STM JOURNALS

International Journal of
Electrical and Communication
Engineering Technology

Volume 2, Issue 1, 2024
January–June
DOI (Journal): 10.37591/IJECET

https://journals.stmjournals.com/ijecet

Review IJECET

Natural Language Processing-Based Plagiarism Detector

Sri. G. Kalyan Chakravarthi[1], G. Harsha Vardhani[2,*], Praveen Kumar. J[2], M. Nishita[2], P. Satish[2]

Abstract

Plagiarism is the act of taking another's work and claiming it as his/her own without giving the credit. Plagiarism is a grave concern in many fields like education, literature, business, etc. The project aims for text and image plagiarism detection employed by a web application containing four modules. Text plagiarism is detected when the input is either text or PDF. For PDF, the text extraction is done from the PDF and then detected for plagiarism. Image plagiarism detection is performed by taking an image as input. Image plagiarism detection has an extension that takes input as an image containing text. The text extraction is done from the input and compared for similarity with the documents of the dataset and the output generated. Text plagiarism percentage detection uses Jaccard Similarity and Image plagiarism percentage detection uses Cosine Similarity. Image plagiarism detection uses Levenshtein Distance if the input is image-containing text. Additionally, some features are added to this project like Citation status detection, AI content detection, and Type of Plagiarism detection.

Keywords: Plagiarism detection, web application, Jaccard similarity, cosine similarity, levenshtein distance

INTRODUCTION

Plagiarism is a major problem defined as taking another's models, documents, pictures, paintings, etc., and presenting them as their own. It is mainly faced by publishers, researchers, and educational institutions. An act of dishonesty in academics will occur if plagiarism is not prevented. Plagiarism can be a reason for harming a person's name and fame, resulting in strict action, and even result in problems in professional and academic scenarios. Therefore, In the era of available web content, it is more important to recognize and prevent plagiarism, especially.[2].In this paper, a tool is introduced for plagiarism detection that combines Natural Language Processing (NLP) text pre-processing, cosine similarity, Jaccard similarity, and Levenshtein distance techniques to appropriately check for plagiarism inside a document by comparing it with the document dataset, detect plagiarism of images by comparing it to image dataset. The tool first pre- processes the text by performing stop word removal, tokenization, cleaning, and lemmatizing the text. The extraction of text from the document is performed by the tool and checks for similarity within the dataset. For image plagiarism, the tool first extracts feature from the images and checks the similarity of features with images in the dataset. The tool then computes the jaccard similarity between the two sets which measures by dividing the size of the intersection of the sets by the size of the union of the sets. For Image plagiarism, the tool calculates the cosine similarity which checks the similarity between two non- zero vectors. Additionally, for text plagiarism, the output also includes the AI content status, Citation status, and Type of Plagiarism. Since originality is an essential factor, plagiarism detection inside a document plays a vital role in many fields and industries.

***Author for Correspondence**
G. Harsha Vardhani
E-mail: vardhanigandreddi@gmail.com

[1]Assistant professor Computer science Department, Gayatri Vidya Parishad College for Degree and PG courses, Vishakhapatnam, Andhra Pradesh, India
[2]Student, Computer science Department, Gayatri Vidya Parishad College for Degree and PG courses, Vishakhapatnam, Andhra Pradesh, India

Received Date: May 21, 2024
Accepted Date: June 11, 2024
Published Date: June 20, 2024

Citation: Sri. G. Kalyan Chakravarthi, G. Harsha Vardhani, Praveen Kumar. J, M. Nishita, P. Satish. Natural Language Processing-based Plagiarism Detector. International Journal of Electrical and Communication Engineering Technology. 2024;2(1): 27–35p.

