Plagiarism Detection in e-Learning Systems

A Thesis Submitted for the Degree of Master of Philosophy

R.V.S.P.K. Ranatunga

University of Colombo School of Computing

May 2012

Declaration

The Thesis is my original work and has not been submitted previously for a degree at this or any other university/institute. To the best of my knowledge it does not contain any material published or written by another person, except as acknowledge in the text.

Author’s name: R.V.S.P.K. Ranatunga Date: ………………

Signature: ……………………………………….

This is to certify that this thesis is based on the work of Mr. R.S.P.K. Ranatunga under our supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by

Supervisor 1 Name: Dr Ajantha S. Athukorale Date: ………………...

Signature: ………………………..

Supervisor 2 Name: Dr K.P. Hewagamage Date: ………………..

Signature: …………………………

Acknowledgement

First and foremost I offer my sincerest gratitude to my supervisor Dr. Ajantha S. Athukorale a senior lecturer of the University of Colombo School of Computing who has always supported and guided me throughout my research with his vast experience and valuable knowledge. It is impossible to carry out this research without his encouragements and patience. Besides I owe sincere and earnest thankfulness to my co-supervisor Dr. K.P. Hewagamage a senior lecturer of the University of Colombo School of Computing whose encouragement, guidance and support from the initial to the final level enabled me to fulfill this task.

I am very grateful to Mr. Dulip Herath for the stimulating science discussion we had in UCSC and gives very important suggestions, advices, feedbacks and great encouragement when I was in difficult stages on my research.

I acknowledge the National e-Learning Project of Sri Lanka for their funding facilities to do this research successfully and I gratefully thank Mr. Henrik Hanson, Mr. Lars Glimbert and the staff of the computer department of the Stockholm University who arranged Sweden visits and grant me great opportunities to obtain vast experience on the field. I also acknowledge the staff of the UCSC including director, and the e-Learning Center of the UCSC with the coordinator for their kind cooperation for arrangements of necessary administrative tasks when releasing funds and other equipments.

I would like to show my gratitude to my colleagues Mr. Viraj Welgama, Mr. Kenath Thilakarathne, Mr. Damith Sandaruwan, Mr Ajith Wickramasingha and Mr. Sedara who always share their knowledge of the field with me and gives valuable suggestions and feedbacks.

Last but not least, I wish to acknowledge the patience and endurance of my loving wife, Crishanthi who never complained when I had to spend many time over nights and weekends to complete this research. Appealing smile of my loving two daughters was also greatly helpful me to be relax for a moment during this hard work.

Finally, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

Abstract

At present, the problem of plagiarism is being increased by widespread use of online documents, the Internet and e - learning systems. It has been identified as one of the most crucial issues to be addressed to maintain the quality and effectiveness of the learning/teaching process especially in higher and university education sector. In order to tackle this problem there should be free, efficient and reliable methods to identify the plagiarized versions of documents among the corpus stored in the large document bases in Learning Management Systems (LMS). The main problem which is addressed in this thesis is detecting the plagiarized versions of documents among the submitted tutorials, assignments and other documents by the students in a LMS.

Other than the traditional plagiarism detection approaches a new framework for plagiarism detection is introduced for detecting plagiarism of such kind of corpus in the LMS which covers all the inherent tendencies of the plagiarizer. It is called MAPDetect. The core of this framework consists of several metrics which give more evidence of plagiarism on different types such as verbatim copying, paraphrasing and collusion, structural changes of the content and change of formatting. Algorithms on the document representation are used to calculate the word level correlation among the documents and it is more related to the surface level document similarity analysis. The deep structure of a document such as its, syntactic and semantic analyses are used to detect paraphrasing and collusion. Formatting structure of a document which gives other area of evidence on plagiarism is also emphatically considered. Authorship verification from the field of intrinsic plagiarism detection is also used in the proposed framework. A modular architecture is used for this framework to implement the plagiarism detection techniques with preprocessing sub systems.

Real document sets submitted by university students have been used for testing the improved surface level detection of the framework. The deep level detection is tested with a manually created corpus. The result of the exploratory experiments on proposed algorithms of each module gives promising results. It demonstrates that the integration of several metrics on different areas gives significant evidence to discriminate the plagiarized documents more accurately. In this context the user is provided a great opportunity to obtain more evidence to prove the identification of the plagiarized segments of the documents.

Table of Contents

List of Tables

List of Figures

Chapter 1- Introduction

1.1 Plagiarism as a Crucial Problem in e-learning Systems

At present, a wide range of extended online learning methods have come into existence with the development of communication technology from blogs to collaborative software, groupware, e-portfolios, and virtual classrooms. Hence, the term learning has simply become e-learning. It was introduced by distance learning and flexible learning which enhanced the traditional face-to-face learning or teaching. This broader interpretation sends up e-learning into a wide range of complex applications. Virtual Learning Environment (VLE) with Management Information Systems called Managed Learning Environment is the widely used application and it handles all the aspects of courses within an institution. Universities all over the world and most of the online only colleges as well as employee training institutions are offering their degrees, diplomas and certificate level educational programmes through these applications in a wide range of disciplines.

Students, lecturers or teachers are working collaboratively and learning is a kind of self-assessment for the students in an e-learning system. Students should be assessed by the lecturer or teacher in various ways. For example, this could be an assignment, a multiple choice question, a quiz, a discussion group or a case study. When such schemes are used for the assessment of students, they are given more flexibility or higher capability to copy the materials which they use during preparation for their assessments.

Plagiarism is an unconventional human ability which attempts to imitate other’s writings, thoughts, presentations and concepts closely. “Plagiarism is the appropriation of another person’s ideas, processes, results, or words without giving appropriate credit, including those obtained through confidential review of others’ research proposals and manuscripts” (Jones, 2006), (Hart and Friesner, 2004). Presently, this phenomenon is known as online plagiarism. Since it is very easy to steal content from the web by simply copying and pasting, the problem of online plagiarism has being grown for many years. This phenomenon, also known as content scraping, has affected both established sites and blogs.

Most of the universities and higher educational institutions use Learning Management Systems (LMS) like *Moodle* [www 1]or *Webex* [www 2]that provide facilities to the lecturers to present the assessments online to the students who in turn can submit their assignments digitally to their lecturers.

As it is so easy to locate information from the Internet, students tend to take subject materials for their tutorials, assignments and case studies etc. and use them as their own thoughts. Sometimes they copy the submitted assignments of their colleagues which include the same structure and the same contents. Finally, the same assignment is submitted digitally as their own.

Before mid-1990's, plagiarism seemed to be comparatively rare. However, during recent times, the higher educational sector observed this crucial problem as a world-wide phenomenon within the academic communities. In this respect, plagiarism has now become a serious and endemic problem [www 3]. Studies of cheating behavior in the United States dated as far back as 1940's reported that 23% of the students admitted some form of cheating behavior (Hart and Friesner, 2004). This is academic dishonesty and could be specified as endemic. They also found that three quarters of a sample of 5000 students drawn from a sample of 99 colleges and universities involved in some degree of academic dishonesty in 1964. A more recent estimate has even claimed a figure as high as 90% in American high school students (McCabe, 2005).

On a broad base, two types of cheating behaviors with the aid of computers can be found in the higher education sector. Firstly, the students take the materials from the web and use them in essays or reports without citing proper references but as their own ideas. This is plagiarism. Secondly, some groups of students who are supposed to submit the same topic of assignment may work together, using resources commonly and submit them as independent works of each student which is returned as collusion (Miguel, 2006), (Barret and Cox, 2005) (Carroll, 2004). In both cases, students pass off work of other's as their own. Whether it is plagiarism or collusion, it is tedious and extremely time consuming for teacher or lecturer to detect and mark plagiarism in these assignments. On the other hand, it would legally be a risk to the students and teachers as well.

Considering the above facts, plagiarism can be carried as a rapidly growing problem in many universities today. Any administrator who expects higher standard of overall quality of education and knowledge he should responsible for avoid this phenomenon from their institutions. The goal of this research is to build a framework to reduce and eliminate the impact of plagiarism on educational institutions that use E-learning and LMS.

1.2 Plagiarism Detection Methods

Avoidance of academic dishonesty and provide genuine knowledge to the world is the main idea of eliminating plagiarism from the higher education sectors and the universities. Two main approaches such as detection of plagiarism and prevention of plagiarism can be used to overcome this problem.

Mainly, plagiarism detection can be broadly divided into two as manual and automated detection. Again, there are two possible aspects in manual detection such as using human exposition on past experience and using the existing physical styles of documents. In the former, the teacher presupposes the sections that could have been plagiarized by the students based on his past experience on the particular subject area. It requires more proofing strategy and knowledge. The teachers should have good memorizing capabilities and spot them as “this is familiar to me”. The latter is more objective and depends on various attributes existing on the document such as layout changes, language changes, level of language differences than the oral presentations, same tables, figures, equations and other representations, same spelling and grammatical mistakes, same structural mistakes, unfamiliar references, etc.

Layout changes could be identified by examining the deference between default layout and the existing layout. Since the students are not matured in linguistic features; generally, their level of language manipulation is poor. So, their composition of sentences in the paragraphs will damage or misinterpret or disturb the idea and plagiarism can be easily spotted by the lecturer or the teacher. The vocabulary changes, the changes of the sentence length also are good indicators for identifying language changes. Most students use inaccurate references and the style of this inaccurate reference will lead to identify the plagiarized portions of a document. On the other hand, plagiarism on program source code can be detected by checking comments, memory allocations and other variable usages and the usage of programming structures etc.

Consequently, plagiarism prevention is also been practically implemented in most of the higher education institutions recently. Three major kinds of prevention techniques are been used by most institutions. Firstly, they use individual assignments as possible and concentrate more on the classroom work. Secondly, the students are made with a good awareness on academic honesty policies, on fair and unfair academic activities by explicitly changing their attitudes to minimize dishonesty. Thirdly, they introduce proper rules and regulations with punishment schemes and educate the students. Then the students are able to understand the final results of academic dishonesty.

However, the above tasks are tedious to exercise practically since the number of assignments per teacher may be high and/or the assignments may be very long. Time constraints and other classroom work will affect the teachers' evaluation process and finally the quality of the assessment may become very poor. Automation of plagiarism detection is one of the solutions for maintaining a good assessment system.

Automated plagiarism detection can also basically be divided into two as External Plagiarism Detection (EPD) and Internal Plagiarism Detection (IPD). In EPD there is a set of suspicious documents and a set of source documents. The suspicious documents may contain plagiarized segments and the detection system would determine those plagiarized contents with the help of source documents. The detection mechanism can be one-by-many or many-by-many. In IPD there is no comparison with the source document set and the evidence of plagiarism is revealed through the document itself. The students' writing styles can be compared with the authors' genuine styles to identify the plagiarized passages. The most significant thing is that the documents have to be divided into segments to compare the unequal styles i.e. by detecting the variation of the writing style of a particular document.

EPD yields with four kinds of major approaches such as information retrieval methods, n-gram base methods, fingerprinting and natural language processing approaches (see section 2.5 and 2.6 of chapter 02). All these approaches are deeply considered in this thesis. Yet, IPD is another theme of this study and presently, it is also an experimental area on plagiarism detection (see section 2.6 and 2.7 of chapter 02).

1.3 A New Framework for Plagiarism Detection

After having plagiarism detection systems and approaches been closely studied a framework that can be implemented to accelerate and improve the efficiency of the human plagiarism detecting process is proposed to address this problem. It can be described as a Machine Assisted Plagiarism Detection System (MAPDetect), which encapsulates a framework of most powerful and most applicable approaches. The user is the final decision maker of such a system. All the components of the framework can be divided mainly into few modules to capture human cheating behavior namely, document similarity checking, natural language processing, and authorship verification. The Information Retrieval Models (IRM) such as Boolean Model, Vector Space Model (VSM) and n-gram model are used to calculate the word level correlation among the documents and it is more related to the EPD (see section 3.3 of chapter 03). The deep structure of a document like, syntactic and semantic analyses are used to detect paraphrasing and collusion. Formatting structure of a document which gives more information of plagiarism is also emphatically considered. Authorship Attribution and Verification from the field of computer forensics can also be used in the proposed framework.

The proposed integrated plagiarism detection framework can be implemented in a free and open source platform and is supposed to adhere to e-learning standards adopted by LMS. A modular architecture will be proposed for this framework to implement the plagiarism detection techniques with preprocessing sub-systems.

1.4 Document Formatting Property Analyzer

Time management of most university students is practically poor and assignments are written in the last minute. On the other hand, they are not adequately instructed to use same formatting properties for their normal class tutorials, assignments and other publications. Thus they use the Internet and copy the particular contents or they may copy down the contents of digital copies from other students and finally submit the assignments on time.

Frequently, In the above process, some formatting structures of the documents may not be changed by the plagiarizer and the suspicious documents themselves contain those formatting techniques This study addresses this phenomenon and a new algorithm is introduced to measure the similarities in documents with the help of document formatting properties.

Presently, sophisticated word processors are being used by the students who submit their assignments to the LMS i.e. MS Word. Large numbers of formatting facilities are provided by these software for various word processing activities. Programmatically, these formatting techniques can be extracted explicitly to the document text and it can be analyzed separately in order to obtain the similarity score among the formatting properties (see section 3.4 of chapter 03).

1.5 Intrinsic Plagiarism Detection with Authorship Verification

Typically, plagiarism detection software may consist of tools for relative comparison of documents and make decisions relatively with other documents. This may cause some problems such as, the need for a corpus to obtain similarities and that it should be subjectively related to the entire document. On the other hand, the plagiarizer may use hard copies of some books which are not included in any softcopies of a corpus or the Internet. Then it is difficult to find the document which is exactly used by the plagiarizer. The absence of such tools to cover all the inherit abilities of a plagiarizer is another crucial problem. In order to overcome these problems intrinsic plagiarism detection is proposed in the above framework pertaining to the internal plagiarism detection methods.

There is a significant difference between authorship attribution and authorship verification. Every student has his/her own diverse and individual version of his/her own idiolect. One example is the vocabulary of any person. It may be different from one another. Based on this universal truth the signature of an author can be detected and be made use of for detecting the authorship of a document or at least to obtain the basic semblance of the authorship. This is called authorship Attribution. However, the authorship verification concerns whether the text is written by the same author or not. Here, a critical problem may arise as to how a very small variation can be taken into account to verify the author as the author's shallow changes have to be caught.

Generally, in intrinsic plagiarism detection there should be a machine learning part in order to cluster the different styles of the given attributes of a document. Authorship verification rather than the authorship attribution is the technique used in the framework with Stylometry and an unsupervised learning approach called Self Organizing Maps (SOM) has been used for clustering. In this study several new stylistic features have been introduced as Stylometry features (see section 2.8.4.1 of chapter 02).

The main and the most critical factor of the intrinsic plagiarism detection is the segmentation of documents. It is identified that the performance of the detector directly depends on good segmentation. This thesis introduces a new parameter for improving the segmentation process.

1.6 Research Problem, Objectives and Scope

Even though many plagiarism detection systems are available, they have their own limitations and restrictions. Most of the commercial plagiarism detection systems have enough facilities to detect plagiarism but their services are restricted to registered users. This is not a feasible solution for many educational organizations especially in the developing world. The obvious solution to this problem is relying on open source systems. Unfortunately, the existing open source systems are currently not integrated with the e-learning systems and cannot be used with the purpose of educational activities. In addition, most of these systems consume more computational resources that are not available at the average educational environments of developing countries. Since the commercial products always hide their valuable and efficient algorithms the study had to explore a correct and efficient algorithm for detecting plagiarism to develop plagiarism detection software by improving the existing open source algorithms. To fulfill this requirement it is necessary to conduct a research of this type.

The research aims to build a system which contains proper algorithms to avoid plagiarism in higher educational institutes which use e-learning systems. In doing so, it is expected to use the University of Colombo School of Computing as the base institute to test the framework with its e-learning system and LMS.

The following objectives are identified during the course of the initial stages of the research.

- To identify the issues in plagiarism detection in Information Technology domain.
- To develop/Improve algorithms to detect plagiarism.
- To implement them in the UCSC LMS to detect plagiarism in assignments and theses submitted by the students.
- To define a document similarity measure and evaluate the proposed algorithms.
- To eliminate/minimize plagiarism through the LMS of UCSC
- To contribute the implementation of proposed algorithms to the open source community so that others interested in plagiarism detection can improve the system.

There are several document representation methods available and these can basically be divided into two, namely, VSM and other syntactic and semantic representations.

In a VSM, a document can be represented in terms of its words, substring, or n-grams. It has been reported that substring matching and n-gram matching outperform word based algorithms significantly.

In VSM based methods, the contextual information of words are ignored and this is one of the major weaknesses of these methods. In order to capture this useful contextual information documents can be compared at higher levels such as phrases, clauses and sentences. These can be extracted by utilizing freely available parsers developed for English. Phrase level or sentence level comparison help identifying document similarity more effectively. Further, these syntactic structures can be converted into semantic structures so that the documents can be compared at a more abstract level. However, more computational power is required in order to analyze the syntactic and semantic structures of the given document. It is a crucial problem in plagiarism detection. Intrinsic plagiarism detection is used to fill the gap of these failures. A clever combination of the techniques mentioned above can produce a highly accurate optimal algorithm which can be used for automatic plagiarism detecting.

1.7 Thesis Outline

The rest of this thesis is structured as follows:

Chapter 2 provides the general background of e-learning, plagiarism, plagiarism detection, plagiarism detection tools and different approaches of electronic plagiarism detection while Chapter 3 provides the proposed new framework for plagiarism detection including a new approach called Document Formatting Property Analyzer, outlining the methods which have been carried out for intrinsic plagiarism detection with unsupervised learning. In Chapter 4, details of experiments and the results are presented. The most important conclusions drawn from the results are presented in Chapter 5, together with a summary of the contributions of this work and ideas for future work.

Chapter 2 – Related Work

2.1 Introduction

The literature review map of the research problem is explained by this chapter. Mainly, literature on the plagiarism and its nature in the higher education sector, plagiarism detection tools and approaches, the extrinsic plagiarism detection and the intrinsic plagiarism detection are deeply reviewed in this chapter.

2.2 What is Plagiarism?

The word plagiarism has been derived from the old English word plagiary ('one who wrongfully takes another's words or ideas') and plagiary is derived from the Latin word plagarius ('kidnapper, seducer, plunderer, literary thief'). Plagarius is derived from plagium (kidnapping) from plaga (snare, net) (Barnhart, 1988).

Plagiarism is the unauthorized use or close imitation of the language and thoughts of another author and representation of them as one's own original work. Carroll (2002) defined "Plagiarism is passing off someone else's work, whether intentionally or unintentionally, as your own for your own benefits". Another definition has been given by McNaughton which has a broad illustration of Carrols's definition. He has mentioned "Plagiarism can be defined as the attempt to gain advantage for yourself academic advantage, financial advantage, professional advantage, advantage of publicity by trying to fool someone, such as teacher, an editor, an employer, or reader, into thinking that you wrote something, thought something, constructed something or discovered something which, in actual fact, someone else wrote, thought, constructed or discovered" (Carroll, 2002). "Plagiarism is the appropriation of another person's ideas, processes, results, or words without giving appropriate credit, including those obtained through confidential review of others' research proposals and manuscripts" (Hart and Friesner, 2004). The word plagiarism was defined by the US Office of Research Integrity (ORI) as 'the theft or misappropriation of intellectual property (Anonymous, 1995). Commonly, some authors wrote plagiarism as a malpractice and many others view it as poor practice. (Howard, 2000) said that the plagiarism is a kind of a mental illness.

2.3 Plagiarism in the Higher Education Sector

Eventually, plagiarism is one of the crucial problems in the world and it is continuously growing bigger as an epidemic disease in the academia. Most of the academic institutions, especially universities all over the world have been affected with this formidable problem and it has been proved by several researches. The evidence has been obtained from many countries, including the UK (Ashworth et al, 1997), the USA (White, 1993), Finland (Seppanen, 2002), and Southern Africa (Weeks, 2001). Several academic institutions such as universities with both undergraduate and postgraduate students and some public and private higher education institutions have been covered by these researchers.

According to the past evidence of the world literature, journalism, politics and science, academia should not be taken by the monopoly of the plagiarism. Several most famous authors of the world also have been accused of plagiarism. Embracing William Shakespeare, Mark Twain, George Orwell, Alex Haley, Samuel Beckett, and Edgar Allen Poe, Song writers like Celine Dion as well as Michael Jackson and film directors such as Steven Spielberg were punished by the court on plagiarism cases (Chris, P. 2003).

Plagiarism may occur in several ways among students. It is mostly seen as copying necessary paragraphs or the whole document from other sources and using them as their own thoughts. The source may be the essay bank, term paper mill or any other knowledge base from the Internet, from the hard copy of a book or a paper, or a soft copy of any other student. Attempting to submit tutorials or assignments of other students as their own work is another form of plagiarism mostly among the university students. Some students use other’s ideas without quotation marks or proper citations of the origin to make such portions to bloom as their own thought. Some students paraphrase documents or hard copies of other students and trading it as their own thought (Chris, 2003). Thus, the act of cheating with the aid of computers can be broadly divided into two categories in the higher educational practice of those students. Firstly, it is observed that their practice is reproducing the materials of the web and willfully using them in their own tutorials, assignments, study papers or reports to appear them as their own thought. This is nothing other than plagiarism. Secondly, if the student is supposed to submit an assignment individually on a common topic for the whole class where they are supposed to work together using learning resources commonly and submit them as independent work. It is collusion. In both cases, students pass off work of other’s as their own. It is a very difficult and extremely time consuming endeavor on the part of the teacher or the lecturer to detect and mark plagiarism manually to judge whether it is plagiarism or collusion.

Conversely, some legal aspects are established by most of the higher education institutions on such cheating behaviors. Hence, the teachers or lecturers are expected to shoulder more responsibility to catch the students who may have been involved in cheating intentionally or unintentionally. Thus, it will be a risk legally to both students and teachers (Miguel, R. 2006), (Barret and Cox, 2005) and (Carroll, 2004).

2.4 Categorizing Plagiarism Detection

Figure 2.1 shows a wider abstract categorization of plagiarism detection which is identified in the literature review.

