Journal of Advanced Research in Computer Technology and Software Applications
Volume 7, Issue 1 - 2023, Pg. No. 12-14
Peer Reviewed Journal

Review Article

Advancements in Plagiarism Detection: Exploring the Development and Evaluation of an IEEE Paper

Ashish

Department of Computer Science & Engineering Chandigarh University, Gharuan, Punjab, India.

INFO

E-mail Id:

How to cite this article:
Ashish. Advancements in Plagiarism Detection: Exploring the Development and Evaluation of an IEEE Paper. *J Adv Res Comp Tech Soft Appl* 2023; 7(1): 12-14.

Date of Submission: 2023-05-01
Date of Acceptance: 2023-06-12

ABSTRACT

In the current project, I plan to create a plagiarism detector that will inform us how unique the article is. As students, we submit assignments, homework, other academic materials, thus we must guarantee that our work does not duplicate with that of others. To avoid this, we must utilize a plagiarism checker so that we do not become involved in benign plagiarism. Teachers must know if students are completing tasks on their own while evaluating them. As a result, a plagiarism detector is extremely useful in educational contexts.

Keywords: Detector, Plagiarism

Introduction

In the rapidly developing world, we are moving towards online education. During the pandemic people have moved towards different online platforms. E-learning is great which provides a platform where we can study on our own leisure.[1] As the students are submitting assignments and tests online, it is necessary to check if someone has copied or plagiarized the content. There are cases of innocent plagiarism where a person has unknowingly used some text. So, to avoid that and to check if someone has stolen the content of another manuscript or copyright content, plagiarism checker comes into play.[2-4]

Plagiarism Checker is made in python language. It uses natural language processing libraries and google API to know how much content is copied. Plagiarism Checker project is economic and highly beneficial, as far as the cost of development is considered. No extra costs were incurred apart from the software used while making of the project.[4-7]

For the creation of GUI of the app I have used Tkinter. Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI. Tkinter is included with standard GNU/Linux, Microsoft Windows and macOS installs of Python.[8-14]

In phase one we use rake library to perform keyword extraction from the original text document. RAKE (Rapid Automatic Keyword Extraction) algorithm is a domain independent keyword extraction algorithm which tries to determine key phrases in a body of text by analysing the frequency of word appearance and its co-occurrence with other words in the text Figure 1.

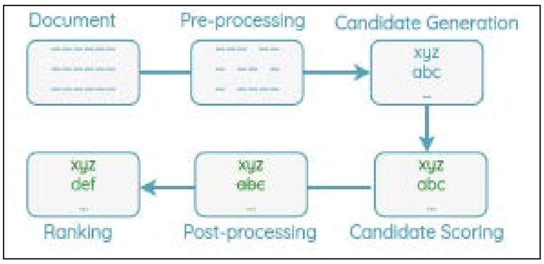

Figure 1

In phase two, I have used custom search engine from google to search for those keywords and get the top results. To achieve this, I have used google API client library which is a third-party library officially recognized by Google. Using this library, I create an object of client and store the web data acquired from search engine in this object. Then I extract links from the data Figure 2.

In phase three, I have used requests library to get the data of the whole website from the links. Further I extract only the text data from it and store it in the object of beautiful soup. Now I extract all the data present in the paragraph tags of the websites and store it in a text file Figure 3.