RELATED WORKS

In this section, we have gone through different plagiarism detection systems to propose a new plagiarism detection system that detects plagiarism for both text and image in one tool. In [1], the system uses machine learning and deep learning techniques to detect source code plagiarism in C programs. Furthermore, the structure, syntax, and code presentation are focused mainly on detecting plagiarism issues that contain code obfuscation. The current state-of-the-art plagiarism detectors face some issues which shows the incapacity to detect multiple- source plagiarism is solved using Machine learning and program analysis techniques. Feature extraction pipeline uses a transfer network for feature extraction from every function. First, the various functions are suitably mapped considering every function's similarity scores that create the program also considering all the other functions of the competing programs. This work comes under the text-based approach of plagiarism detection. In [2], a plagiarism detection tool is proposed which concentrates on the document similarity. The algorithms and techniques used come under Natural Language Processing. Firstly, the text is pre-processed by stop word removal, tokenization, and cleaning, and further, the text is extracted. The generated TF-IDF matrix for the document is further converted into a high-dimensional space vector using VSM. Finally, for checking text similarity cosine similarity is computed for the two sets of vectors. An appreciative and authentic solution is offered by the introduced tool for plagiarism detection inside a document against web content. In [3], an image plagiarism detection tool is proposed which follows three steps: pre- processing, feature extraction, and comparison. After following the three steps the result will be generated containing true and false and similarity index in ascending order. In [4], The proposed system follows three phases: preprocessing, seeding, and post-processing. In the first step, pre-processing strategies like removing stop words, tokenization, lemmatization, and parts of speech tagging are performed [5-9]. In the seeding process, the plagiarized cases are extracted, the syntactic, lexical, and semantic features are determined, the most effective features are selected, the training database is created, and the support vector machine model is constructed. It follows two different paths to detect similarity in sentences, the traditional paragraph-level comparison is the first path, and the constructed SVM classifier has a hyperplane equation which is the second path. The best-plagiarized segment between source and suspicious documents is the final step and it is based on filtering seeds, combining adjacent detected seeds, flexible behavior, and filter segment strategies. In [10], the main aim is to detect plagiarism in diagrams. This system consists of three parts: pre-processing, feature extraction, and comparison. For pre-processing, grey scaling, boundary detection, thresholding, and cropping are used. [11-12] In feature extraction, the color and shape are extracted from the diagrams. In comparison, the Euclidean distance formula is used to compare two images.

Research Methods

Three strategies are proposed for four different inputs namely Jaccard similarity for text-based plagiarism, cosine similarity for image-based plagiarism, and Levenshtein distance for plagiarism check of input as text within an image. These algorithms compute similarities and dissimilarities and they come under Natural Language Processing. In this section, various algorithms used in this work are explained. Figure 1 represents the proposed structure.

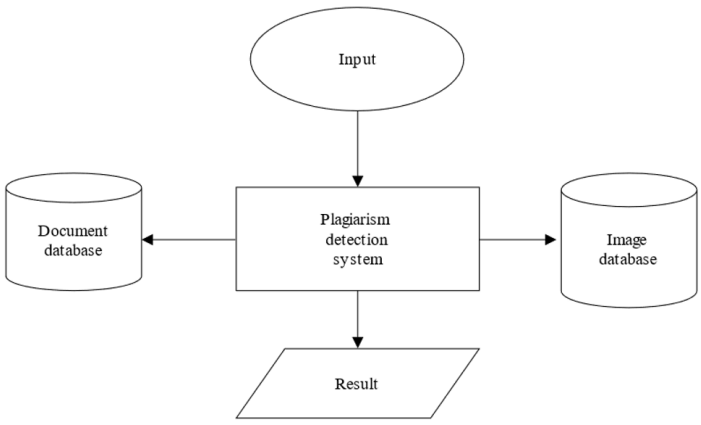

Figure 1. System architecture

Jaccard similarity

Jaccard similarity is a similarity metric between two sets. It is defined as the division of the magnitude of the intersection of the sets and the magnitude of the union of the sets. In our work, Jaccard similarity is used as a measure to check the similarity of text for two inputs like text or documents. [13–16] Firstly, we will import all the nltk resources useful for similarity check and then we will pre-process the text by tokenization, stop word removal, lemmatization, etc. Now we will extract text from the document if the input is a document else take the text and convert it into sets. Further, we calculate Jaccard's similarity using the formula. For the document as input, the similarity check is performed by comparing the input with the dataset which contains few documents. For the text as input, the similarity check is performed by comparing the text with the web content as shown in Figure 2.

$$\mathcal{J}(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|}$$

(Formula of Jaccard Similarity)

