PLAGIARISM

Plagiarism has been identified as theft of intellectual property by Lancaster and Culwin. It can further be defined as turning of someone else's work as your own without reference to the original source. Commonly observed plagiarism methods are listed below

1. Copy - Paste Plagiarism (Directly copying word to word)
2. Paraphrasing (Restating same content in different words)
3. Translated Plagiarism (Translating Content and then using it without reference)
4. Artistic Plagiarism (Presenting same Content but with different media)
5. Idea Plagiarism (Using similar kind of ideas)
6. Code Plagiarism (Using directly available code snippets with permission)
7. Misuse of references (Added incorrect or non-existing sources)

It is hard to believe that plagiarism is now becoming an integral part of our societies but people are becoming aware of the same and they are now also coming to a conclusion that a society with high ethical standards needs to be plagiarism free. The roots of plagiarism in India can be found way back in schooling and education. Students start to copy work from high scoring individuals at a very early stage perhaps due to the fact that we value grades more than actual knowledge. But at a later stage, we see that plagiarism problem's extent is increasing because of the recent advancements in the IT sector. Everyone has access to the vast knowledge bank available on the web and students especially tend to use them as it is.

There are in general two solution possible for the problem

1. Plagiarism Prevention
2. Plagiarism Detection

And as we say "Prevention is better than Cure", plagiarism prevention is an important class of solution which aims to destroy the problem from its root cause and provides long term results whereas Plagiarism Detection is something which provides short term solution. So we think that we should be using a combination of both Detection and Prevention as Detection will help in erasing the current problem and prevention works on it not becoming a problem again in future.

Method	**Attributes of method**	
	Implementation work – intensity	Duration of positive effect
Plagiarism prevention methods	Require more time to implement	Positive effect isn't momentary, but it is long – term
Plagiarism detection methods	Require less time to implement	Positive effect is momentary, but it is short – term

Plagiarism Detection Methods

Detection is nothing but assigning a similarity score to a pair of documents and then using this score value as a metric to decide whether they are similar enough or not. This score can be based on different metrics which are nothing but parameters and aspects in a document. Some of the general purpose metrics which are used are described below. Firstly, the metrics can be further classified in two major categories which are

1. Based on Number of Documents involved in the metrics calculation process
    a. Singular or Paired
    b. Corpal or Multidimensional (Operates on an entire set of corpora)
2. Based on Computational Complexity of the methods employed to find similarities.
    a. Superficial (Similarity measure gauged by simply looking, not involving knowledge of linguistic features of natural language)
    b. Structural (Similarity measure requiring knowledge of the structure of documents)

One another classification of metrics is based on how the content in the documents is analyzed, for instance

1. Semantical Methods
2. Statistical Methods
    a. Here, we have no need to understand the meaning of a document
    b. Common approach might be using the Bag of Words Model to represent Documents as a vector and then using, maybe, cosine similarity on these vectors. Other similar approaches are compression metrics, lancaster word pairs etc.
    c. These methods can be language sensitive, and are quite famous due to their simplicity.

TOOLS AVAILABLE

Judging from the extent of the problem, it is quite obvious that academic institutions require tools to automate and enhance plagiarism detection. There are a number of tools available like

1. Turnitin
2. Eve2
3. CopyCatchGold
4. WordCheck
5. Glatt
6. Moss
7. JPlag

Attributes	Detection tools							
	Specific tools							*Alternative tools*
	Turnitin	Eve2	CopyCathGold	WordCheck	Glatt	Moss	Jplag	Google Yahoo AltaVista
Type of text tool operates on								
Checks source code?	-	-	-	-	-	**Y**	**Y**	-
Checks free text?	**Y**	**Y**	**Y**	**Y**	**Y**	-	-	**Y**
Type of corpus tool operates on								
Operates intra-corpally?	**Y**	-	**Y**	**Y**	-	**Y**	**Y**	-
Operates extra-corpally?	**Y**	**Y**	-	-	-	-	-	**Y**
Other attributes								
Designed for use by students?	**Y**	-	-	-	-	-	-	**Y**
Designed for use by teachers?	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
Instant response?	-	**Y**	-	**Y**	-	-	-	**Y**
Free?	-	-	-	-	-	**Y**	**Y**	**Y**

The two main attributes which are common among all the tools are

1. Type of text the tool operates on, two types are possible here
    a. Tools operating on non structured data (Free text)
    b. Tools operating on Structure data (Source Code)
2. Type of Corpus the tool operates on, three types are possible
    a. Tools that operate intra-corpally
    b. Tools operating extra corpally
    c. Tools operating on both

From the available descriptions of some detection tools it may be concluded that the great majority of tools uses statistical methods to detect plagiarism due to the fact that these methods are easily implemented and are relatively easier to understand. Despite the advancements in plagiarism detection tools it is still not fair to directly classify a paper as plagiarized based on the conclusion of a tool because these tools are still not able to distinguish cited text from plagiarized text.

