Available ONLINE www.ijart.org

International Journal of Advanced Research In Technology

IJART, Vol. 1 Issue 1, 2011,64-70

ISSN NO: 6602 3127

RESEARCH ARTICLE

A Study of Plagiarism Detection Tools and Technologies

Sindhu.L, Bindu Baby Thomas, Sumam Mary Idicula
Department of Computer Science ,CUSAT, Cochin

ABSTRACT

This paper discusses in detail plagiarism and its detection .The increase of material now available in electronic form and improved access to this via the Internet is easily allowing plagiarism that is either intentional or unintentional. Due to increased availability of On-line material, people are finding checking for plagiarism very hard. Common techniques for detecting plagiarism are discussed in this paper. Plagiarism detectors are programs that compare documents with possible sources in order to identify similarity and so discover plagiarism

Keywords: ***Plagiarism, Plagiarism detection, corpus, syntax-based, semantic-based.***

1. INTRODUCTION

When the work of someone else is reproduced without acknowledging the source, this is known as plagiarism. The most frequent cases are in academic institutions where students copy material from books, journals, the internet, their peers etc. without citing references. there are many cases where students actually plagiarise unintentionally simply because they are not aware of how sources should be used within their own work. plagiarism is not just limited to written text, but also regularly found in software code where chunks are copied and re-used without reference to the original author. there are different plagiarism methods. some of them include [1]

- copy – paste plagiarism (copying word to word textual information);
- paraphrasing (restating same content in different words);
- translated plagiarism (content translation and use without reference to original work);
- artistic plagiarism (presenting same work using different media: text, images etc.);
- idea plagiarism (using similar ideas which are not common knowledge);
- code plagiarism (using program codes without permission or reference);
- no proper use of quotation marks (failing to identify exact parts of borrowed content);
- misinformation of references (adding reference to incorrect or non existing source).

Many methods to fight against plagiarism are developed and used. These methods can be divided into two classes [2]: (1) methods for plagiarism prevention, and (2) methods for plagiarism detection. Some examples of methods in each class are as follows: plagiarism prevention – honesty policies and/or punishment systems, and plagiarism detection – software tools to reveal plagiarism automatically. Each method has a set of attributes that determine its application. Two main attributes which are common to all methods are

1) work – intensity of method's implementation;
2) duration of method's efficiency.

Work – intensity of method's implementation means amount of resources (mainly time) which is needed to develop this method and bring into usage. Plagiarism prevention methods are usually time consuming in their realization, while plagiarism detection methods require less time. Duration of method's efficiency means the period of time in which positive effect of method's realization exists. Implementation of prevention methods gives a long-term positive effect. But, implementation of detection methods gives short term positive effect. To achieve momentary, short term positive results plagiarism detection methods must be applied at problem's initial stages, but to achieve positive results in long time period plagiarism prevention methods must be used. Plagiarism detection methods can only minimize plagiarism, but plagiarism prevention methods can to a great extent decrease it. That is why plagiarism prevention methods are more significant measures to fight against plagiarism. But, plagiarism prevention is a problem which can not be solved by efforts of one university or its department and so plagiarism detection methods and tool have been developed.

2. IDENTIFYING PLAGIARISM

A collection of submitted work is known as a corpus. Where the source and copy documents are both within a corpus this is known as intra-corpal plagiarism, or collusion. Where the copy is inside the corpus and the source outside, this is known as extra-corpal plagiarism. An important difference between source code plagiarism and free text plagiarism is that the methods used to detect them both differ. Source code plagiarism detection is easier to detect than free text plagiarism since the language that can be used is constrained to a set of defined key words and also intra-corpal in nature. Free text plagiarism contains an effectively unlimited number of possible words that can be used and plagiarism may be intra or extra-corpal.

3. IDENTIFYING PLAGIARISM IN WRITTEN TEXT

Factors that could be used to distinguish authors of written text and detect plagiarism include:

- Uses of vocabulary – analyzing the vocabulary used for an assignment against previous vocabulary could help determine whether a student had written the text.
- Changes of vocabulary – if the vocabulary used changes significantly within a single text, this can indicate a possible cut-and-paste plagiarism.
- Incoherent text – if the flow of a text is not consistent or smooth, this can could indicate the author has either not written with thought or consistency or that part of the text is not their own work.
- Punctuation – it is unlikely that two writers would use punctuation in exactly the same manner.
- Amount of similarity between texts – there will always be a certain amount of similarity between texts written about the same topic such as names, domain-specific terms etc. However, it is unlikely that independently written texts would share large amounts of the same or similar text.
- Common spelling mistakes –. It is very unlikely that independently written texts would have the same spelling mistakes, or same number of mistakes.
- Syntactic structure of the text –if two texts share exactly the same syntactic structure. It is likely that the most common syntactic

rules used by separate authors would be different.