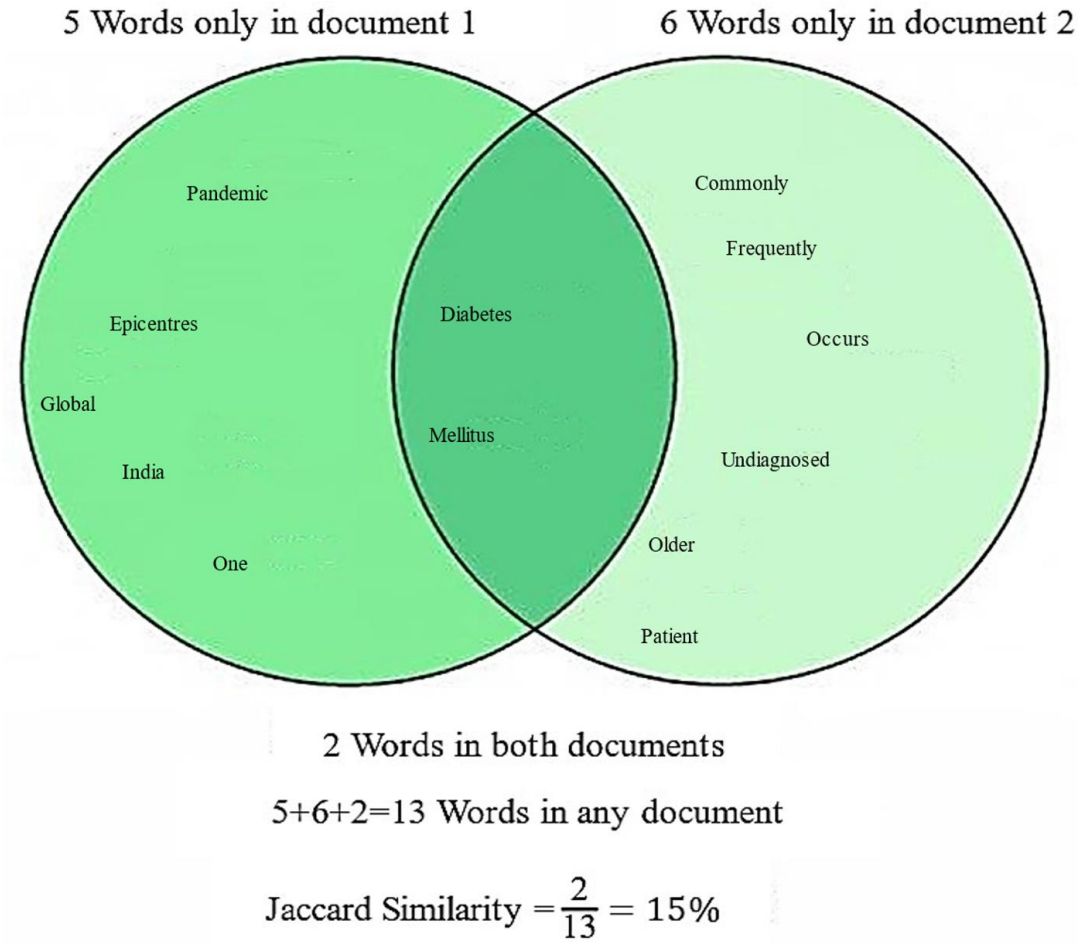

Figure 2. Example of jaccard similarity

Cosine Similarity

Cosine similarity is a similarity metric used to check the similarity between two non-zero vectors and this technique is used for image-based plagiarism in this paper. Firstly, the feature extraction of the input image is performed using VGG16, a pre-trained model that extracts features from each image. The features are represented as vectors and for these vectors, the Cosine similarity is computed as shown. The input image will be compared with the dataset which contains few images. One example of cosine similarity are shown in Figure 3.

$$similarity\ (A,B) = \cos(\theta) = \frac{A.B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \sum_{i=1}^{n} B_i^2}}$$

(Formula of Cosine similarity)

