ARCHITECTURAL DESIGNING AND ANALYSIS OF NATURAL LANGUAGE PLAGIARISM DETECTION MECHANISM

[1]Amandeep Dhir , [1]Gaurav Arora, [2]Anuj Arora
[1]Software Engineer, Newgen Software Technologies, New Delhi, India
[2]Student, Information Technology Engineering, India
E-mail: amannewgen@gmail.com

ABSTRACT

We proposed an Architectural model for detecting plagiarism in natural language text and presented the analysis of various detection processes followed for effective plagiarism detection. Other plagiarism detection mechanisms are based on parsing techniques where sentence and word chunking is performed to extract phrases which are searched on internet in comparison to that we performed sentence and work chunking but our method is more detailed and comprehensive and further bifurcated these two techniques on the basis of the rearrangement of the text or word pattern. We tried to put emphasis on principle behind performance of any search result and developed an efficient plagiarism detection mechanism on the basis of the unitization process which are tested for various input data and has shown considerable output. Our purposed Unitization processes are different from the already available. Challenges faced during the plagiarism detection are discussed with their purposed solutions. Results obtained after experiment reveals empirical study of the performance factor

Keywords: *Chunking, parsing, plagiarism, unitization*

1. INTRODUCTION

Plagiarism is a very common phenomenon now days and it is center of discussion at various educational events. Plagiarism is defined as the practice of claiming or implying original authorship of (or incorporating material from) someone else's written or creative work, in whole or in part, into one's own without adequate acknowledgement. E-learning programs has touched every corner of this small world due to the advanced in the Information Technology , this in turn led to easier availability of the research papers , books , technical and non technical papers etc which are easiest source for making plagiarized documents. Copying and pasting of paragraphs or even entire essays now can be performed with just a few mouse clicks. Researcher are focusing on inventing newer ways for secure information flow so as the confidential information can be transformed freely so as e learning programs not prove harmful for the books publishers and arise cases of copy rights violations. This seems to be a temporary solution for this problem because eventually we are trying to control the possible cases of plagiarism. Most of the research in this field has till now concentrated in the type of text extraction which in turn can be sentence or word chunking, no one has yet bi-furcated the extraction into sub parts such as rearrangement to explore the performance issues. Current systems or tools are effective on the cost of time as if you need more performance system will work slowly even they give effective output till a threshold

level based on the document size .This paper focuses on the architectural details of plagiarism detection mechanism along with the type of the functionality we choose for higher performance of the system.

Natural language processing is a field related to the artificial intelligence, this is of great importance as detecting plagiarized phrases in the given document becomes easier after the semantic analysis of the document. Various Natural language principles make it easier for carrying out the text analysis and for parser generation in the given document. Plagiarism is defined as the practice of claiming or implying original authorship of (or incorporating material from) someone else's written or creative work, in whole or in part, into one's own without adequate acknowledgement.

We have considered the problem of plagiarism which is one of most publicized form of text reuse around us. The ultimate goal of this research is to devise an automatic plagiarism detection which can distinguish among the derived and non derived texts. Most techniques have concentrated on finding unlikely structural patterns or vocabulary overlap between texts, finding texts from large collections and collaboration between texts. Some plagiarists are very clever and generally while copying data they make complex editing so that even the sophisticated methods of analysis are unable to detect plagiarism. Various procedures for text analysis are semantics, parsing, structure or discourse, morphological analysis. The most dominant methods which can enhance research in plagiarism detection are lexical overlap, syntax, semantics and structural features.

In academia, recent interest has shifted towards identifying plagiarism between natural language texts. Particular areas of concern includes identifying verbatim cut-and-paste having minor or major changes from Web-based sources and identifying the same content but paraphrased. This is reflected by the increase in the online services such as turn tin and plagiarism.org. Services to track and monitor commercial content have also received increased popularity as the media report more cases of stolen digital content (e.g. contentguard.com). Few researchers reveal distinctive methods such as paraphrased comments and misspelled identifier in detection plagiarism easily. The process of automatic plagiarism detection involves finding quantifiable discriminators which can be used to measure the similarity between texts. The complete design process has taken many set of assumptions which includes the text which is assumed as source text is plagiarism free or original text, greater similarity means greater are the chances of possible plagiarism. So far the focus has been on lexical and structural similarity in both program code and natural language, but these become less effective when the degree of rewriting or form of plagiarism becomes more complex. The process of automatic plagiarism includes developing suitable methods to compare those discriminators, Finding suitable measures of similarity and Finding suitable discriminators of plagiarism which can be quantified.

The goal of an automatic plagiarism detection system is to assist manual detection by: reducing the amount of time spent comparing texts, making comparison between large numbers of multiple texts feasible and finding possible source texts from electronic resources available to the system. The systems must minimize the number of incorrectly classed as plagiarized and those incorrectly classed as non-plagiarized and maximize the number of true positives.

2. CONTRIBUTION OF THE PAPER

Our approach to plagiarism detection is through the knowledge of key challenges faced during detection, Present plagiarism detecting tool in the market, Experimentation and verification of the techniques and selecting the best out of the available options. Plagiarism Detection Mechanisms can broadly classified as the Web based and Desktop based. The first category classifies the plagiarism detection through web interface where the fed document is searched on the internet to find the possible clues for the plagiarism. The document searching is done through the Document cycle generator, while later is plagiarism detection mechanism through a standalone application on the computer where two documents are compared with each other to possibly locate the percentage of the plagiarism or simply document similarity. We consider plagiarism detection mechanism as document analysis problem that can be understood through the knowledge of the following aspects as detecting plagiarism in natural language is very difficult to locate.

2.1 Challenges and Language Related Issues

Detecting plagiarism in natural language can be better understood through the knowledge of the language issues related to it such as the writing style of any author is it unique and it depends upon author to author, this can be standardized to some extend but not generalized for every case, other issues such as lack of information, changing the words or paraphrasing and lack of standardization in common definition for plagiarism.

Plagiarism can take several distinct forms which includes

- **Word-for-word plagiarism:** direct copying of phrases or passages from a published text without quotation or acknowledgement.
- **Paraphrasing plagiarism:** when words or syntax are changed (rewritten), but the source text can still be recognized.
- **Plagiarism of secondary sources:** when original sources are referenced or quoted, but obtained from a secondary source text without looking up the original.
- **Plagiarism of the form of a source:** the structure of an argument in a source is copied (verbatim or rewritten).
- **Plagiarism of ideas:** the reuse of an original thought2 from a source text without dependence on the words or form of the source.
- **Plagiarism of authorship:** the direct case of putting your own name to someone else's work

Several challenges exist to find original sources for plagiarized documents. If no collection is given against which a suspicious document can be compared, it is reasonable to search for original sources on the Internet. When search engines like Google are employed, the question is which keywords from the suspicious document deliver the most promising search results. Supposed that a keyword extraction algorithm is given, queries have to be generated that combine extracted keywords with respect to a selection strategy. The search results of the generated queries form a candidate document base. All documents from the candidate document base are represented through a document model that serves as abstraction for the analysis with one or more plagiarism detection algorithms. Several methods for plagiarism analysis have been proposed in the past. Known methods divide a suspicious document as well as documents from the candidate base into chunks and apply a culling strategy to discard undesired chunks, e.g. too long or too short chunks. A hash function computes digital fingerprints for each chunk, which are inserted into a hash table: A collision of hash codes within the hash table indicates matching chunks.

2.2 Evaluation of the existing Models and Methodology

It plays an important role in the designing process for plagiarism detection system. Evaluating the already existing models or the commercial tools actually help in determining their performance and underline process they are following. This also helps in designing a new model by improving the already algorithms based on the empirical analysis through the comparison of the performance, number of the valid links obtained etc.

Two major properties of Natural Language are ambiguity and unconstrained vocabulary. For example, words are replaced by their synonyms but because word senses are ambiguous, selection of the correct term are often non-trivial. The flexibility and complexity of natural language has driven researchers in many language engineering tasks, not just plagiarism to apply simpler methods of similarity involving a minimal amount of natural language processing. As with detection between different programs, various methods of comparing texts have been investigated, as well as defining suitable discriminators. The natural language plagiarism detection is very difficult and involves Web-based sources which need to be investigated using methods of copy detection. Methods of detection originating from file comparison, information retrieval, authorship attribution, compression and copy detection have all been applied to the problem of plagiarism detection. Plagiarism detection in case of multiple text involve finding similarities which are more than just coincidence and more likely to be the result of copying or collaboration. The second stage is to then find possible source texts using tools such as web search engines for unknown on-line sources, or manually finding non-digital material for known sources.

