

**TITLE: A METHOD AND A SOFTWARE IMPLEMENTED TOOL FOR
DETECTING PLAGIARISM IN DOCUMENTS**

FIELD OF THE INVENTION

The present invention relates to a method for detecting plagiarism, in particular the present invention relates to a software-implemented method for detecting plagiarism. The invention also relates to a software implemented tool for detecting plagiarism.

BACKGROUND AND THE PRIOR ART

Plagiarism is the close imitation of the language and thoughts of an individual and the representation of them as one's own original work. With the advent of internet and the widespread use of online resources among the common masses, plagiarism in various educational research domains and publishing industry is a growing menace. Electronic versions of various books, tutorials, research papers and patent documents found on the internet serve as readily available resources for a plagiarist. Such a trend of imitation of original work has escalated with time especially in recent years, thereby depriving a deserving person of the economic and moral rights bestowed by the law of copyright (Detecting Duplicate and Near Duplicate Files as described in US 6, 658,423 B1, Dec 2, 2003 which is discussed hereinbelow in details) and also acting as a hindrance in promotion of research. Till date, no effective tool or method has been designed to deal with this problem of plagiarism, and the so-called practice of 'cut' or 'copy' and 'paste' is going ahead unabated. For instance, many organizations may initially be convinced that their work force is doing lot of copyrightable creative work, but the same may prove to become a big liability later, for which the organizations may be forced to compensate the original creators by way of damages, penalties, court fees, imprisonments and loss of reputation.

Thus, there has always been a need for developing an effective means for detecting plagiarism. Many prior art pertain to the same need but none proved effective enough to detect textual as

well as semantic plagiarism, i.e. none have a holistic approach. The prior art pertaining to the same problem has been discussed below in the following paragraphs.

The earlier known systems for plagiarism detection were based on feature vectors. Ottenstein . (Method for detecting Plagiarism as described in US 6, 976, 170, B1, Dec 13, 2005 which is discussed hereinbelow in details). Further -“an algorithmic approach to the detection and prevention of plagiarism” SIGCSE Bulletin, Vol. 8, no. 4, pp. 30-41, 1976, came up with such a system . Later, some more system were developed but their performance was not so remarkable. Another system developed by Savil Sehleimer, Daniel S. Wilkinson and Alex Aiken (Local Algorithms for Document Fingerprinting- Recognizer of text based work as described in US 7,356,188 B2, Apr 08,2008 which is discussed hereinbelow in details). Another system reported was named MOSS (Measure of Software Similarity). Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data used for detecting plagiarism in computer software codes which uses Rabin-Karp Algorithm with Windowing. However, it is not of much use for detecting content plagiarism, as it fails to detect the semantics associated with the document. The system developed was based on selecting fingerprints of k-grams. A k-gram is a contiguous sub-string of length k. Document is divided into k-grams, where k is a parameter chosen by the user. Each k-gram is hashed and some subset of these hashes is selected to be the document's fingerprints.

There are some online plagiarism detection tools and plagiarism reporting resources [5] readily available on the internet like MyDropBox, Turnitin, Wcopyfind, www.plagiarismchecker.com and www.plagiarismdetect.com. However all of these tools fail to compare vast resources on internet, inclusive of synonyms and thus semantics, and for providing side by side columnar display of target document vis-a-vis the resources from which they are copied and to what extent.

Most of the well-known Plagiarism Detection Systems use Running Karp-Rabin Matching and Greedy String Tiling (RKR-GST) [6] Pattern Matching Algorithms – IBM Journal of research and Development March 1987 or just counting frequencies of words between two documents. The algorithm was first made for comparing amino acid bio-sequences. The RKR-GST algorithm is frequently used in various plagiarism detection systems available today. It works by finding the longest possible string common in both of the documents.

There are other methods, which don't rely on fingerprinting methods like SCAM [7- A Real Life Instance of Plagiarism Detection by SCAM- Retrieved April 28, 2004] from Stanford University <http://www-db.Stanford.edu/NEShiva/SCAM/plag.html> , a copy-detection system. It says that if two documents are similar, the distance between feature vectors representing the two documents is small. The features are defined to be words here, and the notion of distance is a measure of variation on standard information-retrieval algorithms.

However, both of these algorithms may not form an efficient approach. The biggest disadvantage is that they are based on string matching and not comparing the semantics of the text. It only matches two texts that have similar strings and not two texts that have similar meaning using different words. So, if k-grams is being used to develop a plagiarism detection system, it can only detect plagiarism in documents that are exactly copied as texts from some source or are just slightly modified. Even if a person copies some text and replaces most of the words using their synonyms, the k-grams approach will fail to detect the plagiarism. However, the present system handles all such possibilities successfully.