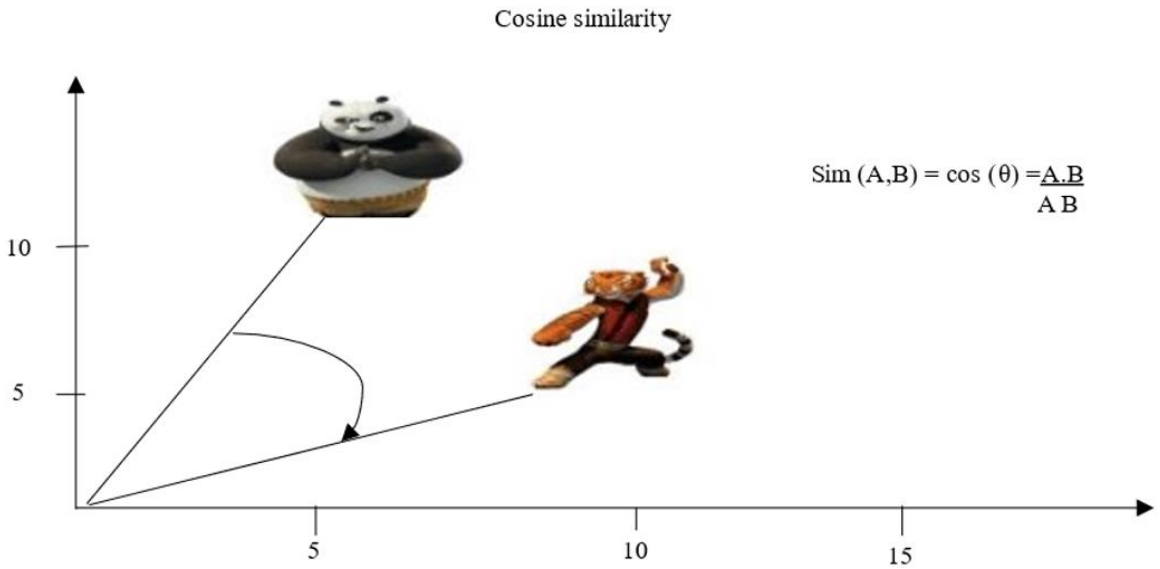

Figure 3. Example of cosine similarity

Levenshtein distance

Levenshtein distance, also called edit distance, is a similarity metric that calculates the least number of single-character edits required to change one string into the other. It is mainly performed to check the similarity between two strings. Firstly, the text from the image is extracted using easy OCR and the Leveshtein distance is computed using the formula. The extracted text is compared with the dataset which contains documents for similarity as shown in Figure 4 and Figure 5.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Figure 4. Formula of Levenshtein Distance

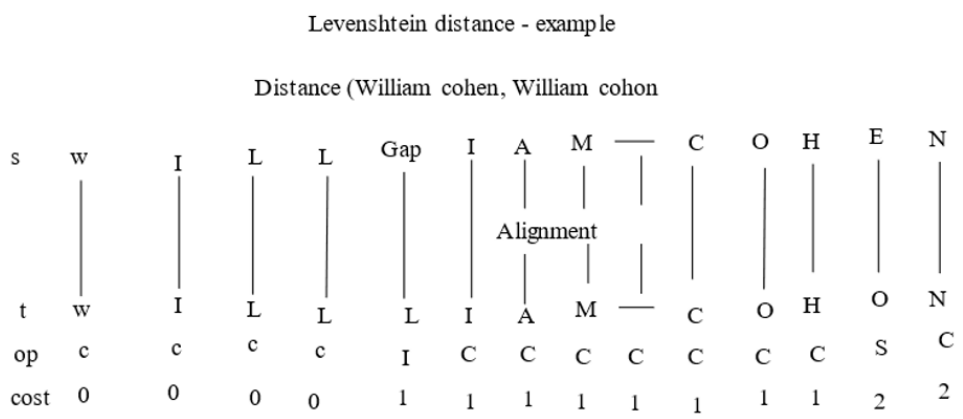

Figure 5. Example of Levenshtein Distance

EXPERIMENTATION

In this section, we will explain the experiments done for different input types. Firstly, for text as input, we used Jaccard similarity to check the similarity between the input text and online or web content. In this, at first, the text is pre-processed by removing stop words, tokenization, etc. Now the text similarity is checked with the web and provides the plagiarism percentage and sources. It additionally provides whether the text contains any AI content or not. For the pdf as input, we used Jaccard similarity to check the similarity between the input document and the dataset containing documents. In this, at first, the text is extracted and pre-processed removing all the stop words, tokenization, lemmatization, etc. Now the pre- processed text similarity is checked with the document dataset and generates output containing plagiarism percentage, citation status, AI content status, and Type of plagiarism. For image as input, we used cosine similarity to check the feature similarity between the input image and the dataset containing images. In this, at first, the features are extracted from the image using VGG16 and now the feature similarity is checked with the image dataset features and provides similar images. For images containing text as input, we used Levenshtein distance to check the similarity of text extracted from the image with the dataset of documents. In this at first, the text is extracted from the image using easy OCR and next the text is pre-processed removing stop words, tokenization, etc. Now the pre-processed text is checked with the text in a dataset of documents and generates an output of similarity ratios of text with few documents and also overall plagiarism percentage. For AI content detection we trained a dataset containing sentences of AI content and non-AI content using the BERT model. Based on this we get the output of AI content status.

RESULTS

The results differ for different inputs. For text, the output consists of plagiarism percentage and sources of plagiarism. For PDF, the output consists of plagiarism percentage, AI content status, citation status, and type of plagiarism. For the image, the output consists of similar images that match the different features of the input image. For image-containing text, the output consists of the extracted text and the similarity ratio when compared with the document dataset and the overall plagiarism percentage.