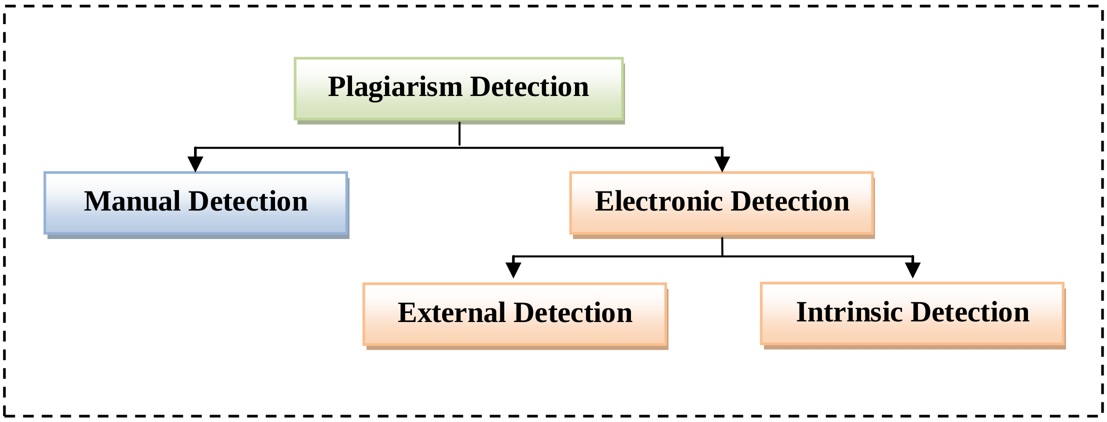


Figure 2.1: Categorization of Plagiarism Detection Methods

Integral manual detection of plagiarism is a very labor intensive and time consuming effort and, it is obvious that it is definitely inefficient. If a lecturer or a teacher attempts on manual detection of plagiarism he/she should examine a large number of documents. Conversely, a person who involves in this effort should possess a good knowledge of such documents as well as they should have practical experiences of the behavior of such students.

Electronic plagiarism detection is a new development of the information technology which is known as Automatic Plagiarism Detection. Generally, the goal of the automatic plagiarism detection is to automate the above task and hence, it should be the identification of the plagiarized segments of a suspicious document electronically without human intervention. More practically, human beings are the decision makers in these kinds of systems and all detection measures are made by the computer which will support to make such decisions.

There are two major distinctions of electronic detection devices, namely, extrinsic (external) detection and intrinsic (internal) detection. External plagiarism detection is more related to techniques of information retrieval, such as vector space model, n-grams and fingerprinting. Intrinsic plagiarism detection rather depends on analyzing the variations of the Stylometry features of a document and authorship verifications (see section 2.7 and 2.8).

2.5 Plagiarism Detection Tools and Algorithms

Automatic plagiarism detection has been growing through several decades and it has not yet reached maturity. Researchers who have struggled with the plagiarism as a worldwide have designed various tools and algorithms to detect them. It is essential to study the literature of previous studies in order to carry out more reliable and sophisticated algorithms or tools. Mainly, the plagiarism detection tools can be divided into two, such as program source code plagiarism detection tools and natural language plagiarism detection tools (Jurriaan, et. al. 2010).

Although the research concerns on natural language plagiarism detection, especially English, it is important to mention the tools of source code plagiarism detection. JPlag is a web based system which is especially used to detect program source code plagiarism by uploading the documents to its website [www 19]. It presents results in HTML format and can input C, C++, Java, and C# source codes. Greedy String Tiling is the base algorithm used by this tool (Prechelt, et al. 2000). MOSS (Measure of Software Similarity) is another sophisticated web based source code plagiarism detection tool which supports a wider range of computer languages than JPlag such as C, C++, C#, Java, JavaScript, Python, Visual Basic, FORTRAN, ML, Haskell, Lisp, Scheme, Pascal, Modula2, Ada, Perl, TCL, Mat lab, VHDL, Verilog, Spice, MIPS assembly, a8086 assembly, and HCL2. Winnowing algorithm which is based on fingerprinting approach is used by this tool. The output will be presented as HTML and there are links to the user to obtain more information [www 20]. Plaggie is different from the above tools and it is an open source. It should be installed locally to the machine. It can accept only java source code to detect plagiarism and will give plain text output about the result. Plaggie also uses Greedy String Tiling algorithm and does not use any optimization algorithm like JPlag. SIM plagiarism detection tool can be used to detect plagiarism on both programming source code and natural language texts (Jurriaan, et. al. 2010). C, Java, Pascal, Modula-2, Lisp, and Miranda are been accepted by SIM tool as computer languages successfully. The tool is launched by command line arguments and the output is written on text file.

Plague was built to detect these reusing or copying the program codes. Plague consists of three main phases such as (Prechelt, et al. 2000):

1. In the first phase, a sequence of tokens is produced for each file, as well as a list of *structure metrics*, reworked as a *structure profile*, which summarizes the structures used in the program. The component structure metrics represent iteration and selection statements, and statement blocks.

2. In the second phase O(n^2), the structure profiles are compared and pairs of nearest neighbors are determined using a combination of language specific distance functions. It is expected that at the end of this phase, the majority of submissions will remain unpaired; if any paired submissions remain, they move forward to the next phase.
3. In this phase, the token sequences are compared using a variant of the longest common subsequences algorithm.

Plague endows a number of problems:

1. Plague is currently applicable to programs written in Pascal, Prolog, Bourne Shell and Llama (a Pascal translator generator); writing versions for each new language requires considerable effort, starting with the construction of a parser for the target language and the selection of distance metrics to be used in the second phase.
2. The results are returned in the form of two lists which are ordered by indices, H and HT, which need to be interpreted. The Plague manual provides guidance on how to do this.
3. Parts of Plague are written in Pascal, and while good quality C implementations are common, good quality Pascal implementation is rare and implementation is dependent on features abound.

Yet, Another the Plague (YAP) moves through two phases such as generation phase which converts the file into tokens and the other which creates token files for each submission. In the second phase the pairs of token files are analyzed and finally these files are compared. While the files are being tokenized the inbuilt functions, the language structures, identifiers etc. are removed.

In other words, YAP is able to counter (or at least limit the effect of) all the ploys used by students to disguise the copied work. In general, it appears that YAP is at least as accurate as Plague in detecting significant similarities and at least as good as Plague in avoiding finding matches where plagiarism is non-existing (remembering that both are a sliding scale). YAP is more portable than Plague, and its output is easier for users to understand. In terms of each system’s performance, YAP is clearly much slower than Plague, which may prevent its use for large classes.

GLATT is another stand alone non-web based plagiarism detection tool which is used in a different way to detect plagiarism. It accepts the student’s assignments or reports and makes standard sizes of blank words. Students are requested to fill up the blank word. The time consumed for their responses, comparisons with the correct answers and some other factors will be determined in calculating the plagiarism index for the submissions and finally give the ranked result [www 8].

The SCAM tool of plagiarism detection uses common keywords from the document by scanning the union of word sets. Count of these key word occurrences are used to detect plagiarism. This system searched thousands of DBWorld research papers and detects some serious cases of plagiarized documents (Shivakumaran, 2003).

The SNITCH (Spotting and Neutralizing Internet Theft by Cheaters) which uses an algorithm with the Google Web API has advanced ability to detect cut and paste plagiarism. Sliding Window Approach is used by this system. Firstly, it obtains the windows one by one from the document and secondly, it reads each window containing words and measures the number of characters per each word and thirdly, uses the average number of characters on which the algorithm assigns a weight for the window and memorizes it. The system repeats this procedure for all windows. When one searches for plagiarism he should first eliminate the overlapping windows. Then all the windows should be ranked in the order of weights and the top weighted windows are selected for searching in the Internet. The tool gives good performance on the tested document set without any false positives and the comparison with EVE tool also is made use of in the research (Sebastian and Thomas, 2006).

CHECK anti-plagiarism detection tool uses a different method for detecting plagiarism with recursively applied information retrieval techniques and finally tries to extract semantic meaning of the texts. According to the comments given by the authors, the method may give good results since the processing time is reduced by eliminating the most unnecessary text segments. Document recognition, keyword extraction, and generating the structural characteristics are the key features of the tool (Antonio et. al. 1997).

TurnitIn is the leading plagiarism detection software in the world and it is a web-based system. In 2007 there were more than 50 countries that used this system which can be used individually, department wise, by a single university or by many universities. Turnitin has more than 10 million pre- submitted papers as its database and it is updates more than 40 million web pages per day. There are more than 10 million users using Turnitit all over the world and more than 20000 papers are processed per day. The users are provided with various facilities for identifying the plagiarized segments in the documents as matching percentage, links, highlights of the plagiarized sections, sources of the plagiarized documents as well as printing facilities. LMS, which are called Blackboard, Angel and Moodle can be integrated with this tool and should pay extra payment for integration. Turnitin provides excellent facilities for training and helping on using it with video and printed materials. Online training programs are also included [www 4].

MyDropBox, also, is one of the web based plagiarism detection systems which reads nearly 10 billion documents from MSN. Both Academic and non-academic institutions are serviced by this tool and all the reporting facilities in Turnitin are also included. Online training with animations and onsite training are included in this system. It can be integrated with Blackboard, Sakai, and Moodle LMSs. More than 300 institutions from 30 countries covering more than 2 million users have registered for this tool. Especially, this tool supports to detect plagiarism on more than 150000 papers from paper mills [www 5].

WCopyFind is another tool and it is free software developed by the University of Virginia in 2002. This tool can be used for detecting plagiarism of a submitted document set. The user should provide the document set to the system and the documents are checked against each document of the provided set. A HTML report will be generated by the system with matching phrases underlined. It has only online FAQ as help for the users [www 6].

CopyCatch is commercial downloadable plagiarism detection software and it can be used department or individual wise. Google API is used by the tool for web plagiarism. It is the same as WCopyfind and the user must submit the documents to the system. The tool includes the side by side comparison of the document and can save the reports as requested by the user. The company provides one day training on the tool as requested by the registered organizations [www 7]. GLATT plagiarism detection tool built by Dr. Babara S. Glatt in 1987 is distributed as CD copies. It is the same as WCopyfind and the users are provided only with the final result of the detection. The software can be used individually and no supportive facilities are available [www 8].

Urkund is also an anti-plagiarism software which can be used on any plagiarized text with more than 400 characters. This is a commercially available web-based system and all the administration work is handled by the company. No web or any other server is maintained in the registered organization and all the security features on documents are handled by the company. It uses e-mail to upload the student’s documents and the results also will be received by e-mails. According to the company “Urkund was born from the academic world. A team of teachers developed the idea of a web based service that would help them detect and deter plagiarism and Urkund was born in the fall of 2000. The problem of plagiarism received much attention in the media and more and more began realize the scope of the problem and the need of a tool to support the pedagogical work. Urkund continued to grow and develop over the years and came to be recognized as Sweden's foremost anti-plagiarism service." Urkund uses several data sources in order to detect plagiarism. Such as the Internet (over 10 billion web pages), published material and previously submitted student texts (over 2 million) are the three sources. The source area contains scientific and popular articles, references books and databases and more. It is also possible for clients to add their own sources, such as internal databases to the plagiarism checks. When using Urkund program vendor is responsible for all detections. All the students are given ID number for authorization to submit their submissions. This ID is generated by the lecturer on given interface to him. Students should log in to the system by using the given number and then submit their assignments to the system. They are stored at the company and the lecturer can obtain the plagiarized documents on his/her interface. The system changes the color of the text portions in order to indicate the plagiarized documents. To identify plagiarism, the number of hits in the document will be counted and that will decide whether the document is plagiarized or not. Another tool called Genuine Text also provides same facilities as Urkund does and both tools are mostly used in Sweden Universities with Sakai LMS. Plagiarism.org [www 8], Paperbin [www 9], Text Ranker [www 10], FindSame [www 11], HowOriginal.com [www 12], Plagiserve [www 13] are other commercial and free web based software available for plagiarism detection.

2.6 Different Approaches of Plagiarism Detection

Water Mark based approach is one of the passive copy detection approaches (Hiary, 2005). The original document contains a water mark on it and the original user of the document is recognized by that mark. There are some drawbacks of this approach, as well. As the watermark of a document can be either deleted or corrupted by applying compression techniques, one can reproduce the same document without its original watermark. Furthermore, if one copies a portion of a document without inserting the watermark this approach will not be able to detect plagiarism (Hiary, 2005).

From the beginning of the information era storing and retrieving information become a highly necessary requirement. Facilities of storing very large amounts of information are rapidly increased after the invention of computers and, again, searching for necessary and useful information from such collections also has become another requirement. The field of such kind of activities has been opened by Vannevar Bush in 1945. He tried to access a large amount of information automatically. In 1957 the first algorithm was explained by H.P. Luhn and he proposed indexing the terms and measuring the word overlaps can be used as a searching technique (Singhal, 2001). During the last fifty years the field of information retrieval has been considerably developed by thousands of researchers all over the world with sophisticated and efficient algorithms. These improved algorithms have also been employed as searching mechanisms of the World Wide Web from 1990's and now it has become matured.

Document ranking with information retrieval (IR) is one of the popular approaches of plagiarism detection. These IR models basically depend on the given query and there are two categories of this approach; conventional and non-conventional. The former is based on Boolean model and the latter can be explained with probabilistic model or VSM or by using fuzzy matches (Salton, 1983). Generally, there are three steps to be followed to make a raking result. On the first step, the document collection is indexed and on the second step, the query is processed with each document of the collection with computing weighted values in order to obtain the similarity matrices. Finally, the documents are stored in descending order and the topper most ranked documents are selected as identical to the query document.

The Boolean Model is a simple IR model which is based on a set theory and Boolean algebra. It is a non-weighted approach of information retrieval and is easy to implement. The main drawback of this algorithm is that too few or too many documents can be given by exact matching of terms and finally it will be difficult to ranking the results. However, it may be useful in some types of plagiarism detection like copy paste plagiarism. In the VSM a document is represented by a vector and each item of the vector is a term of the document. Each item of the vector is assigned to a value and this value is a function of term frequency in which such term occurs in the document collection. It gives the importance of the term which associates with the given document.

The VSM has very important properties. The document vectors can geometrically be compared by using their angles and it is easy to measure the similarity among documents. Cosine is used to quantify this correlation in normalized forms. If the angle of the two vectors d and q gets high, the cosine value is related to 0. It indicates that the documents are totally different and the correlation will be very low. If the angle of the two vectors is identical, it has a higher correlation and the cosine value reaches 1. The property of finding correlation among given documents and the query document is utilized with the intrinsic plagiarism detection by (Mario et. al. 2009).

Another partial copy detection approach is document Finger Printing. Dividing the document into Ngram and assigning hash values is the first step. Then an algorithm is used to select the hash values as fingerprints and finally evaluates the hash value for each Ngram of the two documents to compare the documents for detecting plagiarism. There are several algorithms used to select the hash values to represent fingerprints. Shivakumaran (2003), Schleimer and Wilkerson (2003) guaranteed that the winnowing algorithm is efficient and matches of the certain lengths are detected. However, document fingerprinting approach does not consider the behavioral patterns of the plagiarists (Heintze, 2000). There are several algorithms for dimensional reduction in fingerprinting and the Winnowing algorithm is used successfully. (Schleimer and Wilkerson, 2003). In this algorithm, all hashes of n-grams are divided into windows of size n and value of n is assigned by the user. A guarantee is given by this algorithm for at least one fingerprint selected from every window. It means that at least one n-gram is selected by every shared substring of length $n + k - 1$. According to Schleimer and Wilkerson (2003), the method of selecting hash values from window after defining the window size n is “From each window select the minimum hash value and if there is more than one minimum hashes, select the rightmost occurrence”. They have proven that this algorithm is more efficient than other algorithms by using 20000 web pages. This algorithm was used to select the hash values for normal documents created by the students. The implementation of algorithm was extended more efficiently by hash table data structure.

The behavioral patterns of the plagiarists have been considered in the Multi-Level Text Compression approach. The core idea of this approach is Levenshtein distance. It is based on analyzing the document structure rather than specific word or word frequencies. Plagiarists can do three things like insertion, deletion or addition of some information. This approach uses this phenomenon to model the behavior patterns of the plagiarists. The algorithm marks the similarities according to the minimum distance which is affected by primitives such as insertion, deletion, substitution. If the edit distance generated by the plagiarism function on the above primitives is greater than the threshold the document will be a plagiarized one. This property can be applied not only on the word level but also on other levels such as paragraph and period levels. This approach uses recursive plagiarism functions to identify plagiarized documents and the chunk pairs are selected by using the given criterion in order to minimize the time consumed on unnecessary comparisons among documents (Manuel et al, 2006).

According to Sebastian and Thomas (2006) there are 40% students recently involved in cut and paste plagiarism with their studies with at least one writing assignment and 77% do not feel that it is a serious illegal action. This type of plagiarism can be detected by manually copying the area to search engine and search the plagiarized documents in the web. It is a tedious and time consuming effort. Mostly, in technical papers and in most scientific papers it is more difficult to find out the plagiarized portions because the most technical abbreviations are the same. They also discussed the cost and time consumption of some commercial plagiarism detection tools which are difficult for the ordinary users.

- The EVE2 software costs $29.95 for license and takes from 2 to 45 minutes to scan a typical 5-7 page paper, depending on the scan type.
- TurnItIn spends four to six hours for one submission and the cost for annual membership is $3000 or a license fee and per-student charge ($530/year plus $1/student)
- MyDropBox services the same to TunItIn and within 24 hours the reports are provided for one submission.

The authors of the paper consider and address some technical issues when they designed a plagiarism detection tool. The identification (concern on vocabulary of students), thoroughness (concern on the degree of plagiarism) and the usability of detection tool by both students and the instructors are their three major concerns.

Variations of greedy matching algorithms are used by most detection systems. For example, Running Karp-Rabin Greedy String Tiling (RKRGST) is used in YAP3, JPlag, Plaggie, and a similar approach is used in FPDS. These algorithms appear with some heuristic values such as the minimum length of matching substring etc. This takes some time to increase the false rate and hence the reliability is low. This limitation gives inaccurate results and sometimes the plagiarized document may be indicated as a non-plagiarized document. Consequently, the plagiarizer may swap the text into different positions, and then the system may not be able to understand that it is a form of plagiarism.

Most of the plagiarism detection approaches are based on file-to-file comparison methods and usually string matching algorithms are implemented to get the correlation among documents and ranking the documents in order to get higher similarities. These approaches give significant performance on plagiarism with direct copying sources to the suspicious document. However, it does not concern the internal structure of the document and hence, string similarity approaches are unable to determine the attempts of plagiarizer which are related to paraphrasing (Chi-Hong and Yuen-Yan, 2007).

Generally, plagiarizer can rename the variables, change the program control structure, and modify the lexical structure. Some techniques like tokenization, parameterized matching are used to prevent this. Such techniques may not appropriate for documents which are created by using natural languages. Mozgovoy et al. (2007a) presented how to use Natural Language Processing (NLP) in order to overcome the hiding behavior in plagiarism. Consequently, his findings have been illustrated how to detect split matching in plagiarized documents.

All the natural language sentences have syntactic and semantic structures. NLP parsers can detect and divide natural language sentences into syntactic structure of a sentence and it will get the actual idea of the sentences other than the order of the words. This utility can be used to detect plagiarism more accurately. In order to experiment, they use *probabilistic context-free grammars* (PCFGs), with *Stanford Parser.* This parser uses a *Cocke-Younger-Kasami* (CYK) search algorithm and can output both dependency and phrase structure analyses. Klein and Manning reported labeled precision and recall figures of 86.9 and 85.7 respectively for this parser. They use Java transformation of Stanford Parser output to get grammatical dependencies into alphabetical order.

There are some technical issues. Natural language parser uses ordinary text documents and on the first step it creates the parsed file. On the second step this parsed file is used by the plagiarism detection software to detect the plagiarism (This will allow checking the efficiency of deferent parsers and preprocessors). However, the basic drawback is that parser attempts to change the word order of the sentences and the detector cannot identify the position or sentence block of the original document. Two methods are suggested to be adopted to overcome this drawback. Firstly, the system should highlight the entire paragraph other than the word chain and secondly, the parser should maintain metadata of the document. To evaluate the system they use 128 BBC News Massages and divide those massages into four categories as *Business*, *Europe*, *Science/Nature*, and *Technology*. The median size of each message (after removing all formatting) is about 2 KB. They also use several files with plagiarism including copy & paste with subsequent change of words and phrases.

In practice, it is possible to get rid of incorrectly matched pairs by raising a similarity threshold for the final file pair list. For the plagiarized free-form essays the similarity ratios have been increased significantly up to 50%-80%. The results are also noticeably affected by the value of the “shortest string length to match” constant. The smaller the constant is, the lesser is the effect of the use of the parser. High constant values cause higher effects. This experiment shows that the technique is more effective on intentional plagiarism other than the normal similar documents. If there are more swap words it gives higher accuracy. Since the copy & paste plagiarism contains more swapping of words the plagiarizer will motivate to hide the copy & paste plagiarism by using this method. Swap is a good parameter of plagiarism to consider when developing a system and a wider difference in numbers may indicate the presence of intentional word swaps, and, therefore, of plagiarism (Mozgovoy et al., 2007 b).

Most of the plagiarism detection systems use some methods of similarity detection. Hence the quality of the system depends on the selected method and the similarity calculation. If the detection is fast, less precise result will be provided. Mozgovoy et al. (2007b) represented a fast and reliable approach and offline plagiarism detection was concentrated with it.

Fingerprinting is a rarely used system and it deals with special fingerprints such as average line length, file size, average commas per line etc. Most newly implemented plagiarism detection systems use content-comparison techniques and parse trees which detect similarity on file pairs. This approaches consume $O(f(n)N^2)$ time to perform the detection. N is the number of file in collection and f(n) is the time needed to compare the two files length n. The use of Suffix Arrays FPDS will improve the algorithmic performance. Search routine also will be improved as special heuristic search routine. The complexity is $O(nN\gamma +N^2)$ time. Where N is the number of the file in collection, n is the average file length and γ is a special fine-tunable constant.

