IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117114

Plagiarism Detection using Natural Language Processing and Support Vector Machine

Nikhil Sandilya[1], Rishabh Sharma[2], and Merin Meleet[3]

R.V. College of Engineering, Bengaluru, Karnataka, 560059[1,2,3]

Abstract: Plagiarism is the practice of using someone else's words or ideas as one's own. In many nations, plagiarism is considered to be a violation of moral rights. The unacceptable act of plagiarism has been rising significantly in today's environment of developing technology and expanding Internet usage. It is frequently seen in a variety of academic contexts, including research papers, blogs, essays, assignments, etc. In this paper we employed two ways of finding plagiarized text. One method focuses on building a plagiarism detector that examines a specified response text file against a source text file and, depending on the similarities between the two text files, identifies the answer text file as original or plagiarized. In order to create a binary classification model and identify plagiarism, a Support Vector Machine (SVM) was employed. Another method focuses on creating a web application that can identify plagiarism in text, offering a sentence-by-sentence analysis with the percentage of plagiarism and a link to a potential source article, including a method to check for source code plagiarism within a directory.

Keywords: SVM, NLP, Machine learning, Plagiarism Detection, n-grams containment.

1 INTRODUCTION

Plagiarism is defined as "the stealing of another person's words or ideas without properly referencing them and hence without accurately attributing the original author." Depending on how much the original wording was altered.
Plagiarism occurs in a variety of contexts, including books, music, software, academic papers, newspapers, commercials, websites, etc. At spite of the penalties imposed for plagiarism and cheating in Bulgarian institutions, more than 50% of the lecturers think that these measures are ineffective. With more people using the internet, maintaining academic integrity in schools and institutions is becoming increasingly difficult. Thus, it has become imperative for many higher education institutions to deploy effective plagiarism detection software. However, the success of these systems for detecting plagiarism rests on their capacity to identify various fraudsters' tactics for modifying the text without altering its meaning.
A straightforward supervised machine approach called Support Vector Machine (SVM) is employed for text categorization. Text classification is a key component of natural language processing and involves the act of classifying or categorizing text data into categories. We are surrounded with text in the digital world we currently live in, whether it be on our social media accounts, in advertisements, on websites, in Ebooks, etc. Since much of this text data is unstructured, categorizing it may be quite helpful.
The system proposed is a machine learning-based model. It uses two methods, One method compares a given response text file to a source text file and determines whether the answer text file is original or plagiarized based on the similarities between the two text files. A Support Vector Machine (SVM) was used to develop a binary classification model to detect plagiarism. Another method focuses on creating a web application that can identify plagiarism in text, offering a sentence-by-sentence analysis with the percentage of plagiarism and a link to a potential source article, including a method to check for source code plagiarism within a directory.
Through online lectures, homework projects, and exams, the COVID-19 pandemic epidemic has demonstrated how reliant on technology the whole educational system is. By using this idea, it would be simpler to detect plagiarism in student projects and online tests.

2 LITERATURE REVIEW

A system for detecting plagiarism was proposed by El Mostafa Hambi and Faouzia Benabbou in [1] and is based on three deep learning models: Doc2vec, Siamese Long Short-term Memory (SLSTM), and Convolutional Neural Network (CNN). The discovered model is capable of detecting not only the presence of plagiarism but also the likelihood that each form of plagiarism would exist.

With the use of pre-existing machine learning libraries, Hiten Chavan, Mohd. Taufik, Rutuja Kadave, and Nikita Chandra set out to create a plagiarism detector in [2]. It utilizes the Tf Idf Vectorizer and the SciKit package to transform text into vector form, after which the dot product or cosine is calculated. This outcome shows us how the two text files, or in this

DOI: 10.17148/IJARCCE.2022.117114

case, vectors, share the same data. The system has been displayed using a UI, according to the results. The text file input is accepted. Then plagiarism detection software scans the submitted files. The system outputs a "Similarity Score" that illustrates how similar the two text files are to one another. The score might be anywhere from 0 to 1.