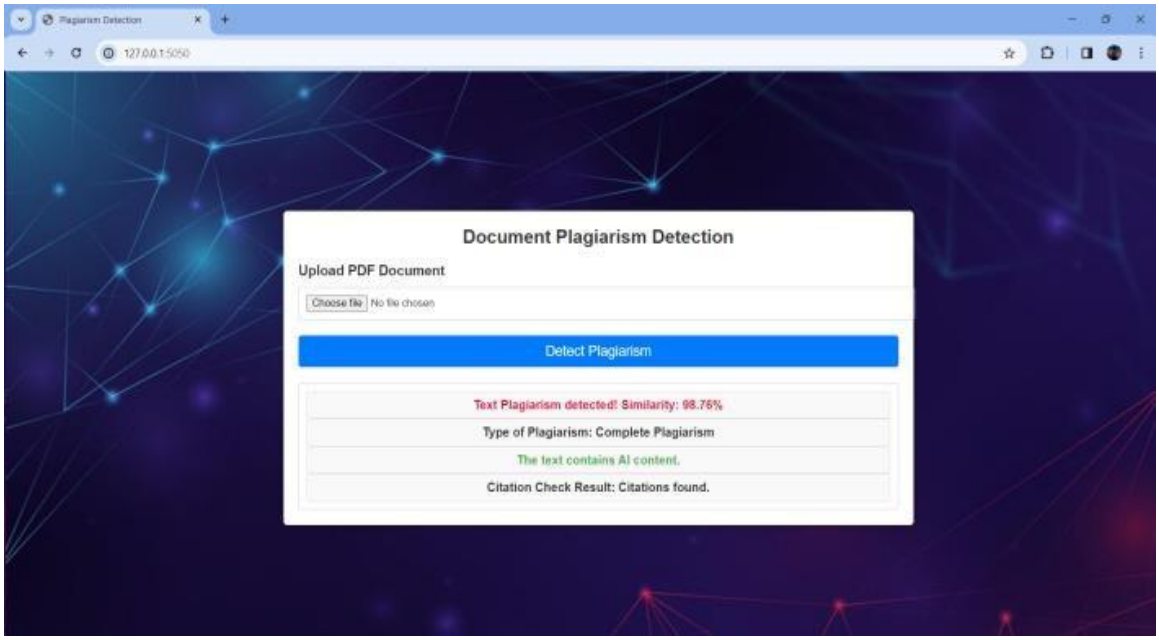

Figure 6. Output of uploading a PDF

Figure 6 shows the output when a PDF is uploaded and this output contains the plagiarism percentage, type of plagiarism, AI Content status, and Citation status

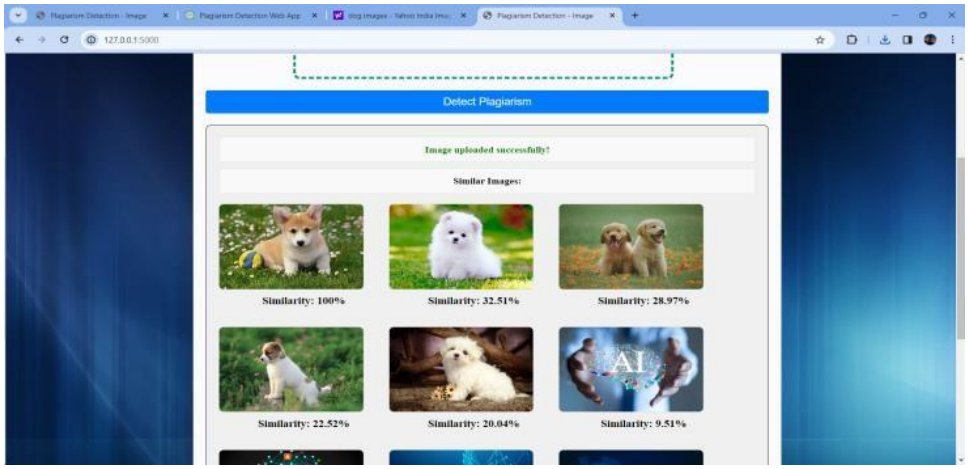

Figure 7. Output of uploading an image

Figure 7 depicts the output when an image is uploaded, and the output consists of a few images similar to the input along with the similarity percentage

Figure 8. Output of uploading an image (Human)

Figure 8 shows the Output of uploading an image (Human) and the output consists of different images similar to the input along with the similarity percentage