Plaggie in JPlag project uses Running-Karp-Rabin Greedy-String-Tiling (RKR-GST) algorithm which is used in YAP3 tool. Empirically in most cases FPDS gives reliable results of plagiarism detection. FPDS uses two steps Firstly, Using RKR-GST it searches easily detectable matches and skip those matches. Secondly, the algorithm does not consider finding continuous matches, so the similar chunks can be uniformly spread inside the files which are being analyzed. Plaggie is highly reliable and result reporting capability and FPDS are speedier than Plaggie. The approach combines these two characteristics together. FPDS gives output as lists of file-to-file similarity and generate special documents. Then Plaggie uses such documents to maximize the similarity within a given threshold.

Two assumptions can be established by using this combined approach. Firstly, the combined system should be noticeably faster than Plaggie and secondly, FPDS generally should not exclude files from the input set. The experiments prove these assumptions are correct. Finally with empirical results the approach concludes that the higher correlation among Plagie and FPDS and also FPDS is a good filter for Plaggie (Mozgovoy, et al. 2007b).

The main objective of the report which was written by Clough (2000) is giving a comprehensive introduction of plagiarism and introducing the plagiarism tools available to avoid this crucial misconduct.

Among students there are two types of plagiarism as collusion and direct plagiarism, sometimes word-to-word, yet, sometimes paraphrasing (Carroll, 2002). The project called METER is assigned for detecting paraphrasing. In Universities, plagiarism detections are difficult since the lecturers have too many assignments to mark or different academics mark the assignments or universities keep the student work as confidential things. And, on the other hand, plagiarism is difficult to prove because the texts concerned are written under one topic and the most linguistic words like in English words look alike as high as 50% of the vocabulary even in paraphrasing. Hence most of the systems used in detecting some unusual patterns like long sentences, use of common propositions, same spelling mistakes, some identical comments etc.

One of the plagiarisms avoiding effort is to teach how to cite the copied items and to implement the rules and to govern the rules correctly. An automated program for teaching students is GLATT plagiarism teaching program. Several important plagiarism detection methods for written texts are suggested by the author. Uses of vocabulary, changes of vocabulary, incoherent text, punctuation, amount of similarity between texts, common spelling mistakes, distribution of words, syntactic structure of the text, long sequences of a common text, order of similarity between texts, dependence on certain words and phrases, frequency of words, preference for the use of long/short sentences, readability of the written text, dangling references are some suggested aspects to consider.

The determination of author style on one or more of the above mentioned characteristics is of vital importance in detecting plagiarism. The system called Computational Authorship Attribution caters to this phenomenon and the following statistical techniques are suggested by them:

- the average length of sentences (words),
- the average length of paragraphs (sentences),
- the use of passive voice (expressed as a percentage),
- the number of prepositions as a percentage of the total number of words,
- the frequency of “function words” used in each text

Clough (2003) presents some important ideas on plagiarism and their detection. He concerns plagiarism detection as a problem to be solved; not to cover other aspects of plagiarism, how much ever important they are, such as: surrounding ethical and moral issues, suggestions for practical steps that the individuals or institutions can take to detect, reasoning behind student plagiarism, or guidance for writers on how to prevent themselves unintentionally plagiarizing their sources. Academics motivate students and researchers to do academic work that depend on the others’ ideas. But it seems that they do not teach them on fair citations or acknowledging the others’ ideas or materials.

The detection of plagiarism varies from other forms of text analyses like authorship attributions based on lexical and syntactic classifications. Plagiarism detection is sometimes like authorship attribution but is deeper than the information retrieval. It concerns the contents other than the author and his lexical and syntactic measures. The researchers should consider two aspects of such as detection within a single text or between multiple texts. The research addresses three areas of automatic plagiarism detection other than the cut-and-paste or simple rewriting such as finding suitable discriminators of plagiarism which can be quantified and develop suitable methods to compare those discriminators and finding suitable measures of similarity. Detecting plagiarism in natural languages is more difficult because of (1) ambiguity, and (2) unconstrained vocabulary. Clough (2003) suggests that all the approaches such as file comparison, information retrieval, and authorship attribution, compression and copy detection be applied to overcome these problems. He recommends five research areas - Multi-lingual detection, a text collection for plagiarism detection, use of natural language processing, use of techniques from machine learning, detection within single texts, and comparing texts using Dotplots.

Viper is another free and online plagiarism detection tool. It provides some capabilities including scanning a document with a document set created by the user, scanning documents against billions of web pages, detecting copied contents from online sources such as books and journals, comparing documents with the millions of students essays provided by the company. Further, it highlights the actual plagiarized segments of the document and it is easier to the user [www 21].

Most of the plagiarism detection systems use file-to-file comparison with words. The decision whether a document is plagiarized or not is based on a count of lexical similarities. The problem is, if the plagiarizer intentionally changes the words (not meaning) the system does not identify it as a plagiarized document. Chi-Hong and Yuen-Yan (2006) use NLP in order to overcome this problem. They identify some common situations in plagiarism. Copying from non-electronic sources cannot be identified by the automatic detection. However, some situations like copying from web, changing the voice or tense of the sentence structure, and applying synonyms can be detected by the automatic detection systems. Conversely, normal string matching techniques are unable to determine this type of plagiarism because string matching uses the surface structure of a sentence other than the internal meaning. A word of a sentence only has lexical meaning and according to their meanings they can be categorized into Nouns, Verbs, Adjectives, and Adverbs. WordNet represents the meaning as related to the particular word. Nouns include Hypernyms, Hyponyms, Coordinate terms, Holonym, and Meronym. Verbs contain Hypernym, Troponym, Entailment, and Coordinate terms. Adjectives have Related Nouns and Participles of verb. Finally, Adverbs have Root Adjectives. Their detection of word replacement by the plagiarizer is based on such concepts. They suggest context free grammar and semantic representations of sentences as powerful techniques in NLP in order to identify similarities of documents as well as finding out paraphrasing.

2.7 Authorship Attribution and Verification

Authorship attribution and verification was born with the linguistic investigations on authorship or forensic purposes (Coulthard, 1993). It is one of the oldest and newest problems in information retrieval (Juola, 2006). Mainly, the task of authorship attribution is to identify the author of a given text. Finding which author from a_1, a_2, … a_n writes the document d is the main procedure of the authorship attribution. Using a set of another known documents of a particular author and identify his idiolect and it is used to classify the author of another text is the manual procedure of the authorship attribution. Currently, the methodologies of statistics are applied broadly in authorship attribution. The approaches which are used in this field can be broadly divided into three as unitary invariant, multivariate analysis, and machine learning classifications. All those approaches add some portions of development to become the modern technologies of authorship attribution and verification. Similarly, the modern technologies and algorithms of computer science with very large corpuses also have affected to develop the techniques of authorship attribution via an information retrieval background. Another sophisticated approach called corpus linguistics also has affected this development. The corpus linguistics authorship attribution can be defined as presuming the attributes of an owner of a piece of linguistic data (Juola, 2008). Computer assisted authorship attribution aims to classify documents among authors using training linguistic data from documents and identify authors of a given document set automatically.

In the means of computer science, there is another deviation of document clustering with pattern recognition contrariwise; authorship verification concerns mainly determining whether a particular text is written by the given author or not and rather does not attempt to determine the ownership of a particular document. The main idea of authorship verification is concluding whether the document d has been written by author a and whether author x also has involved in writing document d. It is essential to identify the very small changes of the author's style in authorship verification than the procedure of authorship attribution. Consequently, it is typically a classification or pattern recognition which depends on finding the discriminate features to identify the particular author clearly.

Two different approaches called writer-dependent and writer-independent have been introduced by researchers for authorship verification. Former is the standard approach and a large number of sample datasets has been employed to generate a specific model for the author. The main drawback is when dealing with a different author, the classifier should be trained with a new sample every time. Otherwise, a reliable model will not be generated by the classifier. Generally, in practical situations, if enough samples are not there for a particular author, then some classification errors with less performance will be presented. Modeling probability distributions of classes in the classification is the major task of the writer-independent approach and hence, the classifier gives the verification according to the predefined probability distributions. Pattern recognition also has been used for authorship verification under the writer-independent approach and it will be a good opportunity to reduce the n - class problem to 2- class problem such as genuine document and forgery (Daniel, et al. 2008).

2.8 Stylometry

2.8.1 What is Stylometry?

The study of analyzing the unique styles of linguistic and idiosyncratic writing behavior of an individual person is the primary goal of the Stylometry. The basic assumption of Stylometry is the "core of author's style can be quantified" and those are the discriminators of a mixed origin. Currently, Stylometry is being used in various fields such as literary works, authorship attribution, authorship verifications, music lyrics, music melodies, paintings, forensics, plagiarism, electronic mail, instant messaging, identification of terrorism, finding the origin of computer viruses etc. Several tools have been utilized in doing such verifications on Stylometry, particularly identifying unknown e-mails and massages on chatting and blogging systems, identification of authors' signatures etc. However, the most popular area where most researches have been done on Stylometry is authorship attribution and verification.

2.8.2 Related Work of Stylometry on Authorship Attribution and Verification

The history of Stylometry started from the 14th century. In 1439 Lorenzo Valla proved that the Donation of Constantine was a forgery and it was not a book pertaining to the historical era

(4th Century) and it was the book of 8th century [www 14]. The first approach of statistical and mathematical analysis appeared in literature written by Thomas C. Mendenhall, an American physicist in the 18th century [www 15]. He used the word frequencies of one letter, two letters and so on which called word spectrum to analyze Charles Dickens' Oliver Twist and William Thackeray's Vanity Fair. The true mathematical approach called Stylometry was invented by Wincenty Lutoslawski and he used 500 numerical attributes to analyze Plato's Dialogues to distinguish their chronology. At the beginning, analysis of frequency spectrum of simple words such as pronouns, conjunctions, prepositions, and so on were the heart of the Stylometry and afterwards machine learning approaches have been used by most researchers. Juola (2008) has suggested that the previous work on Stylometry in authorship attribution can be divided into three categories such as classical approaches, federalist analyses, and controversies: like Qsum and the Elegy and, Foster and the Elegy. Mainly, the classical approaches comprise with the work done by Holmes and he proposed that “word-length might be a distinguishing characteristic of writers”. Most of the classical approaches tried to create authorial fingerprint. However, these studies at large were not successful, so, later, other statistical approaches have emerged including average sentence length (Yule, G. U. 1938), Yule’s “characteristic K” (Yule, 1944), other measures of “vocabulary richness” such as Simpson’s D index (Simpson, 1949), an average number of syllables per word (Fucks, 1952), distribution of parts of speech (Somers, 1972), type/token ratios (Tallentire, 1976), average word length (Kruh, 1988). Reliably sufficient results have not been provided by many of these classical approaches (Juola 2008), (Stamatatos, 2009). The federalist analysis was started by Mosteller and Wallace. They tested a new approach called synonym pairs of federalist documents. However, they understood that the method would not give proper results with the use of function words as features to categorize the federalist papers. (Mosteller and Wallace, 1964). Most of the post-federalist analysts have followed the method of Mosteller and Wallace and the federalist papers have become a milestone of authorship attribution. The next era began with graphical representation of correlation between the characteristics by using a technique called QSum which is an abbreviation of cumulative sum (Farringdon, 1996). But the approach was not accepted mostly since the obtained result did not provide good shelter and the researchers who involved with this approach had to undergo severe criticism.

Vocabulary is a good cue for identifying the writer of a document. Some special words in the document tell not only about the period of the document but also exhibit the group or the country of the author (Johnson, 1996). However, Juola (2008) mentions that this kind of approach is more problematic on two reasons that data can be faked and the particular word may not appear itself in the most documents. In order to overcome this problem he has suggested calculating a large scale of simple statistics and vocabulary. Since every word of the document has simple measures like length, syllables, language of origin, part of speech etc. it is possible to have more sophisticated measures which simply calculate the word distribution like Zipf distribution.

Another variation of the measures which have been used was vocabulary richness measures as style markers on authorship attribution. These measures attempt to quantify the vocabulary of the document according to the density of various functions of features. The type/token ratio, once occurring words V1 (hapax legomena) or twice occurring words V2 (hapax dislegomena) are usually used metrics (Honor´e, 1979), (de Vel, et al. 2001). The problem of these vocabulary richness measures is that they depend on the document length and present unreliable results for short documents. Most researchers have suggested various functions to avoid this dependency on text length. Flecsh Index which is calculated by using average sentence length and the number of syllables per word is useful for measuring the easiness of the reading text (Clough, 2000). If this index is higher in value it denotes that the text is easy to read. Kincaid is another index which uses same data to calculate the index. The FOG Index is another variation of readability score which further uses the complex words which have more than three syllables (Tweedie and Baayen, 1998), (Johnson, 1998). The other resolutions of these kinds of measures are SMOG' formula, FORCAST formula and Fry Readability Graph (Johnson, 1998). The Average Frequency Class measure has been used by Mayer and Stain which calculate the vocabulary of the author and it does not depend on the length of the document. A document's average word frequency class tells the style complexity and the size of an author's vocabulary. It has very less variance between document lengths (Meyer, et al. 2007). Arriving at more and more electronic document bases and media, researchers attempt to introduce more sophisticated and computationally complex approaches such as using part-of-speech tags with syntactic and semantic analyses (Stamatatos, et al. 2000), (Kim and Walter, 2008) in more advanced, applied Ngram based approaches to obtain more accurate results. N-grams of syntactic labels from partial parsing have been used as features of authorship attribution by Ol'ga Feiguina and Graeme, (2007). Although more reliable results have been given by these approaches the measures of those approaches are more computationally complex and cannot be used for general purposes. Broadly, two types of such measures can be defined in the above approaches including application of specific measures and structural measures (Stamatatos, et al. 2000). All those measures pertaining to the lexical, character, syntactic, or semantic are application specific measures and greetings, farewell, indentations, length of paragraphs, font type, font size; font color, etc. are the structural measures (de Vel et al. 2001). However, modern Stylometry on authorship attribution and verification going towards machine learning accompanying the above mentioned more reliable lexical, syntactic and semantic features.

2.8.3 Style Markers

Basically, five categories of style markers have been categorized by Stamatatos et al. (2000) including lexical, character, syntactic, semantic and application specific. Lexical style Markers mostly depend on word and sentence level features such as word length, sentence length. Although these style markers are noisy in short texts like e-mail messages they can be used as language independent features more preciously. Style markers on vocabulary richness on lexical category which are used to measure the diversity of word structure used by the author or in short it exhibits the complexity of the sentences. There are several old and new algorithms for making styles on vocabulary such as type/token ration, hapex legomena, hapex dislegomena (McEnery and Oakes, 2000), Flesch Reading Ease score (Clough, 2000), Flesch-Kincaid Formula (Johnson, 1998), Gunning FOG Readability Test (short: FOG) (Johnson 1998), Powers-Sumner-Kearl formula, McLaughlin 'SMOG' Formula and FORCAST Formula and Fry Readability Graph (Johnson, 1998), Yule's Characteristic K measure (Yule, 1944), Honor´e's R measure (Honor´e, 1979), Sichel's S measure (Tweedie and Baayen, 1998), Brunet's W measure (Holmes and Forsyth, 1995), Average Word Frequency Class (Meyer, et. al., 2007). Frequency vector is another lexical style marker category of words such as using articles, propositions, pronouns which can be used as styles to discriminate authors (Argamon and Levitan, 2005). Most frequent words are also one of the lexical style markers which has been successfully used by Burrows (1987, 1992). Adjoining contextual information also can be useful for lexical style markers and the approach of word Ngram was used by Stamatatos (2006). With the availability of correct spelling checkers for natural languages, some researchers use spelling and grammatical mistakes and other formatting mistakes as style markers of authorship attribution (Koppel and Schler, 2003).

Various features have been utilized under character level style markers particularly the alphabetic characters count, digit characters count, uppercase and lowercase characters count, letter frequencies, punctuation marks count, and so on (de Vel et al. 2001), (Zhang et al. 2006). Consequently, more efficient approaches were born with character n-gram algorithms and those are rather efficient (Matsuura and Kanada, 2000). As style marker the most frequent character n-gram is more important to get a good discrimination among authors especially, character bigrams and trigrams give good performances on federalist papers (Stamatatos, 2006), (Keselj et al. 2003).

Next major category of style markers is syntactic level and generally authors try to use same language patterns. These style markers can be used to identify the same language patterns among authors. Conversely, syntactic level styles can be extracted by a more sophisticated language parser. Hence, it will depend on the language which the author has used. In 1996 syntactic style markers were used by Baayen, van Halteren, and Tweedie for the first time for English language and they used syntactically annotated English corpus (Baayen, et al. 1996). Stamatatos et al. (2000) introduced more realistic syntactic style markers called analysis level measures. Parts-of-speech tags and building n-grams on those tags also have been used by some researchers as syntactic style markers (Koppel and Schler, 2003). However, since the POS tags will not be able to make the complete phrases and language structures of the natural languages. Karlgren and Eriksson (2007) presented adverbial expressions and the occurrence of clauses within sentences as syntactic style markers and eventually it has given more reliable results which do not stand with the traditional syntactic approaches.

Mayer and Stain (2006) have divided the style markers into five categories such as (i) character level statistics, (ii) sentence-level text statistics, (iii) part-of-speech features, (iv) count of special words, (v) structural features and finally they introduced a new feature called Average Word Class Frequency which is successfully used as vocabulary richness measure.

Under syntactic level measures which depend on parts-of-speech tags also has been used as style markers in authorship attribution (Stamatatos, et. al. 2000). Mostly used measures are the count of the number of passives and the count of the frequency of various categories of parts-of-speech tags (Kim and Walter, 2008). N-grams of syntactic labels from partial parsing also have been used as features of authorship attribution (Hirst and Feiguina, 2007). Functional lexical features also have to be the reliable markers of style (Argamon, et al. 2007). The basic disadvantage that has been detected in these measures is computational complexity to calculate. These measures have created good resulst in both long and short texts.

2.8.4 Stylometry with Machine Learning

The modern machine learning approaches disseminated a turning point of Stylometry. Analyses of Stylometry was depends on numerical vectors and learning methods which have been extracted the class boundaries of the styles. These class boundaries including a specific style of the particular author depends on the learning method and it can be used to determine the styles of new vectors by analyzing the minimum distance with such boundaries. Various types of neural networks have been utilized with function words as style features at the first time and they have obtained good results in such experiments (Tweedie et al. 1996), (Zheng et al. 2006). Naïve Bayes classification approach was used by Kjell (1994) and again k-nearest neighbor approach was also used by Kjell et al. (1995). Similarly, rule learners as well as support vector machines tested for Stylometry and authorship attribution recently by De Vel et al. (2001), Koppel and Schler (2003) and Zheng et al. (2006). Recently, Bayesian regression also has been used as learning method for Stylometry (Argamon, et al. 2007). Support vector machine comparatively gives good classification results on authorship attribution with stylistic features rather than other learning approaches (Zheng, et al. 2006). However, some recent findings have explained the variations of Bayesian classification and Winnow approach is promising in the field (Koppel, et al. 2003). Function words, parts–of-speech, prepositions, pronouns and modal verbs, number of common words, n-grams especially tri-grams etc. have been used as the Stylometry features by most of the above mentioned studies.

2.8.4.1 Self Organizing Maps

According to the incoming sense perceptions of our body the brain cells self-organize themselves in groups and make decisions. This incoming perception is received by more than one cell of the neurons and the neighbouring cells are arranged according to the cell which receives the incoming massage. A kind of network map is created by these adjustments of neighbourhood cells where neural cells with similar functions are arranged close together. SOM mechanism is based on this principle.

The input data which contain similar attributes are grouped together by the clustering techniques. Although the input feature space will be high dimensional, SOM produces a similarity network graph of such input space and commute this hyperplane into simple typological relationships on two dimensional space (Kohonen, 1990) and (Kohonen, 2001). When arranging the output space, SOM makes the map with similar neurons geometrically together and hence it can be used to identify unknown clusters of the input space easily. Figure 3.7 shows the architecture of the SOM.

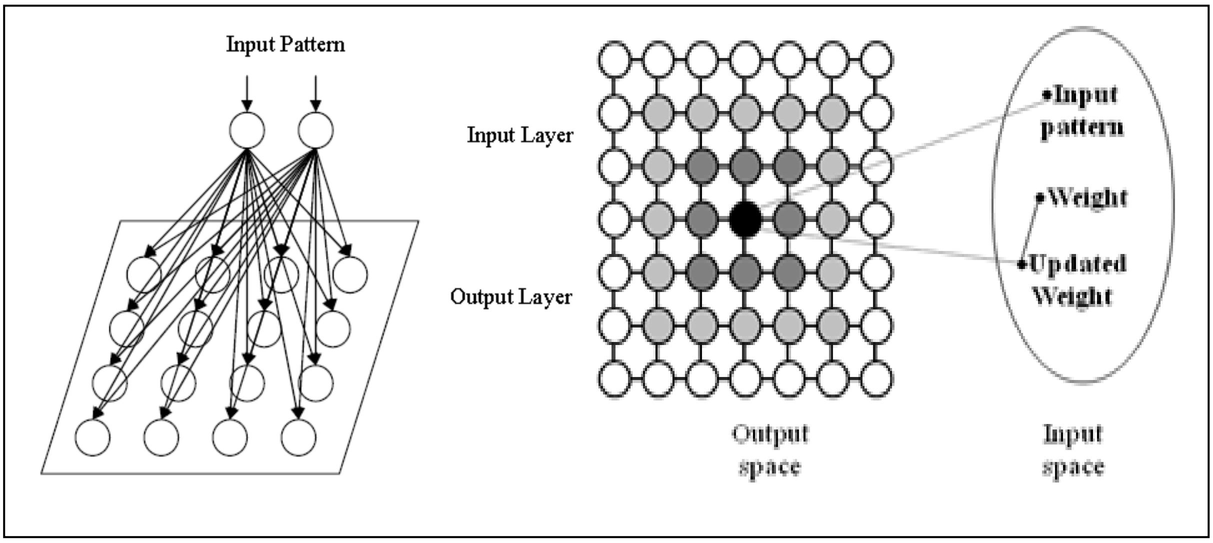


Figure 2.2: The Architecture of SOM

Initialization, training and visualization are the three steps of classifying and clustering using SOM. In initialization, each vector of input space is considered as n dimensional and for each neuron in the map (7 X 7 map in figure 3.6) is assigned to a prototype vector from the data set which is initialized randomly or linearly. After training these prototype vectors it behaves as an exemplar for the entire vectors that is associated with the neuron.