This work's contribution was examined by Siddharth Tata, Suguri Charan Kumar, and Varampati Reddy Kumar in [3]. It is an assessment method that automatically checks student assignments for plagiarism. utilizing the Winnowing, Rabin, and Karp algorithms, as well as Jaccard similarity. With a consistent window size and a change in n-gram size, it clearly demonstrates the variance in the proportion of plagiarism. demonstrates how the proportion of plagiarism varies as the window size is changed while the n-gram size remains fixed.

A semantic plagiarism detection technique was proposed by Ahmed Hamza Osman and Omar M. Barukab in [4] in light of Semantic Role Labeling (SRL) and Support Vector Machines (SVM). Compares the SRL-SVM with graph-based approach, fuzzy semantic string similarity, and LCS for semantic similarity detection in terms of time efficiency. The recommended method's time effectiveness was also determined, and it belongs to the O(n^2) Class.

This paper presents the results of a comparison analysis on plagiarism checking techniques with the technology employed by Keerthana T V, Pushti Dixit, Rhuthu Hegde, Sonali S K, and Prameetha Pai in [5]. It can see how automatic, algorithm-based plagiarism detection has advanced from manual plagiarism detection. can see how utilizing local client apps changed over time to using web-based applications, and now cloud-based applications. The basic feature-based approach, structure-based approach, similarity measurement, and string-matching algorithms were the initial detection techniques. Recent efforts have improved the efficiency and automated the process of plagiarism detection using machine learning and deep learning algorithms and methodologies.

To emphasize further on text matching systems, Tomá Foltynek, Dita Dlabolová, Alla Anohina-Naumeca, and Salim Razi in [6] concentrate on analyzing the performance of the systems on single-source and multi-source documents. System performance was improved for a certain language or language family. The system's performance varies depending on where the copied text was found. The computers seem to be more adept at detecting similarity in papers with several sources than in those with only one.

In [7], Marwah Najm Mansoor and Mohammed S. H. Al-Tamimib tried to address the problem of plagiarism in the academic world. Check Papers that compare non-textual content elements such citations, photos, tables, and mathematical equations using lexical, syntactic, and semantic similarity analysis. Techniques may quickly find hundreds of code lines using the C, C++, and Java programming languages. It stores many academic resources in its database system and generates a thorough report whenever plagiarism is found.

To record non-contiguous interaction inside n-grams, Jitendra Yasaswi, Suresh Purini, and CV. Jawahar collaborated in [8]. classification of assignment programme submissions as copies, partial copies, or non-copies using effective deep features. a job of binary classification (including only copy and non-copy cases). a three-way categorization (comprises copy, non-copy, and partial-copy situations).

3 PROPOSED SYSTEM

Since the advent of artificial intelligence, several techniques—from supervised, unsupervised machine learning methods to deep learning—have been suggested and effectively used in a wide range of industries. Models with numerous processing layers that can learn data representations with many degrees of abstraction are provided by in-depth learning. Many deep learning applications for NLP domains have recently been presented, and their performance in areas like chatbot programming, sentiment analysis, and question-answering has been highly encouraging.
In this context, we propose an online plagiarism detection system based on n-grams containment, Support Vector Machine, WebScraping and NLP Tools.

When the software starts a window comes up which is the UI. A text area is provided where the user can enter the text to be checked whether it is plagiarized or not from the Internet, or put the document in the directory to examines a specified response text file against a source text file and, depending on the similarities between the two text files, identifies the answer text file as original or plagiarized. The UI has been created using Flask.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117114

