

PLAGIARISM CHECKER

Submitted by:

Prachika Kanodia (5th Sem, J.K. Lakshmipat University, Jaipur)

Tanmay Agarwal (5th Sem, J.K. Lakshmipat University, Jaipur)

Objective:

The objective is to achieve is to design a system which when provided any doc or pdf file, check for Plagiarism in the content and revert for the same, along with the sources, from where the content has been copied.

Overview:



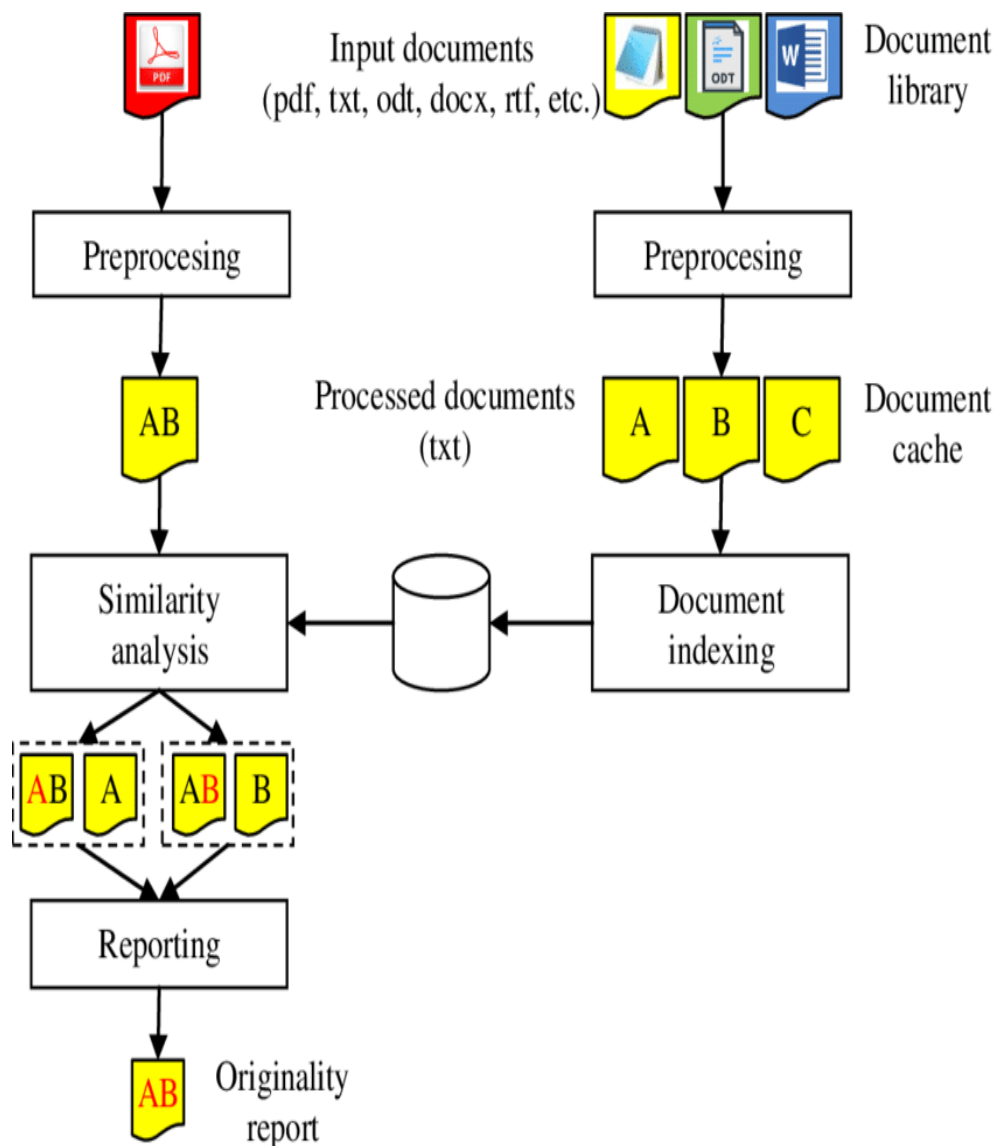
Plagiarism is defined as to take or copy some work and present it as one's own work. Plagiarism affects the education quality of the students and thereby reduce the economic status of the country. Plagiarism is done by paraphrased works and the similarities between keywords and change of sentences from one form to other form, which could be identified using plagiarism detection system. This plagiarism detector measures the similar text that matches and detects plagiarism. Internet has changed the student's life and also has changed their learning style. It allows the students to get deeper in the approach towards learning and making their task easier.