In the training process suppose i be a neuron in $n \times n$ grid and m^i be the prototype vector associated to i and $x \in R^n$ be an arbitrary vector. Now, the task is to map this x to any one of the neuron. For each neuron compute the distance

$$D_i =^{min}_{i} (\|x - m_i\|) \qquad \text{------------------ (2.1)}$$

Better statistic is:

$$D_i =^{max}_{i} (x \,. m_i) \qquad \text{------------------- (2.2)}$$

Neuron satisfying the above statistic is the winner and it is denoted by b. According to the winning neuron b the neighbor neurons in the typology will be adjusted by using the following function.

$$m_i(t+1) = \ m_i(t) + \alpha(t)h_{bi}(t)[x - m_i(t)] \qquad \text{------------------ (2.3)}$$

t is desecrate time coordinate and $m_i(t+1)$ at $t+1$ and neighborhood kernel $h_{bi}(t)$ is defined as

$$h_{bi}(t) = \exp\left[-\frac{\|r_b - r_i\|^{\,2}}{2\sigma^2(t)}\right] \qquad \text{------------------ (2.4)}$$

Where r_b and r_i radious vectors of b and i neurons and $\sigma(t)$ and $\alpha(t)$ are monotonically decreasing with the time.

This learning process is repeated and adjustments are made iteratively until the SOM makes up a sufficiently accurate map. After the training process, prototype vectors containing the cluster of an orderly map are formed and neurons can be labeled with the cluster means or classes of the associated prototype vectors (Vesanto et al. 2000). Visualization of clusters is done by projections, U-matrices and other distance matrices in such a way that the topology of the original data is preserved. Component maps and scatter plots can be generated to aid in inspecting possible correlations among dimensions in the input data. Each component map visualizes the spread of the values of a particular component (or dimension) (Alfred, 2003). As a result, possible correlations are revealed by comparing different component maps with one another. However, the clustering property of the SOM can be efficiently used with computer forensics (Fei, et al. 2005)

2.9 Chapter Summary

Plagiarism detection mainly classifies into two, such as manual detection and electronic detection. The literature on extrinsic and intrinsic detection which are two methodologies pertaining to the electronic plagiarism detection has been explained in this chapter. Most of the plagiarism detection tools and algorithms are based on the extrinsic plagiarism detection. These tools can be divided into two including source code plagiarism detection tools and natural language plagiarism detection tools. Several approaches which have been utilized in the plagiarism detection also discussed in this chapter. Finally, the techniques used in authorship attribution and verification and some machine learning approaches used for authorship verification have also explained.

Chapter 3 – A New Framework for Plagiarism Detection

3.1 Introduction of the Architecture

Academics as well as educationists argue that the prevention of plagiarism is more important than detection of plagiarism (Carroll, 2004). Plagiarism is a behavioral pattern of a person. It is true that somebody can change such behavior by eliminating the factors which affect the phenomenon. However, the attitudes of people naturally vary at a higher degree. Same ideology or methodology may not be applied to change the attitudes of all persons correctly. Such nature may exhibit that the prevention does not bring about sufficient avoidance of plagiarism. It is shown that, although there are some implemented programs on plagiarism prevention, especially in European countries, America, and Australia they are still facing such problems in their universities. Conversely, such kind of attitude changes cannot practically be applied in our countries. There are several considerations which affect the problem including the current educational environments in their primary education, economic conditions, future expectations etc. These explicitly involve with this crucial problem and hence, there should be a proper detection system, rules and regulations to overcome plagiarism.

Since human beings are unable to handle large numbers of documents at once plagiarism detection with integral human involvement will be time consuming and inefficient. Conversely, only automated systems which can be used to obtain such decisions will be an injustice for the submitter due to various reasons. One example is scientific or mathematical definition. Such a definition will neither be changed over the author nor the theme. However, if some of the documents contain the same definition then the detection system decides it as plagiarized and cannot obtain a presumptive decision. Similarly, one of the other arguments is that the detection system will not be capable of detecting all the aspects of plagiarism since the cheating behavior of human beings is more complex (see chapter 02 section 2.3).

Although the plagiarism detection systems are available in both commercial and free software but all the cheating behaviors of the students may not be considered and hence, such misconducts may not be covered by the systems as necessary. It is essential to affiliate all those student cheating behaviors and include the detection capabilities of all in one system. In order to address these issues a detection system with a rich framework and appropriate algorithms should be created to cover the plagiarists' behavior. Such a system should detect direct document-to-document copying, direct copying from the electronic documents from web, copying from the electronic documents and modifying it etc. and simultaneously it will assist the user who uses such a system. In this approach this system is called a Machine Assisted Plagiarism Detection System (MAPDetect) (Ranatunga, et al. 2009).

Two basic characteristics are considered in this framework. Firstly, covering all the above mentioned misconducts of the students and secondly, giving proper assistance to the user to detect and get decisions according to the available evidences. The front end of the proposed framework is similar to the functions of a spelling checker or a grammar checker and the detection system will not give decisions. It will provide a ranked list of plagiarized documents in the wake of different matrices given by several approaches which especially conduct the detections pertaining to each type of misconduct. The user is able to provide his/her input and finally select the plagiarized documents according to an underlining policy.

String similarity algorithms followed by the information retrieval algorithms are used by most of the plagiarism detection systems. In the proposed architecture both of these approaches are employed and in addition, natural language processing (NLP) approaches and authorship attribution especially, authorship verification are also considered. Figure 1 illustrates the architecture of the proposed framework.

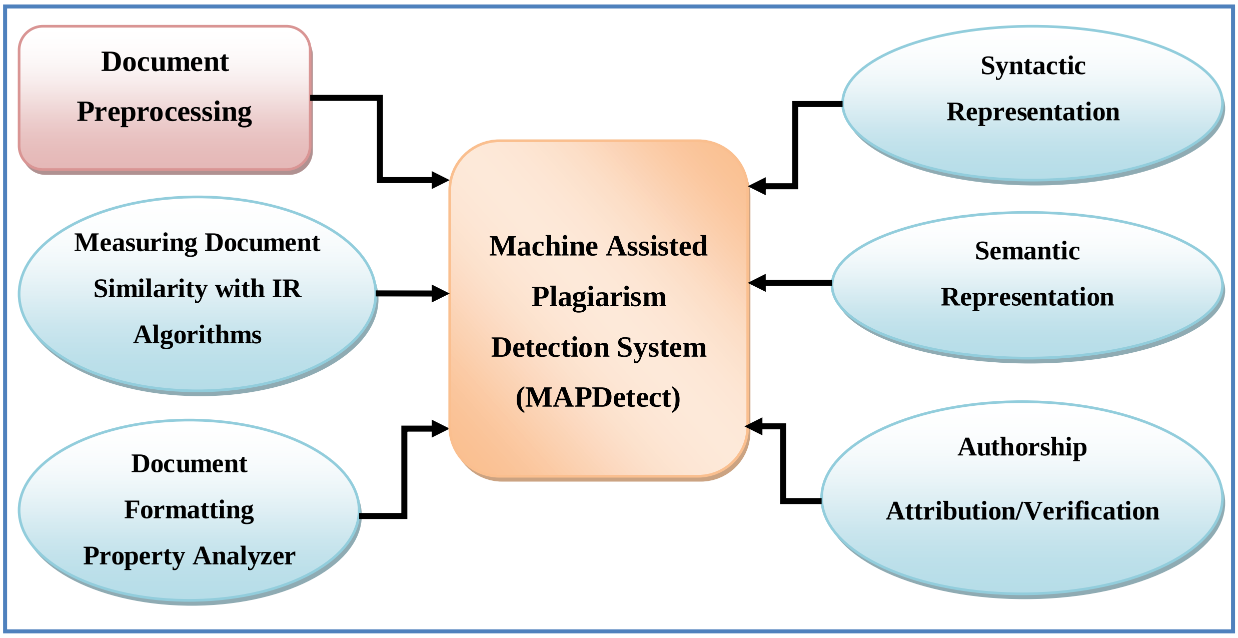

Document Preprocessing
Measuring Document Similarity with IR Algorithms
Document Formatting Property Analyzer
Machine Assisted Plagiarism Detection System (MAPDetect)
Syntactic Representation
Semantic Representation
Authorship Attribution/Verification

Figure 3.1: Architecture of the MAPDetect framework

3.2 Preprocessing the Documents

Several aspects must be considered on documents which are going to be put into the detection system. The documents which are submitted by the student to the LMS may vary in formats such as MS Word, PDF, and Rich Text Format etc. Similarly, all the contents of the documents are not important factors of plagiarism detection. Thus, unnecessary elements should be eliminated. Finally, documents should be fettled for various feature extractions. A bundle of these kinds of activities is called document preprocessing and it is a very essential part of the plagiarism detection system. During the preprocessing, different document structures are generated by the preprocessor. There are four kinds of structures namely, the document vectors, the n-gram representations, XML representations and feature vectors. Documents can be represented as a vector of different terms which are the different words including throughout the document with their frequencies (See Section 3.3.2). Again, the documents can be represented as a collection of n-grams. The basement of the n-grams may be character level or word level (See Section 3.3.4). XML representation of the document is included not only by the content of the document but also properties of the document containing various formatting attributes and Meta data of the document (See Section 3.4). Forty nine features are extracted from each document during the preprocessing and it is also another representation of the documents (See Section 3.6.2).

Processing large documents with unnecessary contents is computationally hard work, very resource consuming and hence, inefficient. Lots of potential duplicates will be compared by the system and thus the user will be directed towards unnecessary decisions. In the methodologies of information retrieval there are some techniques to abolish such contents which are not important to detection contrivance. After converting the document into a text format and before preparing the two different structures, the following preprocessing activities should be applied to obtain maximum efficiency.

1. Eliminating all stop words and common terms ("a", "in", "of", etc).
2. Stemming terms to their roots
3. Limiting the vector space to nouns and few descriptive adjectives and verbs.
4. Using small signature files or not too large inverted files.

In the English language, there are some words which can be used to make the sentence structure and to build up relationship of the presented contents. These are called stop words or common-class words. These words themselves have no meaning and ignoring these words from the documents will reduce the processing time noticeably and it may not affect the final result of the comparison (Witten, et al.1999). Conversely, English words contain suffixes to the original word and create the multiple form of the same word. In the context of the similarity detection, the form of the word is not a significant factor. Stemmers can be used to remove these suffixes from the words and by using the base form of the word will also reduce the time consumption and the computational resources.

However, the XML representation will be made with the original document without converting to text file format. Especially, Ms Word docx files are good examples and all text and other formatting properties should be obtained by the preprocessing. Conversely, while the documents are being preprocessed there are several features which can be extracted as auxiliary determinants to recognize the similarities among the documents such as word frequency, function word frequency, the number of punctuations used, the distribution of words, the average sentence length, the average length of paragraphs, etc. These feature vectors are used for analyzing authorship verification.

3.3 Measuring Document Similarity with Information Retrieval Algorithms

Ranking documents according to the given query text is the final objective of most of the information retrieval methods (see section 2.6 of chapter 02). Although the query oriented document ranking techniques are being used with short quires, it can also positively be used with the plagiarism detection domain with large sizes of suspicious documents, especially, in verbatim plagiarism. Clustering all the documents which are very similar to the given query document will be the final objective in electronic plagiarism detection. It depends on using a kind of metric for the ranking process that can be measured to obtain the highest similar cluster from the entire document set. Basically, query oriented document ranking methods can be categorized into two. Figure 3.2 illustrates these categories.

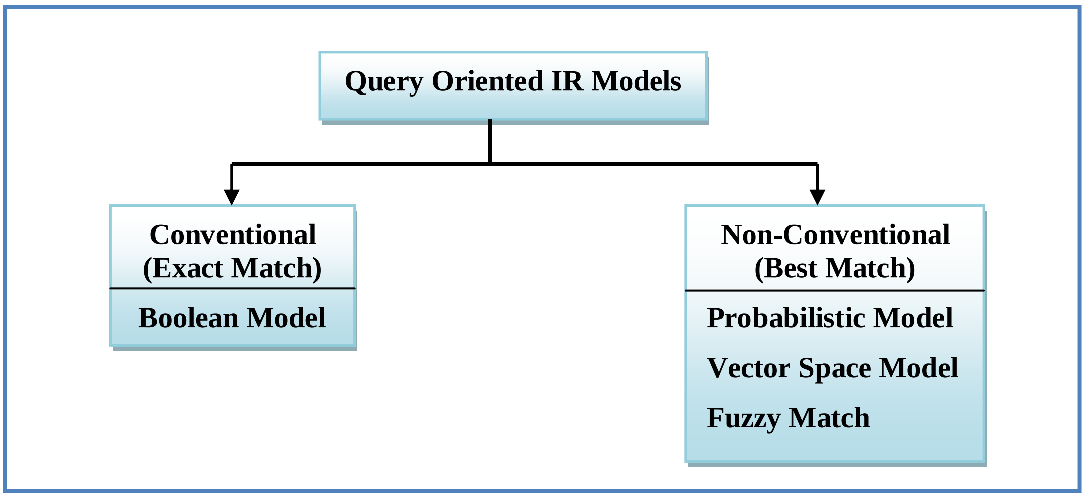


Figure 3.2: Query oriented information retrieval models

Generally, there are three steps to be followed to make a raking result. At the first step document collection is indexed. At the second step, another three main activities such as query is processed with each document of the collection and the similarity matrices are obtained and the weighted values are computed, and finally, the documents are stored in the descending order and the topmost ranked documents are selected as identical to the query document. However, all the above models are employed by finding nearest neighbor search in the hyper plane of a vector space.

3.3.1 Boolean Model

The Boolean Model is a simple IR model which is based on set theory and Boolean algebra. In the Boolean model there is no weighting factor and weights are binary, either 0 or 1, and similar weights are used for all terms in the document. Although the query expression with Boolean operators AND, OR, or NOT is used in the Boolean Model framework and uses one document as a query with the OR operator. In the experiments, D number of documents in the collection and d_i is one of the documents among the collection. The query document is denoted by$Q$. The ranking result of each document in the collection is given by the ranking function$R(Q, d_i)$, after computing the exact matching of each term of the document d_i with query$Q$.

Boolean model encapsulates clear context and implementation is easy than other IR models. Conversely, exact matching of terms may give too few or too many similar documents and it will be difficult to ranking the result. However, it may be useful in some type of plagiarism and this characteristic is shown by the results of the experiments.

3.3.2. Vector Space Model

A document is represented by a vector of a vector space and each dimension of a vector is a term of the document in the VSM. Each item of a vector is assigned a value and it gives the importance of such terms which associates with the given document. This value is a function of term frequency in which such term occurs in the document collection. Suppose that a table contains a total of D documents which is described by T terms of vocabulary can be represented as a $T \times D$ terms by document matrix called M. Practically, D documents are represented in the columns of the matrix and T terms are represented in the row. In other word, document vectors are column wise and term vectors are row wise. Hence, the element m^{ij} is the weighted frequency of the term i that occurs in document$j$. There can be a correlation between the document vector d and the query vector$q$. This correlation should be measured by a normalized form. According to the Section 2.6 Cosine has been used to quantify this correlation in a normalized form. If the angle of two vectors d and q is high, then the cosine value is related to zero. It indicates that the documents are totally different and the correlation among such two documents is very low. If the angle of the two vectors is identical, it has more correlation and the cosine value reaches one.

There should be a single value to measure the similarity of the two vectors d_j and q. In mathematics vectors are multiplied by dot product and can represent the multiplication of the two vectors as;

$$d_{jq} = (w_{1j} \times w_{1q} + w_{2j} \times w_{2q} +, \cdots, + w_{tj} \times w_{tq}) \quad \text{---------- (3.1)}$$

and

$$Sim(d_{j,q}) = \frac{\sum_{i=1}^{t} w_i j \times \sum_{i=1}^{t} w_i q}{\sqrt{\sum_{i=1}^{t} w_i^2 j} \times \sqrt{\sum_{i=1}^{t} w_i^2 q}} \quad \text{---------- (3.2)}$$

In this model the weighting factor w is very important. The elements of M matrix can be weighted by several schemes. It can be calculated by term weighting with the clustering technique. Term frequency means how many times the k_i keyword appears in the document d_j. The number of times k_i appears in a particular document $(freq_{ij})$ is calculated first and the maximum appearance of term $(max\ (freq_{ij})$ is then calculated. Then the normalized frequency f_{ij} of term k in document $d_j$given by n_j;. Is

$$F_{ij} = \frac{freq_{ij}}{Maxfreq_{ij}} \quad \text{--------------------- (3.3)}$$

Since the result as a higher value for more common words and less value for less common words are given by this term frequency tf of the above equation, another normalization factor called inverse document frequency idf should be used to overcome the problem. idf is calculated by using the total number of documents N and the total number of documents that appear where the word n_i appears. After applying logarithm to get inverse of the result, the equation can be derived as

$$idf = \frac{log\ N}{n_i} \quad \text{--------------------- (3.4)}$$

and the integral tf-idf equation is

$$w_{ij} = f_{ij} \times \frac{log\ N}{n_i} \quad \text{--------------------- (3.5)}$$

Apart from the above basic tf-idf weighting scheme Salton and Buckley (1988) represented another variation which uses not only the local information but also global information as the following equation.

$$w_i = tf_i \times log \frac{D}{df_i} \qquad \text{---------------------- (3.6)}$$

Where

- tf_i = term frequency (term counts) or the number of times the term i occurs in a document. This accounts for local information.
- df_i = document frequency or the number of documents containing term i
- D = the number of documents in a database.

The IR models explained in the above are principally employed with short queries. However, the aim is developing the similarity matrices that can be effectively utilized to compare two full length documents. In this problem, maintenance the complexity of the algorithm should be focused and special attention is given for the plagiarism detection domain. The complexity of the algorithm gets linear in the two dimensional searching matrix. Thus, the asymptotic complexity lies on the number of terms T and the number of documents D and it is $O(TD)$.While reducing the terms in documents by preprocessing it substantially decreases the execution time for large documents.

3.3.3. N-gram Comparison and Fingerprinting

Obtaining fingerprints from the document and measuring similarity among those fingerprints can be used for plagiarism detection (see chapter 02 section 2.6). When designing the fingerprinting process it is necessary to take four significant considerations.

1. The method of dividing and selecting substrings from the document
2. The size of substring
3. Fingerprint resolution
4. The method for comparing fingerprints to detect similarities

As the first step an efficient method should be used for dividing and selecting substrings from the document. In this research n-gram method is used to divide the document into substrings. A n-gram has no linguistic meaning and consists with either character level or word level. The character level n-gram substrings are employed as the dividing technique and the statistical patterns of letters in the words are identified by this technique. For example, the word ‘plagiarism’ has following n-grams:

2-gram: pl la ag gi ia ar ri is sm

3-gram: pla lag agi gia iar ari ris ism

The significant advantage of using character level n-gram is when the plagiarizer attempts to copy portions of sentences into different location of the document and character level n-grams can be used to detect those chunks of words efficiently.

The second is the size of the substring or n-gram that is selected from the document. It is called *granularity*. Determining the correct and the suitable value of the granularity will be more important since, if the granularity is very less then it directly affects the computational time and if it is a large value then it is unable to identify the patterns correctly. Typically, the word level substrings are used and the number of words is counted in the substring as granularity. In this approach character level n-grams are used and hence, it is based on the number of characters. Section 4.4 of Chapter 04 illustrates the results of the different levels of this value and finally selects a good limit for the correct identification of the similarities of the two documents.

The third is to see how many minutiae are used for creating a fingerprint. It is called fingerprint *resolution*. Mainly, there are several classes regarding resolutions such as full fingerprinting, positional strategies, frequency based strategy, and structure based strategy. Typically all these classes are implemented with a word level minutiae selection for fingerprinting. However, as mentioned above, it is necessary to identify the positional changes of the chunks among the documents. Hash values of n-grams as minutiae are used in order to identify the positional changes of the plagiarizer and the selected subsets of hash values for creating the fingerprint. Very large vector space will be generated unnecessarily by taking all hash values. It will directly affect the processing power and time especially in large documents. Dimensional reduction in fingerprinting is done with Winnowing algorithm (see section 2.6 of chapter 02).

The final step is implementing a proper method for comparing the selected fingerprints of documents. In this approach there are two methods. The first is 1 to n match and the second is n to n match. First, all preprocessed documents are fingerprinted. While creating the fingerprints, its position in the document is also detected and stored in another file. In 1 to n match a query document also is subjected to the same procedure. Then each pair of fingerprint and the position of that fingerprint in the query document are matched with each document in the document collection $d_1, d_{2\ldots} d_n$ and count both number of matches f and number of fingerprints in query document q. The normalization of the result is done by using these values as f/q and all the results from each file are sorted in descending order. Finally, the highest ranked file may take as most similar file to the query document. In n to n match this procedure extends to each document in the collection one by one and the final result can be large because all the n number of results is included in it.

3.4 Document Formatting Property Analyzer

There are two behavioral and physical considerations that can be made to detect plagiarism among university students. The first is, practically, most university students do not attempt to manage their time. Assignments are done in the last minute of the allocated period. The learning materials for the assignments are not collected gradually and thus, they use the Internet and copy the contents or copy the contents of digital copies from other students and finally submit the assignment as ideal copies of such documents on time. Secondly, most of the normal tutorials and assignments are not considered in a particular format. Students can submit those documents independently by using their own formats and most of the lecturers and tutors are not aware of the formatting and they only mark the contents of such documents. Since all the contents cannot be memorized by the marker the documents will not be identified as plagiarized.

During the copying of texts from other sources, especially, from the internet which contains a formatting structure which is previously used in the web content itself. For example, some table formats, hyperlinks, boarder styles etc. and mostly, as these formatting are encapsulated in the document insidiously the user cannot see such things in the document normally. The mostly available practical example for this incident is the hyperlinks of the students' submissions. When copying text portion from the web it may contain some hyperlinks and those links may appear as blue letters with underlined texts. Normally, students are only aware of the appearance of the text and change the color and underline the fonts without removing hyperlinks. If another student copies the same document and submits it as his/her own document and the hidden truth is included with their submissions.