CMU - CASE STUDY

It has been found in a survey that the problem of plagiarism is of major concern in the CS department of Carnegie Mellon University. Professors have deployed a number of methods so as to control plagiarism. Few of them are

1. Making students answer oral questions based on the assignment.
2. If the performance in examinations and the assignment is not aligning with other then this is a highly probable case of plagiarism.

But the major drawback is these methods are highly time consuming therefore sounding almost unrealistic for courses having large strength of the students. However smaller courses with strength upto 30 are mostly conducted completely by the professor without the involvement of the teaching assistant and it is kind of easy to spot the direct occurrences of plagiarism. The cherry on the cake is such courses tend to be complex enough that students either do by themselves or they directly copy it. But it has been found in the survey that when teaching assistants are employed (for large courses) then they are unable to spot plagiarism occurrences.

4-TUPLE APPROACH

One of the approaches to detect the candidates for serious investigation is to use this 4 tuple approach where we represent a pragram as a 4 tuple (n1,n2,N1,N2), where

- n1 : the number of unique operators
- n2 : the number of unique operands
- N1 : the total number of occurrences of operators
- N2 : the total number of occurrences of operands

The four-tuples of programs are then compared and if they match then they should be further investigated. The probability of matching the above parameters for two programs is really low unless they match to a significant extent. However this method is not well suitable for courses involving introductory assignments where the entire code length is fairly small.

ITAP : Instructional Tool for Program Advising

This system builds a graph to represent the structure of each student's program. It then compares pairs of programs by counting attributes of this representation.

A New System

Students in introductory courses tend to copy by using rather simplistic methods, some of them are:

1. Renaming Variables
2. Transposing statements when the ordering of the statements does not affect the result
3. Altering Format Statements
4. Breaking up single statements such as declarations and output statements.

This new system would be able to tag two programs similar if they match on one or more of the 4 ways listed above. By doing so this method is able to indicate likely cases of plagiarism.

This system is designed to accept the source code version of each student assignment from a file. Then for each assignment a table is created from the structure and content of the assignment. After creating a table for all the assignments, every table is compared with every other table and the results of these comparisons are printed for the instructor's use. This system supports the following languages

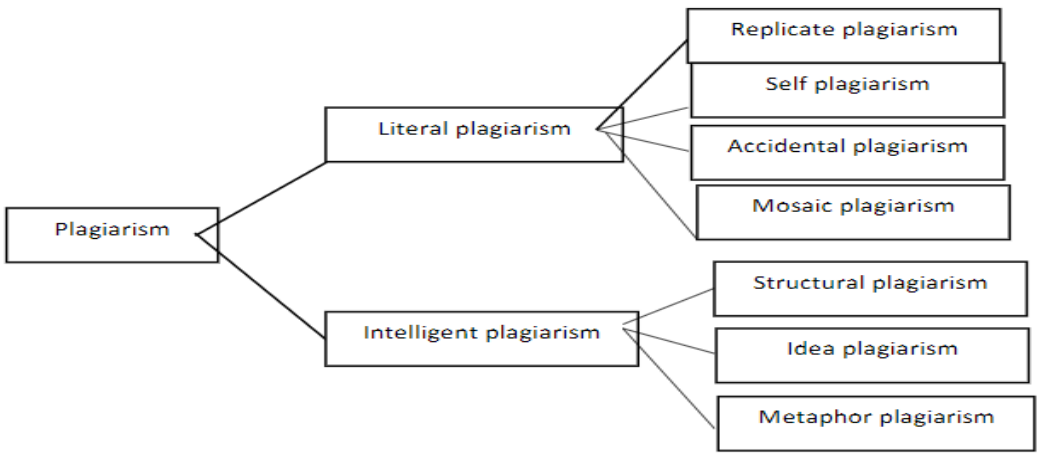
1. FORTRAN
2. COBOL
3. BASIC

DESCRIPTION

The working comprises of 2 phases

1. Data Collection : Every assignment is read and information is gathered on the characteristics of the assignment. Assignments are characterized in 2 ways generally
    a. The number of times certain types of statements occur is computed. We maintain a count of every distinct statement type and we store it in a two-dimensional array. This newly built array is supposed to be used in the next phase.
    b. The order in which statements occur in an assignment

