College Project Plagiarism System

Mayank Patil[1], Prachit Raut[2], Adnan Shaikh[3]
[1, 2, 3] Dept of Information Technology
[1, 2, 3] Theem College of Engineering, Boisar.

Abstract- *Plagiarism is the act of stealing someone else's ideas or writings. Over the course of years, various plagiarism Software have claimed that they are 100 percent accurate. However, these soft- ware's do nothing but use a series of hand-picked thresholds of metrics that ultimately determine the basis of whether or not we can claim that the said author is guilty of committing plagiarism. Since these thresholds are manually determined, they are not good enough to be able to detect all forms of plagiarism, which include unique situations. In this project, we will be making use of Recurrent Neural Networks (RNN) and the LSTM algorithm whose specialty is that it not only processes single data points (such as images), but also entire sequences of data (such as speech or video). Therefore, we will be detecting plagiarism using a pre-classified data set, and make use of deep learning to enable authorities without any knowledge of the methodology to detect plagiarism.*

Keywords- LSTM, Deep learning, plagiarism, RNN

I. INTRODUCTION

The word" PLAGIARISM" refers to a piece of writing that has been copied from someone else and is presented as being your own work. Today with the huge popularity of internet, so many documents are freely accessible. Now internet is an extensive source to collect data. Students can easily get their required information or data from internet. Plagiarism cases are an everyday topic, for example, in academics. So far, textual plagiarism has been the most common form of plagiarism. In this project, we will be working on textual as well as code plagiarism.

Detecting plagiarism is an active field of research with a wide variety of available tools. Detecting textual plagiarism is fairly easy, especially if the text has been copied verbatim. How- ever, nobody copies texts verbatim anymore. But some smart student rearranged text from an active voice to a passive voice, or the other way around; it becomes difficult for the automated systems to detect the plagiarized material. A way to get rid of this issue would be to detect the meaning of the plagiarised text and compare it with the texts found on the web. However, detecting the semantics of the text is challenging for an automated system despite making use of structural and lexical similarities. Over the course of the years, many different approaches have been attempted to overcome these unique situations and challenges, but only a little progress has been made so far as to achieve high levels of accuracy in the complex plagiarism methods

To combat this, now we introduce a system which will provides an easy approach to keep a track of plagiarism in document of student by taking input like code and proposal/report. Now before college select a project all the draft for the projects will be sent to a specific college UI and from there they will be able to run the algorithm and get the similarity score between the given and past documents using ML and also tell whether to allow or not to allow that project. Machine learning is used to help users find settings that fulfil their requirements with an automatic procedure that is solely based on statistics from a data set made up of pre-classified plagiarized and non-plagiarized cases

II. MAJOR CHALLENGES

- Lack of search skills – Detection efficiency depends on the scope and size of the reference collection, but also on the language of the code and proposal/report
- Unable to detect – Due to excessive manipulation of ideas the system is unable to detect plagiarism
- Easy availability of resources – It becomes a major challenge to detect plagiarism due to many resources available anytime from anywhere nowadays
- Accuracy – To find algorithm or methods to increase the accuracy of our system

III. PROBLEM STATEMENT

To develop a 'College Project Plagiarism System' which can increase the accuracy by finding similarity score between the present and past texts and codes by comparing them using a machine learning algorithm like LSTM and RNN. The system will also provide instant results and the percent of plagiarism.

Every college have certain restriction on not choosing a project from certain years in past. But some smart student changes the name of the project and choose same old projects and also busy college staff doesn't notice it in manual selection and also it takes a lot of time. So now we introduce a

system that before college select a project all the draft for the projects will be sent to a specific college UI and from there they will be able to run the algorithm and get the similarity score between the given and past documents using ML and also get instant results whether proposal/report is allowed or not to allowed to enter the presentation round.