By addressing these insidious phenomena, the framework implements another technique called Document Property Analyzer (DPA) which analyzes the formatting techniques which are used by the author with other documents. Figure 3.3 shows the basic architecture of the method.

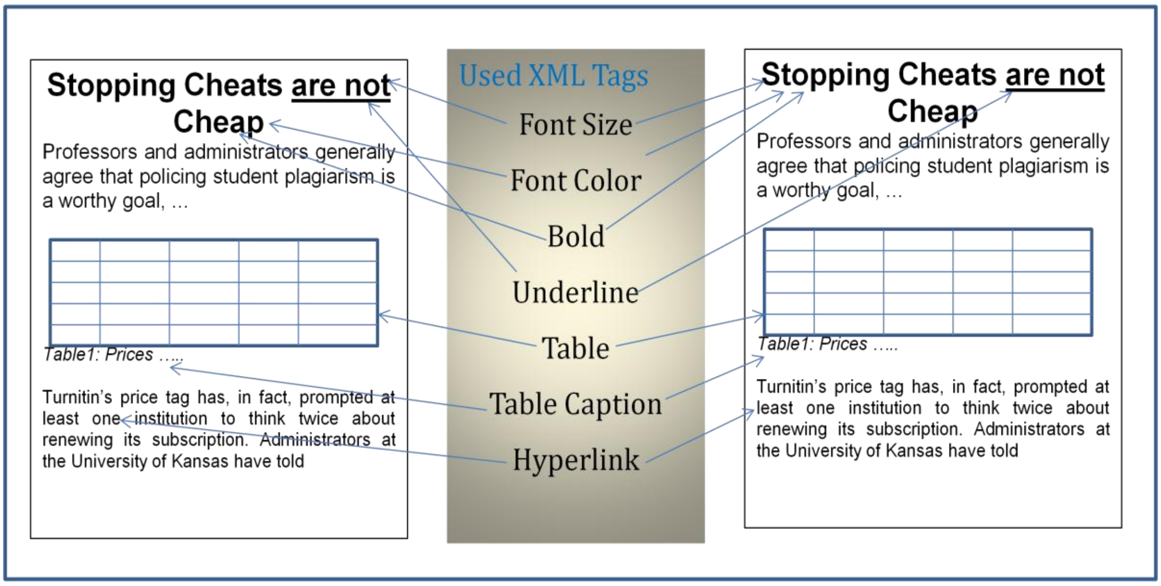
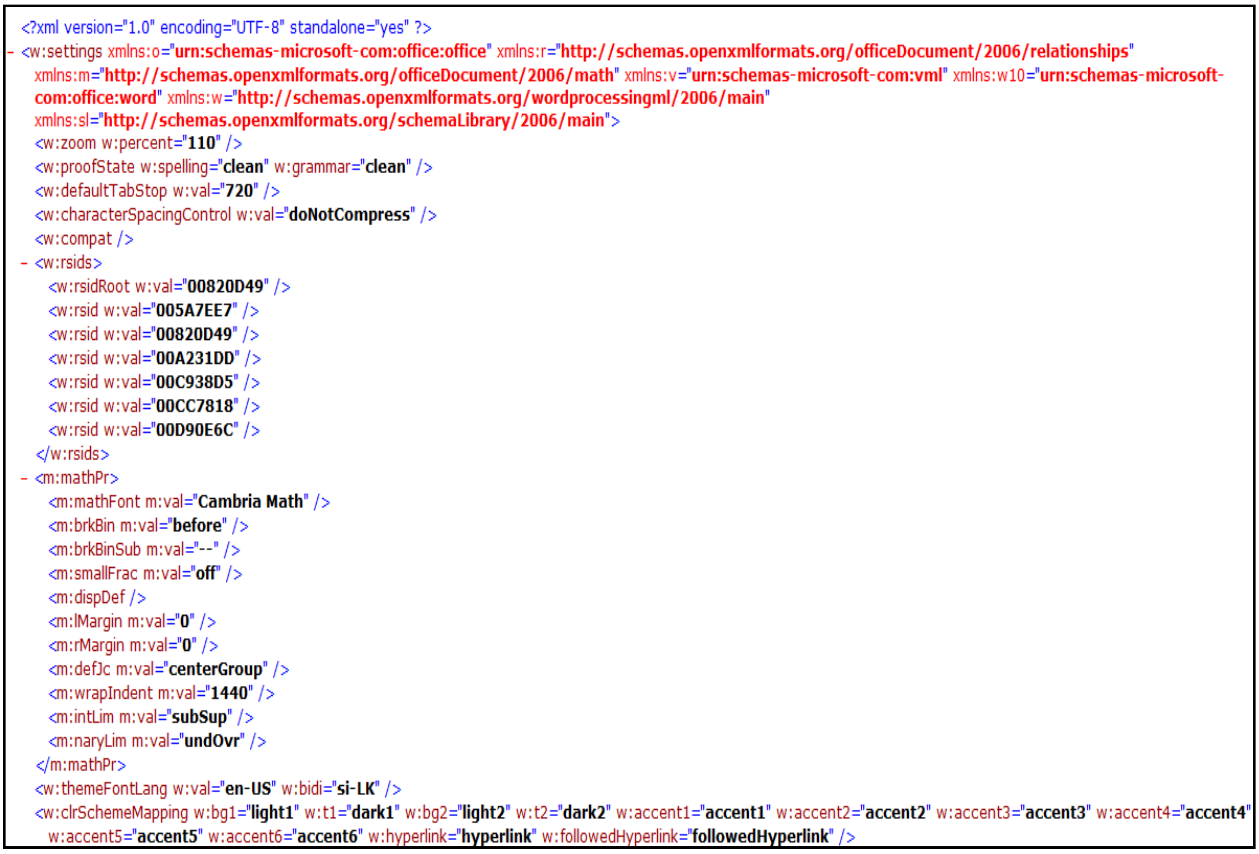

Stopping Cheats are not Cheap
Professors and administrators generally agree that policing student plagiarism is a worthy goal, ...
Table1: Prices .....
Turnitin's price tag has, in fact, prompted at least one institution to think twice about renewing its subscription. Administrators at the University of Kansas have told
Used XML Tags
Font Size
Font Color
Bold
Underline
Table
Table Caption
Hyperlink
Stopping Cheats are not Cheap
Professors and administrators generally agree that policing student plagiarism is a worthy goal, ...
Table1: Prices .....
Turnitin's price tag has, in fact, prompted at least one institution to think twice about renewing its subscription. Administrators at the University of Kansas have told

Figure 3.3: How Document Property Analyzer works

In order to implement the concept the document collection $\mathrm{d}_1, \mathrm{d}_{2\ldots}\mathrm{d}_\mathrm{n}$ are stored without any preprocessing activities like the above algorithms. Since only the formatting features are extracted from the document the preprocessing does not need to apply. Most sophisticated word processors generate an Extreme Markup Language (XML) file with the original document i.e. Microsoft Word, Open Office. This file can be used to extract the XML tags which include all elements such as document property tags, formatting attributes tags, text elements, and standard file attributes, etc. An XML-text parser has been used to identify and store each important formatting property of a document. Figure 3.4 illustrates the XML representation of portion of a document.

The document d_i contains n number of formatting tags p_n and firstly, all the tags of each document are extracted and those vectors are stored separately. Each item of the vector is a formatting property of the document and the frequency of each property is also calculated.

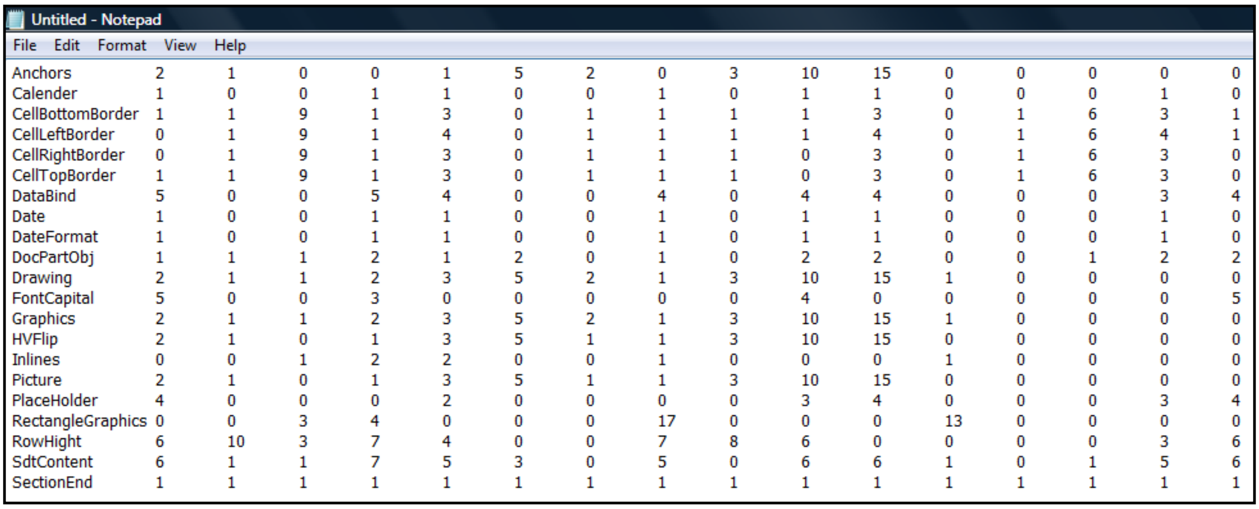
Each vector contains large dimensions and the processing of these large dimensions consumes more processing power. Conversely, the incorrect discrimination of documents also occurs. There should be a method for dimensionality reduction in order to increase the efficiency and obtain the proper results. Since some formatting properties such as bold, underline, etc. are commonly used in all documents. A weighting factor is used to eliminate these unimportant properties called Formatting Frequency Inverse Document Frequency ff-idf. The occurrences of each property in all documents are calculated in ff-idf and adjustable threshold value t is used for dimensionality reduction. The formatting property vector of each document then assigns pairs of names and frequencies which are subjected to removing unnecessary distortions for discrimination.



```
  <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <w:settings xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:r="http://schemas.openxmlformats.org/officeDocument/2006/relationships"
    xmlns:m="http://schemas.openxmlformats.org/officeDocument/2006/math" xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w10="urn:schemas-microsoft-
    com:office:word" xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main"
    xmlns:sl="http://schemas.openxmlformats.org/schemaLibrary/2006/main">
    <w:zoom w:percent="110" />
    <w:proofState w:spelling="clean" w:grammar="clean" />
    <w:defaultTabStop w:val="720" />
    <w:characterSpacingControl w:val="doNotCompress" />
    <w:compat />
  - <w:rsids>
      <w:rsidRoot w:val="00820D49" />
      <w:rsid w:val="005A7EE7" />
      <w:rsid w:val="00820D49" />
      <w:rsid w:val="00A231DD" />
      <w:rsid w:val="00C938D5" />
      <w:rsid w:val="00CC7818" />
      <w:rsid w:val="00D90E6C" />
    </w:rsids>
  - <m:mathPr>
      <m:mathFont m:val="Cambria Math" />
      <m:brkBin m:val="before" />
      <m:brkBinSub m:val="--" />
      <m:smallFrac m:val="off" />
      <m:dispDef />
      <m:lMargin m:val="0" />
      <m:rMargin m:val="0" />
      <m:defJc m:val="centerGroup" />
      <m:wrapIndent m:val="1440" />
      <m:intLim m:val="subSup" />
      <m:naryLim m:val="undOvr" />
    </m:mathPr>
    <w:themeFontLang w:val="en-US" w:bidi="si-LK" />
    <w:clrSchemeMapping w:bg1="light1" w:t1="dark1" w:bg2="light2" w:t2="dark2" w:accent1="accent1" w:accent2="accent2" w:accent3="accent3" w:accent4="accent4"
      w:accent5="accent5" w:accent6="accent6" w:hyperlink="hyperlink" w:followedHyperlink="followedHyperlink" />
```