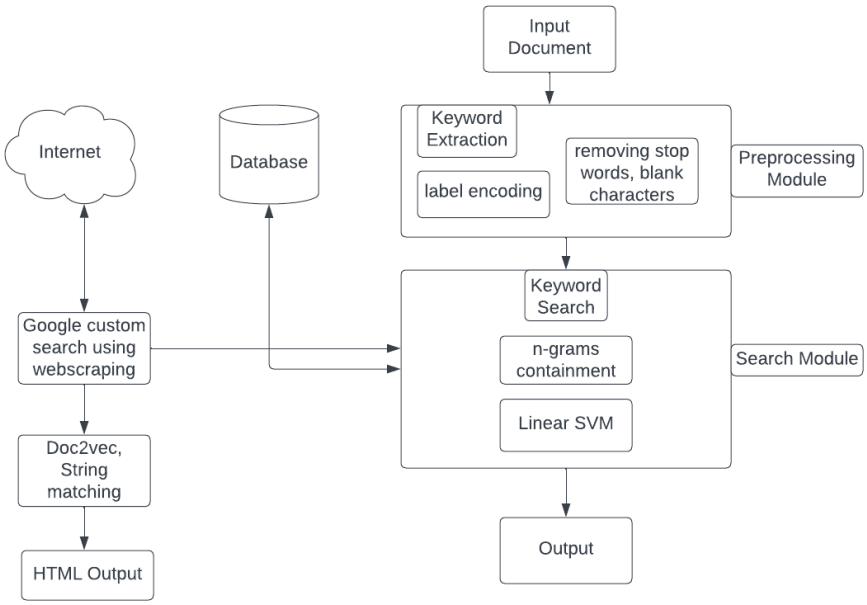

Fig 1. Architecture Diagram for Plagiarism Detection System

4 METHODOLOGY

A built-in library for machine learning tools is called Sci-kit-learn. The suggested approach for feature extraction from the text uses this library. The ability to interpret and scrape HTML and XML documents is made possible by the Python web scraping package known as Beautiful Soup. A parser allows you to search, explore, and edit data. It is adaptable and considerably reduces time. Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
The methodology comprises three major steps which are described as follows:

1. Data collection and preprocessing- For the first method the dataset created by Clough and Stevenson [9] in which created a corpus of responses with fake plagiarism in them. The inclusion of 4 different categories of plagiarism—near copy, light revision, heavy revision, and non-plagiarized—is the dataset's key advantage. 100 text documents make up the data file, of which 5 have the answers from the original source. As a result, the participants gave 95 answers, split across 5 tasks and 5 instances of plagiarism.
One of the five learning tasks (A-E) that each txt responds to is listed in the Task column.
If the participant was instructed to utilize a Near Copy (cut), Light Revision (light), Heavy Revision (heavy), or Non-plagiarized (non) technique to answer the question, it is indicated in the Category column. The 'orig' category in this column serves as a reminder of the original texts participants used as sources for their responses.
For the second method web scraping of whole google is done with the constraints applied. Web Scraping is done using BeautifulSoup python library, Web scraping is the process of extracting data and processing it from numerous websites. In other words, it's a method for extracting unstructured data and storing it in a database or a local file.

2. Model building and training- Text contained in the dataset was divided into non, cut, heavy, and light categories. It was divided into a testing set that made up 20% and a training set that made up 75%. It is labeled non=0, heavy=1, light=2, cut=3, and orig=-1 first.
To prevent plagiarism, data is preprocessed by constructing a vector subset and storing the data as a pair of index and data type variables. Additional spaces, tab characters, extra lines, and other unused white spaces are eliminated.
This is accomplished using Similarity Features based on N-grams Containment. The ratio of the n-gram word count of the Student Answer Text(A) to the n-gram word count of the Wikipedia Source Text(S) is used to compute the intersection of the n-gram word counts of the Wikipedia Source Text and the Student Answer Text. The confinement values are computed using N-gram similarity. It was essentially divided into the following categories: c 1, c 2, c 3, c 4, c 5, c 6, c 7, c 8, c 9, and c 10. Two sets of data, Train x and Train y, are constructed and tested before being saved in Train x and fed

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117114

into Train y. The containment will be 0 if there are no n-grams shared by the two texts, but 1 if all of their n-grams intersect.

$$\frac{\sum count(\text{ngram}_A) \cap count(\text{ngram}_S)}{\sum count(\text{ngram}_A)} \quad (1)$$

3. Interface Development- An interactive interface using flask was developed. Input text given is tested against the keywords searched with the help of web scraping and beautifulSoap python library. Users will get their plagiarism percentage, plagiarized content, individual source from which text is being copied

Taking care of methodologies is very important so is having clarity on it. Designing is the basic requirement for any kind of project development. Hence, having a detailed plan helps in the smooth development of the project.