The typical discriminators factors that indicate the act of plagiarism in the text document might include the following:

1. If text stems from a source that is known and it is not cited properly then this is an obvious case of plagiarism.
2. Use of advanced or technical vocabulary beyond that expected of the writer.
3. If the references in documents overlap significantly, the bibliography and other parts may be copied. Dangling references, e.g. a reference appears in the text, but not in the bibliography, a changing citing style may be a sign for plagiarism.
4. Use of inconsistent referencing in the bibliography suggesting cut-and-paste.
5. Incoherent text where the flow is not consistent or smooth, which may signal that a passage has been cut-and-pasted from an existing electronic source.
6. Inconsistencies within the written text itself, e.g. changes in vocabulary, style or quality.
7. A large improvement in writing style compared to previous submitted work.
8. Shared spelling mistakes or errors between texts.

2.3 Purposed Prototype Model

This is based on the comprehensive approach for detecting plagiarism. We have analyzed and studied various plagiarism detections available. We have performed intensive practical testing of the systems available to actually understand where they lack and where they are ahead. On the basis of our analysis we have collected all the requirements that we need to put in the form an advanced detection system.

2.4. Experiment and Comparisons

While studying and designing the architecture of the plagiarism detection mechanism we feel parsing or the document analysis is one of the critical area upon which the performance of the system depends hence we focused on the type of the documents analysis, searching module details and report generation section which is sufficient for the formal verification of a prototype and for the experimental verification of our approach. The prototype that we are

going to discuss is helpful in carrying out a set of experiments on various set of documents.

3. CHALLENGES AND LANGUAGE RELATED ISSUES

Identical sentences are easily modified to look different. This is a very common phenomenon that is followed in any research and teaching community. According to www.thefreeDictionary.com Paraphrasing is defined as a restatement of a text or passage in another form or other words, often to clarify meaning, the restatement of texts in other words as a studying or teaching device. Paraphrasing leads to serious cases of plagiarism General guidelines that have been issues by various research groups, universities, educational institutes are as follows:

a. If the author is taken any help from any of the sources like internet, books, newspaper he/she needs to put those phrases in quotes and put reference number hence in turn we are acknowledging the actual work of the author.
b. Paraphrasing and writing in your words must be clearly differentiated as paraphrasing is reformation of the text but writing in your own words means that you are taking help in the sense you wants to understand the meaning of the phrase and using that idea in your own work. Even if you are using your own words you need to acknowledge the author from whose work you have taken help as many times it happens that authors blame and raises questions on each other for plagiarizing the work done by any of them.
c. Acknowledging helps the reader of your work to actually know the source from where you have taken the help as this in turn helps him to better understand whole idea of your work.

Hence overall idea for this, always reference the work from where you have taken the help as this is to avoid the ambiguity about the actual source of the work done. Do not paraphrase at any cost as this will spoil your image and your work done. Everyone should know how to write in your own way this is referred as "Stylometry is the application of the study of linguistic style, usually to written language, is often used to attribute authorship to anonymous or disputed documents. It has legal as well as academic and literary applications". Imitating someone's work is not productive as understanding and learning are more important [2].

3.1 Lack of Exact Definition

There is no objective, quantifiable standard for what is and what is not plagiarism? Plagiarism lacks in exact definition as still we don't have an exact definition periphery for this word. Many research groups have contributed to this but still we are struggling for achieving the precise definition. According to S.E. Van Bramer [5], Widener University we can classify the Plagiarism according to the study level concerned like for Undergrad students plagiarism means straight forwardly copying the data available from various web sources, Research Paper, books etc while they are expected to take conceptual help from these and write in your own way, But this doesn't mean that we should take them for granted. While Post grad are expected to put forward new ideas based on their research work but problem lies if independent creation of some ideas or concept may coincide as we cannot rule out this option as well. Various Journal claim that they don't publish paper that have even small percentage of plagiarism but they fail to give accurate boundary to this meaning. Some want that new paper submitted should have at least 40% new ideas.

3.2 Lack of Knowledge

Computers are not able to understand the Natural language and interpretation their meanings as still various research groups are writing parsers application so as to make computer more intelligent, as the effort to make systems independent and decision makers but this is still a vision. Natural Language processing has achieved a new pinnacle such as various language parsers have been made, efficient grammar checkers, efficient algorithms for text parsers but still we are far away from the automatic language processor [1].

3.3Stylometry Issues

Plagiarism can be easily detected if we have an automated system for Stylometry comparisons as Stylometry of any author cannot be standardized. Moreover, it's very difficult to maintain the database of the Stylometry actions of all the authors as Stylometry may or may not be same. This technique is only useful where we have less number of the authors example Author A and Author B have done certain work now if want to see how much percentage A has copied from B and how much B has copied from A , in that case we can use Stylometry but it's not for the generalized cases.

4. EVALUATION OF THE EXISTING MODELS AND METHODOLOGY

There are many tools available that claims to provide you exact information regarding any act of plagiarism in the submitted documents. Plagiarism detectors are of two types depending on the type of the source fed to it. One is source code plagiarism detection mechanism and second is simple text plagiarism detector. Our scope is limited to the Simple text plagiarism detection mechanism. Several academic institutions use various commercial/non commercial tools for ensuring fair submission of the assignment. This software is generally web based which actually have large repository of the text documents in the form of the journal, Research papers, Books etc. They regularly update their databases to include more and more latest papers. Some even have third party tools deployed such as proquest, word net etc. Some of the commonly available plagiarism detections are as follows

4.1 Plagiarism Finder

This is a standalone Application and is available on the internet having 30 days free trial license [10]. We have downloaded this tool and tested on various set of the text files. The results that we got have been interpolated using graphs. Our research includes the intensive testing by taking different text chunking having length range from 100-1000 words text. We have taken different articles from various fields such as medical science, space research, English essay, e-tutorials etc so as to get a general over view of the results. Our results are plotted as word length versus accuracy of the result obtained and word length versus number of the valid links obtained. Analysis shows that when the number of words are increased then percentage accuracy decrease and number of the valid links returned increases. Figure 1 we have plotted graph between the word length and number of the links returned where as the document size increases the number of the possible web links increases hence the accuracy decreases. We have chosen various set of parameters on the basis of which we performed our analysis Document size, number of the Links, percentage accuracy, Relevance factor etc. The results initially seem to be the empirical results but when the same operation is repeated number of times than that becomes a result. We have deduced all these results after intensive testing on about 80 different documents.

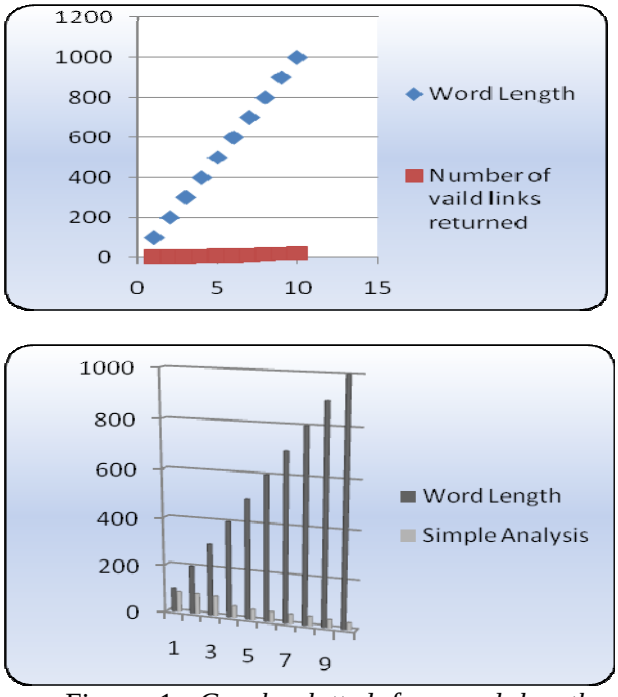


Figure 1: *Graph plotted for word length versus Number of the valid links returned*