IV. PROPOSED METHODOLOGY

In this plagiarism system, the user has to login into the UI where the system will authenticate the user. If the user is an admin, it will be taken to the admin page, where it can manually view the accepted report and it can also reject and accept the report. If the user is not an admin, the report will be processed by Django middle layer, extract the text data using OCR. Then it will clean the data and convert the text data to vector format, loads trained machine learning model. Before loading trained machine learning model, it gets the dataset, clean and pre-process the dataset, vectorize the data before training and uses machine learning algorithm to train the model and then saves the model for prediction. After loading trained machine learning model, it passes the pre-processed data to the model and gets the output. Output will be stored to the database and the then will be sent to the user. Result will also be displayed on UI.

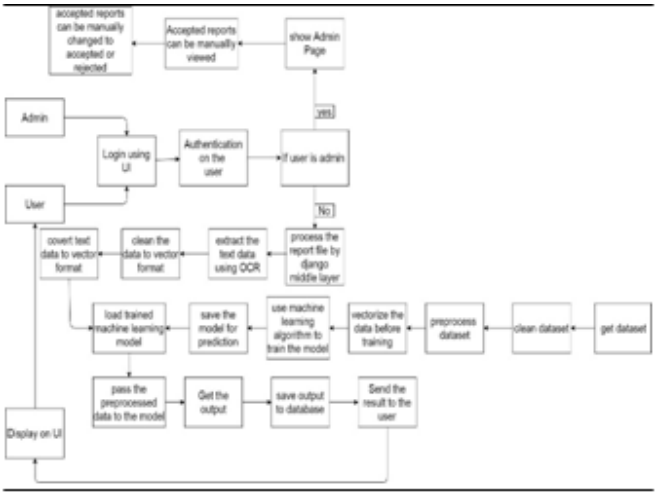

Fig 1. *Architectural Diagram*

V. PROPOSED ALGORITHM

- **RNN model-** A Recurrent neural network or an RNN is an architecture of Neural Networks which is used with use cases that comprise sequential or contextual data. Recurrent neural networks have been used widely with speech recognition, language translation and even image recognition. RNN are neural networks that are good at modelling sequencing data. For example, if we take a still snapshot of a ball moving in time, and if we want to predict the direction that the ball will go to; with just this much information, is it possible to predict the accurate answer? Without no prior knowledge of where the ball has been - there won't be enough data to predict where the ball is going. If there are many snapshots taken of the ball in succession - we can make a better prediction. This is a sequence of a particular order in which one thing follows another. With this information, you will be able to tell what direction the ball is moving to. Sequential data comes in many forms. For example, audio is a form of sequential data that can be divided into partitions. Similarly, a text is a form of sequential data that can be separated into a sequence of characters or a sequence of words. These sequences are then fed into an RNN - which is known for its ability to be good at processing sequences for predictions.
- **LSTM model:** As the name suggests, Long Short-term Memory is a specialized neural network that is used to combat the problem of Short-term memory faced by a typical Recurrent Neural Network (RNN). An LSTM will essentially perform just like an RNN, except they are able to mitigate the problem of long-term dependency using the "gate" mechanism. These gates know what information to add or delete to the hidden state and act as different tensor operations. This mitigates the problem of short-term memory. By simple logic, it is evident that prediction can be made easier if there is sequential logic to the sentence. In a typical prediction problem, where we have a bunch of characters, the goal is to predict the next character of the sequence. However, this prediction is impossible to make without knowing the context.

VI. WORKING OF THE PROJECT

Our proposed system uses LSTM algorithm to find similarity score between the given and the past texts and codes by comparing them. The system will also provide instant results in selection process whether proposal/report is allowed or not to allowed to enter the presentation round.

Implementation involves the following steps:

Step 1: Authentication module

In authentication module, it consists of user, teachers and super admins it collects information like ID and password and compares it with the entries stored in database. If the data provides matches with the database, then the user is validated and directed to the specified UI. If he data provided does not meet the criteria then the validation is denied.