Figure 3.4: Portion of a document represented in XML

The document property matrix is created by using all the formatting properties F and the document collection$D$. Thus, the $F \times D$ matrix is created and the document vectors are presented in columns and the formatting vectors are presented in rows. Figure 3.5 illustrates the actual text file with matrix created by the program. The frequency of the property i occurring in document j is denoted by P^{ij} element in the matrix. Same procedure is also followed by the query document.



```
Anchors           2    1    0    0    1    5    2    0    3    10   15   0    0    0    0    0
Calender          1    0    0    1    1    0    0    1    0    1    1    0    0    0    1    0
CellBottomBorder  1    1    9    1    3    0    1    1    1    1    3    0    1    6    3    1
CellLeftBorder    0    1    9    1    4    0    1    1    1    1    4    0    1    6    4    1
CellRightBorder   0    1    9    1    3    0    1    1    1    0    3    0    1    6    3    0
CellTopBorder     1    1    9    1    3    0    1    1    1    0    3    0    1    6    3    0
DataBind          5    0    0    5    4    0    0    4    0    4    4    0    0    0    3    4
Date              1    0    0    1    1    0    0    1    0    1    1    0    0    0    1    0
DateFormat        1    0    0    1    1    0    0    1    0    1    1    0    0    0    1    0
DocPartObj        1    1    1    2    1    2    0    1    0    2    2    0    0    1    2    2
Drawing           2    1    1    2    3    5    2    1    3    10   15   1    0    0    0    0
FontCapital       5    0    0    3    0    0    0    0    0    4    0    0    0    0    0    5
Graphics          2    1    1    2    3    5    2    1    3    10   15   1    0    0    0    0
HVFlip            2    1    0    1    3    5    1    1    3    10   15   0    0    0    0    0
Inlines           0    0    1    2    2    0    0    1    0    0    0    1    0    0    0    0
Picture           2    1    0    1    3    5    1    1    3    10   15   0    0    0    0    0
PlaceHolder       4    0    0    0    2    0    0    0    0    3    4    0    0    0    3    4
RectangleGraphics 0    0    3    4    0    0    0    17   0    0    0    13   0    0    0    0
RowHight          6    10   3    7    4    0    0    7    8    6    0    0    0    0    3    6
SdtContent        6    1    1    7    5    3    0    5    0    6    6    1    0    1    5    6
SectionEnd        1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

Figure 3.5: $F \times D$ matrix of Document Property Analyzer

The final step measures the correlation between each document vector d_i and the query document vector q against the frequencies of formatting properties. The similarity ratio which is based on correlation is calculated by using the following equation.

$$sim(q, d_i) = \frac{\sum d_i q - \frac{\sum d_i \sum q}{n}}{\sqrt{\left(\sum d_i^2 - \frac{(\sum d_i)^2}{n}\right)\left(\sum q^2 - \frac{(\sum q)^2}{n}\right)}} \qquad \text{----------- (3.7)}$$

Finally, the similarity measures of all documents are sorted in descending order in order to rank the collection of documents. The highest ranked is selected according to the assigned threshold value.

3.5 Syntactic and Semantic Analysis of Documents

Paraphrasing may be carrying out in several ways by the plagiarizer mainly changing the syntactic and semantic structure of the sentences (see section 2.6 in chapter 02). Modifying words in the sentence, change the word order, change the tense of the sentence, change the voice of the sentence are some paraphrasing techniques mostly used by the plagiarizers. However, finally, these kinds of activities will not affect the basic idea of the sentence. If there is a method to identify similar ideas of the document it will be more appropriate to detect plagiarism in this context. The next methodology of this research is concerning on identifying the chunks of documents which are intentionally modified by the plagiarizer. The most familiar modifications are changing the sentence structure and replacing with synonyms for some words. For example, suppose the original sentence is John told a story to Mary which can be modified as

- A story was told by John to Mary.
- John tells a story.
- John asked a story to Mary.
- John told a poem to Mary.

According to the above example same meaning is given by the first sentence but the *voice* of the sentence has been changed. The second sentence also has the general meaning but change the *tense*. The third and fourth sentences also have same meaning but with a *synonym* to derive the meaning. If the plagiarism detection algorithm concerns only the surface structure of the sentence it will not be able to identify none of the above sentences as same.

Instead of comparing sentences as morphological combinations of words, it is essential to analyze the syntactic structure of the sentence which does not depend on the morphological representation of it. Syntactic knowledge of a language is based on the structure of word arrangement through a sentence and it does not consider the meaning of the words. Semantic structure of a sentence concerns the meaning of lexical components, its argument structure, and the grammatical functions such as subject, object etc. There are several types of grammatical frameworks for analyzing natural languages. According to the language rule classification by Chomsky, there are three types of grammatical rules exists. Context-free grammar, Context- Sensitive grammar, and Unrestricted-grammar are the three which can be used to represent any language (Chomsky, 1956). Among these three grammatical categorizations, context free grammar is given significant performance in this context since it has capabilities to extract the most of the structures in the natural languages and there are efficient parser available for analyze the sentences with context free grammar.

In Natural Language Processing (NLP) and computational linguistic, another grammar framework is Lexical Functional Grammar (LFG) (Bresnan 1995). The syntactic structure of the sentence is divided into three different ways of representations in the LFG called Argument structure or a-structure, Constituent Structure or C-structure and the Functional structure or f-structure. The a-structure consists of a predicator and its other components such as argument roles like agent, beneficiary, experience/goal, instrument, patient/theme, and locative. Constituent structure is the prominent and more concrete level of linear and hierarchical mapping of words into phrases according to the grammatical functions. The f-structure is the descriptive and abstract functional syntactic mapping of the sentence into functional relations on predicate, subject and object (Pienemann, 2005). The following example demonstrates two different sentences that have been plagiarized in a certain submission which contains the same meaning. This can be derived with predicate argument structure as:

Click (user, application icon)

S_1: A user clicks on the application icon and the application starts the session…

S_2: The application icon is clicked by the user and the application starts the session…

After mapping the two sentences above similar a-structure can be generated and the three structures are illustrated by figure 3.6.

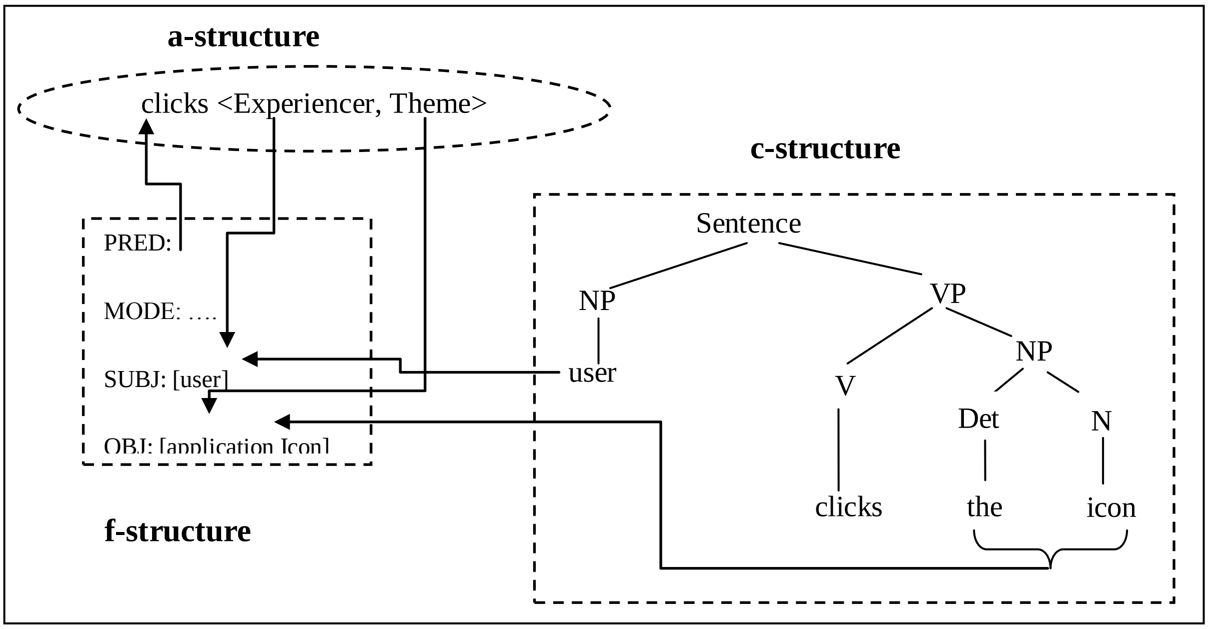


Figure 3.6: Three structure representation of a sentence

There are some built-in parsers which can be used to obtain the parse tree of the sentence. Quite contrary to most other parsers based on *probabilistic context-free grammars* (PCFGs), *Stanford Parser* which is a dependency parser is used to obtain the parse trees of sentences of the given documents of the corpus (Klein and Mannin, 2003). As the searching algorithm *Cocke-Younger-Kasami* (CYK) is used by this parser and both dependency and phrase structure can be obtained by the Java implementation [www 16] of the parser (De Marneffe, et al. 2006).

There are collections of documents D_1, D_2 … D_n and those documents contain a different number of sentences S_1, S_2 … S_n in each document d_{j}. The methodology which is used in the experiments is divided into two stages. The first is the preprocessing stage and it mainly concerns on generating the parse structures of each sentence of the document and store each document containing parse trees separately. During this process sentences are separated and for each word of the sentence the possible syntactic parts-of-speech tags are stored i.e. the above mentioned sentence has been converted as following syntactic tree by the Stanford parser with probabilities which is as same as the c-structure of the figure 3.5. This tree structure also contains different hierarchical levels of the sentence.

(ROOT (S [287.338] (UCP [102.091] (NP [63.468] (NP [30.716] (DT [4.555] A) (NN [8.796] user) (NNS [11.443] clicks)) (PP [32.230] (IN [3.103] on) (NP [28.725] (DT [0.650] the) (NN [9.009] application) (NN [15.328] icon)))) (CC [0.157] and) (S [33.845] (NP [11.822] (DT [0.650] the) (NN [9.009] application)) (VP [21.256] (VBZ [6.557] starts) (NP [9.892] (DT [0.650] the) (NN [7.260] session))))

Although good paraphrasing is acceptable by quoting the source, most of the plagiarizers, especially students, do not attempt to do paraphrasing in an acceptable manner. The main idea of the method is to identify the paraphrasing attempts of the plagiarizer. The plagiarizer can use several techniques in order to hide the plagiarized chunks of texts. These obfuscations are generated by changing the basic elements of the sentence such as subject, verb and object. Therefore, the method which is going to be applied to splash the obfuscations should consider the above three main elements.

- Verb changes

In English, every sentence contains two parts: a subject and a predicate. The subject illustrates what the sentence is about and the predicate tells something about the subject. Normally, the identification of the subject is based on the predicate and further, by the identification of the verb. Without any changes of the other parts of the predicate the plagiarizer can change the verb that does not affect the meaning of the predicate itself. Changes of the verb of the sentence can be basically varied in four ways.

- Changing the tense of the original sentence

Sentences can be changed without distorting the basic ideas which they generate by applying the different tenses of the same verb. For example, according to the above sentence it can be written as “A user clicked on the application and the application started the session” without changing the context and the concept which is to be conveyed. In this case, although the similar verbs such as “clicks” and its past tense “clicked” may be identified by the text matching technique, and if the different form of a verb like “go” can be changed to “went” in the past tense it will not be identified by such techniques.

- Changing the voice of the original sentence

Similarly, the voice of the sentence also can be changed without any distortion of the basic idea and this obfuscation also cannot be easily identified by the text similarity technique. According to the above example the sentence may be written as “The icon is clicked by a user and session is started by the application” and this may again be complex if it combines with tense changes, too.

- Using similar verbs

In any language one verb can be used to obtain different ideas. Conversely, an idea also can be formed with different verbs. Mostly, plagiarizers move with this utility for shading the original sentence. For example, “give” can be changed as “grant” and the words are totally different in lexicon. This phenomenon also cannot be identified by the text similarity algorithms.

- Transforming singular into plural or plural into singular

Transforming sentences from singular to plural or from plural to singular is also popular in paraphrasing. The general idea to be expressed by the sentence will not be damaged by this transformation. However, some verbs may totally change its lexicon when transforming it to singular or plural. For example, “go” is the base verb and its singular form “goes” or plural form “go” will be used. The text matching techniques are not capable of identifying these two words as the same.

- Subject and object changes

Most of the plagiarizers use synonyms for the subject and the object of a sentence. If the plagiarizer is flaunt in paraphrasing it is very difficult and complex to identify because he can use not only the synonym of one word but also several words. It may be one to many words (clause) or reduce many words to similar one word. Additionally, such synonyms may be used with some adjectives. This kind of lexicon changes may not be identified by the text matching algorithm and hence, basically it should obtain the meaning of the sentence for further analyzing.

The second step is more sophisticated and mainly the subject, verb and the object of each sentence has been established in another location. This process should be carried out on both documents such as original documents and the query documents.

Get the verb of the original document. If the verb contains several parts i.e. am, is, are, was, were, has, have, will, etc. split them and obtain the main verb. If the verb contains suffixes, like “ing”, “ed”, “s” or any other form like “went” is converted into the base verb. For example “went” will be converted into “go”. The obtained base verb will be passed into the WordNet and get the senses of the particular verb.

Match the verb of the query sentence with the obtained senses. If there is a similar sense with the query verb it will be marked as plagiarized and the total of plagiarized verbs will be used to calculate the similarity ratio of verbs.

The subject and the object of each sentence also are subjected to check similarities. First, each subject is extracted and stored accordingly in a text file. Query file is also subjected to this process. The subject of a sentence of both original and query file can be varied on one-to-several words. Hence, the process of getting the similarity ratio of subject and the object is more complex than that of the verb. It contains two parts, such as direct similarity checking and the deep similarity checking. However, there should be one-to-one, one-to-many or many-to-many comparisons. In direct similarity checking, processing each subject of the original file with the subjects of the query file the similarity ratio is given for each subject according to the number of words that match with the query subject and the total number of words pertaining to such subjects. Finally, the average similarity ratio is obtained according to the number of subjects in the original file.

Conversely, in deep similarity ratio each word of a subject of the original file will be passed to the WordNet and obtain the synonyms of such words and then they will be compared with simultaneous words of the query subject. Removing some common words like “A”, “a”, “The”, “the”, etc. may be used to maximize the efficiency and reduce the processing time of the deep checking. The objects of the original document are also considered in the same process of subject and obtain the similarity ratio.

Changing the voice of a sentence is a common practice exercised by plagiarizers. With regard to this incident the framework uses the NLP approach again by extending the above mentioned subject-object analysis. When changing the active voice sentence to passive voice the object of the active sentence becomes the subject and simultaneously, the subject of the active voice sentence becomes the “agent” of the passive voice sentence. After analyzing the subjects and the objects of the two documents the system makes four files which contain subjects and objects of both the original and suspicious documents. Cross references of these four files can be made in order to detect the passive voice and active voice changes.

3.6 Authorship Attribution/Verification

All the methodologies explained above may consist of algorithms for a relative comparison of the document with other documents and then relatively make decisions. Eventually, there are four significant drawbacks should have been faced in this context. Firstly, there should be a large corpus to compare the documents. Secondly, the corpus should be subjectively related to the document which is to be checked. Thirdly, the availability of electronic source document(s) which might have been used by the plagiarizer i.e. the plagiarizer may use the hard copies of some books which are not available in any corpus or in the Internet. Eventually there are no such tools to cover all the inherent abilities of the plagiarizer. The framework proposes a distinct method for this including authorship attribution especially authorship verification and it is generally called Intrinsic Plagiarism Detection (IPD).

Mainly, IPD is based on Stylometry. In section 2.8.1 of chapter 02 describes the Stylometry and its variations which are used in authorship verification. 49 Stylometry features are used in the method proposed by the framework of authorship verification pertaining to the IPD.

In IPD, only one document is used for identifying plagiarism. Because of this nature, the identification of a correct idiolect of the author is difficult. Conversely, if the writer uses documents of several other authors it exactly is a hindrance in identifying several authors and will be a crucial problem. However, in order to simplify the problem, all the attributes of the authorship attribution or authorship verification are not taken into account. In the experiments, the main focus is to get a decision whether the document is plagiarized or not by establishing a simple hypothesis called "*if more than one idiolect exists in the document, it may be a plagiarized document*".

According to the established hypothesis the document may contain different classes of idiolects and these are not known. Excavation of unknown classes from the given input is the task of unsupervised learning in machine learning (Barlow, 1989). Thus, unsupervised learning approach is more appropriate to check the above hypothesis. Typically, as the definition of the unsupervised learning do not give the class of the examples which the machine is going to learn. The feature set is entered to the system and the learning system should cluster the available classes according to the variations of the features and the Self Organizing Map (SOM) is selected for the clustering purpose (see section 2.8.1 of chapter 02).

The above hypothesis emphasizes that; a given document may contain different classes of significant idiolects. These idiolects are not known and an exact representation of these unknown classes of idiolects is the task of unsupervised machine learning, especially SOM in this context. Typically, as the definition of the unsupervised learning do not give the class of the examples which the machine is going to learn. The feature set is entered into the system and the SOM should cluster the available classes according to the variations of the features.

3.6.1 Input Space of the Model

While the document is being preprocessed it is converted to a text file and some unnecessary elements such as figures and figure captions, tables and table names etc. will be removed by the preprocessor. Since the IPD clearly bases on the analysis of one document and all the evidence should be extracted from the entire document, the segmentation of the document is done. This is the critical point of the whole process and should be done very carefully because the author's writing styles may vary. The segment size θ has been declared as threshold on the experiments. The number of words is used to measure the lengths of both the document and the segment. The number of segments of the document d should be proportionate with the document length. Then $n = \frac{d}{\theta}$ where n is the number of segments in document $d.$ If s denotes a segment of a document $d,$ feature extraction is made on each s_i segment of $d.$

The author's writing style attributes can be quantified by the style makers and let $s_1, s_2, .. s_n$be the segmentation of d into n contiguous, non-overlapping segments. Let m denote the number of styles makers and $\tau_1, \tau_2, .., \tau_m$ be the quantified styles of the segment s and $s = (\tau_1 \cdots \tau_m)$ denotes the segment of the document d and then the input feature space of the model can be represented in $n \times m$ vector space per document d as

$$d = \begin{pmatrix} s_1: \tau_1, \tau_2, .., \tau_m \\ \vdots \\ s_n: \tau_1, \tau_2, .., \tau_m \end{pmatrix} \qquad \text{------------------ (3.9)}$$

3.6.2 Classification of the Used Feature

Stylometry features were explained in literature section 2.8.3 of chapter 2. There are forty-nine features used in the experiments covering almost all the aspects of the previously defined features in the literature. As simple ratios, such as the total number of characters, average length per word, number of sentences, words per sentences etc are included. Six word based features are used, such as words longer than six characters, the total number of short words, the number of syllables, the syllables per word, the number of complex words (more than 3 syllables) and the number of specific words. Nine features are used to measure the vocabulary richness as standard authorship attribution like Hapax legomena, Hapax dislegomena, Yule's K measure, Simpson's D measure, Sichel's S measure, Harden's V measure, Brunets W measure, Honore's R measure, and Average Word Frequency Class. Syntactic and POS features are also used and there are nineteen such features including the number of nouns, the number of passive verbs, the number of base verbs, the number of adjectives, and also the number of clauses and the number of phrases. Since many authors irregularly attempt to use adverbs on their own, other than these features pertaining to the POS features adverbs are also extracted as domain, duration, frequency, focus, locating, manner, promina, and sequence. The number of articles, the number of prepositions, the number of coordinate conjunctions and the number of auxiliary verbs are also used as the syntactic features. With regard to readability measures, Flesh Index, Kincaid Index, and Fog Index are used. The study introduces the punctuation measures which is not in the literature as important and they include the number of commas, the number of single quotes ('), the number of double quotes

('), the number of colons (:), the number of semi-colons (;), the number of question marks (?), the number of exclamation marks (!) and the number of "etc.".

3.6.3 Experimental Setup for IPD

Experiments have been carried out on a document collection which includes freely downloadable genuine 19th century English Books [www 17]. Table 3.1 shows the corpus setup according to each author.

Author	Document	No of Words Selected
Thoreau	Walden	5000
	A Week on Concord River	5000
Emerson	Conduct of Life	5000
	English Traits	5000

Table 3.1: Selected documents and the document size of the selected text

Four books of two authors from American essayists called Thoreau and Emerson are selected. 5000 word segments from the beginning of each document are used for creating the test data. Firstly, the selected word segments from the two different authors are mixed and two documents called "Walden with Conduct" and "Concord with English Traits" are created. Secondly, 5000 word segments from same author are selected and two documents are created by mixing such segments and these documents are named "Walden with Concord" and "Conduct with English Traits". These four documents are used in the experiments.

Tables and figures are removed from all documents while preprocessing. AutoSOME tool is used for conducting the experiments [www 18]. The tool consists of several parameters which can be used to enhance the clustering performances such as Ensemble Runs, SOM Iterations, SOM Grid Size, SOM Topology, SOM Error Exponent, SOM Distance Metric, Cartogram XY Size, Clustering Method, MST P-value. There are several normalization techniques other than those parameters that can be applied for normalizing the dataset. Log2 Scaling, Unit Variance, Median Center, Sum of Squares=1 for both columns and rows are some normalizing facilities available in the tool.

Four documents are segmented into 100, 150, 200, 250, 300, 350, 400, 450, 500 word segments. Each document under each segment is fed to extract the features and these extracted feature values are analyzed by using a genetic algorithm based feature selection to filter the best feature set from all features of each document. Finally, each selected feature file under each number of segmentation is been put into the AutoSOME tool and the result is obtained for analysis.

3.7 Chapter Summary

The core of the proposed framework for plagiarism detection encapsulates six methods. Boolean Model, Normalized Vector Space Model, Fingerprinting model, Document Formatting Property Analyzer model, Syntactic and Semantic Analyzer Model and Authorship Verification Model for Intrinsic Plagiarism Detection are the six methods which are deeply explained in this chapter. These models do not work individually in the framework and are designed for covering the detection of all the cheating attempts of the plagiarizer. First three models are designed for detecting copy and paste plagiarism and Document Formatting Property Analyzer concerns similarities of the structure of the document. Conversely, Syntactic and Semantic Analyzer Model presents the methods for detecting illegal paraphrasing. Finally, Intrinsic Plagiarism Model gives a method for detecting plagiarism without having a source document.

Chapter 4 – Experiments and the Results

4.1 Introduction

The framework introduced in the chapter 3 has practically been implemented as a system called MAPDetect. This chapter presents the experiments and their results which are obtained from the implemented modular architecture. The proposed algorithms have been implemented and tested in four modules:

- Information Retrieval / Fingerprinting Module
- Document Property Analysis Module
- Syntactic and Semantic Analysis Module
- Authorship Verification Module (Intrinsic Plagiarism Detection)

The result of the experiments are analyzed with the Precision, Recall and the ‘F’ value measures which are obtained by contingency tables derived from testing results of each algorithm. First and second algorithms directly compare with manually calculated plagiarism ratios. The *Paired T-Test* has been used for obtaining the significance of the result over the manual ratios.

4.2 Data Sets for Experiments

The proposed algorithms of the above module one and two are tested with the customized corpus which is built by using real assignments submitted by the university students of UCSC. It is expected that the word variance of the matched pair of documents may significantly affect to the final result of these algorithms of module one and two. This expectation is addressed by using two datasets. Firstly, sixteen plagiarized Microsoft Word documents are selected randomly from one subject without considering the document length and it is called as dataset 01. Secondly, another twenty Microsoft Word documents have been selected by considering the word difference of the document pairs from 500 to 1000 words as dataset 02. A portion of university registration number of the students who submit the document is assigned as a document name for shortening the file names. Last module is tested with the document collection mentioned in the section 3.6.4 in chapter 03.

4.3 Experiments and the Results of Boolean Plagiarism Detection Model (BPDM)

The model described in section 3.1 of chapter 03 is implemented and tested with two experiments. Initially, the documents of two sets are evaluated manually for verbatim coping.

Sentence level similarities are considered and the number of words included in each similar sentence is counted in this n to n document comparison. The most similar document is identified by comparing one document with all the others. A ratio value is assigned to each document as the following method. This ratio depends on the total count of similar words in identical sentences or paragraphs. The percentage value is based on the number of similar words in the most similar document relative to the total number of words in the original document. Table 4.1 shows the paired documents and the given ratio of each document with the highest similarity rate. Hence, the given ratio for each document is denoted by Plag % for all experiments and it represents the percentage of the plagiarism. The manually detected Plag % of each data set is used as a benchmark for testing all the algorithms of modules one and two.

Original Document	**Highest Plagiarized Document**	**Total Number of Words in OD**	**Total Number of Words in Highest CD**	**Manually Identified Number of Similar Words**	**Plag %**
581.txt	635.txt	1998	4506	207	10.36
591.txt	613.txt	2629	3769	180	6.84
594.txt	607.txt	2714	2042	55	2.02
595.txt	613.txt	2338	3769	684	29.25
600.txt	635.txt	2285	4510	29	1.26
604.txt	643.txt	3738	2272	79	2.11
607.txt	617.txt	2042	2568	168	8.22
613.txt	595.txt	3769	2338	684	18.14
617.txt	604.txt	2568	3728	170	6.61
627.txt	633.txt	2093	1692	209	9.98
633.txt	627.txt	1692	2093	209	12.35
635.txt	613.txt	2285	3769	101	4.42
640.txt	604.txt	3560	3728	221	6.20
643.txt	604.txt	2272	3738	320	14.08
644.txt	613.txt	2720	3769	335	12.31
649.txt	607.txt	3132	2042	128	4.08

Table 4.1: Results of the manual calculation with the highest Plag % of paired documents (OD = Original Document and CD = Copied Document)

Each file is preprocessed before applying the BPDM according to the methods explained in the section 3.2 of chapter 03. BPDM algorithm creates the $n \times n$ matrix and Table 4.1 of Appendix I illustrates this 16×16 matrix which shows the highest similarity file according to the number of exact matches on words. Table 4.2 summarizes the results obtained from the table 4.1 of Appendix I. It represents the paired documents according to the highest similarity as number of similar words given by this Model. Calculation of Plag% relative to the original document is not fare because of the preprocessing has been done on these documents and the number of words actually used in the original document may reduce by the preprocessor i.e. stemming and eliminating common words. This problem is addressed by using the diagonal of the above matrix which represents the number of hits on the same document and it is the number of words used by the algorithm for original document.

Original Document	**Highest Plagiarized Document**	**Number of Words in OD**	**Number of Words in CD**	**Number of Hits on OD**	**Number of Matched Words**	**Plag %**
581.txt	635.txt	1998	4506	700	311	44.42
591.txt	613.txt	2629	3769	985	408	41.42
594.txt	635.txt	2714	4506	952	369	38.76
595.txt	613.txt	2338	3769	804	484	60.19
600.txt	635.txt	2285	4506	859	381	44.35
604.txt	635.txt	3738	4506	1165	466	40.00
607.txt	613.txt	2042	3769	859	387	45.05
613.txt	595.txt	3769	2338	1182	484	40.94
617.txt	604.txt	2568	3738	920	378	41.08
627.txt	635.txt	2093	4506	607	299	49.25
633.txt	635.txt	1692	4506	578	293	50.69
635.txt	604.txt	2285	3738	1328	466	35.09
640.txt	604.txt	3560	3738	696	355	51.00
643.txt	635.txt	2272	4506	885	384	43.38
644.txt	613.txt	2720	3769	715	348	48.67
649.txt	635.txt	3132	4506	543	218	40.14

Table 4.2: Summary of the obtained Plag % with paired documents and similarity score (OD = Original Document and CD = Copied Document)

The experiments only study how many documents are classified by the model as plagiarized. The basic matrices of Receiver Operator Characteristics (ROC) are used for analyzing the performance of the generated results (Richard, et al. 2001). Four outcomes are generated in the two-by-two matrix for one instance of a test such as true positive (TP): if the instance is plagiarized and it is identified as plagiarized, false negative (FP): if the instance is not plagiarized and identified as plagiarized, true negative (TN): if the instance is not plagiarized and it is identified as not plagiarized, and false positive (FN): if the instance is plagiarized and it is identified as not plagiarized. The matrix is also called contingency table and the table 4:3 illustrates the sample. Precision and recall (sensitivity) measures are calculated on the data which is provided by the contingency tables derived from each algorithm.

		True Class	
		Positive **P**	**Negative** **N**
Tested Class	Y	**TP** (true positive)	**FP** (false positive)
	N	**FN** (false negative)	**TN** (true negative)

Table 4.3: A Sample of two-by-two confusion matrix

In this context, Recall is defined as the number of actual plagiarized documents detected by the algorithm divided by the total number of existing plagiarized documents. Precision is defined as the number of actual plagiarized documents detected by the algorithm divided by the total number of hits on the document set. Equation 4.1 illustrates the calculation of precision by using data provided by the contingency table while recall calculates using equation 4.2

$$Precision = \frac{tp}{tp + fp} \qquad \text{------------------ (4.1)}$$

$$Recall = \frac{tp}{tp + fn} \qquad \text{------------------ (4.2)}$$

F measure can be obtained by combining these two measures. The equation 4.3 can be used to measure the F value.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \qquad \text{------------------ (4.3)}$$

Obviously, the number of words actually concerned on manual detection and number of word selected by the algorithm for calculating Plag % can be differed. Hence, the separation threshold of whether a document is plagiarized or not is set for more than 7.0 for manual experiment and more than 40.0 for the results obtained by this algorithm. The contingency table is derived by this basis and the table 4.4 shows the results of the true class and the tested class. There are nine actually plagiarized documents. Although there are some similar sentences containing in other documents, they are actually not plagiarized. Conversely, eleven documents are detected by the algorithm as plagiarized and other five documents are testified as not plagiarized.

		True Class	
		P	N
Tested Class	P	8	3
	N	1	4

Table 4.4: Contingency table for the first experiment on BPDM

Only 8 documents are detected as highest plagiarized documents by the BPDM according to the manual experiments. The performance of the algorithm on this experiment is shown in table 4.5.

Precision	**Recall**	**F Measure**
0.73	0.89	0.8

Table 4.5: Performance matrices obtained from the experiment by BPDM

BPDM detects 89% of such plagiarized documents while detecting 73% of correct hits from the total detection. Calculated F measure is 0.8.

The second corpus is used in the same experiment. This experiment is focused on to identify the effect of word difference between the document pairs for the final result. Experiment is carried out following the same method and the manually calculated result is shown by table 4.6.

Original Document	**Highest Plagiarized Document**	**Number of Words in OD**	**Number of Words in CD**	**Number of Matched Words**	**Plag %**
09MS032	09MS069	540	879	419	77.59
09MS066	09MS006	887	805	147	16.57
09MS069	09MS102	879	614	498	56.66
09MS080	09MS016	830	989	305	36.75

09MS003	09MS069	504	879	58	11.51
09MS005	09MS078	692	699	435	62.86
09MS006	09MS085	805	963	790	98.14
09MS012	09MS016	733	989	406	55.39
09MS016	09MS078	989	699	474	47.93
09MS024	09MS051	560	565	553	98.75
09MS025	09MS069	700	879	30	4.29
09MS030	09MS006	563	805	235	41.74
09MS039	09MS078	965	699	453	46.94
09MS047	09MS016	802	830	14	1.75
09MS051	09MS024	565	560	553	97.88
09MS058	09MS085	729	963	284	38.96
09MS061	09MS039	593	965	156	26.31
09MS078	09MS039	699	965	453	64.81
09MS085	09MS006	963	805	790	82.04
09MS069	09MS102	879	614	498	56.66

Table 4.6: Manually calculated results of second experiment (OD = Original Document and CD = Copied Document)

BPDM has been used in the second experiment too and the results are presented in table 4.7. Since the most identical two documents named 09MS024 and 09MS051 are found in this corpus, a special attention is made in this experiment. Additionally, only four documents are found as non-plagiarized and the other sixteen documents are testified as verbatim plagiarism by at least two paragraphs.

Original Document	**Highest Plagiarized Document**	**Number of Words in OD**	**Number of Words in CD**	**Number of Hits on OD**	**Number of Matched Words**	**Plag %**
09MS032	09MS069	540	879	293	203	69.28
09MS066	09MS006	887	805	436	153	35.09
09MS069	09MS102	879	614	438	205	46.80
09MS080	09MS016	830	989	397	158	39.79
09MS003	09MS069	504	879	237	100	42.19
09MS005	09MS069	692	879	377	197	52.25
09MS006	09MS085	805	963	423	290	68.56

09MS012	09MS016	733	989	368	184	50.00
09MS016	09MS078	989	699	463	192	41.47
09MS024	09MS051	560	565	268	268	100.00
09MS025	09MS069	700	879	353	110	31.16
09MS030	09MS006	563	805	275	154	56.00
09MS039	09MS078	965	699	415	209	50.36
09MS047	09MS006	802	805	389	87	22.36
09MS051	09MS024	565	560	268	268	100.00
09MS058	09MS085	729	963	385	174	45.19
09MS061	09MS039	593	965	311	152	48.87
09MS078	09MS039	699	965	359	209	58.22
09MS085	09MS006	963	805	426	290	68.06
09MS102	09MS069	879	614	319	205	64.26

Table 4.7: Results of the second experiment by the BPDM (OD = Original Document and CD = Copied Document)

The calculated contingency table is shown in tables 4.8 and 4.9 which show the performance obtained by the algorithm in the second experiment.

		True Class	
		P	N
Tested Class	P	16	1
	N	0	3

Table 4.8: Contingency table for the second experiment on BPDM

Precision	**Recall**	**F Measure**
0.94	1.00	0.97

Table 4.9: Performance matrices obtained from second experiment by BPDM

After minimizing the variation of the number of words the BPDM produces good results. A precision rate of 0.94% and 100% recall rate have been obtained. The calculated F measure is 0.97. Although both datasets contains very similar documents according to the F measure of the two experiments, the second experiment gives a greater significant result.

This experiment highlights some significant incidences according to the result tables. The most identical documents remarked by the manual calculation are also ranked highest by the algorithm. The other higher rank is shown over the document 09MS085 in manual calculation is again identified as third highest by the BPDM.

Empirically, it is identified that the variation of the obtained value and the actual value of the similarity usually become high except for the similarities approximate to 100%. Practically, this incidence is true since the Plag % ratio is calculated by the number of hits on the similar pairs of words and all those similar pairs are not considered as plagiarized in manual calculations. Even though the pattern of the identification is similar, the error of the two functions should be measured. Therefore, a *Paired T-Test* has to be used to measure the difference between the actual detection and the detection from the algorithm. There are three significant reasons for using this kind of test to measure the variation ,firstly, the incidence has matched pairs of scores (e.g., two measures per document). Secondly, each pair of scores is independent of every other pair and thirdly, the sampling distribution is assumed as normal.

Null hypothesis is placed as "*There are zero differences between the mean value of manual ratios and the BPDM ratios*" and an alternative hypothesis is used as "*There are differences between the mean value of manual ratios and the BPDM ratios*". Table 4.10 shows the calculated statistics of the two tailed paired t-test which is done with a high confidence level of 99%.

t-Test: Paired Two Sample for Means		
	Manual	*BPDM*
Mean	54.49828	52.39723
Variance	398.0862	949.5928
Observations	20	20
Pearson Correlation	0.887585	
Hypothesized Mean Difference	0	
Df	19	
t Stat	0.586979	
P(T<=t) one-tail	0.282064	
t Critical one-tail	1.729133	
P(T<=t) two-tail	0.564128	
t Critical two-tail	2.093024	

Table 4.10: Calculated statistics of the two tailed paired t-test.

According to table 4.10 the null hypothesis is accepted and it determines that this conclusion is highly significant. It is evident that both statistics 2.09 and 0.564128 respectively say there is no mean difference with the given confidence level of 99%.

4.4 Experiments and the Results of Normalized Vector Space Model for Plagiarism Detection (NVSM)

The NVSM is run on the first document and compared against the first benchmark. The obtained result of the highest similarity document for each file is illustrated in the table 4.11. The ranking process is done according to the cosine similarity described in the section 3.3.2 of chapter 03. Figure 4.1 of Appendix I represents a segment of calculated term vectors by the java implementation on the vocabulary of the document set. According to the method explained in the section 3.3.2 of chapter 03 tf-idf is also calculated and figure 4.2 in Appendix I illustrates the segment of the text file.

Original Document	Most Suspicious Document	Plag %
581.txt:	644.txt:	11
591.txt:	613.txt:	14
594.txt:	607.txt:	06
595.txt:	613.txt:	26
600.txt:	635.txt:	13
604.txt:	644.txt:	07
607.txt:	640.txt:	18
613.txt:	595.txt:	26
617.txt:	640.txt:	12
627.txt:	635.txt:	9
633.txt:	613.txt:	8
635.txt:	604.txt:	7
640.txt:	607.txt:	18
643.txt:	604.txt:	9
644.txt:	613.txt:	15
649.txt:	613.txt:	5

Table 4.11: Results of NVSM on the first document set

Same threshold as 7.0 for manual detection is selected for separating the documents to determine whether they are plagiarized or not. The Plag % given by the NVSM model does not reflect the level as same as that of the BPDM. So the same threshold can be used to separate the plagiarized document placing it above 7.0. Values of the contingency table are selected on this thresholds and table 4.12 shows those values.

		True Class	
		P	N
Tested Class	P	9	3
	N	0	4

Table 4.12: Contingency table for NVSM on first document set

Table 4.13 presents the obtained performance by the NVSM algorithm on the first document set.

Precision	Recall	F Measure
0.75	1.0	0.72

Table 4.13: Performance obtained by NVSM on first document set

Considering the above performance of the first document set the second document set is also tested with the NVSM algorithm. Table 4:14 shows the detected most suspicious documents and its Plag % by the algorithm.

Original Document	Most Suspicious Document	Plag %
09MS032	09 MS069	23
09MS066	09MS006	20
09MS069	09MS102	23
09MS080	09MS069	12
09MS003	09MS069	11
09MS005	09MS078	18
09MS006	09MS085	45
09MS012	09MS016	19
09MS016	09MS078	20
09MS024	09MS051	100
09MS025	09MS102	10
09MS030	09MS006	28
09MS039	09MS078	23
09MS047	09MS066	8
09MS051	09MS024	100
09MS058	09MS085	22
09MS061	09MS078	14
09MS078	09MS039	23
09MS085	09MS006	45

09MS102	09MS069	22

Table 4.14: Results of NVSM on second document set

Considering the same fact in the previous experiment the threshold (7.0) is used in the manual detection and, since the similarity values get higher in this experiment, the threshold of the NVSM is changed to 12.0 to determine whether the detected documents are plagiarized or not. The contingency table is shown in table 4.15.

		True Class	
		P	N
Tested Class	P	15	1
	N	1	3

Table 4:15: Contingency table for NVSM on second document set

Table 4.16 presents the obtained detection performance by the NVSM algorithm on the second document set.

Precision	**Recall**	**F Measure**
0.94	0.94	0.94

Table 4.16: Performance obtained by NVSM on second document set

In the same way, the three most plagiarized documents identified in the manual detection are also detected by the NVSM with a high significant accuracy. 09MS051 and 09MS024 documents are identical and detected as plagiarized with a ratio of 100% and 09MS006 and 09MS085 documents come third in the manual evaluation and the same result is given by the NVSM.

Same null hypothesis is used as "*There are zero differences between the mean values of manual ratios and NVSM ratios*" and the alternative hypothesis is placed as "*There are differences between the mean value of manual ratios and the NVSM ratios*" to get the significance statistics for these ratios obtained. . The calculated statistics of the two tailed paired t-test done on 95% of confidence level is shown in table 4.17.

The null hypothesis should be rejected and the alternative hypothesis must be accepted according to the given statistics of the table 4.17. The probability of accepting the zero difference is 0.0001 and it is very poor. There is a very strong reason for this difference. Contextually, these two measures are laid in different backgrounds. The manual ratio is built by counted similar sentences but in the NVSM, documents are subjected to preprocessing and the ratio is perfectly based on the correlation which is calculated on the term vectors.

t-Test: Paired Two Sample for Means		
	Manual	***NVSM***
Mean	52.5	29.3
Variance	948.2631579	676.7473684
Observations	20	20
Pearson Correlation	0.7231654	
Hypothesized Mean Difference	0	
df	19	
t Stat	4.804337558	
P(T<=t) one-tail	6.1629E-05	
t Critical one-tail	1.729132792	
P(T<=t) two-tail	0.000123258	
t Critical two-tail	2.09302405	

Table 4.17: Calculated statistics of the two tailed paired t-test on NVSM

Even though the pattern of the detection ratios is similar the error value between two ratios become high and the rejection of the null hypothesis must not be unfamiliar.

However, the error distribution can be used to make an approximation parameter for mapping the two distributions. The mean of the error distribution is 21.59 and the standard deviation is 23.2. It is empirically observed that, pertaining to a given document, the NVSM Plag % may vary by ±23.2 from the mean error. This is a large variance and it is possible to consider the mean difference as not zero but approximate at least by ±23.2 from mean error to obtain a high significance.

Does this approximation significant for all datasets? This question can be answered by testing another different dataset on same method. Randomly selected thirty genuine MS word documents in same course are used to test the above conclusion. The obtained result is shown in tables 4.2 and 4.3 of Appendix I. The Plag% of the new experiment must be laid between the same range of the above ±23.2. The mean error of this experiment is 15.97 and the standard deviation of the error is 19.16. It means that the ±19.1 should be added to the mean Plag % to obtain the actual plagiarism ratio. Although the mean error value is less than the second experiment value and the mean error value is laid on the second experiment value. It empirically proves that less than ±23.2 should be added to the mean error in order to obtain the actual PLag%. Table 4:18 shows the performance of the third experiment.

Precision	**Recall**	**F Measure**
0.96	0.93	0.94

Table 4.18: Performance of the third experiment on NVSM

No significant difference is shown between these two experiments and obtains the same results according to the F measure.

4.5 Experiments and the Results of Fingerprinting Model for Plagiarism Detection (FMPD)

The approach explained in the section 1.3 of chapter 03 is implemented as java program and the above two document sets were utilized to test the proposed algorithm called Fingerprinting Model for Plagiarism Detection (FMPD). The same manual detections of the datasets also are used to compare the results. The experiments are done at different levels of granularity and fingerprinting resolutions according to section 3.3.3 of chapter 03. Figure 4.3 of Appendix I represents the number of correct detections which are obtained on different granularity and fingerprint resolution levels. The selected granularity differs from 3 to 8 and fingerprint resolution also changes into two classes as *granularity-1 class* and *equal number of granularity*.

A low level of granularity and fingerprinting resolution such as three is not suitable according to figure 4.3 of Appendix I. Four and six levels give a high number of hits. The figure 4.3 of Appendix I clearly shows that the optimum detection capability as thirteen documents from sixteen documents is obtained, when using four granularity levels and four fingerprinting resolution levels.

Finally, four by four granularity and fingerprinting resolution is used to compare the performance of the algorithm. The table 4.19 represents the obtained results. The manual threshold (greater than 7.0) is assigned for separating the documents to determine whether they are plagiarized or not and a threshold of FMPD is also assigned as 10.0.

Original Document	Most Suspicious Document	Plag %
581.txt	613.txt	12.5
591.txt	613.txt	11.3
594.txt	607.txt	10
595.txt	613.txt	18
600.txt	635.txt	8.6
604.txt	643.txt	9.7
607.txt	643.txt	11.3
613.txt	595.txt	13.4
617.txt	604.txt	10.7
627.txt	633.txt	11.2
633.txt	627.txt	12.3
635.txt	613.txt	9.6

640.txt	604.txt	11.1
643.txt	604.txt	12.2
644.txt	613.txt	13.4
649.txt	607.txt	9.8

Table 4.19: Results obtained on dataset 1 by FMPD with four granularity and fingerprinting resolution

		True Class	
		P	N
Tested Class	p	9	0
	N	1	6

Table 4.20: Contingency table for FMPD on first document set

Table 4.20 illustrates the contingency table which is derived by comparing the above results with the manual detection.

Table 4.21 presents the obtained detection performance by the FMPD algorithm on the first document set.

Precision	**Recall**	**F Measure**
1	0.9	0.94

Table 4.21: Performance obtained by FMPD on first document set

The second document set is also tested by using the same classes of granularity and resolution and figure 4.4 of Appendix I illustrates the correct number of detections in each class.

The highest number of hits represents in the four granularities and the four fingerprint resolutions which is similar to the first dataset. The results given by the class four-by-four are illustrated by the table 4.22.

Original Document	**Most Suspicious Document**	**Plag %**
09 MS 032.txt	09MS 102.txt	23.71
09 MS 066.txt	09MS 085.txt	10.24
09 MS 069.txt	09MS 102.txt	15.76
09ms080.txt	09 MS 069.txt	12.15
09MS 003.txt	09 MS 069.txt	13.08
09MS 005.txt	09MS 078.txt	19.06
09MS 006.txt	09MS 085.txt	34.3
09MS 012.txt	09MS 016.txt	15.52
09MS 016.txt	09MS 078.txt	15.69
09Ms 024.txt	09MS 051.txt	100
09MS 025.txt	09 MS 069.txt	10.47

09MS 030.txt	09MS 006.txt	16.94
09Ms 039.txt	09MS 078.txt	17.29
09MS 047.txt	09 MS 069.txt	10.01
09MS 051.txt	09Ms 024.txt	100
09MS 058.txt	09MS 085.txt	15.42
09MS 061.txt	09Ms 039.txt	11.76
09MS 078.txt	09Ms 039.txt	21.42
09MS 085.txt	09MS 006.txt	33.97
09MS 102.txt	09Ms 069.txt	20.13

Table 4.22: Results obtained on dataset 2 by FMPD with four granularity and fingerprinting resolution

Table 4.23 represents the contingency table comparing the results of both manual detection and the results of FMPD in table 4.22.

		True Class	
		P	N
Tested Class	P	16	0
	N	0	4

Table 4.23: Contingency table for FMPD on the second document set

Table 4.24 presents the obtained detection performance by the FMPD algorithm on the second document set. Although the BPDM and the NVSM show the variations of obtained performance among the two document sets the FMPD provides more stable performance on two document sets and finally, the second document set gives100% optimum results.

Precision	**Recall**	**F Measure**
1	1	1

Table 4.24: Performance obtained by FMPD on the second document set

Although the ratios given by the FMPD are low and the manual ratios are relatively high the two distributions take the same pattern.

The mean error of the above two distribution is 26.6 and the standard deviation also is 22.9. It is expected that it should be a greater value than the NVSM value. Because the document is highly preprocessed and divided into n-grams for creating fingerprints and the correlation between these vectors of fingerprints are the similarity ratios of plagiarism detection. It must not be the same ratio values of one to one word mapping of the sentences. However, there are no arguments that the FMDP ratio must be as same as the manual ratio for actual plagiarism detection. It shows that the actual value must be added to or subtracted from the mean error value to get the manual ratio. Empirically, at least 23.5 should be added or subtracted for getting high accurate and significant measures and statistics for the test. The significance of the test is obtained on this consideration and the null hypothesis is placed as "*There are 23.5 differences between mean values of manual ratios and FMPD ratios*" and the alternative hypothesis is placed as "*There are more than 23.5 differences between the mean value of manual ratios and FMPD ratios*" to get the significance statistics for this obtained ratios. The calculated statistics of the two tailed paired t-test done with 99% of confidence level is shown in table 4.25.

t-Test: Paired Two Sample for Means		
	Ratio Manual	*Ratio FPDM*
Mean	52.5	25.846
Variance	948.2632	688.969
Observations	20	20
Pearson Correlation	0.687643	
Hypothesized Mean Difference	23.5	
df	19	
t Stat	0.61524	
P(T<=t) one-tail	0.272848	
t Critical one-tail	2.539483	
P(T<=t) two-tail	0.545695	
t Critical two-tail	2.860935	

Table 4.25: Calculated statistics of the two tailed paired t-test on FMPD

It is proven that the null hypothesis can be accepted in high probability as 0.54. It means, empirically the difference should be considered around the value of 26.6.

Third experiment is also done for proving the obtained consideration of the FMPD ratio. The Plag% of the third experiment of FMPD poses in the same range of the above value (26.6). The obtained result is shown in table 4.4 of appendix I. The mean error of this experiment is 24.32 and the standard deviation of the error is 22.62. It means that the 22.62 should be added to obtain the actual plagiarism ratio of the FMDP. Although the value is less than the previous value it is laid in par with the previous and it empirically proves that less than 23.5 should be added to obtain the actual Plag%.

Precision	**Recall**	**F Measure**
1	0.93	0.96

Table 4.26: Performance of the third experiment on FMDP

4.6 Experiments and the Results of Document Property Analyzer (DPA)

The basic idea of the DPA is comparing the formatting properties of a document. As mentioned in section 3.4 of chapter 03 the documents are not subjected to preprocessing in these experiments. The first and second document sets given above are used to conduct the experiments of DPA. The first document set is run in the java implementation of the DPA and the obtained $F \times D$ matrix is represented in figure 3.4 of chapter 03. Table 4.27 shows the results of the experiment which gives the obtained ratio of each document.

Original Document	Most Suspicious Document	Plag %
581.txt:	644.txt:	0.81
591.txt:	613.txt:	0.84
594.txt:	607.txt:	0.79
595.txt:	613.txt:	0.89
600.txt:	635.txt:	0.70
604.txt:	644.txt:	0.91
607.txt:	640.txt:	0.69
613.txt:	595.txt:	0.89
617.txt:	640.txt:	0.76
627.txt:	635.txt:	0.91
633.txt:	613.txt:	0.68
635.txt:	604.txt:	0.52
640.txt:	607.txt:	0.78
643.txt:	604.txt:	0.84
644.txt:	613.txt:	0.52
649.txt:	613.txt:	0.81

Table 4.27: Results of DPA on the first document set

The same threshold is used to separate the documents in manual detection as 7.0 and discrimination threshold of the DPA is considered as 80. Since totally different facts are used to calculate the ratio the Plag % given by the DPA are higher values for all documents. Table 4.28 shows the contingency table for DPA on the first document set.

		True Class	
		P	N
Tested Class	P	6	2
	N	3	5

Table 4.28: Contingency table for DPA on the first document set

Precision	Recall	F Measure
0.75	0.67	0.71

Table 4.29: Performance obtained by DPA on first document set

Table 4.29 presents the obtained detection performance by the DPA algorithm on the first document set. A good performance is not maintained by the DPA in this experiment relative to the BPDM, NVSM and FMPD.

The second document set is tested and figure 4.5 of Appendix I illustrates the $F \times D$ matrix of the document collection. This matrix contains the formatting properties and its weights on ff-idf for each document in the collection. There are some formatting properties according to the figure 4.5 of Appendix I which are not important for discriminating the documents. Some properties obtain zero values for most of the documents and 0.85 is empirically used as dimensionality reduction threshold as mentioned in section 3.4.of chapter 03.

Original Document	Most Suspicious Document	Plag %
09MS032.txt	09MS025.txt:	0.9
09MS066.txt	09MS006.txt:	0.85
09MS069.txt	09MS047.txt:	0.97
09MS080.txt	09MS016.txt:	0.93
09MS003.txt	09MS080.txt:	0.76
09MS005.txt	09MS006.txt:	0.86
09MS006.txt	09MS005.txt:	0.86
09MS012.txt	09MS069.txt:	0.78
09MS016.txt	09MS085.txt:	0.95
09MS024.txt	09MS051.txt:	1
09MS025.txt	09MS069.txt:	0.96
09MS030.txt	09MS006.txt:	0.8
09MS039.txt	09MS047.txt:	0.95
09MS047.txt	09MS069.txt:	0.97
09MS051.txt	09MS024.txt:	1
09MS058.txt	09MS078.txt:	0.75
09MS061.txt	09Ms039.txt:	0.96
09MS078.txt	09MS058.txt:	0.75
09MS085.txt	09MS016.txt:	0.95
09MS102.txt	09MS016.txt:	0.94

Table 4.30: Results of DPA on the second document set

Table 4.30 shows the results obtained by comparing the correlation of the document vectors illustrated in figure 4.5 of Appendix I by using the equation 3.8 mentioned in section 3.4 of chapter 03.

The given ranking values of the DPA vary in a small range from 0.75 to 1.00. The first document set also shows this phenomenon. The similarity ratio does not vary in a large range because most of the properties of the document set are used by the authors. This experiment is also subjected to measure the performance and table 4.31 presents the contingency table.

		True Class	
		P	N
Tested Class	P	13	3
	N	3	1

Table 4.31: Contingency table for DPA on the second document set

Table 4.32 presents the obtained detection performance by the DPA algorithm on the first document set.

Precision	**Recall**	**F Measure**
0.81	0.81	0.81

Table 4.32: Performance obtained by DPA on the second document set

The DPA gives out better results in this experiment than the experiment on first document set. A significant feature is that the two identical documents – 09 MS 024 and 09 MS 051 are detected as identical by other algorithms and this algorithm also detects that they resemble. In the manual checking also on these documents give the same result. When the ratios given by the DPA are compared with the manual detections they are relatively similar in most documents but in documents 09MS006, 09MS016, 09MS025, 09MS030, and 09MS061 give a different pattern.

4.7 Experiments and the Results of Syntactic and Semantic Analyzing Model for Plagiarism Detection (SSAMPD)

The analyzed models given above heavily depend on the verbatim copying while the documents are ranked by using a particular string matching algorithm. However, the FMPD has another capability like detecting positional changes on some segments of the document. Mainly, SSAMPD is introduced to detect the illegal paraphrasing of two documents as mentioned in section 3.5 of chapter 03. The section explains the techniques used by the plagiarizer to hide his/her plagiarism. Since all those techniques cannot be identified in the original documents like the above mentioned document sets one and two the experiments of SSAMPD is run on manually created plagiarized documents.

The method explained in section 3.5 of chapter 03 is basically designed to identify the changes on the verb, subject and object of the sentences of a particular document. It is assumed that the verb changes may be done in four ways as discussed in the same section and subject and the object changes mainly depend on the use of synonyms. The algorithm uses direct and deep checking to identify obfuscations of the subjects and objects of sentences as explained in section 3.5 of chapter 03. An original document and a query document are made manually including all the obfuscations discussed earlier and they are used by the model for the purpose of testing.

4.7.1 Experiments on Verb Changes

An experiment is designed to test the approach on changing tenses, using passive voice, using similar verbs and changes from singular to plural and from plural to singular. One simple sentence is used as the original document and that sentence is changed according to the above mentioned obfuscations.

A sentence “The students go to the school.” Is used for original sentence and the following parse tree is generated by the Stanford Parser. The figure 4.1 shows the result pane of the implementation and it is indicated as number 1.