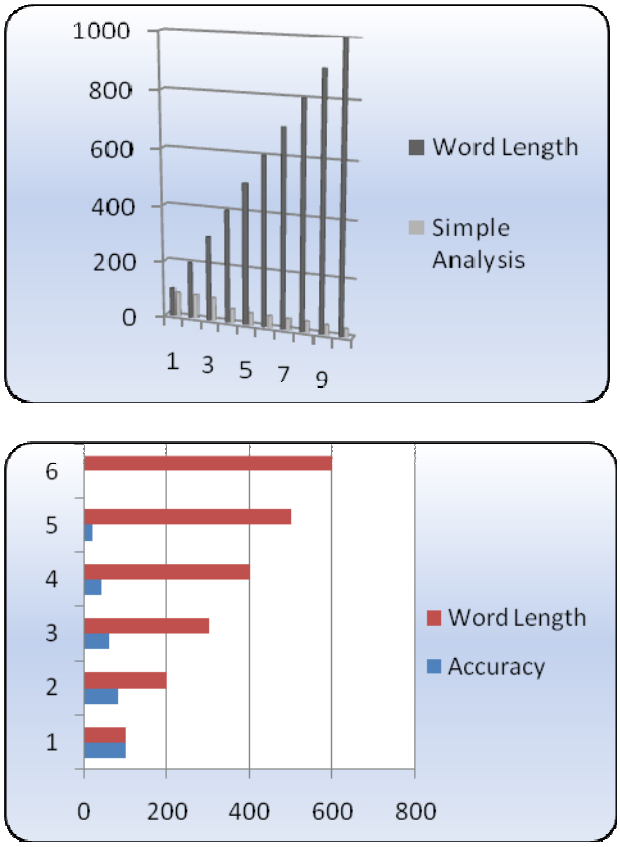


Figure 2 *Word Lengths versus Accuracy graph. This is the representation of the relationship between the document length and Accuracy produced.*

Throughout our finding we were restricted on for just 1000 words because in case of the trial version we are restricted for just 1000 words only. This complete analysis is based on the process that we have followed, test documents taken into consideration, threshold values assumed.

4.2 COPS & SCAM

Both these systems have been developed by the Stanford University; both are based on the simple objective of the detecting plagiarized documents to the copyright recordation and registration authorities. COPS stands for Copy Detection Mechanism for Digital Documents based on the principle of Sentence chunking and finally correlating those phrases produced by the COPS parsing unit. SCAM stands for the Stanford Copy Analysis program which is based on the word chunking algorithms. Both these programs use their own build crawlers to search the links for phrases produced after word chunking [4].

In case of copy Detection mechanism the text is collected from all the sources and converted to single format like ASCII in this case. Then sentence chunking is carried out where sentences are located and they are chunked with the help of hashing and stored in the database. After storing the sentences, query processing is carried out based on a text matching algorithm. Stanford Copy Analysis program is based on the principle of the word chunking where chunking is performed on the basis of word end points. As the parser on detecting the white space assume it to be one word. Hence after collecting, the word chunks are arranged and they are stored in the database. Besides chunking semantic analysis is done to eliminate those words which are not useful from the searching point of view. The database performs the redundancy check. After that query optimizer takes the charge and same processing occurs as that in the COPS. The type of chunking performed largely affects the accuracy of the system [3].

There are many tools available that claims to provide you exact information regarding any act of plagiarism in the submitted documents. Plagiarism detectors are of two types depending on the type of the source fed to it. One is source code plagiarism detection mechanism and second is simple text plagiarism detector. Our scope is limited to the Simple text plagiarism detection mechanism. Several academic institutions use various commercial/non commercial tools for ensuring fair submission of the assignment. This software is generally web based which actually have large repository of the text documents in the form of the journal, Research papers, Books etc. They regularly update their databases to include more and more latest papers. Some even

have third party tools deployed such as proquest, word net etc. Some of the commonly available plagiarism detections are as follows:

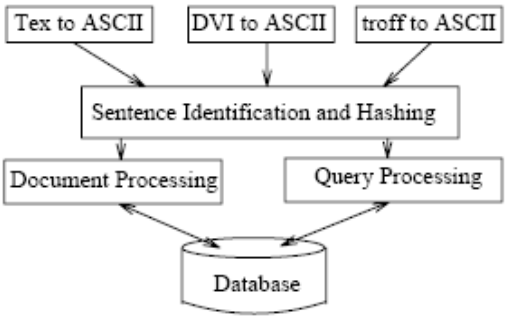


Figure 3 *Copy Detection Mechanisms for Digital Documents Architecture*

Copy Detection mechanism is based on the simple process of sentence chunking as in the diagram above text is collected from all the sources and converted to single format like ASCII in this case. Then sentence chunking is carried out where sentences are located and they are chunked with the help of hashing and stored in the database. After storing the sentences, query processing is carried out based on a text matching algorithm.

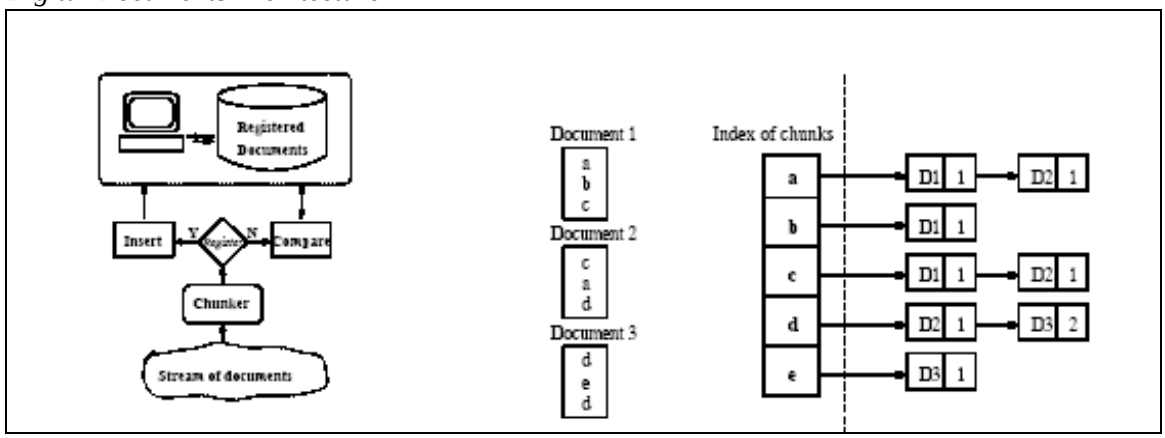


Figure 4:

Stanford Copy Analysis program Architecture

Stanford Copy Analysis program is based on the principle of the word chunking where chunking is performed on the basis of word end points. As the parser on detecting the white space assume it to be one word. Hence after collecting all those words word chunking are arranged and they are stored in the database. Besides chunking semantic analysis is done to eliminate those words which are not useful from the searching point of view. Above figure shows the same that document having a b c as the phrase or paragraph which after chunking produces a ,b ,c which are then stored in the database and database in turn performs the redundancy check. After that query optimizer takes the charge and same processing occurs as that in the COPS. The type of chunking performed largely affects the accuracy of the system.

5. PROPOSED MODEL

This Model is based on the comprehensive approach for detecting plagiarism. We have analyzed and studied various plagiarism detections available. We have performed intensive practical testing of the systems available to actually understand where they lack and where they are ahead. On the basis of our analysis we have collected all the requirements that we need to put in the form an advanced detection system. Various set of requirements were

1. Ideal system should detect possible phrases of plagiarism in the shorter time span.

2. Ideal system gives user an option for the type of analysis he/she wants.
3. Besides performing the intensive searching it first gives user an overview of the possible percentage plagiarism.
4. As after performing various set of analysis on the present system it has been revealed that we need a system that should not fail after a certain numbers of words lengths.

Our plagiarism detection mechanism takes into account the document architecture into consideration for developing efficient plagiarism detection. The success of any mechanism is based on the following factors

1. **Document Analysis Unit**: Implementation of the document analysis unit as if the document analysis unit is really efficient then whole model will be productive [8].
2. **Relevance Factor Calculation** Relevance of the source you are taking into consideration while developing the conclusion on plagiarized document as many times happen the source what we choose for the analysis purpose is in turn a plagiarized source. Example information regarding the yahoo search API development is present on the www.developer.yahoo.com but the same information is available on some web blog in this case web blog is a plagiarized source and www.developer.yahoo.com is the actual source.
3. **Efficiency Factor**: Efficiency and relevance of the report generation phase as this section manipulates the entire results that are produced after the internet search. So the report generation algorithm will help to deduce the actual link from where the document has been prepared.

5.1 Document Analysis Unit

Developing any Plagiarism detection mechanism you need to implement an effective Document/parsing system. Parsing system designing is very crucial from the performance point of view as the results you will display for searches you perform is entirely dependent upon the effectiveness of the text extraction and text synthesis modules. Architectural Model can be discussed as the input in the form of text document is taken and fed to the Parser unit which actually performs the parsing in three different forms particularly sentence chunking, word chunking and random chunking. Unitization may be defined as "the process of the configuration of smaller units of information into large coordinated units". Unitization may also be interoperated as Chunking. Hence Unitization is the process of converting the whole document under consideration into smaller chunks or units where chunk or a unit represents the word or a sentence. Natural language processing is a field which is interrelated to the artificial intelligence; this is of great importance as detecting plagiarized phrases becomes easier after the semantic analysis of the document .Various artificial intelligence principles make it easier for carrying out the text analysis and for parser generation in the given document.