Many software implemented methods have also been patented for some related problems. The patent documents have been discussed below.

United States Patent 6,658,423, which relates to Detecting Duplicate and Near Duplicate Files pertains to information management and retrieval in general. More specifically, it concerns detecting, and optionally removing, duplicate and near-duplicate information or content, such as in a repository of documents to be searched for example. the problem solved by this prior art relates to detection of redundant, duplicate information provided by a search engine on feeding a search query. According to this patent, from a user's perspective, duplicate and near-duplicate documents raise problems.

While conducting a search on any search engine on the internet the user submits a query. The search engine responds to such a query by providing results. Many of these search results, contain redundant, duplicate/repetitions of information.

The patent besides eliminating duplicate or near-duplicate documents to meet user expectations and wishes, eliminating duplicate or near-duplicate documents is desirable to search engine hosting entities to (i) reduce storage requirements (e.g., for the index and data structures derived from the index), and (ii) reduce resources needed to process indexes, queries, etc.

It detects near-duplicate documents for each document, generating fingerprints, determining near-duplicate documents based on the fingerprints, and determining near-duplicate documents based on the (remaining) fingerprints.

The act of generating fingerprints for each document may be affected by (i) extracting parts (e.g., words) from the documents, (ii) hashing each of the extracted parts to determine which of a predetermined number of lists is to be populated with a given part, and (iii) for each of the lists, generating a fingerprint. This patent fails to detect plagiarism if some document is copied only 30% of content from one source. For instance when a document has been copied from many different sources majority of which are of three lines / three sentences or more, this patent will fail miserably to detect plagiarism.

United States Patent 6,976,170, which relates to Method for detecting Plagiarism tries to meet a need for a method to determine whether a document submitted as an individual's own work is actually that individual's work, or whether it has been substantially copied from another source. Such a method claimed to be reliable, is not totally dependent on machines and provides for the possibility of a human making the final decision. The scope of resources checked is also limited herein.

Numerical values are assigned by loading the document in a computer, and assigning the numerical values according to a simple formula in a computer program. The computer program can then automatically check the numerical values and note numerical values from the submitted document that are equal to, or within a preset range of, the numerical values in the archival documents. A reviewer then reviews the document if there are too many matches as determined by the reviewer based on his or her experience with the type of paper or document and assignment being reviewed.

Assigning numerical value to each sentence usually makes the program quite slow. Also, it only does a literal verification of the concerned documents and not a semantic one thereby failing to detect if some minor modifications viz. synonyms have been made by the plagiarist. Essentially this patent is suitable only for detecting cheating within a specified domain such as by students in a school assignment works.

United States Patent 7,356,188, which relates to Recognizer of text based work is directed to a computer programme implemented method which helps in recognizing contents of a text document according to the following steps :

Obtaining a body of text containing textual content in a computer-readable format, removal of superfluous elements,

Formatting the filtered subtext into a defined image-based format, wherein the textual content of the defined image-based formatted filtered subtext is immutable via software tools for manipulation of the textual content of bodies of text;

Deriving a hash value representative of the textual content of the filtered subtext,.

The method indicates suspicion of *plagiarism* between the two filtered subtexts when the compared hash values of the two image-based formatted, filtered subtexts substantially match.

United States Patent 7,139,756 which relates to a System and Method for detecting duplicate and similar documents and it satisfies the need to provide an accurate and efficient algorithm and system for determining a degree of likeness between electronically represented documents.

It further describes a method for processing data representing documents (both individual or group of documents), comprising:

Executing a software program to obtain a list of terms found in each document;

Comparing the list of terms for a first document to the list of terms for a second document;

Declaring the first document to be substantially identical to, or substantially similar to, the second document if some predetermined number of terms(90% of the salient terms) are found in each of the lists of the first document and the second document; and

The step of comparing includes a preliminary step of sorting the documents into a document list in order of increasing size, and where the step of comparing comprises of comparing the list of terms for a given document with the list of terms for the next larger-documents in the document list.

Both these systems fail to detect plagiarism when a particular document has been copied in parts from various others.

Thus, all the methods discussed above deal with the problem of detecting plagiarism but lag behind with respect to considering parts of documents and detecting on basis of semantics thereby not being a foolproof method for detection.