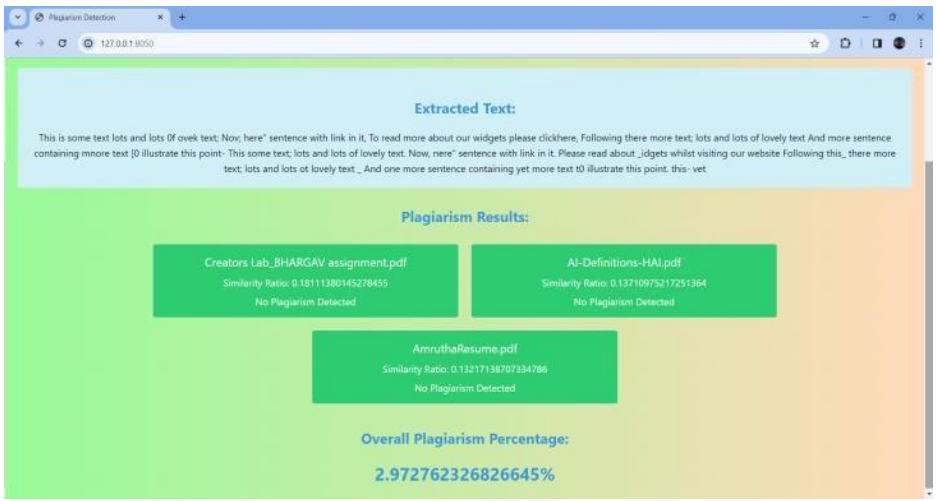

Figure 9. Output of uploading an image containing text

Figure 9 depicts the Output of uploading an image containing text Where the text from the image is extracted and checked for similarity with the dataset to provide a similarity ratio and overall plagiarism percentage

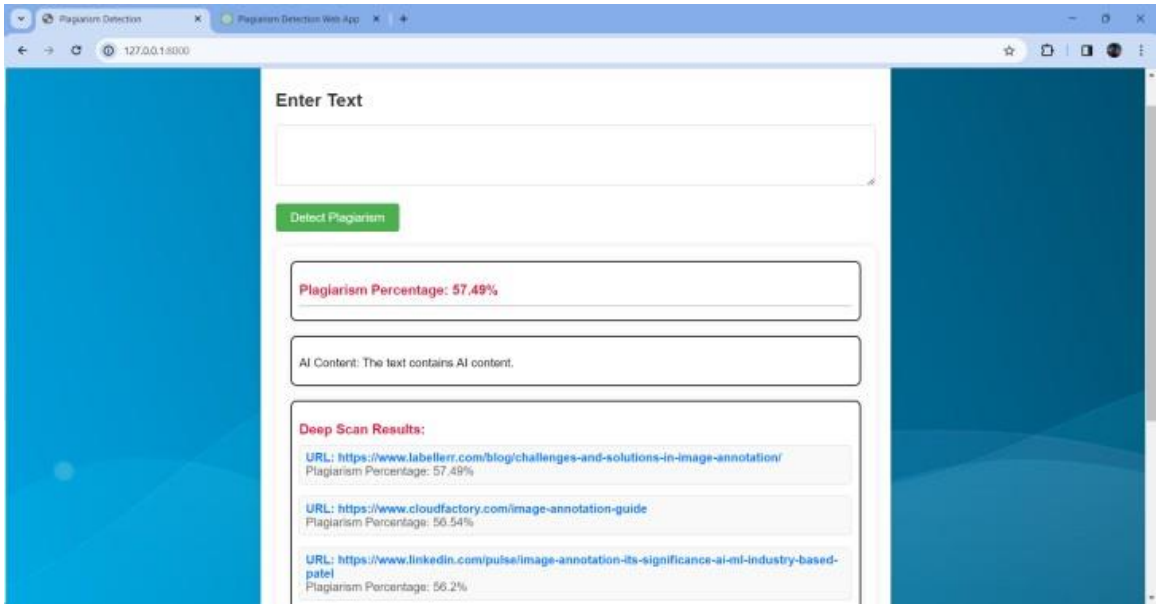

Figure 10. Output of uploading text

Figure 10 shows the Output of uploading text which provides the result as plagiarism percentage and the sources that match with the input text