2. Data Analysis : The characteristics collected in the previous phase are now compared and tables of results are built. It is accomplished in 2 part
    a. Calculating similarity of any two assignments
    b. Determining whether the structure of 2 assignments is similar or not



A COMPARATIVE STUDY OF DIFFERENT METHODS AVAILABLE

Table 1- A Comparative Study [15]

Author	Approach	Illustration	Limitation
Rajkumar Kundu, Kartik. K [16]	Latent Semantic Analysis LSA	Used to find out the semantic SVD and reduction to capture all similar text	A distributed model that is not suitable for nonlinear equations
Alireza Talebpour et al [17]	Plagiarism Detection based on Trie-tree based data structure	Both character- based and knowledge-based approaches are used for comparing data at high speed	A comparison based technique that requires processing of content which is not an efficient solution for large number of files
S.N. Autade et.al [18]	Evolutionary multi-agent system	System synonym recognition and word -generalization is used	A large word dictionary update is required
Jingling Zhao et. Al [19]	An AST-based Code Plagiarism Detection Algorithm	They proposed AST- CC algorithm to generate and compare hash values	Less efficiency for the storage of data structure
Mayank Agrawal et al [20]	A State of Art on Source Code Plagiarism	Various methods like NLP and machine learning	It is hard to locate the plagiarism among different source codes of different languages
Agung Sediyono et al [21]	Longest common consecutive word	A numerical based comparison algorithm that outsources suffix tree algorithm	The drawback of this proposed algorithm is loading time.
Michal Ďuračíka et. al [22]	Detection of clones and methods for determining similarity	Provides anti- plagiarism system which is to handle large size of dataset	Needs to process multiple documents with similar identity which exhibits high execution

CASE STUDY : CHECK

Most existing copy detection mechanisms employ an exhaustive sentence based comparison method in comparing a potential plagiarized document against all registered documents to identify plagiarism. But this is not really scalable and they are not so secure, it is kinda easy to bypass them in terms of security. So a newer approach involves use of information retrieval methods to store the semantic meaning of the sentences such that when a potential plagiarized document arrives we are able to comment on whether it is plagiarized or not by taking the semantic meaning in context.
The rationale which is deployed in practice is if the two docs are on different subjects then further comparisons between the docs could be eliminated. We use this rationale in a recursive way by applying it to individual organizational constructs of different granularities including sections, subsections and paragraphs. We use it until we find two paragraphs which are semantically very similar and finally these paragraphs are compared on a per sentence basis. By taking semantics into account we basically add another layer of security.

ARCHITECTURE

High level architecture of CHECK is as follows, it is composed of 3 modules

1. Document Registration
2. Document Comparison
3. Document Parsing

Module 1 is responsible for registering an original document to the database. And as the name suggests module 2 is the module which takes an input doc and compares it with registered documents. And module 3 helps in building an internal indexing structure called structural characteristic (SC) for each document to be used by the document registration and comparison modules. In this case, we also make use of the ORACLE database to store the SC of each document.

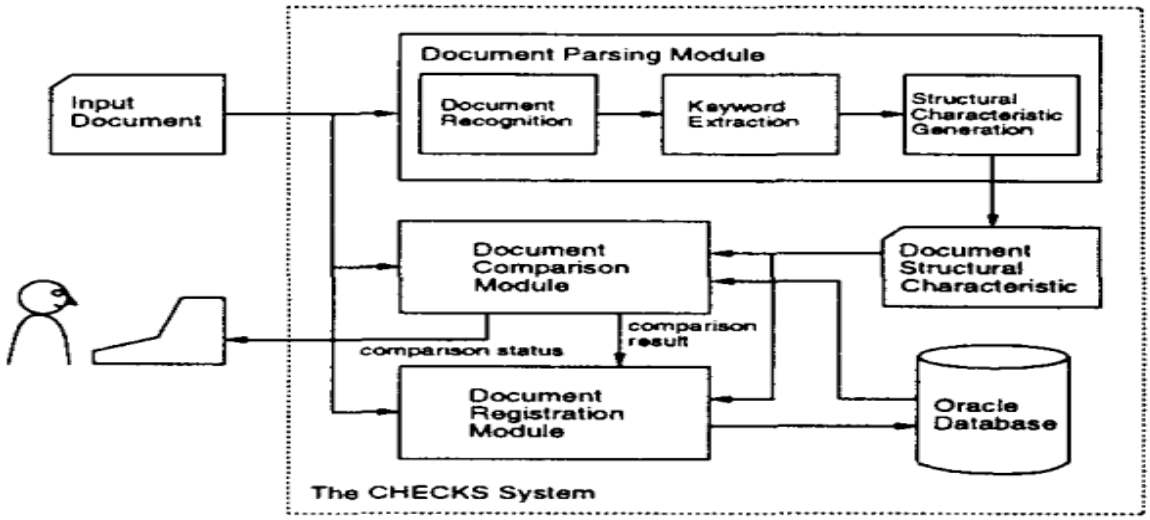


Figure 1: The architecture of CHECK.

