z
International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 05 | May 2021 www.irjet.net p-ISSN: 2395-0072

Plagiarism Detection Tool for Coding Platform using Machine Learning

P.Ashwin[1], M.B.Boominathan[2], G.Suresh[3]

[1,2]B.E students, Department of Electronics and Communication Engineering , Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India - 638401.
[3]Assistant Professor, Department of Electronics and Communication Engineering , Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India - 638401.

Abstract - *Plagiarism is known as illegal use of words or sentences or the whole context from Wikipedia, journals, paper publications, paper presentation, books, research articles, lab assignments or any other source of online sites. Still the accuracy of the plagiarism software is not very perfect in the coding domain as it can be modified and changed easily. So we can use machine learning to overcome this problem. Machine learning is the area of artificial intelligence that uses logical techniques, so the system gains self-learning ability. It is used to design algorithms based on data trends and historical relationships between data. The process of learning begins with observations or data, such as examples, direct experience or instructions, in order to look for patterns in data and make better decisions in future, based on the examples we provide. To allow the computers to learn automatically is the primary aim. There are three kinds of machine learning supervised, unsupervised and reinforcement machine learning. Supervised machine learning systems provide the learning algorithms with known quantities to support future judgment. In supervised machine learning the training data includes a set of inputs and corresponding outputs, whereas in unsupervised learning the data are classified according to their similarities. In reinforcement learning the activity of the machine has an effect on the environment. Thus, machine learning plays a great role in the day–to-day world, which includes recognition of spam in emails, online customers In online coding exam platforms, they use third party own or others plagiarism checker tools. So the required network bandwidth is high. Also still many people use others' code and make some little adjustment and submit it. In order to overcome this problem, we incorporated the plagiarism checker tool inbuilt in the coding site eliminating the need of third party plagiarism checker tool. Also we use several algorithm techniques with help of machine learning to detect plagiarism more accurately.*

***Key Words*: Machine learning, n-gram, Anaconda cloud, Django, Node.js.**

1. INTRODUCTION

Plagiarism is done by many people nowadays. People find different ways to cheat even the plagiarism detector. So new techniques are needed to make the detector more powerful. It does not only include copying of words but also any type of copying others work like copying of images, graphical representation, programming codes and so on. Even translating someone's work in another language without their permission is also said as plagiarism. Plagiarism is copying the content of others ideas and using them in your work like it's their own work without giving any credits to the other. So we use machine learning to make the plagiarism checking tool work efficient and faster. Machine learning is a subset of AI. AI helps to create algorithms based on the trends of data and historical relationships of data. It is one big part of artificial intelligence. The concept of machine learning is nothing but learning the mistakes from data and rectifying it. It helps to find the similarities between the data and make the approach of humans lesser. The machine learning field is in disciplinary, that contains a number of sciences, professions mathematics, computer science, neuroscience, etc. Machine learning play a major role in weather prediction, social media advertisement, movie recommendation and E-commerce etc. Most social media use machine learning to make them user friendly.

2. LITERATURE SURVEY

In [1], the author has done a project on Software plagiarism detection on multiple programming languages using machine learning approach. They have proposed this method by Principal component Analysis (PCA) for feature extraction from the source code. Further they applied multinominal logistic regression model to categorize the source code based on predictions.

In [2] paper, the authors have used Naive-Bayes and Support Vector Machine algorithms for learning. The metric returns the performance of the model in terms of accuracy and certainty for each prediction. They used this method for the learning of similar words, similarity of fingerprint, LSA similarity. SVM algorithms have a higher accuracy of 92.76% than Naïve Bayes which gave an accuracy of 54.29% with use of word similarity features.

In [3], the author created a prototype called Deimos to find the source code plagiarism. Deimos performs in the following ways. One is parsing and making it to token for text pre-processing. Second one is comparing tokens using the Karp-Rabin greedy algorithm. They also created a web application using PHP and the backend of the web application is done with the help of java. They used a web application which is used to eliminate the problem of timeout. The application is also designed to consume less data rate.

3. PROPOSED METHODOLOGY

In the proposed system we used the following software to meet our requirements

3.1 Software Requirements

3.1.1 Anaconda cloud

Anaconda navigator is a virtual environment manager that comes with anaconda distributors and 1,000 other data packages. It eliminates the need of downloading specific libraries and studying about it. It is a desktop graphical user interference that permits the users to run and manage anaconda packages. It needs packages so it searches in the cloud under local repository. It can be used in any OS like Windows, Linux, and Mac OS. It is a package management service where you can surf public and private notes, environments, and packages. Cloud hosts useful python packages, notes and environment for a large variation of applications. We can build new packages through CLI (Anaconda Client Command Line Interference), then it will upload it to the cloud.

