Plagiarism Detection Using Artificial Intelligence Technique In Multiple Files

Mausumi Sahu

Abstract: Plagiarism relates to the act of taking information or ideas of someone else and demand it as your own. Basically it reproduce the existing information in modified format. In every field of education, it becomes a serious issue. Various techniques and tools are derived these days to detect plagiarism. Various types of plagiarism are there like text matching, copy paste, grammar based method etc.This paper proposes a new method implemented in a program ,where we utilise a text set to identify the copied part by comparing with some existing multiple files. Here we put the concept of a machine learning language i.e k-NN. It helps us to identify whether a paper is plagiarized or not.

Index Terms: k-nearest neighbor, machine learning, plagiarism detection, text matching.

1 Introduction

The issue of plagiarism is often discussed in the educational community across the world. It relates to the act of taking another person's work or ideas and passing it off as your own without giving credit to the original writer. Basically it reproduce the existing information in a modified format. Plagiarism is defined by S. Hannabuss as "is the act of imitating or copying or using somebody else's creation or idea without permission and presenting it as one's own [5]. Today with the huge popularity of internet, so many documents are freely accessible. Now internet is a extensive source to collect data. People can easily get their required information or data from internet and make their copy instead of writing their own text document. As recent trends show, the detection of plagiarism becomes more important as it is very easy for a plagiarist to find an appropriate text fragment that can be copied. On the other side it becomes increasingly difficult to correctly identify plagiarized sections due to the large amount of possible sources[7]. Plagiarism cases are an everyday topic, for example, in academics, journalism, scientific research and even in politics. This approach to plagiarism detection is especially useful when no reference collection is available or not all the possible copy sources are present, thus document-to-document comparison algorithms cannot be used. Plagiarism is of various types like literal, integral, intrinsic, extrinsic, exact copy, text manipulation etc [3]. Similarly various plagiarism detection methodologies are present to detect plagiarism. Presently systems which are based on the text manipulation technique are not accurate enough for practical applications. Therefore, we have proposed a new easy method which is based on the text identification technique through file transfer method which uses a machine learning approach in order to detect plagiarism between text sets. It compares two files and identify how many words are similar between two files then we calculate a percentage value according to our threshold value required to detect plagiarism, through which we can get the plagiarised text series and frequency of each copied set of words too. The rest of this paper is arranged as follows. Section. II explains the basic concepts on the topics related on plagiarism and its detection techniques. Section.III explains the proposed work and its objectives. Section. IV explores implementation and result of our work. In section. V, we discuss the related work and depicts a comparison with our work. In Section. VI we mention what would be possible in our future work and finally our last section.VII conclude with the summarizing results, respectively.

2 BASIC CONCEPTS

Plagiarism

Plagiarism in universities is an important problem, remaining as a topic for scientific works for years[11]. Plagiarism is an a mistake mostly in academic field. It takes others contributions without their permission and does not give honour to the originator. Reprobates are rewarded though they are not deserve for that. We can observe plagiarism occur in various field like literature, academic, science, music vastly. It can be also possible that one day we will get our project work in another publication without proper citation. Plagiarism detection techniques are there, which are classified into character based method, structural-based method, classification or cluster based method, Syntax-Based Methods, Cross language-Based Methods, Semantic-Based Methods and Citation-Based Methods. Various tools are available using the above plagiarism methods[10].The later problem turns out to be a technical task in many cases, since plagiarism detection can be effectively done with the help of computer tools. There are three types of plagiarists we can see[12] defined as follows:

- Accidental plagiarist: Due to lack of understanding, the student does poor academic practice and named as cheater.
- Opportunistic plagiarist: Though they are aware of things going ‘wrong’ but does due to some higher authority of pressure or have a belief that it will result good in future.
- Committed plagiarist: Some misrepresentation occurs which called as intentional cheatings.