To meet the need of an effective tool for detecting plagiarism, a software implemented method has been developed which detects plagiarism by comparing a document referred hereinafter as the 'test' document, with existing counterparts on the internet resources and also from documents already present in any soft repository. Besides, it can also pinpoint the source from where the content has been copied. In addition, for both the target (i.e. the test document) content as well as the source where from it is copied, the percentage matches are reported along with the copied contents thereof displayed side by side in a columnar form to enable anyone compare and see what has been copied from where and up to what extent. The tool is of immense importance to enable the owner retain his/her copyright on the work else the same is likely in due course of time automatically land into public domain. Any new organization can also benefit from it to screen out and release for outreach only those documents/research papers/project reports/ thesis/ conference proceedings, which are free from plagiarism and therefore can be publicized for name, fame, honest royalty fee etc.

OBJECTS OF THE INVENTION

Thus an object of the present invention is to meet the above mentioned needs and provide a method for the detection of plagiarism.

Another object of the present invention is to provide a tool implementing said method, to be used by institutes of higher education and research, project works, research papers, thesis, conference proceedings, industrial enterprises including the publishing industry, and government departments to determine whether a document is plagiarized or not.

Yet another object of the present invention is to provide a tool implementing said method, to be used by institutes of higher education and research, project works, research papers, thesis, conference proceedings, industrial enterprises including the publishing industry, and government departments to maintain their copyrighted works.

Yet another object of the present invention is to provide a tool implementing said method, which takes into consideration both the textual and semantic similarity of works.

Yet another object of the present invention is to provide a tool implementing said method, whereby the results of similarity will be displayed in columnar format along with percentage of similarity.

SUMMARY OF THE INVENTION

According to one aspect of the present invention there is provided a method for detecting plagiarism in a given 'test' document, said method comprises the steps of:

generating search strings pertaining to a given document, namely, the 'test' document;
creating a local repository of the search results including the potential target documents of the source document;
uploading said test document in a local database
comparing semantically each potential source document with the test document;

generating a report in a columnar format displaying the plagiarized passages of the source and the target document along with the percent similarity;

wherein the comparison of the source and the target document is carried out by a sliding window technique and wherein prior to comparing the target and the source documents, said documents are converted from a text format to a hashed format.

According to another aspect of the present invention there is provided a tool using the said method for detecting plagiarism in a given 'test' document, said tool comprises :

means for generating search strings pertaining to a given document, namely, the 'test' document;

means for creating a local repository of the search results including the potential target documents of the source document;

means for comparing semantically each potential target document with the source document;

means for generating a report in a columnar format displaying the plagiarized passages of the source and the target document along with the percent similarity;

wherein the comparison of the source and the target document is carried out by a sliding window technique and wherein prior to comparing the target and the source documents, said documents are converted from a text format to a hashed format.

The other objects and advantages of the present invention will be apparent from the description provided hereinbelow with reference to the accompanying figures and detailed description provided herein below.

BRIEF DESCRIPTION OF THE ACCOMPANYING FLOWCHARTS

Figure 1 illustrates the overall procedure followed in the system.

Figure 2 illustrates the Process Flowchart for execution of the System of online searching on the internet for creating a local repository of similar documents.

Figure 3 illustrates the process of conversion of text document into hashed document.

Figure 3A exemplifies the processing of a piece of text.

Figure 4 illustrates the sliding window technique used for comparing the texts.

Figures 5 and 5A illustrate the process of comparison of the corresponding source and the target windows.

Figure 6 illustrates the process of semantic analysis.

Figure 7 illustrates the output of the method, i.e., the result being displayed in a columnar format along with the percentage similarity of the target and the source document.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

The present invention provides a novel method for detecting plagiarism. Unlike the prior methods described above, present method has a holistic approach in dealing with the problems i.e. it takes into consideration both the textual and the semantic similarity of a document.

The working of the present invention can be described by way of the flowcharts mentioned above. An overall view of the entire working of the system is presented in Figure 1. The document to be checked for plagiarism in text format is selected by the user and uploaded in interfacing unit like GUI (Graphics User Interface), in a computer connected to a network or group of networks or group of computers connected in intranet or internet.

On completion of the selection of the target document, the user follows the below mentioned steps:

User uploads the test document onto the GUI and selects desired settings.

The uploaded test document is stored in a local database.

The user is given the option of either searching the local repository or the Internet itself.

On choosing the Internet option, the user is further given the option of searching a few specific websites or the Internet as a whole.