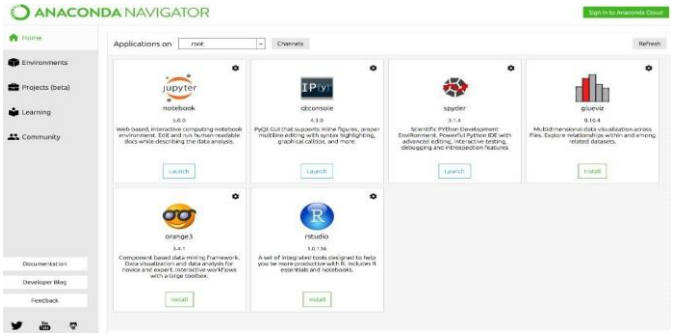

Fig-1: Anaconda Navigator cloud

3.1.2 Django, node.js-Tools and frameworks

This is where users interacts with your online judge. It can either be a website (e.g. code chef). User accounts - people should be able to signup/login, and see his/her submissions history. Problem setter profiles - another set of users/admins who upload problems along with test case. Submit programs - The user should be allowed to submit programs for the problem of his choice. Submissions - a submissions page that tells the user what happened to his program (AC, WA, TLE, RTE... etc.).

3.1.3 Django Model Architecture and API Call

We have used Django and JavaScript from login and logout roots. Finally, the app is integrated with the Django for code execution and submission plagiarism check throughout the documents. The app works on local environment to host online assessments and contest. The same is extended as online platform as a future work.

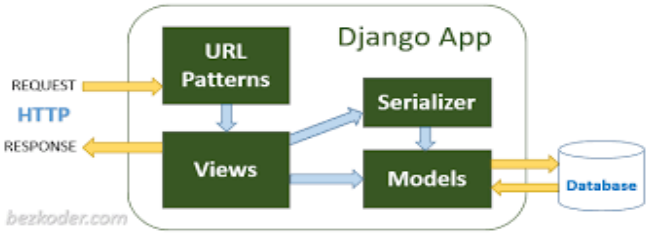

Fig-2: Anaconda Navigator cloud

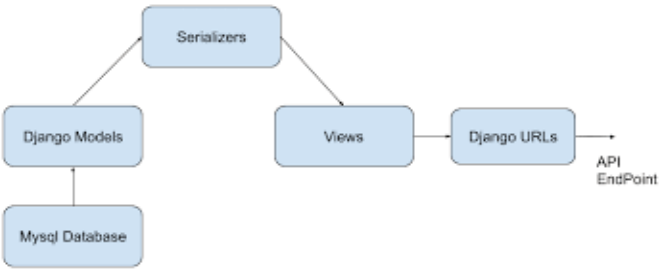

Fig-3: Anaconda Navigator cloud

3.1.4 Code execution API

A multiple code evaluation API engine is built to getting the submission status of the work. The program evaluates the code submitted by the different person and evaluates the input output test case validations. The status of the code is displayed.

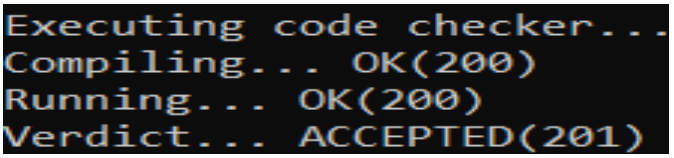

Fig-4: Anaconda Navigator cloud

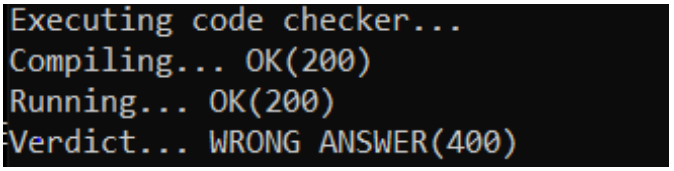

Fig-5: Anaconda Navigator cloud

4. SYSTEM ARCHITECTURE

For testing the dataset is obtained from the data obtained from user. For training and preprocessing the dataset is obtained from inbuilt libraries. There are plenty of libraries available in machine learning to train different types of datasets. The library contains things like stop words which removes unnecessary words like the, which, of, etc., which doesn't needed in plagiarism check and it also increases the running time of checking thus decreases the performance.

z

4.1 Normalization

In Natural Language Processing, the words which cannot used be as data can be removed using stop words. These words can be easily removed by storing the useless words in a database which is termed as stop words in NLP. In NLP, stop words are available in 16 different languages.

Sample text with stop words	Without stop words
He is the best student in class.	Best, student, class
It is a Chinese hotel.	Chinese, hotel
I love walking. So I walk.	Love, walking, walk

Fig-6: Anaconda Navigator cloud

4.2 Tokenization

A phrase, sentence or paragraph is divided into individual smaller parts like word. Such smaller terms is called tokens. The tokenization process is carried out to extract raw text into smaller terms. The token later helps us in understanding the context or meaning of the text. So, the plagiarism tools tends to detect plagiarism on document submissions. In the plagiarism detector we tend to use nltk plunkt model for the tokenization process. Tokenization is done with the help of word boundaries which is nothing but space for word tokenization and full stop for sentence tokenization. Tokenization is very first stage of text pre-processing as it leads to the next stage of pre-processing namely lemmatization and stemming.

