PLAGIARISM DETECTION WITH PARAPHRASE RECOGNIZER USING DEEP LEARNING

Prof. Pritam Ahire[1], Yogesh Wadekar[2], Tushar Shendge[3], Manali Dhokale[4], Vaishnavi Ohol[5]

[1]Professor, Dept. of Computer Engineering, D. Y. Patil Institute of Engineering and Technology, Ambi, Pune, India
[2-5]Student, Dept. of Computer Engineering, D. Y. Patil Institute of Engineering and Technology, Ambi, Pune, India

---***---

Abstract - *Plagiarism is a progressively widespread and growing issue within the educational field. Many plagiarism techniques square measure utilized by fraudsters, starting from a straightforward word replacement, phrase structure modification, to additional advanced techniques involving many varieties of transformation. Primarily human-based plagiarism detection is troublesome, not much accurate, and time-consuming method. In this paper, we tend to propose a plagiarism detection framework supported by 3 deep learning models: Doc2vec, Siamese Long Short-term Memory (SLSTM), and Convolutional Neural Network. Our system uses 3 layers: Preprocessing Layer together with word embedding, Learning Layers, and Detection Layer. To judge our system, we tend to dispense a study on plagiarism detection tools from the educational field and build a comparison supported a group of options. Compared to alternative works, our approach performs an honest accuracy of 97.26% and might notice differing kinds of plagiarism, permits to specify another dataset, and supports to check the document from an internet search.*

***Key Words*: Plagiarism detection, Plagiarism detection tool, Convolutional Neural Network (CNN), Deep Learning, Doc2vec, Long Short-Term Memory (LSTM).**

1. INTRODUCTION

“The Plagiarism can be conceptualized as the theft of others efforts, words or ideas without citing the right reference and therefore while not giving the correct credit to the correct person and original author [9]”. [1] Depending on the depth of transformation performed on the original text, plagiarism can be classified into different categories as Copy paste plagiarism [11], Paraphrasing [12], Use of false references [13], Plagiarism with translation [14], and Plagiarism of ideas [15]. Plagiarism is applied in various areas which include literature, music, software, scientific articles, newspapers, advertisements, websites, etc. As the use of the internet increases plagiarism becomes a big challenge in schools, institutions, and universities to maintain academic integrity. Web search engines become the common point of view to retrieve and find needed information. Hence, evaluating search engine quality [18] is a hot topic that attracts many researchers' attention [18]. People commonly use web search engines to find what they want. However, as search engines become an efficient and effective tool [17], plagiarists can grab, reassemble and redistribute text contents without much difficulty [17]. TensorFlow is an open-source library which is significant for diverse applications of deep learning programming tasks [16]. The deep learning model can be trained using high-level Keras API [16].

1.1 Problem Statement

Design and develop the plagiarism detection system which can detect different types of plagiarisms and the fraud submissions which might be copied from others work using deep learning and machine learning algorithms.

In 2020 El Mostafa Hambi and his team came with a Research Paper entitled "A New Online Plagiarism Detection System based on Deep Learning" [1] at IJACSA. In this paper, Deep Learning based model is proposed which identifies the plagiarism using Doc2vec, Long Short-Term Memory (LSTM) and CNN algorithms and gives respective plagiarism percentage.

Considering the above-mentioned problems in mind we decided to design a system that will help to identify the uniqueness of the document uploaded.

1.2 Literature Survey

“A New Online Plagiarism Detection System based on Deep Learning [1]” paper proposed an online plagiarism detection system which is based on Doc2vec technique for word embedding in coordination with SLSTM and CNN deep learning algorithms. “Code Plagiarism Detection Method Based on Code Similarity and Student Behavior Characteristics [2]” paper proposed the concept of code similarity concentration. We learn to focus on the similarity between two documents from this paper. “Machine Learning Models for Paraphrase Identification and its Applications on Plagiarism Detection [3]” paper identify if a sentence is a paraphrase of another one. Paraphrasing [5] is what where you copy someone’s exact words and put them in quotation marks. From this paper we studied LSTM and RNN algorithms. Plagiarism detection is done using string matching algorithm, k-gram and karp Rabin algorithm. This algorithms are used to detect the originality of students work based on similarity concentration.

“Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer [4]” paper studied a support vector machine based paraphrase recognition system. SVM is one of the algorithm in machine learning, which works by extracting lexical, syntactic, and semantic features from input text has been used. In "Extending Web Search for Online Plagiarism Detection [17]", Yi-Ting Liu and team developed an online plagiarism detection system to reduce misapplication of search engines. They extracted and verify the suspicious documents through the collaboration of plagiarism detection system and search engines.

2. PROPOSED SYSTEM

Plagiarism is nothing but quoting directly another person's language, data, or illustrations without clear indication that the authorship is not your own and due acknowledgment/authorization of the source.

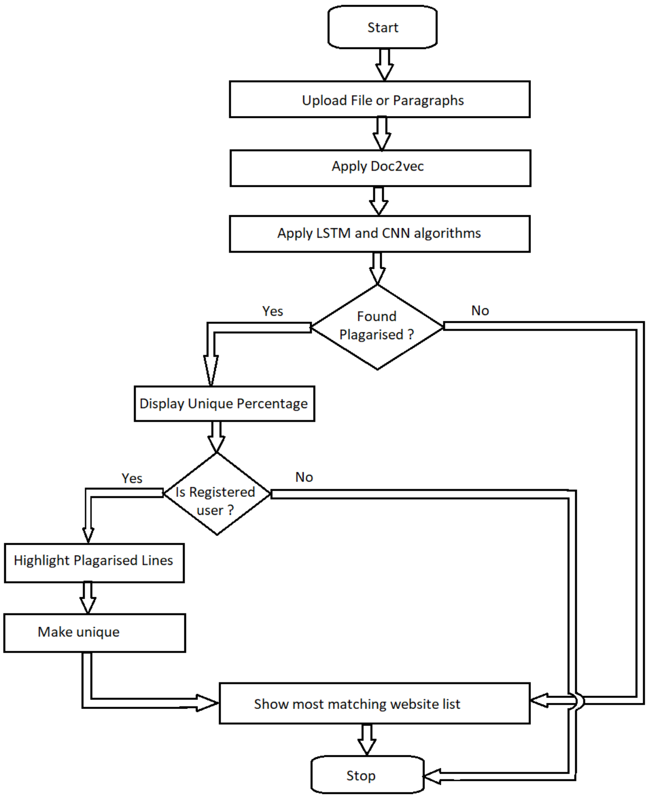
Start
Upload File or Paragraphs
Apply Doc2vec
Apply LSTM and CNN algorithms
Found Plagarised ?
Yes
No
Display Unique Percentage
Is Registered user ?
Yes
No
Highlight Plagarised Lines
Make unique
Show most matching website list
Stop

Fig -1: State Diagram

This project detects plagiarism with paraphrase recognition Using Deep Learning. Unregistered users can perform the plag-check operation without sign-up but cannot view full detailed report with additional features. It includes counting of words, report generation, report download, font changing, highlighting the stolen part, suggestions for words. If user want's to access these features then he has to register himself on our website. To sign up, user needs to do registration using G-mail, phone number, and username. If the user is a registered member then he/she can login using the login credentials. At the end, such users can download the plagiarism report which they can use as a valid document to show uniqueness of their work.

In this Project, Python Framework and Deep Learning Will be used. Python will be used in frontend as well as in backend. Python will also be used to implement deep learning algorithms and which will help in using Doc2vec module for data preprocessing. Also help in building CNN architecture. First of All, we will take paragraph or suspicious document for plagiarism checking from user. Line by line all the sentences from the document will be taken and passed to Doc2vec where data pre-processing will be done. Later on, the sentence vectors which are given as a output after data pre-processing will be shared with first learning phase which is based on LSTM algorithm and then from the CNN algorithm for plagiarism detection.