5 RESULT AND ANALYSIS

This paper demonstrates the use of two methods for plagiarism detection. In the first method SVM model is trained in which on taking the 25 text files for testing an accuracy of 96% has been observed.

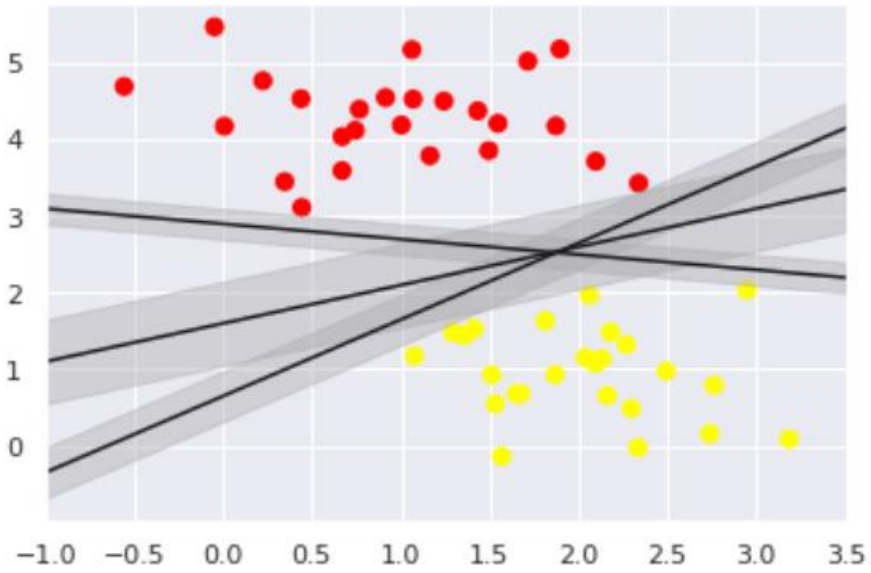

Fig 2. Training and validation accuracies of the SVM model

In the second method, when the software starts a window comes up which is the UI. A text area is provided where the user can enter the text to be checked whether it is plagiarized or not from the Internet. Using the web scraping and Natural Language Processing tools such as doc2vec, the results are generated.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117114