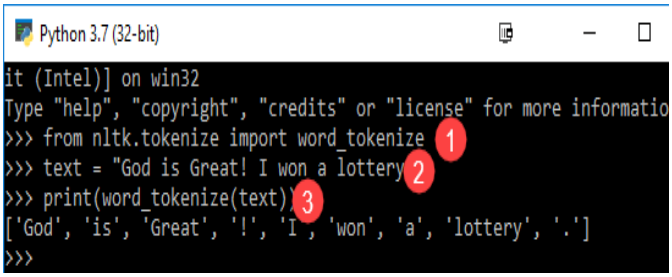

Fig-7: Anaconda Navigator cloud

4.3 Cosine similarity algorithm

After completing pre-processing, the cosine similarity algorithm is used in plagiarism detection. It is nothing but a similarity and differences of vectors between two or among many documents. It finds the distance and angle of separation between two vectors. If the distance between two vectors increases, the angle between the vectors also increases which makes similarity between the vectors decreases and vice versa. It also used in recommendation systems and also used to classify the documents easily. Every time when a code is submitted in the coding platform, it is saved on a library. It is then checked with all other saved codes for the plagiarism check. To implement cosine similarity algorithm, we should first vectorize the target documents after completing all stages of pre-processing.

After vectorization, we find the similarities using the formula. The result of the formula ranges from 0 to 1. The accuracy of the result is more when cosine similarity is performed with stemming feature than without doing stemming feature. When the similarity index is greater than
0.7, it can be said as copied and the person who copied will be rejected from the test. But the cosine similarity is not enough algorithm as it has less accuracy. So we moved on to n-gram algorithm for better and much accuracy.

4.4 N-Gram algorithm

N-gram is nonstop order of N strings which may be word, sentence, letter, syllable or phrase from the given text. The n denotes the number of string pairs.

E.g.: "There are many people in the world"

1) Unigram- n (1) = (There), (are), (many), (people), (in), (the), (world)
2) Bigram- n (2) = (There, are), (are, many), (many, people), (people, in), (in, the), (the, world)
3) Trigram- n (3) = (There, are, many), (are, many, people), (many, people, in), (people, in, the), (in, the, world)

4.5 N-gram N parameter analysis

Cosine similarity cannot detect change in meaning of a sentence. The below grammar and meaning analysis is for N-gram only.

N	N-gram test	cosine test	grammar and meaning n-gram
1	0.9767441860465116	0.9896567391098409	low
2	0.9245283018867925	0.9650608087224455	high
3	0.8867924528301887	0.9401880564188065	high
4	0.8431372549019608	0.9151007705656548	Less compared to 2 and 3
5	0.7959183673469388	0.8865926413116155	reduces
6	0.7826086956521774	0.8783100656536798	Reduces and stable point
7	0.7674418604651163	0.8687219087128831	reduces

Fig-7: Anaconda Navigator cloud

z

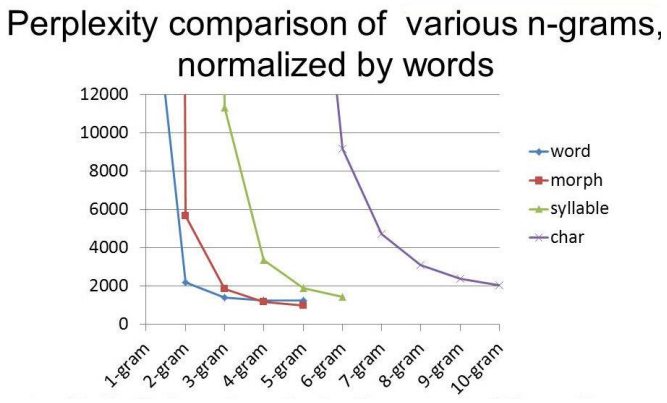

Fig-8: Anaconda Navigator cloud

5. CONCLUSION AND FUTURE SCOPE

From the analysis, the plagiarism in the text documents are measured using trigram and cosine similarity measures in text documents. We have proposed both the cosine similarity and n-gram analysis on the obtained submissions and obtained a most promising outcome. The future work of the project involves scrapping the web content based on the input and take that data in to consideration. And also to integrate MOSS plagiarism API for the best optimal performance and to work on Real-time massive data uploads from the application

ACKNOWLEDGEMENT

The authors acknowledge the Management of Bannari Amman Institute of Technology, Sathyamangalam for providing high standard infrastructure and laboratory facilities to carry out this work.

REFERENCES

[1] Farhan Ullah, "Software Plagiarism detection in multiprogramming language using machine learning approach", Concurrency and Computation: Practice and Experience 2018.

[2] Zakiy Firdaus Alfikri, Ayu Purwarianti, "Detailed Analysis of Extrinsic Plagiarism Detection System Using Machine Learning Approach", 2014 TELKOMNIKA Indonesian Journal of Electrical Engineering.

[3] Cynthia Kustanto, Inggriani Liem, "Automatic Source Code plagiarism Detection", IEEE Xplore 2009.