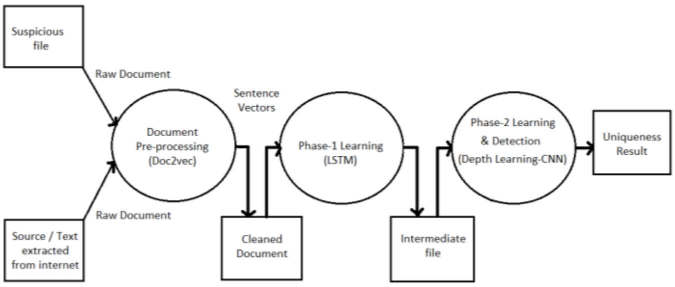

Fig -2: Data Flow Diagram

Here two deep learning approaches are used for more accurate results. Once we get the plagiarism percentage we will display it to the screen and check whether there is active user session or not. If active session is found which means the user accessing our site is registered to our system and then we will display the detailed report with some advance features. If no active session is found then flow is redirected to the registration window after showing the plagiarism percentage to the unregistered user.

2.1 Algorithms

Doc2vec:

Doc2vec is a deep learning technique that is used to represent words as features of vectors with high precision [10]. In this method, a text is considered as bag of words where there is no more order, and with each word we associate a weight which makes it possible to measure its importance in the text. A text is transformed into a vector in a large space where each coordinate corresponds to the degree of importance of a given word in the text.

This new illustration contains a serious a part of syntactical in addition as linguistics rules of the text information. A lot of larger units like "phrases, sentences and documents" ought to be represented as a vector. The paragraph vector learning approach is based on word vector learning methods. The inspiration is that the vector words are asked to contribute to a prediction task regarding consecutive word at intervals the sentence. [6]

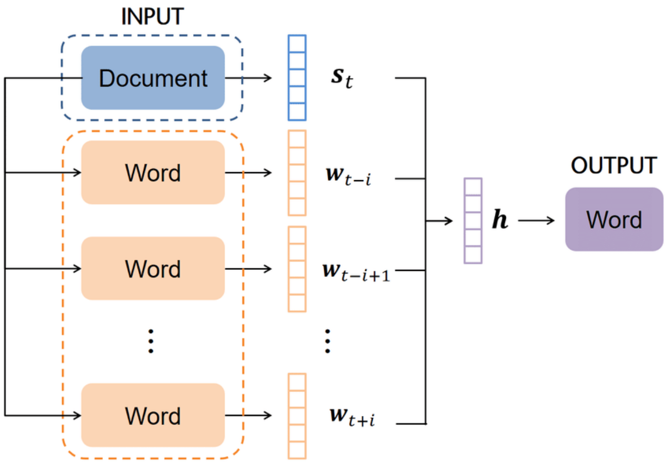

Fig -3: Doc2vec Algorithm

So, despite the fact that the word vectors are randomly initialized, it can eventually capture the semantics as an indirect result of the prediction task. They use this idea in our paragraph vectors in a similar way. Paragraph vectors are invited to contribute to the task of predicting sequential word, in many contexts sampled from the paragraph.