Majority of plagiarism carried out by Students. They normally unaware with the academic requirements, so the majority of accidental plagiarists are shown in students.

k-NN

The k-Nearest Neighbor Algorithm is one of the simplest machine learning algorithms that is suitable for pattern recognition. The k-Nearest Neighbor (kNN) algorithm often performs well in most pattern recognition applications [2]. "k" is a parameter in the kNN algorithm. It is necessary to select the correct k value for the kNN algorithm by conducting several tests with various k values. k-NN assigns its "k" nearest neighbors in the text data set. kNN simply memorizes all the documents in the training dataset and compares the given test document against the training dataset. For this reason the kNN is also known as “memory-based learning” or “instance-based learning”[2]. The k-NN classifier is an instance-based classifier, which classify a set of training document for one particular category and will collect all important words and possible distribution in this category. Sometimes when we do classification some key words used in a text, out of a training set may be assigned to the wrong category or falsely classified. So practically to establish such a training set is infeasible[8]. There are some drawbacks like the related documents and contents are required. So k-NN can be used to overcome the above drawbacks effectively and successfully by using a particular text classifier. Training sets have their different taxonomic standards and use their own concept levels to identify and differentiate the document content. Till now the k-NN algorithm has been applied in various field. It is used in text categorization due to its simplicity and accuracy. We can categorize an unknown document, by using this k-NN classifier and rank the document's neighbours among the training documents. It uses the class labels of the k most similar neighbours in text document. Similarity between two documents may be measured by some existing theories like the Euclidean distance, cosine measure, etc. The similarity measure of each nearest neighbour document to the test document is used as the weight of the classes of the neighbour document. If a specific category is shared by more than one of the k-nearest neighbours, then the sum of the similarity scores of those neighbours is obtained from the weight of that particular shared category. Conceptually KNN is used for classification and we use this concept in our paper to classify a text set from any particular paper and do compare with a existing stored paper from database to find out plagiarism.

Text Based System

Text analysing can be achieved to determine whether the text has been copied or not. Statistics measure can decide whether a text has been plagiarised by identifying the frequency of the words and sentences[5]. To find the plagiarism a text is being compared to other text. Parsing is a technique used to parse the text set. By using parsing we can find how a text set is classified into tokens. It have various ways to implement like string tokenizer , stream tokenizer , scanner class , pattern and matcher class. Mainly tokenizer is used to split a sentence and breaks a string into tokens. A string can be tokenized by a string tokenizer. Some operations advance this current position past the characters processed. A token is returned by taking a substring of the string that was used to create the String Tokenizer object.

3 PROPOSED WORK

We propound a technique to identify plagiarism in any document. We have consider clustering methods k-NN which cluster the string and match each word with it's neighbour words. We thought about the issue involved with plagiarism detection and try to make it easy to find. We have used a method to compare the String. We have used counter to count the string matched in the text files. Also we can find out the percentage of frequency of each word as well as the whole document. We write a code by using the k-NN, which not only find plagiarism but also show how the plagiarism detection work is held. k-NN concept tells us to find out the nearest neighbour of the particular event. Then compare with those neighbour and gather the similar events at a place. Using this concept only we implement our plagiarism program. So first we take any sample paper to test our project. There was some existing papers are saved in the format of file in our database. First the text file is parsing by the use of parser. Split the string then do compare with the other. Then we make a comparison with the query file with the stored files. Then we will get the related similar words at a place, then sentences. Then we give a condition to find out the percentage of each copied or similar word. From that we can find out the percentage as well as frequency of each word. After that we it do the same method for sentences too. So we now have the frequency of each copied word as well as each sentence. From that we can find the percentage of plagiarism occur in the particular given paper. We fix a threshold value to find plagiarism. If the percentage we got are more than 30 or 40 percent, we can tell that there is plagiarism occur otherwise it will show plagiarism not found. Now we propose this much. After wards we are planning to make it more efficient to get the result. Here we also found a way to know how plagiarism tools are working. When we This addresses a significant and clear idea about how to detect plagiarism for multiple files.

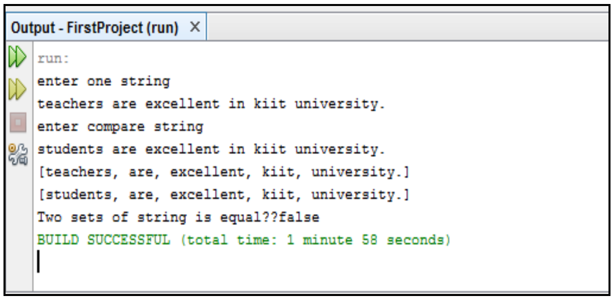
- Here, first we have to retrieve a file and do compare with the other existing files related to it.
- We have used KNN algorithm to find the 'k' nearest neighbour of each word in the text document.
- The set of words which are matched between the target file and related files, are selected as copied words and showed as output.
- It will also find the frequency of each copied set of words in the file and calculates the percentage. As we have already mention that it will give output according to the defined threshold value.
- We can also take multiple files which is the most beneficial. But we have to classify the file first and have to store a lots of related file.
- So we will plan to expand our project in every field to get more appropriate result. From this we can get the best result.
- Our output showed to the user as, plagiarism occured or not in the particular paper. Like this we can say whether a project or a thesis is plagiarised or not.