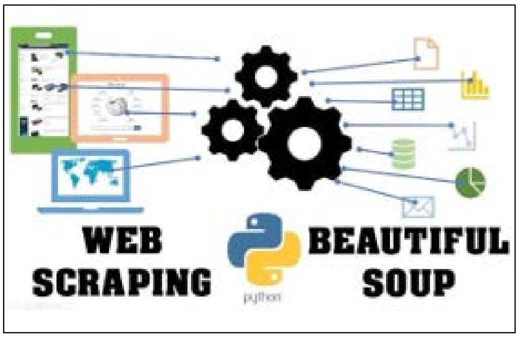

Figure 2

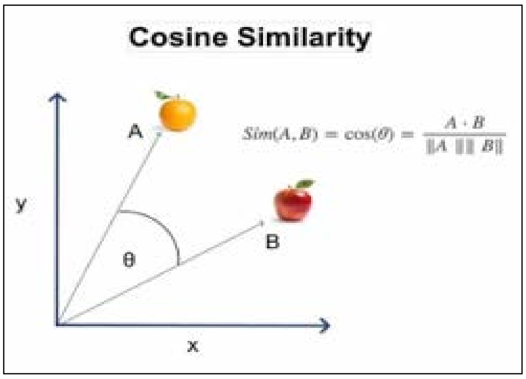

Figure 3

In phase four, I open the original text document and the document in which the data from search result is stored. I tokenise both the documents and convert all the words in lowercase. Now using NLTK library I attempt to remove all the morphological terms and stop words from those documents. After this I change all the document in dictionary. Both the dictionary in a list is now merged. Two vector spaces of both the documents are then created. Then the cosine similarity between two vector space is calculated Table 1.

Table I

S.No.	Methodology
1.	Importing all the necessary modules
2.	Using Tkinter creating the GUI for app
3.	Taking input of the text document
4.	Extracting keywords from the document
5.	Searching those keywords using custom searchengine
6.	Scraping the data from Google's top results using beautiful soup and storing it in a text file
7.	Using NLTK and NumPy we calculate cosine similarity between the documents

Existing System

In most of the existing approaches used at present, all the plagiarism checker use string by string search on the internet. It works fine but if the document is large, it can be slow and inefficient.

Proposed System:

In the development of my project, I have used natural language processing libraries which are constantly being updated as ML is the hot topic globally. I use NLP libraries to extract keywords which are fast and efficient. After that all the keywords are searched on the internet using Custom search api on the web and get the top results. We then use beautiful soup to web scrape the data and store it in the file. Now we compare those files with cosine similarity.

Competitive Analysis

Prior to getting started with the plagiarism checker project, I did some research on the current systems of plagiarism detection out there. I was looking forward to building a unique experience rather than an exact clone of an existing plagiarism platform. I already knew the existence of several existing platforms, a few applications that befitted paid customers. However, never I had done an in- depth analysis of their tools to find out whether they were good enough for us. Soon, I realized that none of the sites were heading in my direction.

Some of the explored sites were missing features which we considered crucial, others had opportunities for further enhancements. Contrary to what many people think, having a few platforms around is not necessarily a bad thing. I was able to get ideas of what to build and how and determine which technologies and strategies to use based on their experience. Often, this was as simple as checking their blogs/websites.

Acknowledgment

If words are considered as a symbol of approval and token of appreciation, then let the words play the heralding role of expressing my gratitude. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. I am grateful to my Project Mentor and Project Teacher

"Er. Kritika Sharma" for her guidance, inspiration and constructive suggestions that helped me in the preparation of the project. She was always there to listen my problems and to give me advice. She showed me different ways to approach a project problem and the need to be persistent to accomplish any goal and guide me about the project and what steps to be followed to make the project. She also taught me how to write project synopsis, mid-term report and final project report and had confidence in me.

References

1. https://www.wikipedia.org/.
2. https://www.geeksforgeeks.org/.
3. https://amitness.com/keyphrase-extraction/.
4. https://laptrinhx.com/how-recommendation- systems-work-1825654218/.
5. https://dev.to/jessesbyers/someone-stole-my-dev-article-how-to-build-a-python-script-to-detect- stolen-content-54fg.
6. https://towardsdatascience.com/the-best-document-similarity-algorithm-in-2020-a- beginners-guide-a01b9ef8cf05.
7. https://www.nltk.org/.
8. https://pypi.org/project/rake-nltk/.
9. https://docs.python-requests.org/en/latest/.
10. https://www.crummy.com/software/BeautifulSoup/bs4/doc/.
11. https://www.tutorialspoint.com/index.html.
12. https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient.
13. https://github.com/googleapis/google-api-python-client.
14. https://console.developers.google.com/.