In this pandemic where online teaching plays a very important role but some of the students take the advantage of it by copying from internet and from each other which makes difficult for the professors to analyse them so to make it easier for them to check and analyse one's knowledge in that plagiarism detection system can be very helpful which can detect the percentage of plagiarism and then accordingly he/she can be evaluated.

Now there are many technologies which can be used to design this plagiarism detection system. We have used Python, Data Science and Machine Learning as a technology stack to

design this system which includes Data Exploration, Engineering, Training and Testing, deploying of the model using support of different libraries. The system designed can able to analyse the document uploaded and give the percentage of plagiarism and other relevant information as an output.

The basic process to be followed to create the model for plagiarism detection system are as follows: -



Libraries used:

- SequenceMatcher module from difflib library which is used to compare any datatypes sequences other than the hashable sequences.
Installation: `pip install difflib`
- re library is used for the textual data analysis in python i.e., for any kind of document analysis.
Installation: `pip install re`
- nltk library is used to implement the plagiarism detection system with the help of N-gram language model for which different modules are used for different process involved in it.
Installation: `pip install nltk`
- `plotly.graph_objects` is used for heatmap plotting to get probability ranging from 0 to 1 of plagiarism in which aesthetics are given using `gaussian_filter` module from `scipy.ndimage` library.
Installation: `pip install plotly`
`pip install scipy`
- `docx2txt` and `doc2txt` library are used for the detection of specific extensions files i.e. `.docx` and `.doc` after uploading it.
Installation: `pip install docx2txt`
`pip install doc2txt`
- `textract` and `pypdf2` library used for the extension of `.txt` and `.pdf` respectively.
Installation: `pip install textract`
`pip install pypdf2`
- `random` module is used to generate random numbers in python program.
Installation: `pip install random`
- Copyleaks SDK which provides API Key along with ID after which the files are scanned for plagiarism detection and gives the percentage and other relevant information in `.json` file for that particular data file provided for scanning.
Installation: `pip install Copyleaks`

Plagiarism detection system for comparing files stored locally:

Using SequenceMatcher module from difflib which is used to compare the any datatype sequence so two files are compared for plagiarism i.e., similarity to get the percentage as an output.

Now to get visual representation of plagiarism detection N-gram language model is created which scores words based on the preceding window of context. To implement it nltk library is used which provide different modules for steps involved in it. It was noticed that window of 4 (N=4 i.e., N-grams) worked well in comparing the words from one file to other as it aligns with the advice of many teachers not to use more than three words in a row from a source.

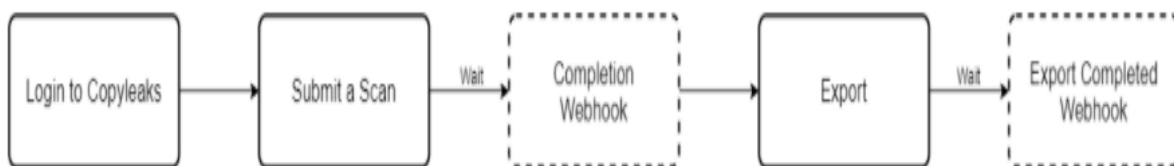
In the training process first all the punctuation and formatting are removed from file provided after that word is tokenize i.e., separating in individual then using .ngrams division is done in the group (N-1=3 i.e., Group of words). And to deal with unseen N-grams MLE i.e., Maximum Likelihood Estimator and WittenBellInterpolated module is used.

After training to do the testing, file is provided and same process is repeated as in training. Now to visualize plotely and scipy lib is used i.e., to get the heatmap representation. In the map in which width and height is specified and we get per line (K=8 i.e., Words per line) result means probability ranging from 0 to 1 is labelled. Many other functions are used to create the visualization in an effective way.

Also, for different extensions of files different lib can be used as per the requirement i.e., docx2txt for .docx, doc2txt for .doc, .pdf, .txt, etc.

Copyleaks API for comparing files from internet:

To check files against internet sources, Copyleaks SDK is used as it is an open source pre built API which can be integrated with the project. It takes up file encoded in base 64 and checks against the Copyleaks database for plagiarism. It also checks the internet by querying searches based upon the phrases from text. The AI automatically picks up sentences and phrases from the text and launches it against the internet to get back search queries. These are used as reference to check for plagiarism.