The next section gives the result obtained during the research:

4 IMPLEMENTATION AND RESULT

In this section, we discuss the implementation of our algorithm in java platform. We first use the parsing technique to parse the text into its constituent data. It returns a set of tokens to be used for pattern matching and compare whether two strings are equal or not.

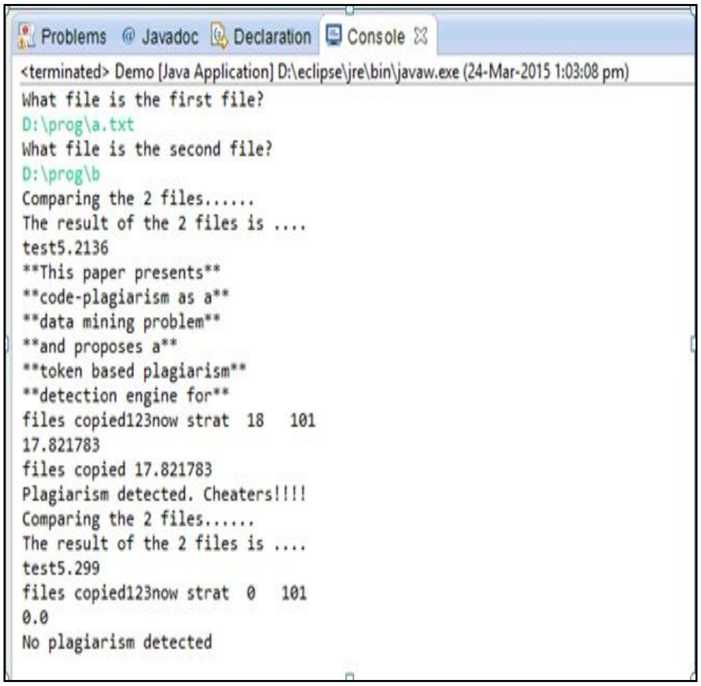
Fig. 1. *Snapshot of the output of parse the text set.*



```
run:
enter one string
teachers are excellent in kiit university.
enter compare string
students are excellent in kiit university.
[teachers, are, excellent, kiit, university.]
[students, are, excellent, kiit, university.]
Two sets of string is equal??false
BUILD SUCCESSFUL (total time: 1 minute 58 seconds)
```

The second phase of our work is to read two set of files and then cluster them into two set of arrays to compare its similarities. Each word of the target file is compared with the each word of second file and by using k-NN the consecutive neighbour words are identified. We have performed the experiment taking different values for 'k'. If the files have matched set of tokens then we conclude plagiarism exists, otherwise it will show no plagiarism occur. But when our plagiarism percentage is more than our fixed threshold value then only it will show plagiarism occur otherwise not. We have used two files to compare and implement this logic for the research and we found the following result. To show our result we take a file to compare a.txt with the file b.txt, where in b.txt we have stored multiple number of file like b1,b2,b3 etc. Now we compare a.txt with b. It compare with b1 first and give the output 17 percent copied so plagiarism detected and the second part compares with b2 and give result as no plagiarism detected. The snapshot of the result is given.

Fig. 2. *Snapshot of final output of text set between multiple files.*



```
<terminated> Demo [Java Application] D:\eclipse\jre\bin\javaw.exe (24-Mar-2015 1:03:08 pm)
What file is the first file?
D:\prog\a.txt
What file is the second file?
D:\prog\b
Comparing the 2 files......
The result of the 2 files is ....
test5.2136
**This paper presents**
**code-plagiarism as a**
**data mining problem**
**and proposes a**
**token based plagiarism**
**detection engine for**
files copied123now strat  18   101
17.821783
files copied 17.821783
Plagiarism detected. Cheaters!!!!
Comparing the 2 files......
The result of the 2 files is ....
test5.299
files copied123now strat  0   101
0.0
No plagiarism detected
```