PARSING

It involves the following sub-modules

1. Document recognition
2. Keyword Extraction : We use information retrieval techniques to identify keywords that would best describe the semantics of a document. In general, each word can be categorized into open-class or close-class. Open class includes nouns, verbs, adjectives, and adverbs that are usually meaning bearing, and are potential candidates for keyword extraction. Whereas close words include pronouns, prepositions, conjunctions and interjections are usually not meaning bearing and are not considered keywords.
3. Structural Characteristic Generation

DOCUMENT COMPARISON AND REGISTRATION

This module compares the SC of an input document, which is generated by the document parsing module, with the SC of each registered document in the database. Recall that the SC is simply a document tree with weighted keywords assigned to each node of the tree. The similarity between the set of keywords for a node of an SC and that of another SC indicates the degree of similarity in content between the modeled organizational constructs. An advantage of comparing the SCs instead of the documents is that unnecessary comparisons can be filtered at any level of SCs recursively, in a depth-first search manner.

The document registration module registers an input document to the system and stores both the document and its SC in the database for future comparison. A user can either input a document to the system and indicate that it is an original document. In this case, the system will simply call the module to register the document. Alternatively, a user can request the system to check the document before registering it. In this case, the system will call the document comparison module to check for the validity of the document. If no similarity is found between the input document and any of the existing registered documents, the system will call the document registration module to register the input document.

The comparison goes like this, consider two SCs of two documents A and B as shown below. In the diagram, the dashed arrows represent the path for a typical comparison process. The diagram illustrates that the root nodes of the two SCs are determined to be similar. The comparison process then proceeds to compare the children of the root nodes. The three nodes of SC_A are compared with the three nodes of SC_B in a pairwise manner. If the node modeling Section 2 of document A is determined to be similar to the node modeling section 3 of document B, the comparison will proceed to compare the children of Section 2 with the children of Section 3. If the node modeling Section 2.2 is determined to be similar to the nodes modeling Section 3.1 and Section 3.2, the children of Section 2.2 with the children of Section 3.1 and 3.2. This comparison process will continue to the leaf nodes or stop at some level if the similarity among the nodes at that level all falls below the similarity threshold.

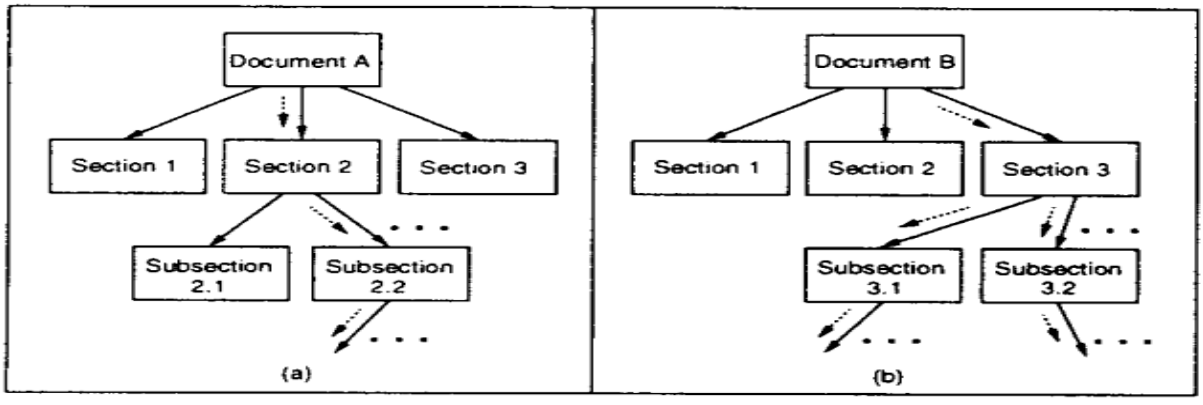


Figure 3: Document comparison process.