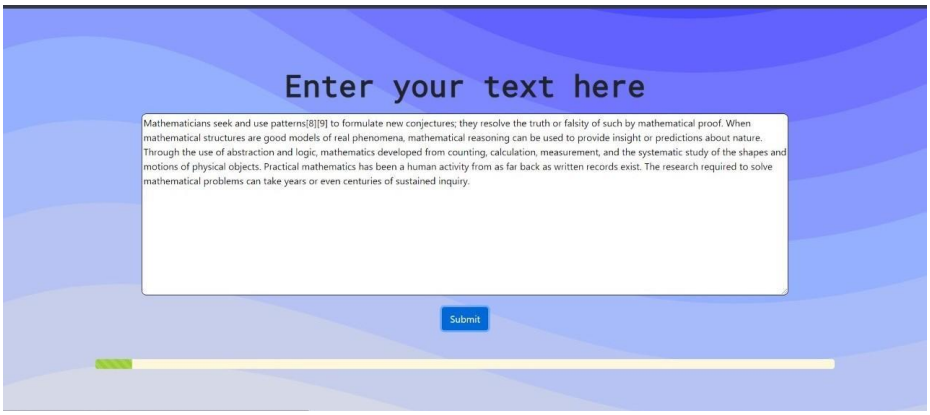

Fig 3. Home Page of application

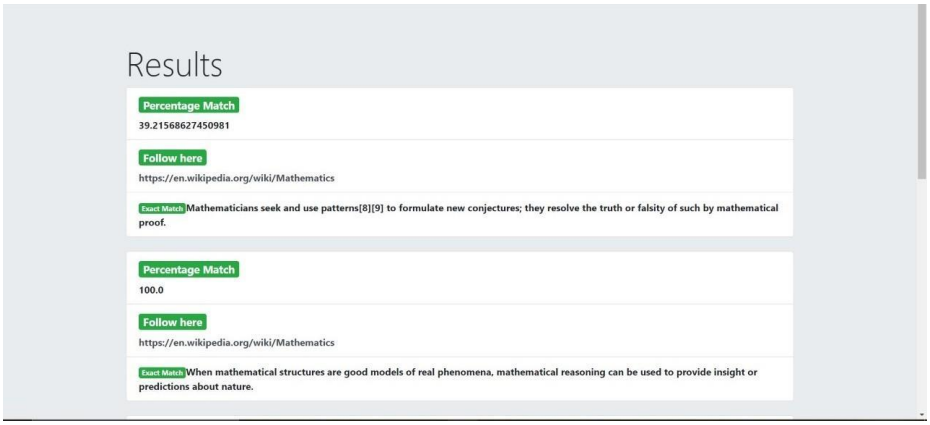

Fig 4. Results Page

Figure 3 represents the UI for the project that has been implemented using python libraries, Flask and NLP Tools.

6 CONCLUSION AND FUTURE WORK

For screening and avoiding plagiarism, the conventional manual methods of detecting plagiarism are insufficient. To avoid plagiarism, one should take into account modern technology and utilize online plagiarism detection tools. Plagiarism must be avoided. It's critical to appropriately acknowledge the efforts and knowledge provided by others. The motivation to extend this model and innovate model was an increase in number of copied or plagiarized code in online coding assignments or online coding competitions.The innovation is aimed at creating an interface which can detect and report plagiarized content between two codes, it is not restricted by languages. A csv file will be downloaded which will represent plagiarized percentages between two code files.

The support vector machine will perform poorly when the number of attributes for each data point exceeds the number of training data specimens. With larger datasets and more potent SVM models, there is still room to further enhance this method. The application might be developed as a legitimate website to increase accessibility. The accuracy of the model may also be improved by extending it to operate with source code and text files in other languages.

REFERENCES

1. El Mostafa Hambi , Faouzia Benabbou, A New Online Plagiarism Detection System based on Deep Learning in International Journal of Advanced Computer Science and Applications (2020).
2. Hiten Chavan , Mohd. Taufik , Rutuja Kadave , Nikita Chandra, Plagiarism Detector Using Machine Learning, International Journal of Research in Engineering, Science and Management Volume 4, Issue 4, April 2021.
3. Siddharth Tata, Suguri Charan Kumar, Varampati Reddy Kumar, Extrinsic Plagiarism Detection Using Fingerprinting, ISSN : 0976-8491.
4. Ahmed Hamza Osman, Omar M. Barukab, SVM significant role selection method for improving semantic text plagiarism detection, International Journal of Advanced and Applied Sciences (2017).

DOI: 10.17148/IJARCCE.2022.117114

5. Keerthana T V, Pushti Dixit, Rhuthu Hegde, Sonali S K, Prameetha Pai, A Literature Review on Plagiarism Detection in Computer Programming Assignments, International Research Journal of Engineering and Technology (2022).
6. Tomáš Foltýnek, Dita Dlabolová, Alla Anohina-Naumeca, Salim Razi, Testing of support tools for plagiarism detection, International Journal of Educational Technology in Higher Education,Article 46(2020).
7. Marwah Najm Mansoor, Mohammed S. H. Al-Tamimib, Computer-based plagiarism detection techniques, Ministry of Higher Education and Scientific Research, Iraq (2022).
8. Jitendra Yasaswi, Suresh Purini, C. V. Jawahar, Plagiarism Detection in Programming Assignments Using Deep Features, 17 December 2018, Nanjing, China.
9. Paul Clough, Mark Stevenson, Developing a corpus of plagiarized short answers, Lang Resources and Evaluation (2011).
10. Zubarev D.V. Sochenkov I.V. Paraphrased plagiarism detection using sentence similarity. Federal Research Center “Computer Science and Control” of Russian Academy of Sciences, Moscow, Russia.
11. Suleiman, Dima & Awajan, Arafat & Al-Madi, Nailah. (2017). Deep Learning Based Technique for Plagiarism Detection in Arabic Texts.
12. E. Hunt et al., "Machine Learning Models for Paraphrase Identification and its Applications on Plagiarism Detection," 2019 IEEE International Conference on Big Knowledge (ICBK), 2019, pp. 97-104.