Step 2: Login module

Login module is a portal that allows users to login using their ID's and passwords. Here the student, teacher and super admins will login into the system through this login module. after your login you will be directed to the UI where you can upload the reports/proposals and also check if it's accepted or not and percentage of plagiarism

Step 3:Authentication module

After uploading the file i.e. word or pdf extraction of text data using Optical Character Recognition takes place where it recognizes text within a digital image and it is commonly used to recognize text in scanned documents and images. The extracted data will be made available in formats like XML

Step 4: Data Extraction

After uploading the file i.e. word or pdf extraction of text data using Optical Character Recognition takes place where it recognizes text within a digital image and it is commonly used to recognize text in scanned documents and images. The extracted data will be made available in formats like XML

Step 5: Data cleaning

After data extraction next phase is data cleaning. Data cleaning is done to identify and remove stalk board, errors, duplicate words, incomplete data. This improves the quality of the training data and will help you to yield better results from the machine learning functions. Also, we streamline our data.

Step 6: Training Model

First, we get data from the dataset then we perform data cleaning, data processing and con- vert it into vector format so that it can be easily read by the computer this all is done before training. Now in training we use Machine Learning algorithm like Long Short-Term Memory and Recurrent Neural Networks. Here RNN uses sequential memory to memorize the previous information and make predictions while keeping that in mind. An RNN is trained using a back-propagation algorithm. The back propagation is what causes the RNN to have short term memory and vanishing gradient. The vanishing gradient causes the RNN to not learn the long-range dependencies across the various time steps. Therefore, the RNN by itself proves to be incompetent in providing an answer. This can be solved by utilizing a specialized recurrent neural network, known as Long Short-Term Memory or LSTM. As the name suggests, Long Short-term Memory is a specialized neural network that is used to combat the problem of Short-term memory faced by a typical Recurrent Neural Network (RNN). An LSTM will essentially perform just like an RNN, except they are able to mitigate the problem of long-term dependency using the "gate" mechanism. These gates know what information to add or delete to the hidden state and act as different tensor operations. This mitigates the problem of short-term memory. By simple logic, it is evident that prediction can be made easier if there is sequential logic to the sentence. In a typical prediction problem, where we have a bunch of characters, the goal is to predict the next character of the sequence. However, this prediction is impossible to make without knowing the context.

Step 7:Data preprocessing

In data pre-processing, the raw data is transformed to a useful and efficient format. Here all the missing data is removed or filled and large amount of data is handled to remove the complexity to make the process easier. By simple logic, it is evident that prediction can be made easier if there is sequential logic to the sentence. In a typical prediction problem, where we have a bunch of characters, the goal is to predict the next character of the sequence. However, this prediction is impossible to make without knowing the context.

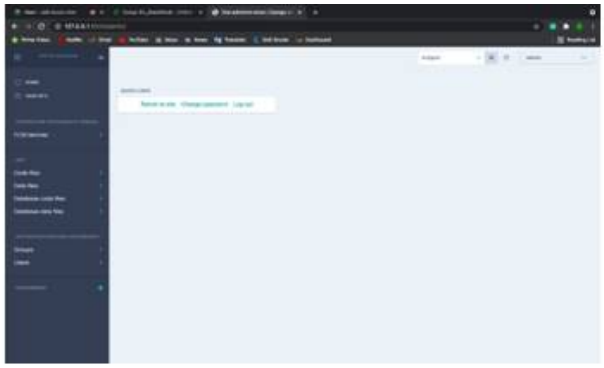

Fig. UI of College Project Plagiarism System

VII. RESULTS AND DISCUSSION

In order to determine the accuracy and suitability of our proposed approach for the plagiarism detection system, we made use of four different split corpora from Kaggle.com. Upon analyzing one of the corpus, we found that there are around 7859 pairs of strange or suspicious passages, out of which 3792 passages are not plagiarized and the rest 4067 are plagiarized. The construction of the corpus was done after performing crowdsourcing on Amazon's Mechanical Turk. In this situation, various texts were extracted from a project and

these texts were paraphrased. These texts passed the green check on the paraphrasing since they were reviewed to be very similar to the original by the system and were hence rejected. On the other hand, cases where the content was grammatically correct and had the same meaning as the source was accepted as a paraphrase.