5 COMPARISON WITH RELATED WORK

To the best of our knowledge, previously comparison with multiple files has not be done using the KNN algorithm, which is implemented in our research. Our logical method intends to find the frequency and accurate result from the multiple files using tokenizer function clustering method. Beside this some more functionalities like each word can give the frequency of repetition we achieve in our project which was not implemented previously. We compare our work with some more papers based on plagiarism which already exist,

- In, plagiarism using machine learning language they used some algorithm to detect plagiarism but when we use some code or tool on it then it will not work properly. But in our work we can take any type of paper and can detect plagiarism.
- Previously there are methods using artificial intelligence technique for plagiarism but always it does not give better performance. So we tried to implement k-NN, an artificial intelligence method to produce better result.
- In grammar rule method also readers face problems like delay in getting result and accuracy in result. Our method detects plagiarism accurately and time consumed is also very less .

6 FUTURE SCOPE

In our project we have implement the program through which we actually know how various plagiarism tools are working. So we are planning to add more artificial intelligence method to get more accurate result in future. We will try to know the internal functions of each plagiarism detection tools. We are also planning to store or make a list of all the files related to similar field, so that searching method is become more easier.

7 CONCLUSION

Plagiarism involves reproducing the existing information in modified format. Today it is found in almost all fields of human activities so a lot of attention is given to identify and detect plagiarism. Some experimental results show that in general there is improvement performance in the use of hybrid machine learning methods in the case of plagiarism. However, the hybrid method does not always produce better performance. So we have designed a process using machine learning method i.e k-NN which will improve the performance. Comparing all methods in this area, we can conclude that the k-nearest neighbour method is much useful in pattern recognition as well as to find copied dataset to detect plagiarism. Our method provide more accuracy and efficiency to detect plagiarism. For this, we have implemented the technique which shows how a text set is parsed and checks a particular file with related existing files for plagiarism detection.

8 REFERENCES

[1] Imam Much Ibnu Subroto and Ali Selamat, "Plagiarism Detection through Internet using Hybrid Artificial Neural Network and Support Vectors Machine," TELKOMNIKA, Vol.12, No.1, March 2014, pp. 209-218.

[2] Upul Bandara and Gamini Wijayrathna ,"Detection of Source Code Plagiarism Using Machine Learning Approach," International Journal of Computer Theory and Engineering, Vol. 4, No. 5, October 2012, pp.674-678.

[3] Salha Alzahrani, Naomie Salim, Ajith Abraham, and Vasile Palade," iPlag: Intelligent Plagiarism Reasoner in Scientific Publications," IEEE World Congress on Information and Communication Technologies, 2011.

[4] Barrón Cedeño, A., & Rosso, "On automatic plagiarism detection based on n-grams comparison," In Advances in Information Retrieval, Vol. 5478. Lecture Notes in Computer Science, pp. 696–700, Springer.

[5] Ahmad Gull Liaqat and Aijaz Ahmad, "Plagiarism Detection in Java Code," Degree Project, Linnaeus University, June 2011, pp. 1-7.

[6] A Selamat, IMI Subroto and Choon-Ching Ng, "Arabic Script Web Page Language Identification Using Hybrid-KNN Method," International Journal of Computational Intelligence and Applications, 2009, pp. 315-343.

[7] Michael Tschuggnall and Gunther Specht , "Detecting Plagiarism in Text Documents through Grammar-Analysis of Authors," pp. 241-255.

[8] Bill B. Wang, R I. (Bob) McKay, Hussein A. Abbass and Michael Barlow, "Learning Text Classifier using the Domain Concept Hierarchy," ACT 2600, pp. 1-5.

[9] Francisco R., Antonio G., Santiago R., Jose L., Pedraza M., and Manuel N., “Detection of Plagiarism in Programming Assignments,” IEEE Transactions on Education, vol. 51, No.2, pp.174-183, 2008.

[10] Ahmed Hamza Osman, Naomie Salim and Albaraa Abuobieda, " Survey of Text Plagiarism Detection," Computer Engineering and Applications , Vol. 1, No. 1, June 2012, pp. 1-9.

[11] Maxim Mozgovoy, Tuomo Kakkonen and Erkki Sutinen, "Using Natural Language Parsers in Plagiarism Detection," University of Joensuu Finland.

[12] Maria Kashkur, Serge Parshutin and Arkady Borisov, "Research into Plagiarism Cases and Plagiarism Detection Methods," Scientific Journal of Riga Technical University, Computer Science, Information Technology and Management Science, Vol.44, 2010.