```
(ROOT (S [37.177] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.946] (VBP [4.947] go) (PP [12.607] (TO [0.003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002].)))(S
```

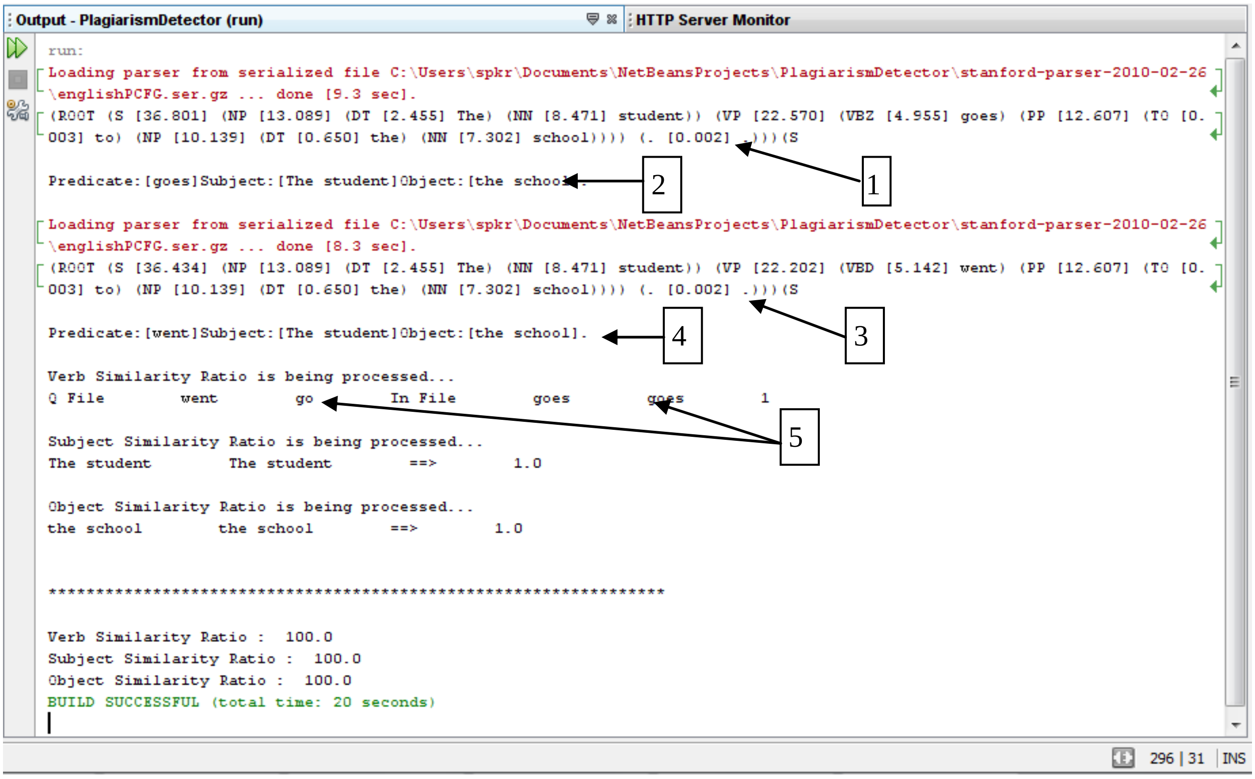
Firstly, the following predicate argument structure has been made by the model and it is stored in the text file. It is indicated as number 2 in the figure 4.1.

Predicate:Verb:[go]Subject:[The students]Object:[the school].

The verb “go” in the original sentence has been changed to “went” in the query sentence. The generated parse tree and the predicate argument structure are mentioned below and they are indicated as number 3 and 4 in the figure 4.1.

