University of St.Gallen

School of Management, Economics, Law, Social Sciences,
International Affairs and Computer Science (HSG)

Individual project

Document similarity checker based on cosine similarity in Python

Dr. Mario Silic

7-789-1.00 - Skills: Programming with Advanced Computer Languages

---

Submitted: 26.12.2021

by

Jan-Philipp Wittmann

16-621-120

GitHub rep:

https://github.com/JPWJPW25/7-789-1.00_Advanced-Programming_Python

Example Outline: Specifications

Documents used:

1) Doc1_FCB
   Description: Snippet from Wikipedia article of FC Bayern Basketball team
   Type: PDF
   Source: https://en.wikipedia.org/wiki/FC_Bayern_Munich_(basketball)

2) Doc2_FCBB
   Description: Snippet from Wikipedia article of FC Bayern soccer team
   Type: Word
   Source: https://en.wikipedia.org/wiki/FC_Bayern_Munich

3) Doc3_Roses
   Description: Snippet from Wikipedia article about roses
   Type: PDF
   Source: https://en.wikipedia.org/wiki/Rose

4) Doc4_Roses_2
   Description: Snippet from Wikipedia article about roses but different paragraph
   Type: Word
   Source: https://en.wikipedia.org/wiki/Rose

Example Outline: Input (in green) and Output

Welcome to the document similarity checker. The program lets you compare two documents and compute their similarity. Based on the similarity you can for example identify plagiats or copies of text that just have been modified by changing word orders.

In a first step, the program loads documents of the type Word and PDF. Second, after loading thedocuments, the program converts the files into text files. Third, based on the created text files, the program preprocesses the data before computing the similarity with Natural Language Processing (NPL) techniques. More specifically, the program computes the similarity based on theso called cosine similarity. Fourth,
the results of the similarity analysis are loaded into dataframe and plotted using a upper triangle heatmap, that can be saved as png by the user. Lastly, the user has the opportunity to start the program again.

Please input the path to the folder that contains the documents you want to compare here:
C:/Users/jan-p/OneDrive/Desktop/MBF/Course Work/03_HS22/MBF_Advanced programming/Example outline

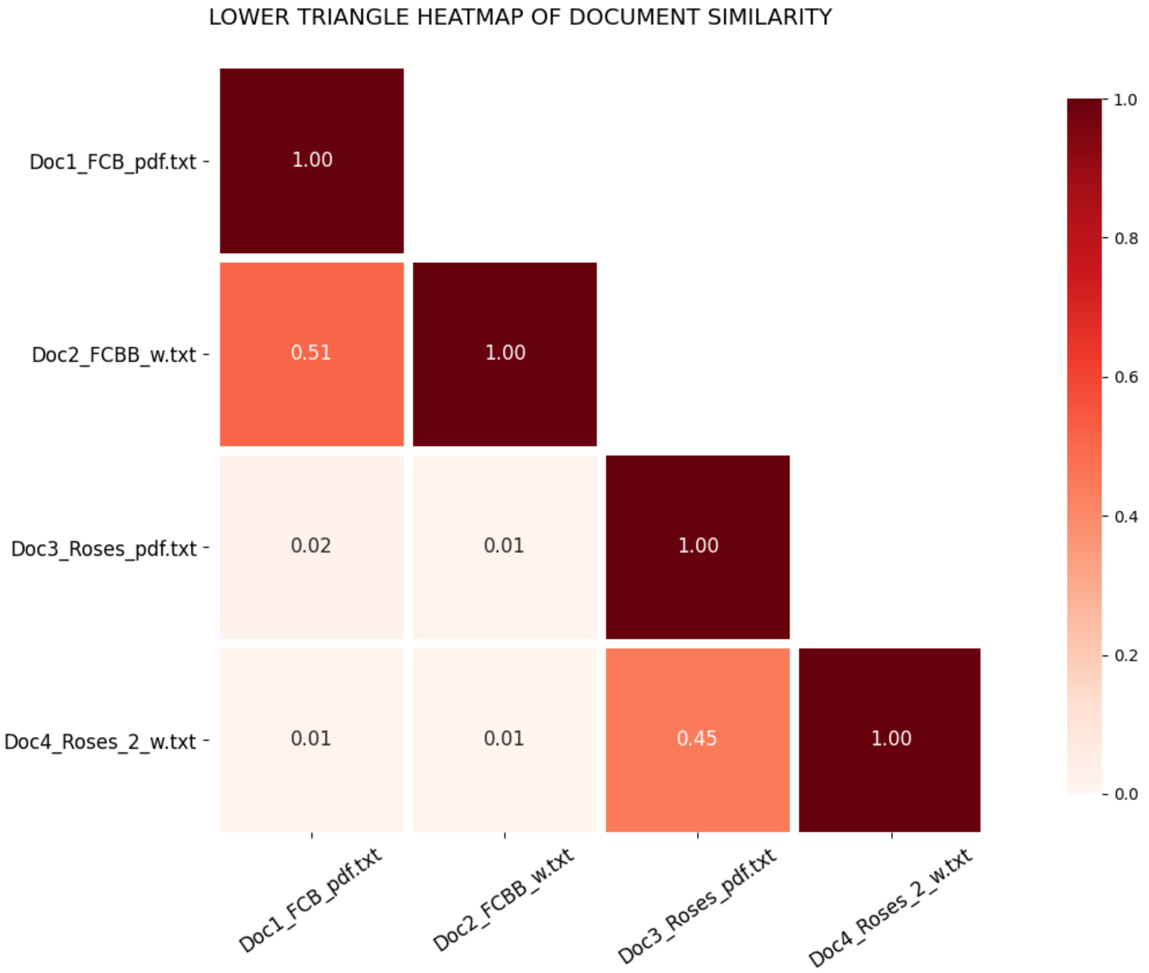
How many documents do you want to compare?
4