- Long sequences of common text – it is unlikely that independently written texts would share long sequences of consecutive characters or words in common.
- Order of similarity between texts – if the order of matching words or phrases between two texts is the same in both texts this may indicate plagiarism. Although taught to present facts in a certain manner (e.g. introduction, body then conclusion), it is less likely that the same facts would be reported in the same order.
- Dependence on certain words and phrases – an author may prefer using particular words or phrases. Consistent use of these words and phrases in a text written by someone else with different word preferences may indicate plagiarism.
- Frequency of words – it is unlikely that words from two independent texts would be used with the same frequencies.
- Preference for the use of long/short sentences – authors may without knowing have a preferred sentence length that would be unusual combined with other features.
- Dangling references – if references appear in the text but not in the bibliography, this may indicate a cut-and-paste plagiarism where the author has not also copied the references.

Names, dates, locations, domain-specific terms, and common knowledge terms.

The taxonomy divides plagiarism into two typical types: literal plagiarism and intelligent plagiarism based on the plagiarist's behaviour[8,9].

There are, however, some words and phrases that are more likely to appear in common between texts written about the same topic, even if written independently. These may include:

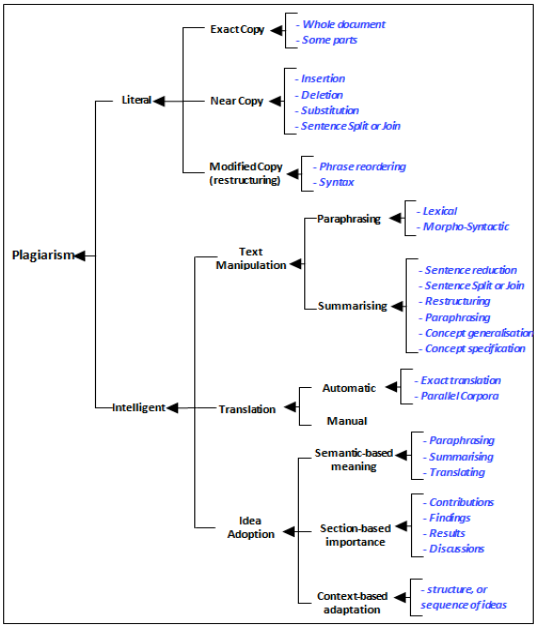

Fig. 1. Taxonomy of Plagiarism

4. PLAGIARISM DETECTION METHODS

With the growing popularity of the Internet, many and various documents are available free. People can easily search for the required documents and make their copy instead of writing the documents themselves. These practices have an enormous impact on the education system. . Document protection techniques, which disable copy paste operations and printing, are insufficient. A large database of existing documents is a better solution because every plagiarized document can be easily identified when compared to the database. Most of the plagiarists only copy a part of a document and do not try to hide this activity. This is an evident case of plagiarism that can be easily identified because a large continuous text is copied. The consistent plagiarists copy some parts of sentences and sometimes exchange several words to cause confusion. This type of plagiarism is difficult to determine

5. CLASSIFICATION OF PLAGIARISM DETECTION METHODS

Almost all current free text or source code copy detection systems are Paired or Singular. This means that every document must be compared with any other possible documents to analyze the whole corpus. Therefore, Paired and Singular methods are suitable for seeking some possibly plagiarized documents, which are related to the concrete tested document. The most general classification of copy detection methods for free text plagiarism detection methods is as in table I[3].

TABLE I
CLASSIFICATION OF FREE TEXT PLAGIARISM DETECTION METHODS

Type of classification		Description
Complexity of the method used	Superficial	The metrics is computed without any knowledge of the linguistic rules or a document structure.
	Structural	The metrics is computed with a partial understanding of documents, e.g. words are converted into their linguistic root, or replaced by a synonym.
Number of documents processed by the used method	Singular	A single document is processed to compute the metrics. Several Singular metrics can be employed to calculate how similar the documents are.
	Paired	Two documents are processed together to compute the metrics.
	Multidimensional	*N* documents from a corpus are processed together to compute the metrics.
	Corpal	All documents contained in a corpus are processed together to compute the metrics

The older systems, such as COPS or SCAM working on the term frequency, are purely Superficial. The current systems, which employ N-grams, are also rather Superficial than Structural. The reason is too time-consuming analysis of sentences whose grammar includes many linguistic rules.

