Приложения

Приложение 1.

[1] – Обработка

from bs4 import BeautifulSoup as bs

import re

paths = [r'C:\Users\dimal\Desktop\DATA\AMD', r'C:\Users\dimal\Desktop\DATA\APPLE', r'C:\Users\dimal\Desktop\DATA\MarathonOil', r'C:\Users\dimal\Desktop\DATA\MasterCard', r'C:\Users\dimal\Desktop\DATA\UHS']

forms = [r'C:\Users\dimal\Desktop\clean\AMD', r'C:\Users\dimal\Desktop\clean\APPLE', r'C:\Users\dimal\Desktop\clean\MarathonOil', r'C:\Users\dimal\Desktop\clean\MasterCard', r'C:\Users\dimal\Desktop\clean\UHS']

years = [i for i in range(16, 22)]

for i in range(len(paths)):

for j in range(6):

with open(paths[i]+'\\'+str(years[j])+'.html', mode='r+') as k:

k = bs(k).text

k = re.sub(r"\s+", " ", k)

with open(forms[i]+'\\'+str(years[j])+'-1'+'.txt', mode='w+', encoding="utf-8") as v:

v = v.write(k)

[2] - Индекс удобочитаемости

from collections import defaultdict

import textstat

forms = [r'C:\Users\dimal\Desktop\clean\AMD', r'C:\Users\dimal\Desktop\clean\APPLE', r'C:\Users\dimal\Desktop\clean\MarathonOil', r'C:\Users\dimal\Desktop\clean\MasterCard', r'C:\Users\dimal\Desktop\clean\UHS']

fogs = defaultdict(list)

labels = ['AMD', 'APPLE', 'MarathonOil', 'MasterCard', 'UHS']

years = [i for i in range(16, 22)]

for i in range(5):

for j in range(6):

with open(forms[i]+'\\'+str(years[j])+'.txt', mode='r+', encoding="utf-8") as k:

k = k.read()

[3] – LM

import glob

import re

import string

import sys

import time

import csv

# Modify to identify path for custom modules

sys.path.append(r'C:\temp')

import MOD\_Load\_MasterDictionary\_v2022 as LM

# User defined directory for files to be parsed

TARGET\_FILES = r'C:\temp\Clean2/\*/\*'

# User defined file pointer to LM dictionary

MASTER\_DICTIONARY\_FILE = r'C:\temp' + \

'\Loughran-McDonald\_MasterDictionary\_1993-2021.csv'

# User defined output file

OUTPUT\_FILE = r'C:\temp\clresults.csv'

# Setup output

OUTPUT\_FIELDS = ['file name,', 'file size,', 'number of words,', '% positive,', '% negative,',

'% uncertainty,', '% litigious,', '% modal-weak,', '% modal moderate,',

'% modal strong,', '% constraining,', '# of alphabetic,', '# of digits,',

'# of numbers,', 'avg # of syllables per word,', 'average word length,', 'vocabulary']

lm\_dictionary = LM.load\_masterdictionary(MASTER\_DICTIONARY\_FILE, True)

def main():

f\_out = open(OUTPUT\_FILE, 'w')

wr = csv.writer(f\_out, lineterminator='\n')

wr.writerow(OUTPUT\_FIELDS)

file\_list = glob.glob(TARGET\_FILES)

for file in file\_list:

print(file)

with open(file, 'r', encoding='UTF-8', errors='ignore') as f\_in:

doc = f\_in.read()

doc\_len = len(doc)

doc = re.sub('(May|MAY)', ' ', doc) # drop all May month references

doc = doc.upper() # for this parse caps aren't informative so shift

output\_data = get\_data(doc)

output\_data[0] = file

output\_data[1] = doc\_len

wr.writerow(output\_data)

def get\_data(doc):

vdictionary = {}

\_odata = [0] \* 17

total\_syllables = 0

word\_length = 0

tokens = re.findall('\w+', doc) # Note that \w+ splits hyphenated words

for token in tokens:

if not token.isdigit() and len(token) > 1 and token in lm\_dictionary:

\_odata[2] += 1 # word count

word\_length += len(token)

if token not in vdictionary:

vdictionary[token] = 1

if lm\_dictionary[token].positive: \_odata[3] += 1

if lm\_dictionary[token].negative: \_odata[4] += 1

if lm\_dictionary[token].uncertainty: \_odata[5] += 1

if lm\_dictionary[token].litigious: \_odata[6] += 1

if lm\_dictionary[token].weak\_modal: \_odata[7] += 1

if lm\_dictionary[token].strong\_modal: \_odata[9] += 1

if lm\_dictionary[token].constraining: \_odata[10] += 1

total\_syllables += lm\_dictionary[token].syllables

\_odata[11] = len(re.findall('[A-Z]', doc))

\_odata[12] = len(re.findall('[0-9]', doc))

# drop punctuation within numbers for number count

doc = re.sub('(?!=[0-9])(\.|,)(?=[0-9])', '', doc)

doc = doc.translate(str.maketrans(string.punctuation, " " \* len(string.punctuation)))

\_odata[13] = len(re.findall(r'\b[-+\(]?[$€£]?[-+(]?\d+\)?\b', doc))

\_odata[14] = total\_syllables / \_odata[2]

\_odata[15] = word\_length / \_odata[2]

\_odata[16] = len(vdictionary)

# Convert counts to %

for i in range(3, 10 + 1):

\_odata[i] = (\_odata[i] / \_odata[2]) \* 100

# Vocabulary

return \_odata

main()

[4] – TextBlob

import glob

import textblob

files = glob.glob(r'C:\Users\dimal\Desktop\Clean2/\*/\*')

for file in files:

with open(file, encoding='UTF-8', mode='r') as form:

sent = textblob.TextBlob(form.read())

print(f'Тональность {file}: {sent.polarity}')

Приложение 2.

Список всех использованных модулей и библиотек.

Необходимо установить отдельно:

1. Pandas. URL: https://pandas.pydata.org/
2. NumPy. URL: https://numpy.org/
3. NLTK. URL: https://www.nltk.org/
4. TextStat. URL: https://textstat.readthedocs.io/en/latest/
5. TextBlob. URL: https://textblob.readthedocs.io/en/dev/
6. Beautiful Soup 4. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Внутренние модули Python:

re, glob, os, collections, csv, string, sys, time