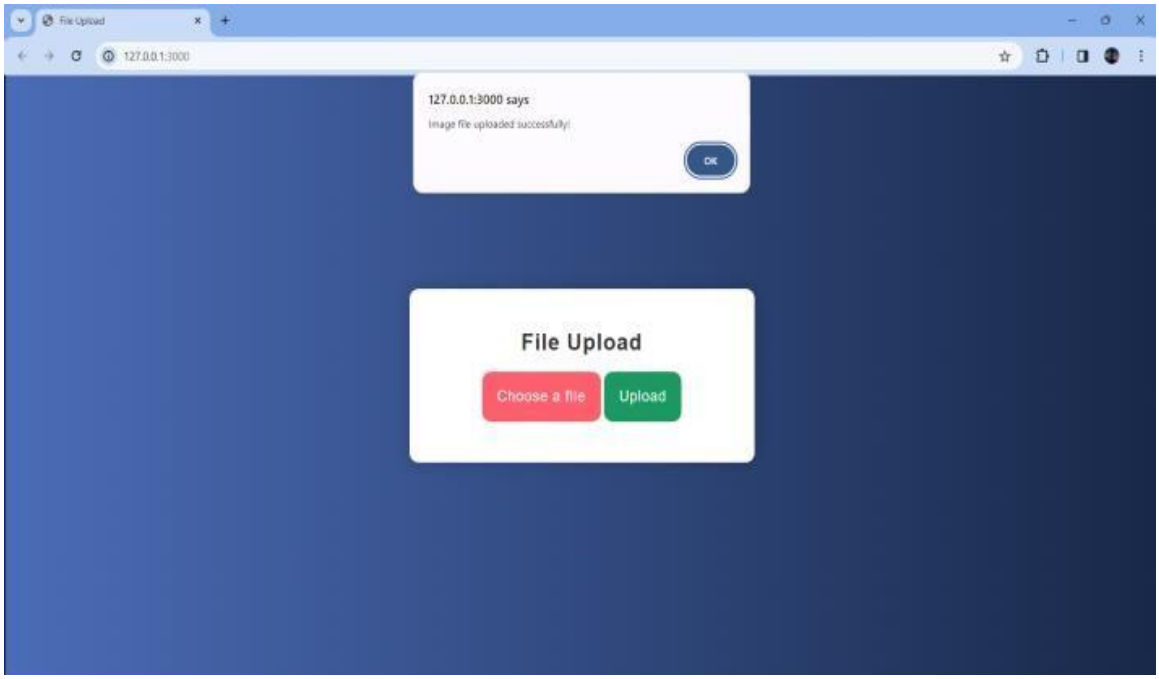

Figure 11. Output of uploading an image by admin to the Dataset

Figure 11 shows the output when an image is uploaded to the dataset by an admin. The dialog box is also displayed as “Image file uploaded successfully”.

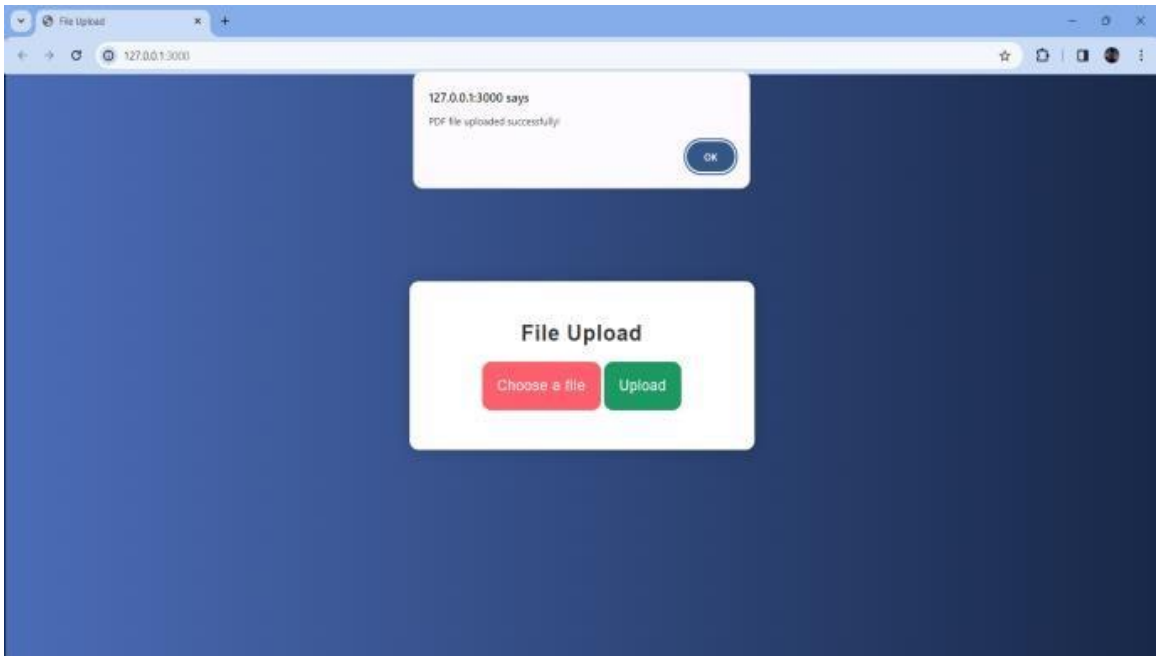

Figure 12. Output of uploading a PDF by admin to a dataset

Figure 12 depicts the output when a PDF is uploaded to the dataset by an admin. The dialog box is also displayed as "PDF file uploaded successfully".