LSTM:

LSTM learns the long-term dependencies of the text. After taking the word embedding as input it captures maximum information from the text. The first step in our LSTM is to come to a decision what data we're getting to throw far away from the cell state. Consequent step is to come to a decision what new information we're reaching to store within the cell state. This has 2 components. First, a sigmoid layer referred to as the "input gate layer" decides that values we'll update. [7]

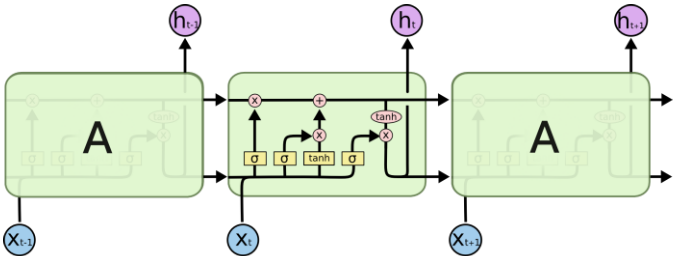

Fig -4: LSTM Algorithm [7]

Next, a *tanh* layer creates a vector of recent candidate values that would be accessorial to the state. In the next step, we will combine these two layers to create an update to the state. Finally, we need to decide what we're going to give as an output. A common LSTM unit consists of a cell, input gate, output gate and a forget gate. The cell remembers values over discretionary time intervals and also the 3 gates regulate the flow of knowledge into and out of the cell.

CNN:

CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) that uses a variation of multilayer perceptions designed to require minimal preprocessing. These are inspired by animal visual cortex. CNNs are usually utilized in computer vision; but, they need recently been applied to numerous NIP tasks sort of a text classification. It reduces the no. of features in dataset by creating the new and reduced dataset gives us information contained in original set of features. It is performed by convolutional layer and subsampling layer. And the classification is performed by dense layer and softmax layer. [8]

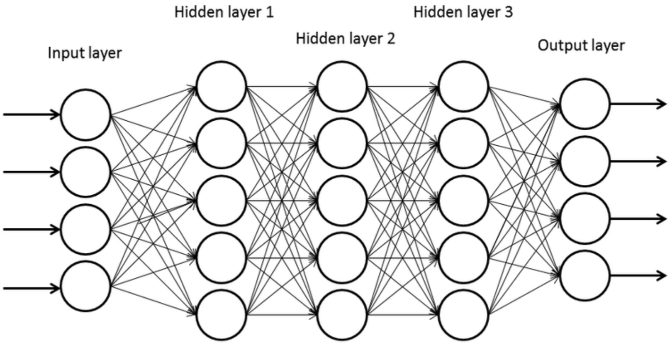

Fig -5: CNN Algorithm

3. CONCLUSIONS

In this paper, we proposed a new system for the detection of plagiarism based on the deep learning methods. Its interest is the extraction of characteristics without losing the sense of the document by using doc2vec word embedding technique. The system proposed has the ability to detect not only that there is plagiarism but also the probabilities of the existence of each type of plagiarism. We presented the different services offered by our system, either at the level of the personalized learning phase or the different ways of detecting plagiarism offered. When compared to the other tools studied, our theory offers more functionalities as adding and training new database or using a special database for comparison. As for our views, we will improve the various interfaces of the application to make it more accessible to the general public and improve the response time due to the learning time. Also it would be interesting to compare the performance of various approaches in a quantitative way.

ACKNOWLEDGEMENT

This paper is supported by many people, some of whom have a direct role and some of them have an indirect way by publishing their research online which helped to understand this concept easily. We express our deepest gratitude, sincere thanks and deep feeling of appreciation to our Project Guide Professor Pritam R. Ahire, his presence at any time throughout the Semester, important guidance, opinion, comment, critics, encouragement, and support greatly improved this project work. We thank the college administration for providing the necessary infrastructure and technical support. Finally, we extend our heartfelt thanks to our friends and family members.