A. Plagiarism Detection Systems

1) COPS (COpy Protection System): is a prototype of copy detection system developed at Stanford. The system is based on unit chunk hashing. A chunk is a sequence of consecutive units; a document may be divided into chunks in a number of ways, as chunks are allowed overlap or not cover the document entirely. A method of selecting chunks from a document is called a chunking strategy. A system following the COPS methodology consists of two main functions. One which obtains chunks from a document via a selected chunking strategy and stores hashes of these chunks into a hash table. The second function is a function that realizes the violation test[4].

2) SCAM (Stanford Copy Analysis Mechanism): is a plagiarism detection system developed at Stanford. Unlike COPS, it operates by assuming a vector space model for the registered documents. It uses a new similarity measure which was developed to accurately characterize copy overlap, while traditional IR systems look for semantic similarity . The SCAM system, as well as COPS, is classified as paired and superficial system[4].

3) MOSS (Measure of Software Similarity): was developed at UC Berkeley in 1994. It is a free available plagiarism detection system for academic usage only. MOSS supports a lot of different programming languages and two platforms, UNIX and Windows. Its primary purpose is to detect programming assignment plagiarism .Its aim is to detect the changing of variable names, I/O prompts, statement spacing and comments.. MOSS offers a script which, whenever run, emails a selected batch of programs to a Berkeley server for analysis. Response is usually obtained within the same day and consists of a set of html documents comprising a report. The report highlights pairs of programs that exhibit suspiciously high mutual similarity[4,5].

4) YAP (Yet Another Plague): token-based system that treats programs as a sequence of strings. The last version of YAP3 introduces a totally novel algorithm to face the presence of block-moves in programs. Namely: the Running-Karp-Rabin Greedy-String-Tiling algorithm. Its aim is to find a maximal set of common contiguous substrings as long as possible, each of which does not cover a token already used in some other substrings[4,6].

5) MDR (Match Detect Retrieval): is a prototype of a system capable of detecting overlapping documents. The basic matching components uses string-matching algorithm based on suffix trees to identify the overlap. The algorithm used for building the suffix tree from the query document is a modification of Ukkonen's algorithm. This system is only capable of locating exact copies of document parts. Once the suffix tree is

built, all registered documents are compared against it[4,7].

6) SID (Software Integrity Diagnosis, or Share Information Distance): is a system developed at University of California, Santa Barbara. Authors of SID consider the sequence similarity from an information theoretic perspective. The metric that measures the amount of information between two sequences is based on Kolmogorov complexity and is universal. The universality guarantees that if there is similarity under any computable similarity metric, this metric will detect it[4].

7) CHECK: is another plagiarism detection system that uses document structure to build a hierarchal representation of the document. Each document is viewed at multiple abstraction levels, which include the document itself, its section, subsection, and paragraphs. For each level, the set of relevant keywords is extracted. Keyword extraction uses keywords frequency as well as italics and boldface formatting information to assign weights to keywords. At query time, the nodes of the query abstraction and that of the referential document are traversed, starting with the root node. Similarity is computed as cosine measure of the two node's keyword weight vectors. If the similarity exceeds a given threshold, the two node's children are processed recursively. The purpose of this step is to obtain pairs of document segments (represented by the lowest level of abstraction, i.e. paragraphs) that are similar to each other. The final step is to analyze these similar pairs of paragraphs sentence-by-sentence and report detected copies[4] .

8) TURNITIN: is the most popular service of plagiarism detection. It was developed by group iParadigms for teachers and educational institutions and was formerly known as Plagiarism.org. The service works on a commercial basis and requires pre-registration. Professors and teachers present student's works on site and in a day or two receive the results. The system compares these materials to the indexed Web content, large databases containing texts, as well as previously reported materials.

B.Plagiarism Detection Techniques

Methods to compare, manipulate, and evaluate textual features in order to find plagiarism can be categorized into the following types: CNG, VEC, SYN, SEM, FUZZY, STRUC, STYLE, and CROSS[11].

1)CHARACTER-BASED METHODS (CNG): The majority of plagiarism detection algorithms rely on character-based lexical features, word-based lexical features and syntax features such as sentences, to compare the query document dq with each candidate document dx"Dx. Matching strings in this context can be exact or approximate. Exact string matching between two strings x and y means that they have exactly same characters in the same order. For example, the character 8-gram string x="aaabbbcc" is exactly the same as "aaabbbcc" but differ from y="aaabbbcd". Different plagiarism techniques featuring the text as character n-gram or word n-gram use exact string matching. approximate string matching shows, to some degree, two strings x and y are similar/dissimilar. For instance, the character 9-gram x="aaabbbccc" and y="aaabbbccd" are highly similar because all letters match except the last one.