On selecting some particular search domain, an input form is generated by the software program on the screen and the URLs of those domains are then accepted as input from the user.

If the user selects to search the whole of the Internet, user is further asked if he wants to block some particular URLs during search.

If the user selects to block/skip some particular URLs during search, an input form is generated on the screen and the URLs of those sites are accepted as input from the user.

After entering all required settings, intelligent and important queries are generated from the document.

The queries generated above are searched on the internet/local repository and URLs of pages from the search results are stored in the database. Closest results to the test document are saved and these subsequently act as the documents with which the system will compare for plagiarism.

Using the URLs stored above, the system further downloads the web pages from the internet with the hope that these documents may be similar in the context to the document being checked for plagiarism.'

After completing the above steps, the system compares the given document with all the documents downloaded from the Internet for detecting textual as well as content plagiarism. The result is displayed as a side by side columnar pattern with respect to the target and the source documents. The result shows the part of the document plagiarized along with the source URL from where it has been copied. When the local repository is chosen by the user for comparing the documents, this step occurs right after the step of choosing the local repository option.

The entire method described above is fully automated as it is software based and the user only has to select the test document to be checked for plagiarism. Herein also, an user friendly "Graphic User Interface (GUI)" is provided.

Steps 3 to 9 of flowchart 1 has been dealt in detail in the subsequent flowchart. Flowchart 2 elucidates the internal processing of the system while downloading relevant webpages from the

internet. It describes the process of a query being searched on the internet using any Search Engine's Application Programming Interface (API). URLs are extracted from the top search results. The extracted URLs are stored in a database. URLs stored in the database are read one by one and the corresponding document is downloaded from the internet and stored in a local repository. Flowchart 3 elucidates the process of hashing wherein the text document is converted to a hashed format, i.e. it is assigned hashed values. Starting with the noise (is, are, am, will, shall etc, all maintaining in a noise database) stemming is done wherein stop words like adjectives, phrases, derivatives and punctuation marks are removed from the 'test' document and the remaining document is converted into its base form or first form as shown in flowchart 3A. Converting into base form or first form allows detecting plagiarism even after change in tense or voice of the original document during copying. After converting into base form each word is given a hash value. The same hash function is used here which was used in the dictionary above. This hash function generates a hash for each word as shown in figure 3. The hash value is an integer with a maximum value of 4×10^7 . For the process of comparing 2 documents for similarity, a preprocessed dictionary and thesaurus is used. Each word in this dictionary is stored in hashed form instead of a string as shown in flowchart 5, the synonyms are also given hashed values which are considered to be equivalent to the base form. This hash value for each word is an integer with a maximum value of 4×10^7 . Then each word in the document is replaced by a corresponding hash value and the document is merely a collection of integers. This gives the hashed document. Flowchart 3A exemplifies the process of hashing as discussed above.

After the conversion into the hashed format, the test and the source documents are compared by the sliding window technique which ensures efficient comparison of the complete document. This particular technique known as the sliding window technique is one of the unique features of this invention which makes the present invention technically advanced over the prior art. Flowchart 4 elucidates the sliding window technique in which a document is divided into windows of fixed size. A particular window of the document shares 50% portion of the preceding window. The overlap region as shown in the flowchart pertains to the 50% share portion. Thus, there is always one overlapping window between two successive windows.

Figures 5 and 5A describe how two corresponding windows from the source and target documents are compared.

Figure 6 illustrates the method used in comparing the documents processed above. After processing the documents, each of the 2 documents is a collection of integers. Sliding windows are used for the 2 documents to be compared with an overlap of 50% as shown in figure 4. An array is used to mark the number of occurrences of each word in the window of document A. If a synonym of a word occurs, then the base form of the word is incremented. The same procedure is followed for the corresponding window of document B. For each word occurred in left window, if it occurs in right window too, then the minimum count is considered. Total count is added and if this total crosses a certain threshold, the paragraph is considered plagiarized.

An Example:

Consider the following two pieces of texts:

1) ***Antique** plays like those of Shakespeare are **really** good to watch and read.*

***Many** books have also been **published** on those and many movies have been made.*

2) ***Ancient** dramas are **truly** good for reading and viewing. Lots of books were **printed** and **pictures** were released on them.*

In the two texts above “**antique**” and “**ancient**” are synonyms. So the synonym detection technique finds that both are the same words. “**Plays**” and “**dramas**” are synonyms of each other and the system detects it. Accordingly, “**really**” and “**truly**” are also synonyms and are caught by the present system. Now “**watch**” and **viewing** may not be synonyms, but **watch** and **view** are almost the same activity. The present system converts “**viewing**” into its basic form, i.e. “**view**” and finds it to be same as “**watch**”. “**read**” and “**reading**” are the same word but different forms and the present system catches them easily since it converts reading into its basic form – “**read**”. Again, “**many**” and “**lots**” are also synonyms. “**Books**” is the only word that has been exactly copied here and so a basic comparison approach can catch this also but one word never proves plagiarism. “**printed**” is another form of “**published**” and the present system detects it. Same is true with “**movies**” and “**pictures**”.