The detection is performed in two phases. First, the parser processes input collection file by file, and generates set of parsed files. Second, detection model checks the parsed files for similarity. The designer needs to perform various set of experiments which includes parsing, tokenizing and preprocessing. Such model have some major drawbacks that parser destroys the initial word order in every sentence of the input text. Therefore, the plagiarism detection system cannot precisely highlight similar text in blocks of text in original file pairs. There are two ways for representing plagiarism which includes either system should be programmed to highlight the whole plagiarized sentences and parser should generate some metadata about the parsed files or it can represent the word chains representing the copied data.

5.1.1 Simple Examination Parser

If no document base is given, a candidate document base has to be constructed using search interfaces for sources like the Web, digital libraries, homework archives, etc. Standard search interfaces are topic-driven, i.e.

they require the specification of keywords to deliver documents. Here keyword extraction algorithms come into play since extracted keywords serve as input for a query generation algorithm. Most of the keyword extraction algorithms are designed to automatically annotate documents with characteristic terms. Extracted keywords shall not only summarize and categorize documents, but also discriminate them from other documents. To automatically identify such terms is an ambitious goal, and several approaches have been developed, where each of which fulfills one or more of these demands. Existing keyword extraction algorithms can be classified by the following properties:

1. A keyword extraction algorithm
2. Document collection on which keyword extraction algorithms relies as these algorithms perform better with respect to the discrimination power of identified keyword extraction.
3. Language dependency as it is very important to the language specifications.
4. Various mathematical and statistical approaches.

5.1.1.1Sentence Unitization/ Chunking

This unit will perform the chunking or the phrase/sentence selection on the basis of the conjunction where we have specifically chosen some of the well known keywords such as comma, dot, exclamation symbol, hyphen etc which certainly help in selecting the phrases these are then stored in the array. Generally difficulty lies when in a simple English sentence these symbol do not appear which leads to long sentence selection but for handling this we added a case if sentence length crosses a certain threshold then sentence breaks into two smaller units. This represents the most basic of all parsing that any basic utility expects [5]. In figure 5 we have simple parsing model where central unit is the conjunction separator which exactly looks for the conjunctions and finally we have the phrase builder which removes the phrases one by one and stores in an array. The phrases that are once stored are fed to the searching unit which in turns performs the search operation. Finally the output is then fed to the manipulation chamber.

5.1.1.2 Modified Sentence Unitization

The case we have so far discussed has certain flaws like the sentence breaking process is entirely dependent upon the occurrences of the conjunction in the text. For optimum results the sentence length needs to be in a threshold range. Threshold value is the region marking a boundary. The worst cases that appear in this scenario are:

1. **Greater Threshold value**: Many times it happened that sentence length crosses certain threshold value like example there can be maximum of 40 alphabets in the sentence chunk but if the length becomes 50 or more. Search engines don't recognize query having more than 8 words or 30 alphabets like in case of MSN Search.
2. **Lesser Threshold value** Sometimes parser based on sentence unitization/Chunking selects sentences haves length one word or 10 alphabets. These types of sentences are irrelevant from the information point of view hence if taken into consideration; these reduce the accuracy of the results.

These cases adversely affect the performance of any plagiarism detection system hence they need proper handling. We have seen that on handling the above two worst cases, accuracy of the system increases [6]. After the introduction of special cases for handling the worst cases, the rate of similarity percentage increases. Now the after certain threshold, the similarity value lies in the range of 30-45. Figure 5 shows the graph for percentage similarity versus the document size in case of sentence and modified sentence unitization as the graph reveals that modified sentence is better as compared to simple sentence from accuracy point of view.

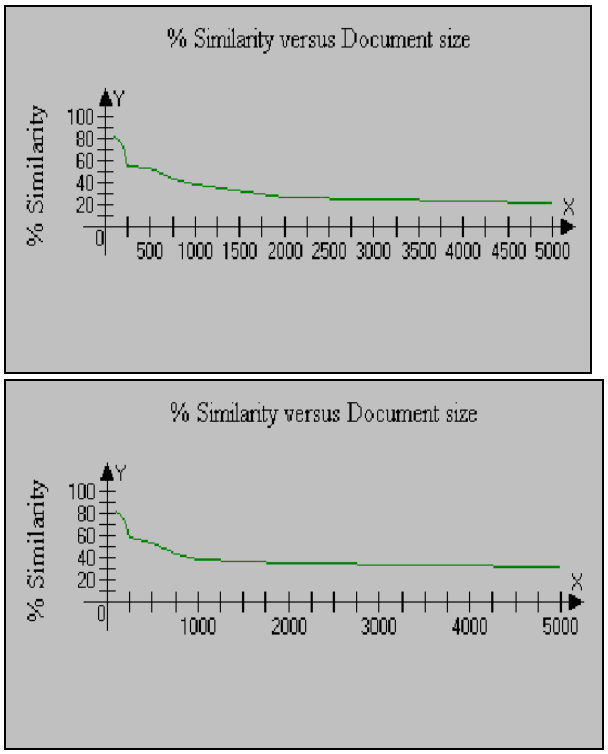


Figure 5 Graphs for Percentage Similarity versus the Document Size for sentence unitization and modified sentence unitization.

5.1.1.3Sentence Unitization/Chunking by External Tools

The trend for Unitization of sentences through external or third party tools has come up in recent years. Unitization process can be improved after understanding the linguistics of any language like in our case English language has verbs, nouns, adjectives, adverbs, and pronouns etc which are very important part of one's speech. Linguistics' categories these into two different classes one is open class and other is close class. Open class denotes verbs, nouns, adverbs and adjectives while close class words are conjunctions, prepositions, pronouns etc. Now this classification is important in the sense open class words contains the real meaning of any sentence but the close words don't contribute to similarity checking principle[9]. We have testing Stanford POS Tagger [12] with our detection mechanism; this has been developed by the NLP Stanford Group which helps in indentifying the open and close words in the speech. This process overall increases the accuracy of the desired system. We also employed Word Net which has been developed by the Princeton University which helps in providing the synonyms for the words that we recollect after POSS Tagger. Hence the overall operation ends with preparing the list of synonyms corresponding to each word in the sentence, this in turn helps in detecting the possible cases of text modification. When we compared simple sentence, Modified sentence and unitization through external tools; we find unitization through external tools has highest accuracy to give optimum results. We plotted the graph same as we have done for the simple sentence and modified sentence unitization [16]. The external tools when added actually help in improving the similarity percentage to some extent but the effect is very minimal and needs further improvement. This empirical analysis helps in knowing the behavior of the similarity percentage. These tools are actually plug and play as we can even add these tool to more sophisticated categories than sentence unitization that have been discussed in later section of this research paper.

5.1.2Detailed/Comprehensive Examination

This part of the parser unit performs word chunking where we find the white space as when parser checks the first word, after reaching the last alphabet of the first word white apace occurs hence first word is recognized and stored in the database. This is process continues till the document or the text is fully parsed. After that chunk builder builds phrases by using these word pairs and fed those phrases to the searching chamber. Detailed parsing is one of the most difficult and complex unit in the whole mechanism. It is difficult as we need to locate the phrases that match our threshold value, for any effective internet search there is certain value of the threshold value for which the results returned are appropriate. Detailed parsing can be subdivided into different mechanisms such as word unitization, overlap word unitization,

hashed break point unitization etc. All these techniques have some unique directions of applicability like in grammar checking etc

5.1.2.1 Word Unitization/Chunking This is second kind of the unitization / Chunking process where word chunks are selected by locating the empty spaces like if we have large text documents then initially points to first alphabet of the first word then it starts traversing the entire string (word) now as it approaches the last alphabet of the word then on next increment white space will occur hence first word will be chunked and stored in the database. This process is repeated and whole document is converted into word units example let W1, W2, W3, W4, W5, W6, W6 - - - be the word sequence generated. Now these word units are collected again and word collection is made such as x= W1W2W3W4W5, y = W6W7W8W9W10 etc and then this word collection is searched on the internet and corresponding search results are stored in the database to generate the final results. We have plotted the accuracy versus word chunks graph to see its performance.