2) VECTOR-BASED METHODS (VEC): Lexical and syntax features may be compared as vectors of terms/tokens rather than strings. The similarity can be computed using vector similarity coefficients. That is, word *n*-gram is represented as a vector of *n* terms/tokens, sentences and chunks are resented as either term vectors or character *n*-grams vectors, then the similarity can be evaluated using matching, Jaccard, Dice's, overlap (or containment), Cosine, Euclidean or Manhattan coefficients. Due to its simplicity, using Cosine with other similarity metrics was efficient for plagiarism detection in secured systems that submissions are considered confidential such as conferences.

3) SYNTAX-BASED METHODS (SYN): Some research works have used syntactical features for gauging text similarity and plagiarism detection. Recent studies have used POS tags features followed by other string similarity metrics in the analysis and calculation of similarity between texts. This is based on the fact that similar (exact copies) documents would have similar (exact) syntactical structure (sequence of POS Tags). The more POS tags are used, the more reliable features are produced for measuring similarity. That is, similar documents and in particular those that contain some exact or near-exact parts of other documents would contain similar syntactical structures.

4) SEMANTIC-BASED METHODS (SEM): A sentence can be treated as a group of words arranged in a particular order. Two sentences can be semantically the same but differ in their structure, e.g. using the active versus passive voice, or differ in their word choice. Semantic approaches have not been used in plagiarism detection which could be due to the difficulties of representing semantics, and the complexities of representative algorithms. Li et al. and Bao et al. used semantic features for similarity analysis and obfuscated plagiarism detection.A method to calculate the semantic similarity between short passages of sentence length is proposed based on the information extracted from a structured lexical database and corpus statistics. The similarity of two sentences is derived from word similarity and order similarity. The word vectors for two pairs of sentences are obtained by using unique terms in both sentences and their synonyms from WordNet besides

term weighting in the corpus. The order similarity defines that different word order may convey different meaning and should be counted into the total string similarity.

5)FUZZY-BASED METHODS (FUZZY): In fuzzy-based methods, implements a spectrum of similarity values that range from one (exactly matched) to zero (entirely different). The concept "fuzzy" in plagiarism detection can be modeled by considering that each word in a document is associated with a fuzzy set that contains words with same meaning, and there is a degree of similarity between words in a document and the fuzzy set. In a statement-based plagiarism detection, fuzzy approach was found to be effective because it can detect similar, yet not necessarily the same, statements based on the similarity degree between words in the statements and the fuzzy set.

6)STRUCTURAL-BASED METHODS (STRUC): All the above described methods use flat features representation. Flat feature representations use lexical, syntactic and semantic features of the text in the document but do not take into account contextual similarity based on the ways that words are used throughout the document, i.e., sections and paragraphs. Most document models incorporate only term frequency and do not include such contextual information. Tree-structured features representation is a rich data characterization, and multi-layer self-organizing maps model (ML-SOM) is very effective in handling such contextual information Chow et al. used block-specific tree-structured representation and utilized ML-SOM for plagiarism detection. The top layer performs document clustering and candidate retrieval, and the bottom layer detects similar, potentially plagiarized, paragraphs using Cosine similarity coefficient.

7)STYLOMETRIC-BASED METHODS (STYLE): Based on stylometric features, formulas can be constructed to quantify the characteristics of the writing style. Research on intrinsic plagiarism detection has focused on quantifying the trend (or complexity) of style that a document has. Style quantifying formulas can be classified according to their intention into writer-specific and reader-specific. Writer-specific formulas aim at quantifying the author's vocabulary richness and style complexity. Reader-specific formulas aim at grading the level that is required to understand a text.

VI.CONCLUSIONS

The problem of plagiarism, one of the most publicised forms of text reuse around us was discussed. Automatic plagiarism detection, the task of identifying quantifiable discriminators able to distinguish derived from non-derived texts was discussed. The use of automatic methods of detection aids the manual inspection of suspect texts by reducing the effort required in comparing large numbers of texts, and finding possible sources from on-line collections. Current anti-plagiarism tools can detect mainly only word-for word plagiarism and do not detect adopting ideas of others. Idea plagiarism is a problem that should be addressed in future research. SEM and FUZZY methods are proper to detect semantic-based meaning idea plagiarism at paragraph level;

REFERENCES

[1] Maurer, H., F. Kappe, B. Zaka. "Plagiarism – A Survey." Journal of Universal Computer Sciences, vol. 12, no. 8, pp. 1050 – 1084, 2006.