After the submitted document is compared with all downloaded documents, or with documents in local repository, the plagiarism result is shown in a columnar display as shown in figure 7. The left column shows the plagiarized paragraph of submitted document and the right column shows the paragraph of the source document from which the text was copied. The URL of the source document is also displayed. At the bottom of the result page, percentage plagiarism is displayed.

The Method/Procedure for Detecting Textual and Content Plagiarisms in the Thesis/Research Papers/Patents etc is as below of a typical institution -

- Remove various formal sections like declaration, candidates undertaking, guides statement and references section from the document.
- Generate website plagiarism report as well as patent plagiarism report and report of plagiarism from e-books, INDEST databases (research papers) etc using the procedure outlined earlier.
- Edit the word document to contain only those columns where three lines plus of a resource material has been copied and it is relevant to the candidate document.
- For all above cases, examine whether figures are also copied. If yes, include them in the plagiarism report.
- Issue plagiarism report in side-by-side printed form with recommendations of the organizations in case an institution of higher education and research, its plagiarism policies should be followed. In its absence, one may directly follow for example IEEE guidelines [8].

- In case percentage of plagiarism is below 20% examine the references section closely. The latest references relevant to the problem solved should be examined and if it is superficial, identify the real references based on which the work has been done.
- Take complete print out of the real references. Based on clues given by the plagiarism report, identify all plagiarized portions. Check for consistency of all sections. If not found, there is likelihood that the body of the research paper is shoddy. Now examine closely the methods section of the research paper in light of the conclusions arrived at and indicate their inconsistencies. Check whether portions copied are within the provisions of “FAIR USE” [1] clause of the Copyright Act/Copyright Rules of the country, for example allow items of similarity if limited to 3 lines/3 sentences whichever is less if it is properly credited through references else follow plagiarism policy of concerned organization who is to own the document(s).

CLAIMS

1. A method for detecting plagiarism in a given 'test' document, said method comprises the steps of:
generating search strings pertaining to a given document, namely, the 'test' document;
creating a local repository of the search results including the potential target documents of the source document;
uploading said test document in a local database
comparing each potential source document with the test document;
generating a report in a columnar format displaying the plagiarized passages of the source and the target document along with the percent similarity;
wherein the comparison of the source and the target document is carried out by a sliding window technique and wherein prior to comparing the target and the source documents, said documents are converted from a text format to a hashed format.
2. The method as claimed in claim 1 wherein said step of creating local repository comprises
searching the internet directly;
extracting URLs of the closest search results;
downloading the corresponding documents and
storing said documents.
3. The method as claimed in claim 2 wherein said extracted URLs are stored in a database.
4. The method as claimed in claim 1 wherein text documents are given hash value by the process of hashing.
5. The method as claimed in claim 1 wherein the said document is divided into windows of fixed size so that two successive windows in the document have an overlap region.
6. The method as claimed in claim 5 wherein said overlap region comprises at least 50% of shared portions of the documents.

7. The method as claimed in claim 1 wherein said report generated having columnar format display in the left column shows the portions of the test document along with line numbers which have been plagiarized and the right column displays the source from which the corresponding text in the left column has been copied.
8. The method as claimed in claim 7 wherein the right column further comprises the URLs of the source documents.
9. The method as claimed in claim 1 wherein prior to step of comparison words of the text documents are reduced to their base form to make the comparison exhaustive.
10. The method as claimed in claim 1 wherein all synonyms of the words are considered and treated same as the base form.
11. A tool for detecting plagiarism comprises
means for generating search strings pertaining to a given document, namely, the 'test' document;
means for creating a local repository of the search results including the potential target documents of the source document;
means for comparing each potential target document with the source document;
means for generating a report in a columnar format displaying the plagiarized passages of the source and the target document along with the percent similarity;
wherein the comparison of the source and the target document is carried out by a sliding window technique and wherein prior to comparing the target and the source documents, said documents are converted from a text format to a hashed format.