The second corpus consisted of 1716 text articles that were extracted from an esteemed Press Association. Since the text articles are a rewritten version of the corresponding PA source: we had to categorize the texts into various categories that described whether or not they were fully derived. For this project, we chose 253 articles with a single source, and were categorized into “Plagiarized” and “Non-Plagiarized”.

The third corpus consisted of 95 short answers that were gathered from participants who answered five different questions. These answers were originally labelled into 4 different categories but were then converted into the “Plagiarized” and “Non-Plagiarized” categories. The measures considered to evaluate the level of plagiarism are based on Accuracy, Recall, and F-measure.

$$Accuracy\ (A) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision\ (P) = \frac{TP}{TP + FP}$$

$$Recall(R) = \frac{TP}{TP + FN}$$

$$F\_Measure\ (F) = \frac{2 * Precision * Recall}{(Precision + Recall)}$$

VIII. CONCLUSION

The system would prove to be an efficient system to obtain Plagiarism results in college projects. In today’s world, plagiarism is a day to day occurrence. With how common it is, there are more and more methods being invented to avoid getting caught by plagiarism tools. One of the ways of doing so is to summarize and paraphrase the content. Due to reasons like this, there is more need for a more suitable plagiarism detection system. It is necessary for this plagiarism detection system to make use of effective mechanisms for the automatic detection of plagiarism. In this project, the approach of paraphrase recognition is used. This is done so as to detect plagiarism in code as well as documented sources. We have utilized Recurrent Neural network using LSTM to combat this problem. The system has been tested used three different scenarios from the website: kaggle.com. The system is also performing better when compared with the best performing system on the previously mentioned corpora. Testing with various subcategories of P4P also gave out excellent results. This leads us to conclude that the paraphrasing recognition techniques shows a lot of potential with regards to plagiarism detection.

REFERENCES

[1] El Mostafa HAMBI; FaouziaBenabbou;, “ A Multi-Level Plagiarism Detection System Based on Deep Learning Algorithms”, IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.10, October 2019

[2] S. Priya; Anukul Dixit; Krishanu Das; Ronak Harish Patil;, “Plagiarism Detection in Source Code Using Machine Learning”, International Journal of Engineering and Ad- vanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-4, April 2019

[3] Ethan Hunt; RitvikJanamsetty; ChananaKinares; Chanel Koh; Alexis Sanchez; Felix Zhan; Murat Ozdemir; Shabnam Waseem; Osman Yolcu; BinayDahal; Justin Zhan; Laxmi Gewali; Paul Oh;,” Machine Learning Models for Paraphrase Identification and its Applications on Plagiarism Detection”, IEEE International Conference on Big Knowl- edge, 2019

[4] MumthazBeegum.M; AjiS;,” A Method for Text Plagiarism Classification Using Deep Learning”, JASC, November 2017

[5] ShubheshAmidwar;, ” Plagiarism Detection Using Supervised Machine Learning Algo- rithm” JETIR, Volume 4, Issue 06 June 2017

[6] MunaAlSallal; Rahat Iqbal; Saad Amin; Anne James; VasilePalade;, ” An Integrated Machine Learning Approach for Extrinsic Plagiarism Detection”, 9th International Con- ference on Developments in eSystems Engineering, 2016

[7] MausumiSahu;, ” Plagiarism Detection Using Artificial Intelligence Technique In Multi- ple Files” International Journal Of Scientific Technology Research vol 5, issue 04, April 2016

[8] A. Chitra; AnupriyaRajkumar;, “Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer” J. Intell. Syst. 2015.

[9] UpulBandara; GaminiWijayrathna;, “Detection of Source Code Plagiarism Using Ma- chine Learning Approach” International Journal of Computer Theory and Engineering,Vol. 4, No. 5, October 2012

[10] Francisco Rosales; Antonio Garc´ıa; Santiago Rodr´ıguez; Jose L. Pedraza; Rafael Me´ndez; Manuel M. Nieto;” Detection of Plagiarism in Programming Assignments”,

IEEE Trans- action On Education, Vol. 51, no. 2, May 2008