Algorithm for

Measuring Corporate Sustainability by using Computer-Aided Text Analysis and Natural Language Processing

The algorithm can be downloaded directly from: https://github.com/nadjadamtoft/Measuring-Corporate-Sustainability-by-using-Computer-Aided-Text-Analysis-and-Natural-Language-Proces

This document provides important information related to the algorithm, as well as explains every step of the code. Please, be aware of the yellow marked words. These are names of files or folders and should be changed according to the files and folder you develop.

Installation requirements:

- python -m pip install --upgrade pip
- pip install --upgrade pip
- pip install -r requirements.txt
- pip install PyMuPDF
- pip-upgrade --skip-virtualenv-check

To-do:

- Create a folder named TextFiles
- Create a folder named Counts
- Save dictionaries under the names: Economicdictionary.xlsx, Environmentaldictionary.xlsx and Socialdictionary.xlsx

Step	Code	Description
1	import os import csv import re import pandas as pd	Imports

2	def cleanText(text):
	  text = text.lower()
	  text = re.sub(r"[^A-Za-z—\-\'\’\,]", ' ', text)
	  return re.sub(r"\s+", ' ', text)
		Function that receives a string. The transformation it applies on the string are the following:
		  1) Sets it to lowercase
		  2) Removes any characters except letters, —, -, ', ’, , and whitespaces
		  3) Replaces multiple whitespaces in just one
		
		  Parameters:
		  text - String
		
		  Output:
		  Cleaned String

3	dfEco = pd.read_excel(
	  "Economicdictionary.xlsx", sheet_name='Ark1', names=['words'])
	dfEnv = pd.read_excel(
	  "Environmentaldictionary.xlsx", sheet_name='Ark1', names=['words'])
	dfSoc = pd.read_excel(
	  "Socialdictionary.xlsx", sheet_name='Ark1', names=['words'])
	
	ecoPattern = "|".join(dfEco['words'].to_list())
	envPattern = "|".join(dfEnv['words'].to_list())
	socPattern = "|".join(dfSoc['words'].to_list())
	
	ecoWORD = re.compile(ecoPattern)
		#read the information in the excels into a dataframe
		
		#extract the words in a list and then transform them into a pattern
		
		#prepare pattern
		
		Re.compile compiles a regular expression pattern into a regular expression object

	envWORD = re.compile(envPattern) socWORD = re.compile(socPattern) WORD = re.compile(r'[A-Za-z—\-\'\']*')	

4	%%time
	with open("Counts/CorporateSustainability.csv", "w+", newline="", encoding='utf-8') as csv_file:
	
	csv_file.write("%s,%s,%s,%s\n" % ('file', 'Economic sustainability',
	'Environmental sustainability', 'Social sustainability'))
	
	for root, dirs, files in os.walk("TextFiles"):
	for file in files:
	if file.endswith('.txt'):
	filePath = open('TextFiles/'+file,
	'r', encoding='utf-8')
	text = filePath.read()
	cleanedText = cleanText(text)
	
	ecoTokens = len(re.findall(ecoWORD, text))
	envTokens = len(re.findall(envWORD, text))
	socTokens = len(re.findall(socWORD, text))
	
	totalTokens = len(re.findall(WORD, text))/500
	csv_file.write("%s, %.4f, %.4f, %.4f\n" % (
	file, round(ecoTokens/totalTokens, 4),
	round(envTokens/totalTokens, 4), round(socTokens/totalTokens, 4)))
		#replace csv file if it already exists, otherwise create
		
		#headers
		
		#go through files
		
		#find the words and count them
		
		#normalization factor of 500