The flowchart shows how to integrate Copyleaks SDK in your project.

Copyleaks provides easy integration and customization of reports as per your accord making it highly versatile. It is also available in almost all platforms making it to be used globally.

Output:

Similarity index between two files stored locally.

```
if extension1 == '.pdf' and extension2 == '.pdf':
    file = open(file1path, 'rb')
    fileReader = PyPDF2.PdfFileReader(file)

    file1 = ''
    file2 = ''

    for i in range (fileReader.numPages):
        pageObj = fileReader.getPage(1)
        filepdata = pageObj.extractText()
        file1 = file1 + filepdata

    file = open(file2path, 'rb')
    fileReader = PyPDF2.PdfFileReader(file)

    for i in range (fileReader.numPages):
        pageObj = fileReader.getPage(1)
        filepdata = pageObj.extractText()
        file2 = file2 + filepdata

    similarity = SequenceMatcher(None, file1, file2).ratio()
    print("Percentage similarity:", similarity*100)

elif extension1 == '.docx' and extension2 == '.docx':
    file1 = docx2txt.process(file1path)
    file2 = docx2txt.process(file2path)
    similarity = SequenceMatcher(None, file1, file2).ratio()
    print("Percentage similarity:", similarity*100)

elif extension1 == '.txt' and extension2 == '.txt':
    with open(file1path) as file1text, open(file2path) as file2text:
        file1 = file1text.read()
        file2 = file2text.read()
        similarity = SequenceMatcher(None, file1, file2).ratio()
        print("Percentage similarity:", similarity*100)

else:
    print("Enter either a docx or pdf file and make sure both files are in same format")
```

Percentage similarity: 100.0




Sample Plagiarism Report in the form of heatmap. (for comparison between 2 files)


CS1105 Design and Analysis of Algorithm Institute of
Engineering Technology IET JK Lakshmipat University Jaipur S
by Submitted to Prachika Kanodia 2019btechcse068 Mr Santosh
Kumar Verma Assignment ASSIGNMENT5 Optimization versus search
the traveling salesman problem TSP Input A matrix
of distances a budget b Output A tour
which passes through all the cities and has
length b if such a tour exists The
optimization version of this problem asks directly for
the shortest tour TSPOPT Input A matrix of
distances Output The shortest tour which passes through
all the cities Show that if TSP can
be solved in polynomial time then so can
TSPOPT Ans Let TSPHb will returns false if
no tour of length b or less than
b exists in H Now TSPOPTh i0 for
all tu belongs to Z i i disttu
return BINARYSEARCHTOURH0i BINARYSEARCHTOURH1t b 1 1 t2 if
TSP Hb false return BINARYSEARCHTOURH1b else return BINARYSE
So this algorithm will do a binary search
using all the lengths So if TSP is
solved using polynomial time then TSPOPT can also
be solved using polynomial time We have to
make polynomial number of calls from varying out
output b using binary search algorithm Binary Search
Algorithm is important right here and we cant
simplify increment the value of b by 1
because the sum of distances is exponentially less
than the length of the input Now using



Sample Plagiarism Report for file from internet.

29469



 **Properties**
Scan Properties

Action: Scanned

Duration: 11 seconds


Status: COMPLETED

Pages: 0


Expiration Date: Dec 19, 2021

Similarity Score: 10%

Total Words: 10

 **Artifacts**
Click to download

[!\[\]\(4cf4858f0f33d9147b4f89d7334365ec_img.jpg\) Crawled version](#)[!\[\]\(b429a63e41c4330a6b4db9bc6268fd87_img.jpg\) Completion webhook](#)

 **Results**
Found 1 results

Result Id	Copied Words	Percentage
2a1b402420	1	10%

The output is shown in two forms. One is the similarity index which shows the similarity between 2 files (multiplied by hundred to give percentage). 0 being the least and 1 being exact copy.

The other heatmap graph shows line by line with each line having 8 words. The probability of plagiarised content ranges from 0 to 1. This is more of a general overview if two files are copied and what parts are copied the most.

The result ID generated shows all the plagiarism information regarding the files detected along with many other relevant information in the .json file.

GitHub Link:- https://github.com/Intern-NetparamTechnologies/Plagrism_Checker

Conclusion:

We were able to build a simple plagiarism detection system and with the help of Copyleaks API we were able to implement plagiarism detection over internet sources.