[2] Romans Lukashenko, Vita Graudina, Janis Grundspenkis, "Computer-Based Plagiarism Detection Methods and Tools: An Overview, ",in Proc. International Conference on Computer Systems and Technologies – *CompSysTech'07*

[3]Ceska Z, "The Future of Copy Detection Techniques" in Proc. The 1st Young Researchers Conference on Applied Sciences, Pilsen, Czech Republic,
13 November, 2007. – Pilsen: University of West Bohemi, 2007. –P. 5-10.

[4] Rehurek R. ,"Semantic-based plagiarism detection": PhD Thesis Proposal
– 2007, P. 6-13.

[5] Clough P.," Plagiarism in natural and programming languages: an overview of current tools and technologies " in The 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico, 13-17 March, 2005. – New York: ACM, 2005. – P. 776-781

[6] Jadalla A., Elnagar A. "PDE4Java: Plagiarism Detection Engine for Java source code: a clustering approach" in International Journal of Business Intelligence and Data Mining – Vol. 3, No. 2 (2008), P. 121-135.

[7] Sorokina D., Gehrke J., Simeon W., Ginsparg P.," Plagiarism Detection in arXiv "in The Sixth International Conference on Data Mining, Hong Kong, Japan, 18-22 December, 2006. – Washington: IEEE Computer Society, 2006. – P. 1070-1075.

[8] G. Stefan and N. Stuart, "Tool support for plagiarism detection in text documents," in *Proc. ACM Symposium Applied Computing*, Santa Fe, New Mexico, 2005, pp. 776-781.

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

[10] Turnitin: Plagiarism Checker to Ensure Academic Integrity. San Francisco: iParadigsm, 1998. [Online].
Available:http://www.turnitin.com/static/index.html

[11] Salha Alzahrani, Naomie Salim, and Ajith Abraham, "Understanding Plagiarism Linguistic Patterns, Textual Features and Detection Methods " in IEEE Transactions on systems , man and cybernetics-Part C: Applications and reviews, vol xx, NO. xx, MMM 2011

[12] Baker B.S. On finding duplication and near-duplication in large software systems // The Second Working Conference on Reverse Engineering, Toronto, Canada, 14-16 July, 1995. – Washington: IEEE Computer Society, 1995. – P. 86

.[13] Clough P. ,"Plagiarism in natural and programming languages: an overview of current tools and technologies " in The 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico, 13-17 March, 2005. – New York: ACM, 2005. – P. 776-781.

[14] EVE: Plagiarism Detection System. USA, 2000. [Online].Available: http://www.canexus.com/eve/index.shtml.

[15] Jadalla A., Elnagar A. PDE4Java: Plagiarism Detection Engine for Java source code: a clustering approach // International Journal of Business Intelligence and Data Mining – Vol. 3, No. 2 (2008), P. 121-135.

[16] Karp R.M., Rabin M.O.," Efficient randomized pattern matching algorithms" in / IBM Journal of Research and Development – Vol. 31, No. 2 (1987), P. 249-260.

[17] Plagiarism detection free detectors at wordchecksystems.com. [Online]. Available:http://www.wordchecksystems.com.

[18] Prechelt L., Malpohl G., Phlippsen M. Finding Plagiarisms among a Set of Programs with JPlag // Journal of Universal Computer Science – Vol. 8, No. 11 (2002), P. 1016-1038.

[19] Rehurek R. Semantic-based plagiarism detection: PhD Thesis Proposal – 2007, P. 6-13.

[20] The plagiarism resource site. Charlottesville: Lou Bloomfield, 1997. [Online]. Available: http://plagiarism.phys.virginia.edu/Wsoftware.html

[21] Wise M.J. String similarity via greedy string tiling and running Karp- Rabin matching: Technical report No. 463 – The University of Sydney, March, 1993. – P. 3-8.

[22] Z. Ceska, M. Toman, and K. Jezek, "Multilingual plagiarism detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. vol. 5253 LNAI, 2008, pp. 83-92.

[23] H. Zhang and T. W. S. Chow, "A coarse-to-fine framework to efficiently thwart plagiarism," *Pattern Recognition*, vol. 44, pp. 471-487, 2011.

[24] M. zu Eissen, B. Stein, and M. Kulig, "Plagiarism Detection Without Reference ollections," in *Advances in Data Analysis*, 2007, pp. 359-366..– 1034, 2007.