Manual comparePDF

jan hellings | manual | 29 juni 2022

Introduction

This manual is about a python program comparing PDF files by converting the files to TXT files and compare the TXT files. It scans through a directory tree of PDF files. The PDF files are converted into TXT files and stored in a separated directory.

These TXT files are then compared and a score is giving to the similarity. The score is from 0-1. Two output files are generated. One with all the scores and one with the high score. This high score value is given by the user.

The output files are CSV files and can be imported into Excel.

System overview

The system consists of two python programs developed in Python 3.9. The first program is the program itself: **comparepdf_text.py**. The second program is the **Run_comparepdf_text.py**. The first program is the functional program, the second gives the input for the program so that you can run the first with one simple command.

You have to change the **Run_comparepdf_text.py** program to meet your local requirements.

System references

*It is develop with Visual Studio Code on a windows 10 PC. It should run with Pyton 3.9. It needs **pdfminer.** To install type:*

pip install pdfminer

Authorized use of system

Use it at your own risk. It has been tested but no guarantee

Points of contacts

Any comments van be sent to

jan@famhellings.nl

Use of the program

*To use the program you have to adapt the file **Run_comparepdf_text.py** (*Figure 1*)*

```
import os

"""
    D:\Surfdrive\MyWork\Leiden\Python> & C:/Users/jan/AppData/Local/Microsoft/WindowsApps/python3.9.exe d:/Surfdrive/MyWork/Leiden/Python/comparepdf_text.py -h
   usage: comparepdf_text.py [-h] -i1 PDFDIR1 -o1 TXTDIR1 -out OUTFILE -sh SIMHIGHSCORE -o3 OUTFILEHIGHSCORE

optional arguments:
  -h, --help            show this help message and exit
  -i1 PDFDIR1, --pdfdir1 PDFDIR1
                        first pdf directory to check similarity
  -o1 TXTDIR1, --txtdir1 TXTDIR1
                        Directory1 to store txt files to compare
  -out RESULTFILE --resultfile RESULTFILE
                        File with similarity score
  -sh SIMHIGHSCORE, --SimHighScore SIMHIGHSCORE
                        High score number 0,0-1,0
  -o3 resultfileHIGHSCORE, --resultfileHighScore resultfileHIGHSCORE
                        File with high score similarity

"""


run_string = ' "F:/Program Files/Python39/python.exe" ' \
    + ' d:\\users\\jan\\Surfdrive\\MyWork\\Leiden\\Python\\comparepdf_text.py' \
    + ' -i1 D:\\Users\\jan\\Surfdrive\\MyWork\\Leiden\\FICT\\controle_oplevering\\'\
    + ' -o1 d:\\users\\jan\\Surfdrive\\MyWork\\Leiden\\FICT\\21-22\\Run02\\outdir_txt\\ '\
    +  ' -out d:\\users\\jan\\Surfdrive\\MyWork\\Leiden\\FICT\\21-22\\Run02\\score.csv\\' \
    +  ' -sh 0.4' \
    +  ' -o3 d:\\users\\jan\\Surfdrive\\MyWork\\Leiden\\FICT\\21-22\\Run02\\highscore.csv'

print (run_string)
os.system(run_string)
```

Figure 1 Run_comparepdf_text.py to run the comparepdf_text.py program

After the modification of the program **Run_comparepdf_text.py** the program can be run:

"F:/Program Files/Python39/python.exe" d:/Users/jan/Surfdrive/MyWork/Leiden/Python/Run_comparepdf_text.py

The output files will be store in the directory given by the **-O3** and the option **-OUT** As you can see in Figure 1.

The high score is given by the **-sh** in the example Figure 1 it is **0,4**

There are two files created

d:\\users\\jan\\Surfdrive\\MyWork\\Leiden\\FICT\\21-22\\Run02\\score.csv\\'

and

d:\\users\\jan\\Surfdrive\\MyWork\\Leiden\\FICT\\21-22\\Run02\\highscore.csv