to extract some sections (which are mentioned below) from SEC / EDGAR financial reports and perform text analysis to compute variables those are explained below. Link to SEC / EDGAR financial reports are given in excel spreadsheet “cik\_list.xlsx”. Please add <https://www.sec.gov/Archives/> to every cells of column F (cik\_list.xlsx) to access link to the financial report. Example: Row 2, column F contains edgar/data/3662/0000950170-98-000413.txt Add <https://www.sec.gov/Archives/> to form financial report link i.e. <https://www.sec.gov/Archives/edgar/data/3662/0000950170-98-000413.txt>

Table of Contents

1.Technologies/Libraries Used

2.Text Analysis

3.Code output

**1.Technologies/libraries used**

Note:This project was developed using Python 3.7 any difference in the version might cause some errors

Also the below mentioned word list and cik\_list.xlsx has to be saved in the directory in which program is being executed specify the exact location of the file in the code

-urllib

-BeautifulSoup

-Xlsxwriter

-xlrd Reading an excel file using Python

-Nltk

This project also makes use of pre made list of the following

taken from <https://sraf.nd.edu/textual-analysis/resources/>

-Positive words

-Negative words

-Constraining words

-Uncertain words

**2.Text Analysis**

# **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 on Financial Texts. It consists of steps:

## **Cleaning using Stop Words Lists**

The Stop Words Lists (found [here](http://www3.nd.edu/~mcdonald/Word_Lists.html)) are used to clean the text so that Sentiment Analysis can be performed by excluding the words found in Stop Words List. Use this url if above does not work <https://sraf.nd.edu/textual-analysis/resources/>

## **Creating dictionary of Positive and Negative words**

The Master Dictionary (found [here](http://www3.nd.edu/~mcdonald/Word_Lists.html)) 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. Use this url if above does not work <https://sraf.nd.edu/textual-analysis/resources/>

## **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

## **Sentiment score categorization**

This is determined by grouping the Polarity score values in the following groups.

**Most Negative**: Polarity Score below -0.5

**Negative**: Polarity Score between -0.5 and 0

**Neutral**: Polarity Score equal to 0

**Positive**: Polarity Score between 0 and 0.5

**Very Positive**: Polarity Score above 0.5

# **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.

# **Passive Words**

Passive Words are the Auxiliary verbs followed by a word ending in “ed” or one of 200 irregular verbs present in the Past form column in this [link](http://www.worldclasslearning.com/english/irregular-verb-forms.html).

Auxiliary verbs are - auxiliary verb variants of “to be” including: “to be”, “to have”, “will be”, “has been”, “have been”, “had been”, “will have been”, “being”, “am”, “are”, “is”, “was”, and “were”.

# **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

**3.Code Output**

This Code outputs a file named Output\_solution.xlsx with 49 columns

The first 6 columns are static data which will be taken from cik\_list.xlsx

rest changes with each document.

It Prints the condition which the code prints according the the sections present in the document

Following are the columns

CIK

CONAME

FYRMO

FDATE

FORM

SECFNAME

mda\_positive\_score

mda\_negative\_score

mda\_polarity\_score

mda\_average\_sentence\_length

mda\_percentage\_of\_complex\_words

mda\_fog\_index

mda\_complex\_word\_count

mda\_word\_count

mda\_uncertainty\_score

mda\_constraining\_score

mda\_positive\_word\_proportion

mda\_negative\_word\_proportion

mda\_uncertainty\_word\_proportion

mda\_constraining\_word\_proportion

qqdmr\_positive\_score

qqdmr\_negative\_score

qqdmr\_polarity\_score

qqdmr\_average\_sentence\_length

qqdmr\_percentage\_of\_complex\_words

qqdmr\_fog\_index

qqdmr\_complex\_word\_count

qqdmr\_word\_count

qqdmr\_uncertainty\_score

qqdmr\_constraining\_score

qqdmr\_positive\_word\_proportion

qqdmr\_negative\_word\_proportion

qqdmr\_uncertainty\_word\_proportion

qqdmr\_constraining\_word\_proportion

rf\_positive\_score

rf\_negative\_score

rf\_polarity\_score

rf\_average\_sentence\_length

rf\_percentage\_of\_complex\_words

rf\_fog\_index

rf\_complex\_word\_count

rf\_word\_count

rf\_uncertainty\_score

rf\_constraining\_score

rf\_positive\_word\_proportion

rf\_negative\_word\_proportion

rf\_uncertainty\_word\_proportion

rf\_constraining\_word\_proportion

constraining\_words\_whole\_report

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*