CONCLUSION

In this text and image plagiarism detection system, good accuracy is achieved which is understood by the results generated. The various techniques employed for both text and image plagiarism are Jaccard similarity, Cosine similarity, and Levenshtein Distance. These algorithms perform with good accuracy as per the requirements of our work. We can see the different outputs for different formats of inputs in the results. The accuracy of the algorithms used depends on the generated outputs. For text plagiarism detection, Jaccard similarity fulfills our requirements and gives good results and for image plagiarism detection, Cosine similarity satisfies the requirements and provides accurate results. The additional features also provide approximate results which allows our work to be different from the past works.

REFERENCES

1. Eppa and A. H. Murali, "Machine Learning Techniques for Multisource Plagiarism Detection," 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2021, pp. 1-5, doi: 10.1109/CSITSS54238.2021.9683752.
2. R. N. Kulkarni, C. Ganesh, D. K. B K, H. B and A. P. Reddy, "Novel Approach to Detect Plagiarism in the Document," 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 2023, pp. 1-6, doi: 10.1109/ICDCECE57866.2023.10150442.
3. Meuschke, N., Gipp, B., & Breitinger, C. (2012). CitePlag : A Citation-based Plagiarism Detection System Prototype.
4. DEEPA GUPTA, VANI K., LEEMA L. M., "PLAGIARISM DETECTION IN TEXT DOCUMENTS USING SENTENCE BOUNDED STOP WORD N-GRAMS" Journal of Engineering Science and Technology Vol. 11, No. 10 (2016) 1403 – 1420.
5. Petr Hurtik, Petra Hodakova University of Ostrava, Centre of Excellence IT4Innovations, Institute for Research and Applications of Fuzzy Modeling, 30. dubna 22, 701 03 Ostrava 1, Czech Republic, "FTIP: a Tool for an Image Plagiarism Detection"

6. Norman Meuschke, Christopher Gondek, Daniel Seebacher, Corinna Breitinger, Daniel Keim, Bela Gipp. An Adaptive Image-based Plagiarism Detection Approach. JCDL '18: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital LibrariesMay 2018Pages 131–140https://doi.org/10.1145/3197026.3197042.
7. A. Ekbal, S. Saha and G. Choudhary, "Plagiarism detection in text using Vector Space Model," 2012 12th International Conference on Hybrid Intelligent Systems (HIS), Pune, India, 2012, pp. 366-371, doi: 10.1109/HIS.2012.6421362.
8. P. Ovhal, "Detecting plagiarism in images," 2015 International Conference on Information Processing (ICIP), Pune, India, 2015, pp. 85-89, doi: 10.1109/INFOP.2015.7489356.
9. S. Dutta and D. Bhattacharjee, "Plagiarism Detection by Identifying the Keywords," 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 2014, pp. 703-707, doi: 10.1109/CICN.2014.154.
10. Hariharan S. Automatic plagiarism detection using similarity analysis. Int. Arab J. Inf. Technol.. 2012 Jul 1;9(4):322-6.
11. Kadir Yalcin, Ilyas Cicekli, Gonenc Ercan. An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding. Expert Systems with Applications. Volume 197, 1 July 2022, 116677
12. Marwah Najm Mansoor, Mohammed S. H. Al-Tamimi.Computer-based plagiarism detection techniques: A comparative study. Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 3599-3611 http://dx.doi.org/10.22075/ijnaa.2022.6140
13. S. K. Pal, O. J. Raffik, R. Roy, V. B. Lalman, S. Srivastava and B. Sharma, "Automatic Plagiarism Detection Using Natural Language Processing," 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2023, pp. 218-222.
14. MAC Jiffriya, MAC Akmal Jahan, RG Ragel. “Plagiarism detection tools and techniques: A comprehensive survey”. Journal of Science-FAS-SEUSL (2021) 02(02) 47-64.
15. “Development of an Algorithm for Plagiarism Detection” https://doi.org/10.58694/20.500.12479/1624
16. Chang, CY., Lee, SJ., Wu, CH. et al. Using word semantic concepts for plagiarism detection in text documents. Inf Retrieval J 24, 298–321 (2021). https://doi.org/10.1007/s10791-021-09394-4