5.1.2.2 Overlapped Word Unitization: This is same as the above discussed category of chunking but it has a major difference that word chunks after generation are recollected to make overlap sequence example like W1 , W2 , W3 , W4 , W5 , W6 , W6 - - - Wn are the word units produced from a document Sn , overlapping sequences can be W1W2W3W4W5 , W4W5W6W7W8 , same sequence is repeated for the rest of the words. We define x = W1W2W3W4W5, y = W4W5W6W7W8 etc. Now x, y and other such parameters are then searched through the search engine. We found that overlapped work units/chunk have higher performance then the simple word chunking method. Figure 6 shows percentage similarity factor in case of overlap word chunking is greater than word chunking.

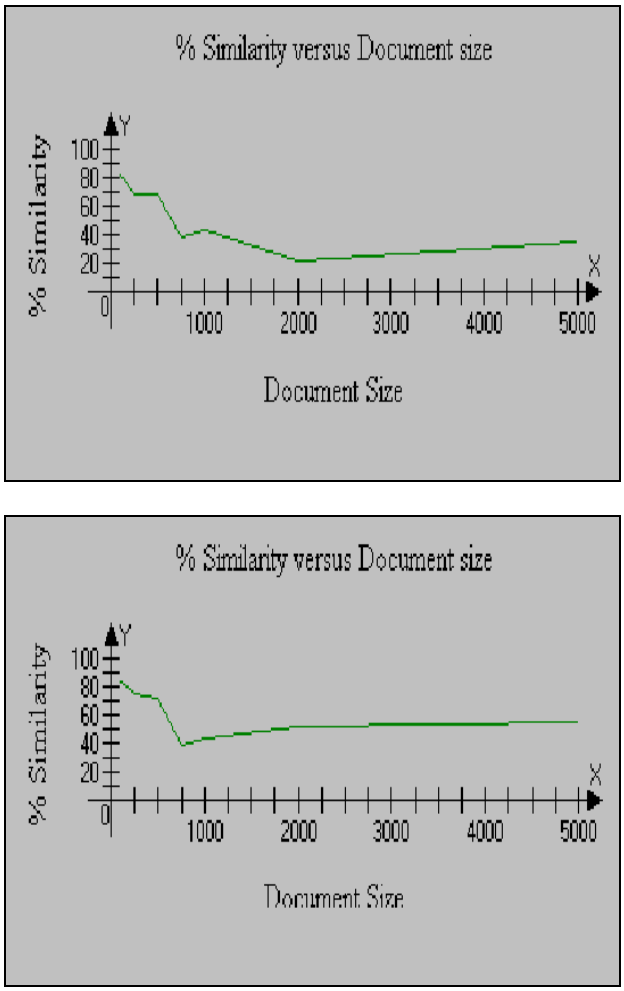


Figure 6 *Graph for Percentage Similarity versus document size for word chunking and overlap word chunking/ unitization.*

5.1.2.3 Hashed Break Point Unitization

Hashing is very common technique used in the document fingerprinting where we prepare the hash values for the words/sentences or keywords in any text documents. These hash values are specific to particular sequence example we want hash value for Plagiarism so in this case we take the sum of all the ASCII numbers in this word [15]. Hashing actually improves our detection process as now for checking its performance we have taken two documents from Natural Language Processing tutorial available on one of the internet website. Now when prepared two documents one was original and second was plagiarized as we have done changes like changes

in the paragraphs, rewording, paraphrasing etc. After all these operation our analysis has shown that this process of hashing has given fruitful results. Hashed Breakpoint chunking is actually the application of hashing but in different way [14]. Here we prepare the list of hash value corresponding to each word. After this we need words to be arranged in a particular way for this we define a parameter n that actually helps in determining the phrase chunk end. Parameter n is chosen such that in a sequence have 8-9 word chunks, fourth chunk gets completely divisible by the n as at least we need three chunks for performing the searching operation. Actually with experiments we find that value of the n is dependent upon the language features, field of the education system to which Paper belongs, size of the document etc. Figure 7 shows the behavior of both hashed and hashed breakpoint chunking to the document size , value of the n parameter selected, type of the document chosen for experimentation.

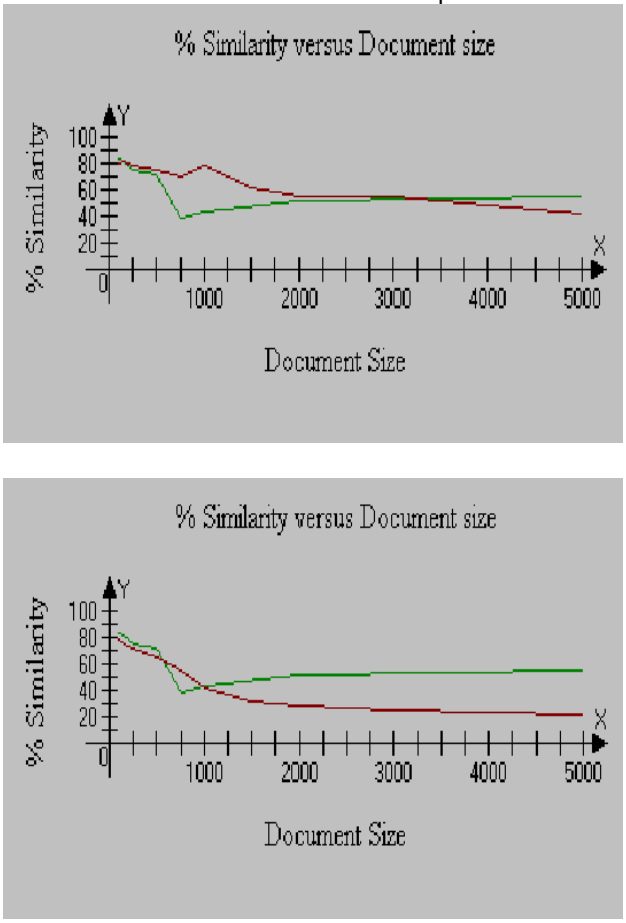


Figure 7 *Graph for Percentage similarity versus document size for hashed chunking and hashed breakpoint chunking.*

5.1.2.4 Over lapped Hashed Break Point Unitization/Chunking

This is modified hashed break point unitization where we choose value of the parameter n more than once i.e. after deciding the parameter n value for first operation we increase the value of n by some factor this operation is repeated after every operation [13] example:

Step 1: Prepare Hash Table for entire words in the document.

Step 2: Choose suitable value for Parameter n

Step 3: Start dividing the words with n.

Step 4: Find word getting completely divisible

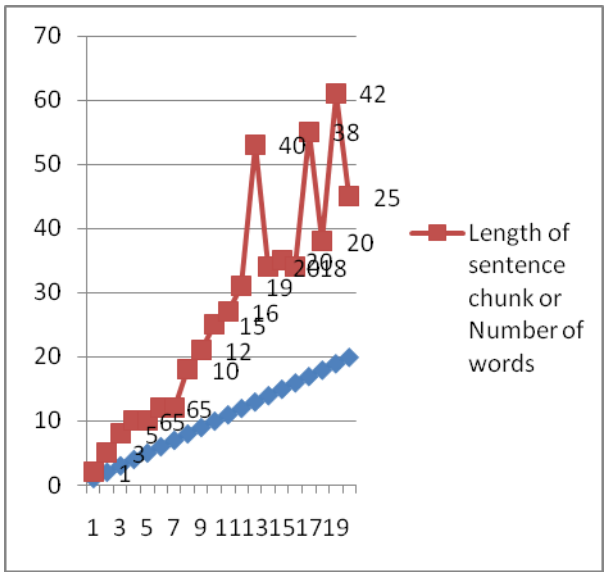
Step 5 First chunk sentence gets selected.

Step 6: Now value of n will be n+2

Step 7: Repeat the operation from the start.

Step 8: Repeat this till last word of the text is detected.

We have done experiment for calculating and verifying the behavior of the hashed breakpoint and overlap hashed break point chunking. We found that as you change the value of the parameter, the value of the accuracy also changes. We found empirically that the accuracy is highest for value 7 and 5 under given test condition. We have also found that accuracy of the system also dependent upon the size of the chunk taken for searching on internet. These results were analyzed through the interpolation of the parameter n, chunk size and percentage accuracy obtained.