REFERENCES

[1] El Mostafa Hambi , Faouzia Benabbou, "A New Online Plagiarism Detection System based on Deep Learning", (IJACSA) International Journal of Advanced Computer Science and Applications Vol. 11, No. 9, 2020

[2] Qiubo Huang, Xuezhi Song, Xuezhi Song, "Code Plagiarism Detection Method Based on Code Similarity and Student Behavior Characteristics", IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2020

[3] Ethan Hunt, Ritvik Janamsetty and team, "Machine Learning Models for Paraphrase Identification and its Applications on Plagiarism Detection", IEEE International Conference on Big Knowledge (ICBK) 2019

[4] Yahia Jazyah, "Open Learning, the Issue of Plagiarism - Efficient Algorithm", International Journal of Computers, Volume 3, 2018

[5] Chitra and Anupriya Rajkumar, "Plagiarism Detection Using Machine Learning-Based Paraphrase Recognizer", J. Intell. Syst. 2016; 25(3): 351–359

[6] Kim, Do-Guk & Ko, Bonggyun. (2019). Investment Universe Construction Based on the Theme Keyword Search. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2939414.

[7] Janardhanan, Deepak & Barrett, Enda. (2018). CPU Workload forecasting of Machines in Data Centers using LSTM Recurrent Neural Networks and ARIMA Models. 10.23919/ICITST.2017.8356346.

[8] Miralles, Luis & Rosso, Dafne & Jiménez, Fernando & García, José. (2017). A methodology based on Deep Learning for advert value calculation in CPM, CPC and CPA networks. Soft Computing. 21. 1-15. 10.1007/s00500-016-2468-4.

[9] Risquez, A., Dwyer, M. O.; Ledwith, A. (2011). «'Thou shalt not plagiarize': from self-reported views to recognition and avoidance of plagiarism». Assessment & Evaluation in Higher Education, vol. 2, no. 1, p. 34-43. http://doi.org/10.1080/02602938.2011.596926. 3 Ruipérez, G.; García-Cabrero, J.C. (2016). «Plagiarism and Academic Integrity in Germany». Comunicar, vol. 24, no. 48, p. 9-17. http://doi.org/10.3916/C48-2016-01.

[10] Suleiman, D., Awajan, A., Al-Madi, N. (2017). Deep Learning Based Technique for Plagiarism Detection in Arabic Texts. 2017 International Conference on New Trends in Computing Sciences (ICTCS). doi:10.1109/ictcs.2017.42

[11] Thomas Lancaster, Fintan Culwin. A Visual Argument for Plagiarism Detection using Word Pairs. School of Computing University of Central England Perry Barr Birmingham B42 2SU United Kingdom. Faculty of Business, Computing and Information Management London South Bank University Borough Road London SE1 0AA. Plagiarism: Prevention, Practice and Policies 2004 Conference.

[12] 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.

[13] Maxim Mozgovo, Tuomo Kakkonen, Georgina Cosma. Automatic student plagiarism detection: future perspectives. University of aizutsuruga, ikki-machi, aizu-wakamatsu, fukushima, 965-8580 japan. Article in journal of educational computing research · January 2010.

[14] Sousa-silva, r. -detecting trans lingual plagiarism and the backlash against translation plagiarists language and law / linguagem e direito, vol. 1(1), 2014, p. 70-94.

[15] Eman s. Al-shamery, Hadeel qasem Gheni. Plagiarism detection using semantic analysis. Published 2016. Computer science Indian journal of science and technology. doi:10.17485/ijst/2016/v9i1/84235 corpus id: 55709933.

[16] Ullah, F., Wang, J., Jabbar, S., Al-Turjman, F., & Alazab, M. (2019). Source Code Authorship Attribution Using Hybrid Approach of Program Dependence Graph and Deep Learning Model. IEEE Access, 7, 141987–141999. doi:10.1109/access.2019.2943639

[17] Liu, Y.-T., Zhang, H.-R., Chen, T.-W., & Teng, W.-G. (2007). Extending Web Search for Online Plagiarism Detection. 2007 IEEE International Conference on Information Reuse and Integration. doi:10.1109/iri.2007.4296615

[18] Shoeleh, F., Azimzadeh, M., Mirzaei, A., & Farhoodi, M. (2016). Similarity based Automatic Web Search Engine Evaluation. 2016 8th International Symposium on Telecommunications (IST). doi: 10.1109/istel.2016.7881901