What is the name of document Nr.1?
Doc1_FCB

What is the type of document Nr.1? (.docx / .pdf)
.pdf

What is the name of document Nr.2?

```
Doc2_FCBB

What is the type of document Nr.2? (.docx / .pdf)
.docx

What is the name of document Nr.3?
Doc3_Roses

What is the type of document Nr.3? (.docx / .pdf)
.pdf

File does not exist. What is the name of document Nr.4?
Doc4_Roses_2

What is the type of document Nr.4? (.docx / .pdf)
.docx

Do you want to save the plot as png file in your folder? (Yes/No)
Yes
Perfect, the file has been saved to your folder.

Do you want to see the plot? (Yes/No)
Yes
```

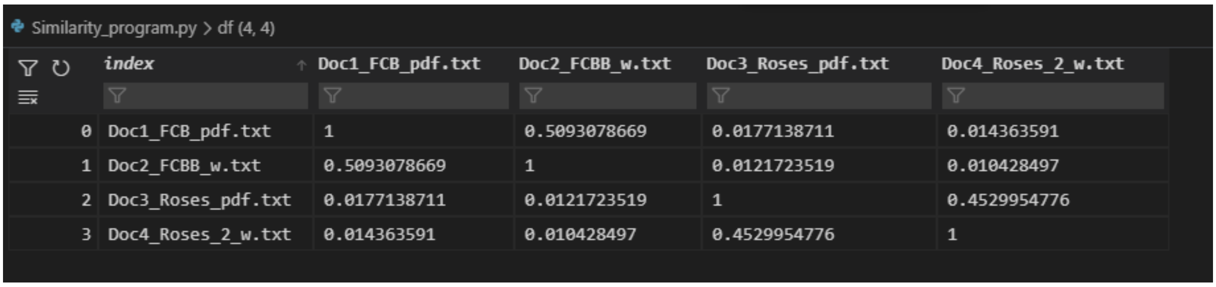


Do you want to restart the program? (Yes/No)
No

Alright, thank you very much for using the program. Goodbye!

Add-on:

If the user changes the code and sets a break point in line 301, the dataframe can be viewed via the Data Viewer function in Visual Studio Code.



Similarity_program.py > df (4, 4)

	index	Doc1_FCB_pdf.txt	Doc2_FCBB_w.txt	Doc3_Roses_pdf.txt	Doc4_Roses_2_w.txt
0	Doc1_FCB_pdf.txt	1	0.5093078669	0.0177138711	0.014363591
1	Doc2_FCBB_w.txt	0.5093078669	1	0.0121723519	0.010428497
2	Doc3_Roses_pdf.txt	0.0177138711	0.0121723519	1	0.4529954776
3	Doc4_Roses_2_w.txt	0.014363591	0.010428497	0.4529954776	1