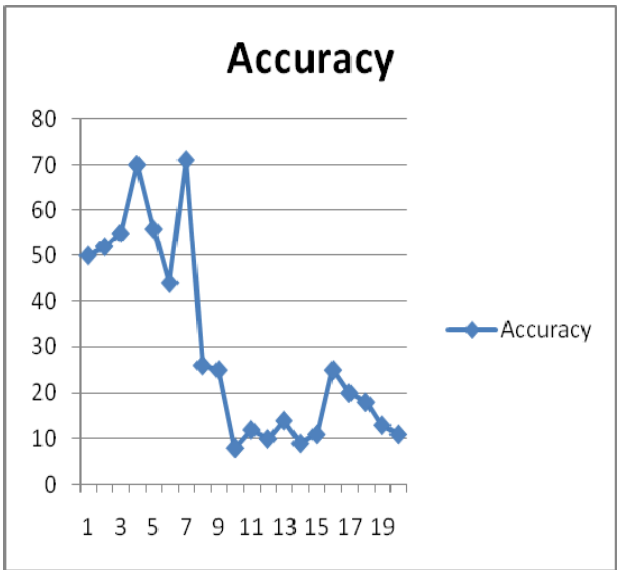


Figure 8 *Graph for Accuracy versus the value of the parameter choosen and length of the sentence chunk versus parameter n and acuracy achieved.*

5.1.3 Random Generator

This is based on the special kind of chunking that is not exactly sentence or word chunking. Here we basically locate for random chunks of the sentences for which there can be numerous ways such as select first and last line of all the paragraphs and chunk them, select only those sentences that have length greater than a particular threshold etc .But this is exactly useful where user need a quick overview of the document .

5.2 Searching Chamber

After the parsing phase is complete the phrases/chunks/chunking combination need to be searched on the internet for which we have used MSN Live search API which is free for non commercial use whose SDK is provided by Microsoft. We use these API which takes our phrases as inputs and provides suitable links for each phrase searched. We choose only first five links out of the approx 180 links returned/per query. We arrange all the links which are returned for whole chunks/sentences etc in our database. Then further processing is carried by the selection unit

5.3 Manipulation Chamber

This whole chamber is further sub divided into three subsections such as:

the total number of words in the paragraph and calculate the number of words that we got on the internet through the search engine , then take their ratio this will give the approximate percentage per paragraph. Other techniques can be through attribute counting like calculate total number of dots , punctuation marks and number of words then compare it with the total number of all the above in the links returned by the search engine .But incremental link gives the exact source from where the plagiarism has been done. Our experimental results show the same

6. EXPERIMENT AND COMPARISONS

In order to access the performance of the proposed approach we have carried out an extensive set of the experiments on a test data which is obtained from the set of documents from different education fields. Search results differ from one search engine to other due to different crawlers and indexing algorithms example Google search results for a query is

different from that of the Yahoo Search. During the experiment session we have realized plagiarism detectors show abrupt changes when size of the document increases. This section mainly focuses on our observations while designing plagiarism detection mechanism. This was certainly one of the major postulates of our research work. We tried to get empirical results from all the experiments that we have carried out. The experimental framework is largely based on the proposed methods of document analysis which in turn relies on the comparison of those methods.

6.1 Observation 1

We have used approx 80 Documents from various fields such as medicine, technical and research paper, universities web pages, educational assignments, new technologies related white papers, work specific assignments etc. According to our empirical analysis random analysis has least accuracy but it's quick. Advanced/Detailed parsing has the highest average accuracy as its worst and the best case are nearly same. Simple parsing takes time less than advanced but more than random analysis. Simple parsing is the default analysis in our detection mechanism. Figure 9 represents the graph for word length versus number of the valid links returned when words are searched on the internet. The graph has been plotted by repeating the experiment through varying the word document size. The study shows that as the size of the document increases the performance factor reduces.

6.2 Observation 2

Second case is comparison of the incremental value calculated per link that are stored in the database example suppose we have a link amandeep.ucoe.com at second position in the database the incremental value is one for this case now when Binary search tree is constructed if in case the same link repeats again then incremental value is set equal to two, the same process is repeated for whole set of links stored in the database. Now graph is plotted between the accuracy level and incremental value. Studies show that if incremental value of a link is high than accuracy will also be highest .These results we have prepared after intensive testing taking 100 different documents from different approximate 15 different fields. We have also calculated the empirical formula for calculating incremental values on the basis of the field taken into consideration.

6.3.1 Selection Chamber

This unit actually proceeds on the links returned by the searching chamber and stored in our database. This unit creates the Binary search tree using all the links stored the database. After creating the Binary Search Tree Selection unit will calculate the link repetition by removing the duplication links from the total links that we got by increasing their count. The link with higher increment value will be the actual link form where the document has been prepared.

6.3.2Percentage Plagiarism Calculator

This takes the output of the selection unit as input and prepares the final out put that need to be shown to the user through our user interface section. Various techniques that are used for the plagiarism calculation can be calculate

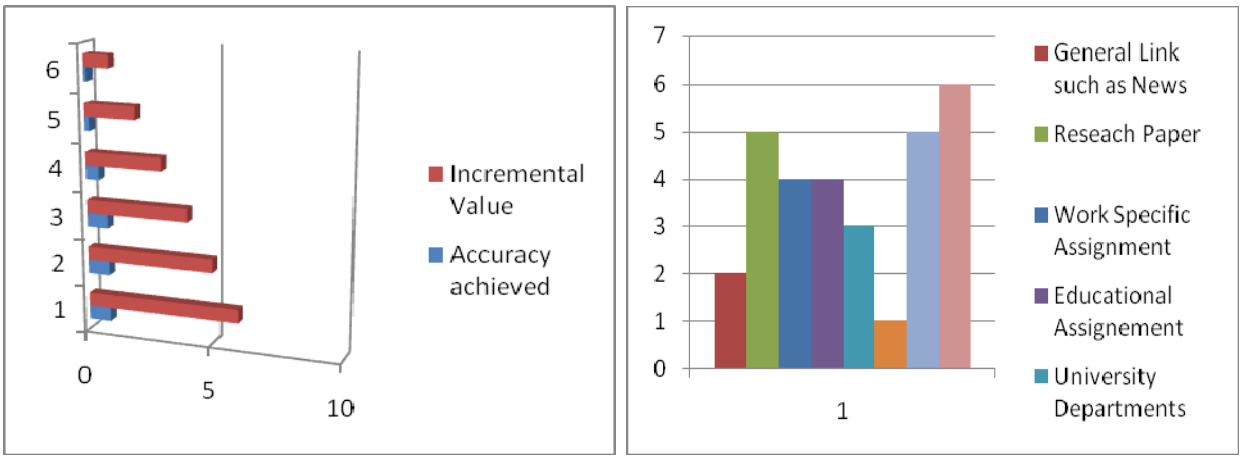


Figure 9

a) Graph for Accuracy versus Incremental value.
b) Incremental value calculated for different fields

Field of the Document	Number of Documents
New Technology Papers	14
Educational Assignments	21
University Departments	12
Work Specific Papers	16
Research Papers	12
Core Research Topics	9
Advertisements	7
General links News	9
Total	100

Figure 10 *Table displays the document number and the corresponding field*

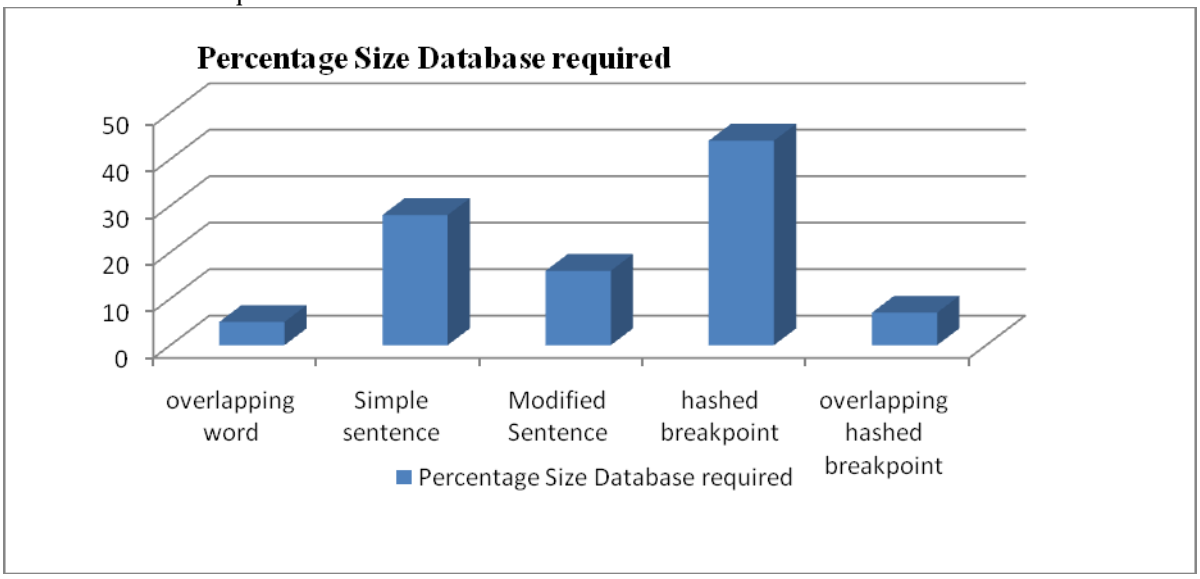
Observation 3

Thirdly we have plotted the graph on the comparison of the incremental values for different fields taken into consideration example we have taken 5 different research papers and taken a specific chunk of the text from all those papers one at a time. This operation was repeated about 15 times and finally we come at an empirical value for incremental value that is 5 while for core research papers the value is 6 hence these operations were repeated for 8 subjects from 15 different fields. Figure10 contains the table representing the documents which we have used for testing the results.