APPROACH

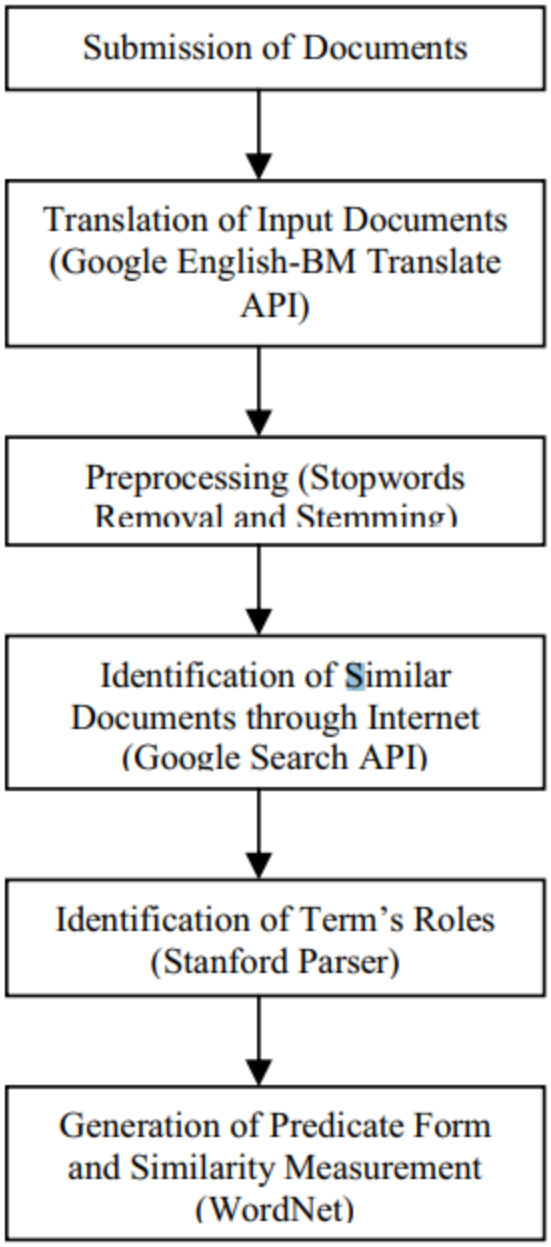
Based on above discussion and other papers which we read, we have found that there are several common techniques which are being implemented by different softwares like

1. Fingerprint Matching Technique
    a. Character-Based
    b. Phrase-Based
    c. Statement-Based
2. Clustering Techniques
    a. Hierarchical Clustering
    b. K-Means Clustering
3. Sylometry Measurement and Comparison Technique

All these techniques are used widely in different available online plagiarism checker tools but they are still not perfect and have some weaknesses and shortcomings in these techniques and tools which will affect the plagiarism detection significantly. During certain situations, the detection process will face limitations which are not capable of detecting the plagiarized documents. Issues which we face are:

1. Paraphrasing Issue : It is generally believed nowadays that fingerprint based approach is quite weak since even slight textual modification can considerably affect the fingerprint of the document. Most tools are not stable against synonyms, if someone copies and systematically changes words by using synonyms or such, all plagiarism tools we are aware of will fail. Therefore using fingerprint based approaches make it difficult to identify similar parts with similar semantic meaning.
2. Translated Plagiarism : Using the work of someone else by not citing it is plagiarism no matter if we change the language of it. It is still someone else's idea. Due to the regular availability of the web and the advancements in the NLP field, it is really easy now to avail translation services on our fingertips and so we find more occurrences of translated plagiarism. These incidents are harder to detect, as translation is often a fuzzy process that is hard to search for, and even harder to stop as they usually cross international borders. All present tools usually do not support stability against translated plagiarism.

So we present a new Approach as suggested in a paper and its architecture is attached below



So the components are

1. Translation : In accordance with the paper we find that two translated docs are difficult to suspect for plagiarism so we first need to find a common language so we intend to translate all input docs to, lets say, english.
2. Preprocessing : We then remove stop words from the text because those are the words which do not add any value to the text. Then we stem the words to get their stem words as dance and dancing are technically capturing the same semantic meaning so stemming is a necessary part and it should not be avoided.
3. Identification of Similar Documents : After preprocessing we inject the docs into the search engine and we get a set of results from the engine. These results

contain the similar documents and we then plan to run analysis over these documents.

4. Detail Comparison Analysis : Now we perform sentence-based similarity analysis between tokenized suspects and candidate documents. Sentences in suspected documents are compared with each sentence in the candidate documents. We aim to detect not only the arrangement similarity between sentences, but also possible semantic similarity between both sentences. Hence, we integrate the use of Stanford Parser and WordNet during the comparison process.

REFERENCES

https://dl.acm.org/doi/pdf/10.1145/1330598.1330642
https://dl.acm.org/doi/pdf/10.1145/800037.800955
https://dl.acm.org/doi/pdf/10.1145/331697.335176
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5701845
https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/3128/1601