Observation 4

We have observed certain phenomena which are very important from the architectural point of view. While testing the reliability of the various ways of the text parsing we tried to calculate the percentage chunking speed

and percentage database size required. We found simple sentence chunking is having the highest speed and comparable to that of hashed breakpoint. We have also calculated database size required where we have found hashed breakpoint requires highest database size. Figure 11 shows the relevance of our results.



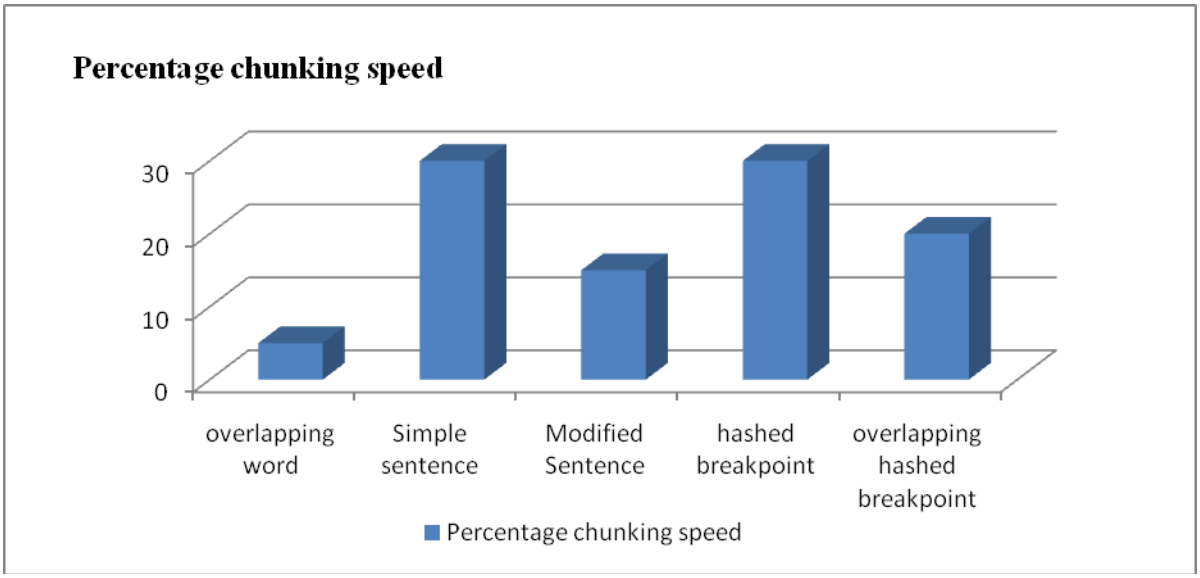


Figure 11 *Graph for percentage chunking speed and percentage database size required versus various document analysis techniques.*

Observation 5

Our foremost objective was to find the plagiarism in the test documents. We have put forward various documents analysis methods and for deciding the best from the given methods we have performed intensive testing for calculating percentage reliability to find plagiarism in large size documents. Our results have shown that overlapping word chunking is having the highest percentage reliability of finding plagiarism. We have also compared the three types of the document analysis i.e. simple analysis, detailed and random. This comparison was

done to find the accuracy versus the number of the links returned taking specific documents under the test environment. Figure 12 shows the graphs for percentage reliability for finding plagiarism and comparison of various document analysis techniques. Our study is empirical but helping enough to study the response of the plagiarism detection mechanism and functional units which constitutes plagiarism detector.

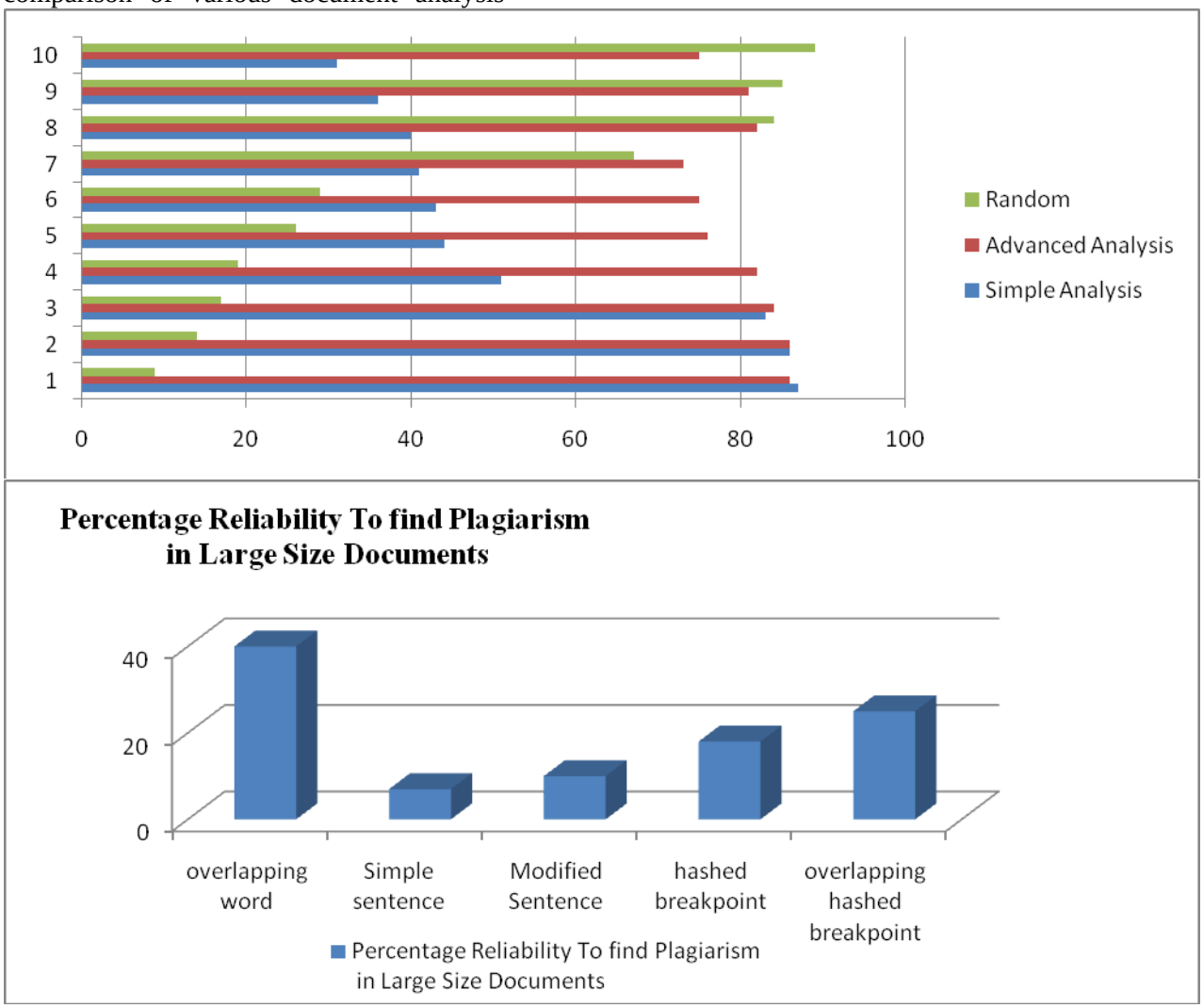


Figure 12. Graph for percentage reliability for finding plagiarism versus various document analysis mechanisms and percentage Accuracy versus various document analysis techniques.

7. COMPLETE MODEL

Complete model which has been constructed through the mechanisms and experimentations which we have so far discussed. The Business model is different from the user interface point of view but the actual internal functionality is same as we representing in the purposed model. The model can be explained as first of all we have the web client which is the user machine through which user will submit the text documents for finding plagiarism hence we can say web client consists of configuration settings, input to the originality checker and viewing of the final report after percentage plagiarism calculation. Second part is web server, now

after submitting document through internet, web server will perform the user access authorization and document is imported with the desired import configuration settings. Third portion is Parser/ Search Server which consists of parsing engine and search engine, now parser engine will perform the document analysis depending upon the desired analysis mechanism selected by the user under configuration setting. After performing parsing the chunks/sentences are stored in the database where the indexing is performed. Indexed chunks are fed to the search engine where we are already using the search API's of the prominent search engines such as Google Search , Yahoo Search etc through the medium of the web services. Finally after finding the web links which are returned corresponding to the chunks searched, all these are fed to the manipulation chamber located in the web server where final plagiarism percentage is calculated and the results are fed to report generation module which will display the final plagiarism percentage.

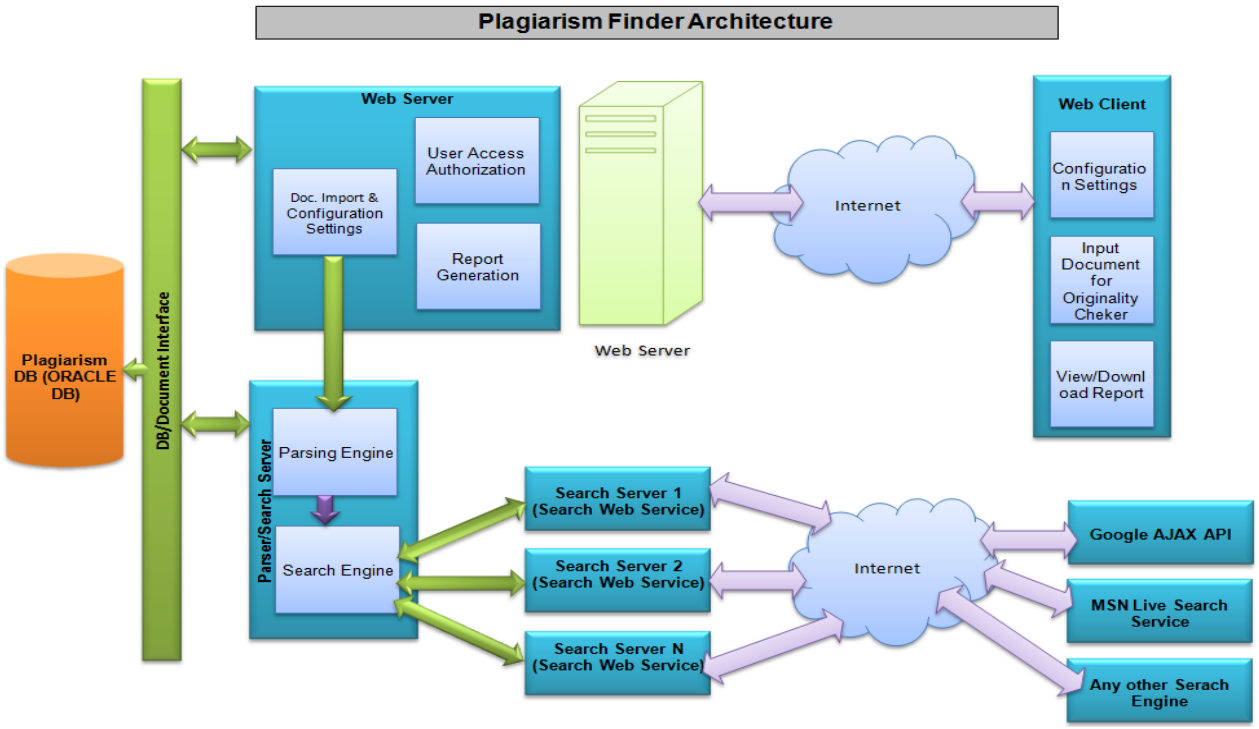


Figure13. Block Diagram of the Business model for plagiarism detection system

8. FURTHER SCOPE OF WORK:

Many Academics and Commercial organizations have been facing the problem of plagiarism from past four decades. There are still many questions which are unanswered. The area of Artificial intelligence which deals with text and text transformation i.e. Natural Language processing poses greater possibilities of inventing a sound mechanism which is capable of detecting plagiarism in any kind of text documents.

Multi-Lingual Support: Till today the research in plagiarism detection is only limited to English language. However, the web access which is known as the chief source of plagiarism has multi lingual nature. It is many time possible that plagiarist translates the source data into different language and in that case it becomes very difficult to detect plagiarism. The researches which are only available in languages such as German, French or in other European language can be easily changed into English

and copied as it is hence, no tool till date can detect. The requirement of the time is to create very stronger Cross Language information Retrieval (CLIR) and Multilingual Copy Detection systems. This is very difficult to attain because of different orthographic structures of different languages.

Size of Text collection chosen: Till date, we don't have any standard collection of text for plagiarism detection in natural languages exists, thereby making comparison between various approaches impossible, unless the same text is used. Many different areas of language analysis such as document analysis, information retrieval, summarization, authorship analysis have been used in the detection process but still researchers are not able to reach at a common size for the text that produces nearest results. It is important as it will enable communities in comparing different approaches. It would help to stimulate research in automatic plagiarism detection.

Detecting Plagiarism in Single text: Detecting plagiarism within a single text is one of the hardest problems ever in the area of language engineering. Till date, we prefer manual inspection and tools which are available in the market can't change manual checking in any way. The approaches such as authorship attribution and style analysis are not able to handle this problem. Many inconsistencies exist in detecting possible source text having single or two word texts.

9. CONCLUSION

We believe that our plagiarism detection mechanism is efficient than the already present plagiarism detectors. The reasons for support of this answer lies in our model, our model supports three types of the views such as detailed analysis , simple and advanced analysis. Our detector is a complete package in itself as it has both sentence and word chunking implemented. Our system still needs improvements like in our analysis we didn't take any noise ratio into consideration as after handling noise issue the system can be made more precise. Plagiarism is increasing with leaps and bounds and we need more comprehensive study to combat this disease. We hope our effort will help the research community to develop more advanced tools than our system.

10. REFERENCES

[1] Council of writing program administrators: Defining and Avoiding Plagiarism: The WPA Statement on Best Practices. http://www.wpacouncil.org/node/9

[2] Virtual Salt - Anti-Plagiarism Strategies for Research Paper <http://www.virtualsalt.com/antiplag.htm>

[3] SCAM: A Copy Detection Mechanism for Digital Documents By Shivakumar and Hector Garcia-Molina -*Proceedings of 2nd International Conference in Theory and Practice of Digital Libraries (DL'95) , Austin, Texas, June '95* <*http://*stanford.edu/pub/papers/scam.ps >

[4] Brin, S.; Davis, J.; Garcia-Molina, H.: Copy Detection Mechanisms for Digital Documents < http://dbpubs.stanford.edu/pub/1995-43 >

[5] How to Avoid Plagiarism in a Research Paper: **By** eHow Education Editor< http://www.ehow.com/how_9265_avoid-plagiarism-research.html >

[6] Plagiarism Detection and Document Chunking Methods < http://www2003.org/cdrom/papers/poster/p186/p186-Pataki.html>

[7] A Decision Tree Approach to Sentence Chunking< http://www.springerlink.com/default.mpx >

[8] Sentence Boundary Detection (Chunking) by Gary Cramblitt <

http://lists.freedesktop.org/archives/accessibility/2004-December/000006.html.

[9] Natural Language Processing – IJCNLP: First International Joint Conference on Natural Language Processing, IJCNLP 2004, held in Hainan Island, China in March 2004 < http://www.springer.com/computer/artificial/book/978-3-540-24475-2 >.

[10] Plagiarism Finder: Tool for Detecting Plagiarism http://plagiarism-finder.mediaphor-ag.qarchive.org/.

[11] The Stanford Natural Language Processing Group http://nlp.stanford.edu/software/tagger.shtml.

[12] Digital Assessment Suite http://turnitin.com/static/home.html.

[13] Steven Bird Ewan Klein Edward Lope Chunk Parsing <http://research.microsoft.com/india/nlpsummerschool/data/files/stevenbird%20-%20chunking.pdf>

[14] Yoshimasa Tsuruoka and Jun'ichi Tsujii Chunk Parsing Revisited:

[15] <http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/papers/IWPT05-tsuruoka.pdf >

[16] Miriam Butt, Chunk/Shallow Parsing

[17] <http://ling.uni-konstanz.de/pages/home/butt/teaching/chunk.pdf>