```
(ROOT (S [36.434] (NP [13.089] (DT [2.455] The) (NN [8.471] students)) (VP [22.202] (VBD [5.142] went) (PP [12.607] (TO [0.003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S
```

Predicate:Verb:[went]Subject:[The students]Object:[the school].

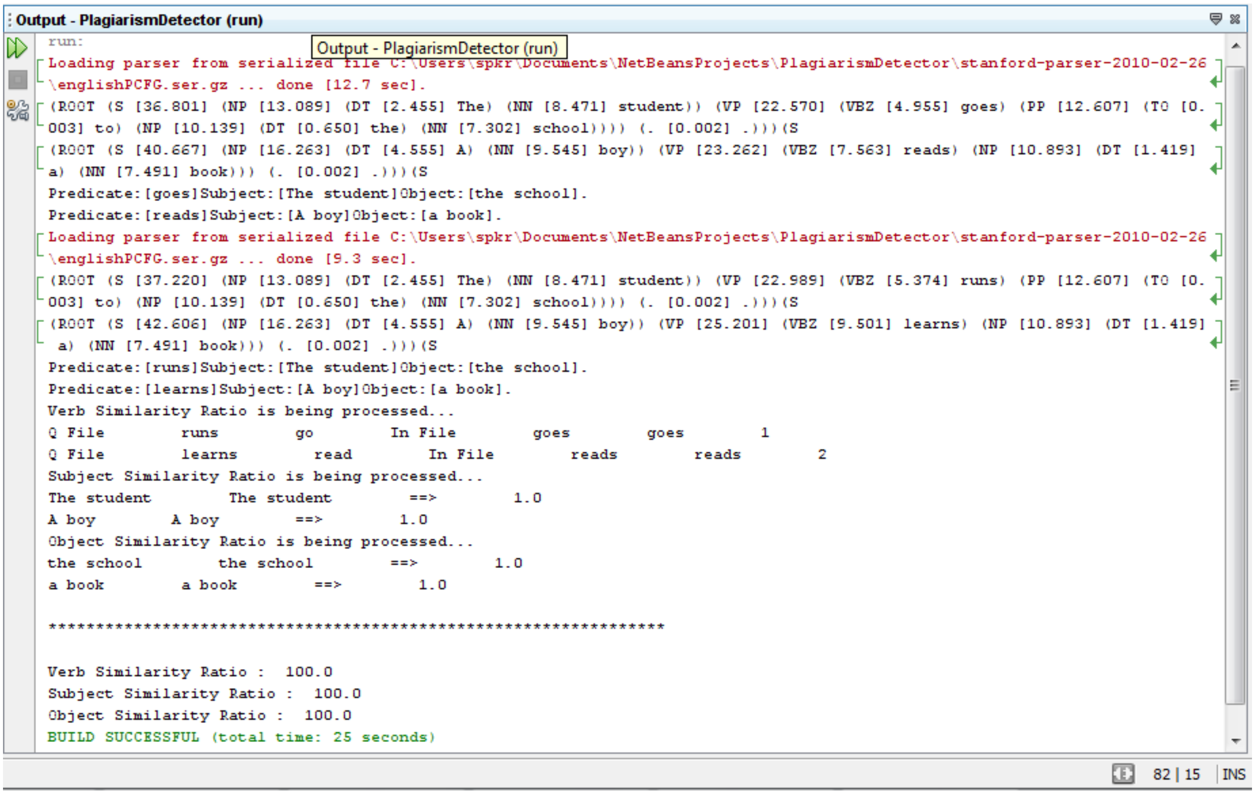


```
run:
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [9.3 sec].
(ROOT (S [36.801] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.570] (VBZ [4.955] goes) (PP [12.607] (TO [0.
003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S

Predicate:[goes]Subject:[The student]Object:[the school].

Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [8.3 sec].
(ROOT (S [36.434] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.202] (VBD [5.142] went) (PP [12.607] (TO [0.
003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S

Predicate:[went]Subject:[The student]Object:[the school].

Verb Similarity Ratio is being processed...
Q File        went        go        In File        goes        goes        1

Subject Similarity Ratio is being processed...
The student         The student         ==>         1.0

Object Similarity Ratio is being processed...
the school         the school         ==>         1.0


*****************************************************************

Verb Similarity Ratio :  100.0
Subject Similarity Ratio :  100.0
Object Similarity Ratio :  100.0
BUILD SUCCESSFUL (total time: 20 seconds)
```

Figure 4.1: Result pane of the java implementation (changing tense of verb)

Although the verb of the query sentence differ in tense model identify the meaning of the sentence and it indicates the verbs are similarly derived from “go”. Number 5 of figure 4.1 shows the actual incident and finally model gives 100% similarity ratio for both sentences.

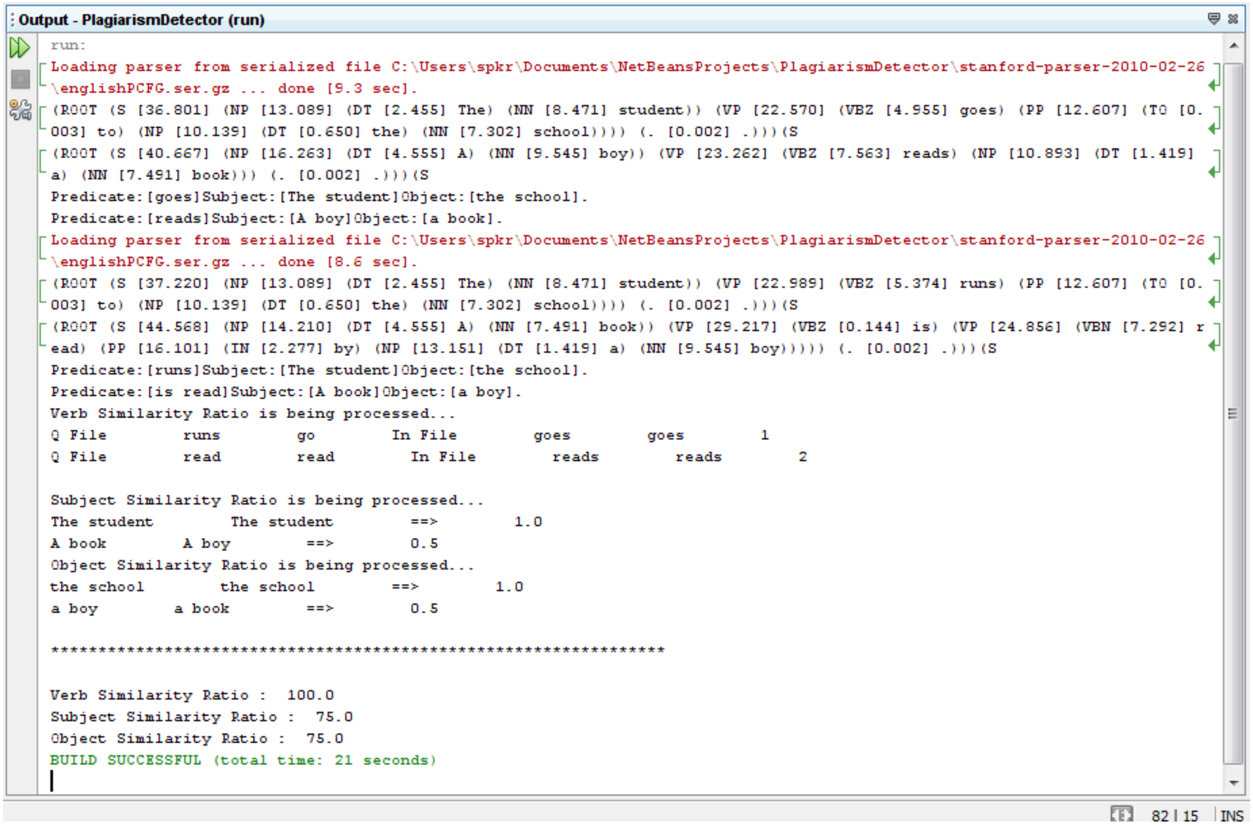
Accordingly, verbs also can be used to express the same idea. The verb “run” is used instead of ‘walk’ in the query sentence. Another two sentences are also used in the experiment. The model uses deep checking with WordNet for this type of sentences and the result pane is presented in the figure 4.2. Although the two combinations of verbs in the two sentences “goes/runs” and “reads/learns” are lexically differed they are identified by the model as similar verbs. The model detects these sentences as identical with a 100% similarity ratio by following the meaning of the two sentences.


Output - PlagiarismDetector (run)
Output - PlagiarismDetector (run)
82 | 15
INS

```
run:
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [12.7 sec].
(ROOT (S [36.801] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.570] (VBZ [4.955] goes) (PP [12.607] (TO [0.
003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S
(ROOT (S [40.667] (NP [16.263] (DT [4.555] A) (NN [9.545] boy)) (VP [23.262] (VBZ [7.563] reads) (NP [10.893] (DT [1.419]
a) (NN [7.491] book))) (. [0.002] .)))(S
Predicate:[goes]Subject:[The student]Object:[the school].
Predicate:[reads]Subject:[A boy]Object:[a book].
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [9.3 sec].
(ROOT (S [37.220] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.989] (VBZ [5.374] runs) (PP [12.607] (TO [0.
003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S
(ROOT (S [42.606] (NP [16.263] (DT [4.555] A) (NN [9.545] boy)) (VP [25.201] (VBZ [9.501] learns) (NP [10.893] (DT [1.419]
 a) (NN [7.491] book))) (. [0.002] .)))(S
Predicate:[runs]Subject:[The student]Object:[the school].
Predicate:[learns]Subject:[A boy]Object:[a book].
Verb Similarity Ratio is being processed...
Q File        runs        go        In File        goes        goes        1
Q File        learns        read        In File        reads        reads        2
Subject Similarity Ratio is being processed...
The student        The student        ==>        1.0
A boy        A boy        ==>        1.0
Object Similarity Ratio is being processed...
the school        the school        ==>        1.0
a book        a book        ==>        1.0

*****************************************************************

Verb Similarity Ratio :  100.0
Subject Similarity Ratio :  100.0
Object Similarity Ratio :  100.0
BUILD SUCCESSFUL (total time: 25 seconds)
```

Figure 4.2: Results pane of the java implementation (using similar verbs)

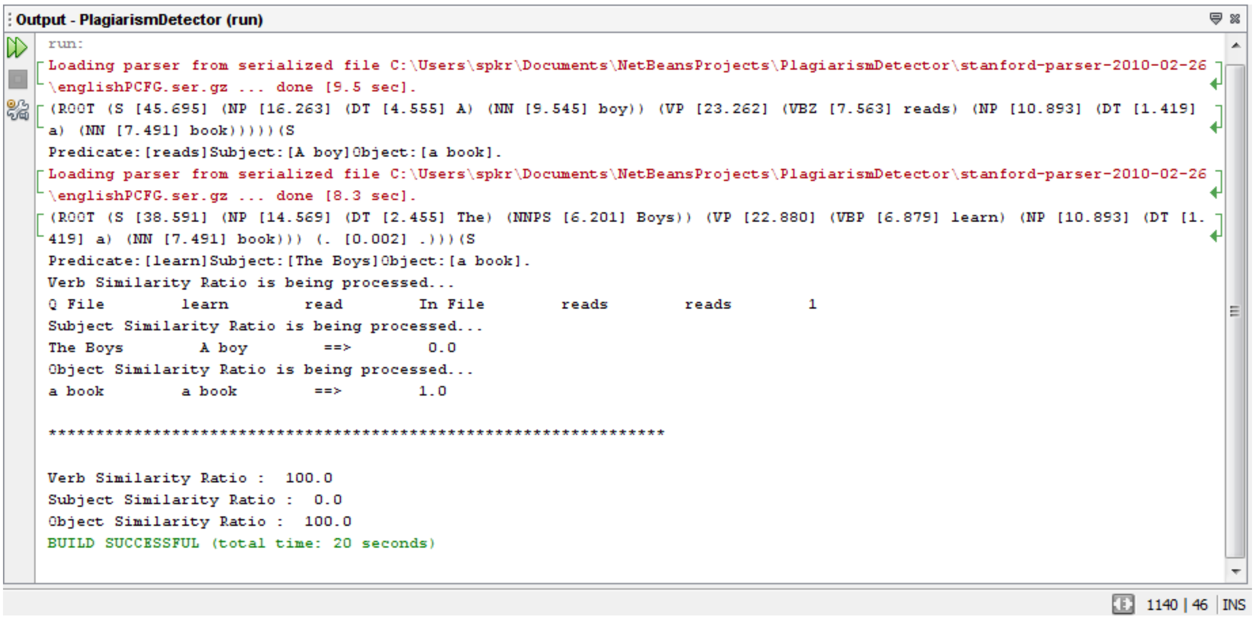
An experiment on changing sentences to passive voice is also checked by converting the second sentence into passive voice. The verb of the sentence is changed while the subject and the object are also changed. Figure 4.3 shows the parse trees and the results obtained from the model. The defined predicate argument structure by the model for “A boy reads a book” is *Predicate: Verb[reads]Subject:[A boy]Object:[a book]*. The sentence changed into passive voice as “A book is read by a boy” and the defined structure is *Predicate: Verb [is read]Subject:[A book]Object:[a boy]*. Although the subject of the first sentence should be changed as agent the model similarly indicates it as object. However, the verb similarity ratio is 100% and the subject and object similarity ratios are changed. Further, the plagiarizer can also change the voice with another similar verb like “is learned”. It will also be identified as same meaning by the model with the above capabilities.



```
run:
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [9.3 sec].
(ROOT (S [36.801] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.570] (VBZ [4.955] goes) (PP [12.607] (TO [0.
003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S
(ROOT (S [40.667] (NP [16.263] (DT [4.555] A) (NN [9.545] boy)) (VP [23.262] (VBZ [7.563] reads) (NP [10.893] (DT [1.419]
a) (NN [7.491] book))) (. [0.002] .)))(S
Predicate:[goes]Subject:[The student]Object:[the school].
Predicate:[reads]Subject:[A boy]Object:[a book].
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [8.6 sec].
(ROOT (S [37.220] (NP [13.089] (DT [2.455] The) (NN [8.471] student)) (VP [22.989] (VBZ [5.374] runs) (PP [12.607] (TO [0.
003] to) (NP [10.139] (DT [0.650] the) (NN [7.302] school)))) (. [0.002] .)))(S
(ROOT (S [44.568] (NP [14.210] (DT [4.555] A) (NN [7.491] book)) (VP [29.217] (VBZ [0.144] is) (VP [24.856] (VBN [7.292] r
ead) (PP [16.101] (IN [2.277] by) (NP [13.151] (DT [1.419] a) (NN [9.545] boy))))) (. [0.002] .)))(S
Predicate:[runs]Subject:[The student]Object:[the school].
Predicate:[is read]Subject:[A book]Object:[a boy].
Verb Similarity Ratio is being processed...
Q File        runs        go        In File        goes        goes        1
Q File        read        read        In File        reads        reads        2

Subject Similarity Ratio is being processed...
The student        The student        ==>        1.0
A book        A boy        ==>        0.5
Object Similarity Ratio is being processed...
the school        the school        ==>        1.0
a boy        a book        ==>        0.5

*****************************************************************

Verb Similarity Ratio :  100.0
Subject Similarity Ratio :  75.0
Object Similarity Ratio :  75.0
BUILD SUCCESSFUL (total time: 21 seconds)
```

Figure 4.3: Results pane of the java implementation (changing to passive voice)

The verb of a sentence is one of the main parts to indicate the singularity or the plurality. A sentence can be changed from singular to plural or from plural to singular without distorting the main idea.



```
run:
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [9.5 sec].
(ROOT (S [45.695] (NP [16.263] (DT [4.555] A) (NN [9.545] boy)) (VP [23.262] (VBZ [7.563] reads) (NP [10.893] (DT [1.419]
a) (NN [7.491] book)))))(S
Predicate:[reads]Subject:[A boy]Object:[a book].
Loading parser from serialized file C:\Users\spkr\Documents\NetBeansProjects\PlagiarismDetector\stanford-parser-2010-02-26
\englishPCFG.ser.gz ... done [8.3 sec].
(ROOT (S [38.591] (NP [14.569] (DT [2.455] The) (NNPS [6.201] Boys)) (VP [22.880] (VBP [6.879] learn) (NP [10.893] (DT [1.
419] a) (NN [7.491] book))) (. [0.002] .)))(S
Predicate:[learn]Subject:[The Boys]Object:[a book].
Verb Similarity Ratio is being processed...
Q File        learn        read        In File        reads        reads        1
Subject Similarity Ratio is being processed...
The Boys        A boy        ==>        0.0
Object Similarity Ratio is being processed...
a book        a book        ==>        1.0

*****************************************************************

Verb Similarity Ratio :  100.0
Subject Similarity Ratio :  0.0
Object Similarity Ratio :  100.0
BUILD SUCCESSFUL (total time: 20 seconds)
```

Figure 4.4: Results pane of the java implementation (changing to singular to plural)

Figure 4.4 shows the results of changing a singular sentence to a plural sentence by using a similar verb as “A boy reads a book” change to “The boys learn a book”. According to the example the string similarity algorithm may not identify the meaning of ‘reads’ and will not take ‘reads’ and ‘learn’ as the same.

4.7.2 The Experiment on an Actual Paraphrasing

Finally, original paraphrased example from the Internet[1] is used to cover all the aspects of the experiments on this model. The original paragraph is:

“*During the last decade, there has been a shift from “instructivist” approaches towards “constructivist” approaches in the field of instructional design. Instructivist approaches keep the belief that the role of knowledge is basically to represent the real world. Meaning is determined by this real world and is thus external to the understander*.”

The paraphrased paragraph is:

“Over the last ten years, there has been a marked change from “instructivist” points of view to “constructivist” points of view among instructional designers. Instructivist points of view hold the belief that the role of knowledge is fundamentally to represent the real world. In this view, meaning is ascertained by the real world and is therefore external to the learner.”

Since the result pane for this paragraphs is too long it is difficult to display as a figure. The copied segments of the original result pane are used to analyze the final result of this experiment. The parse trees of the first paragraph are generated by the model as follows.

```
(ROOT (S [163.279] (PP [19.783] (IN [4.479] During) (NP [14.971] (DT [0.650] the)
(JJ [3.925] last) (NN [7.416] decade))) (, [-0.000] ,) (NP [4.519] (EX [0.433]
there)) (VP [135.848] (VBZ [0.028] has) (VP [132.297] (VBN [0.001] been) (NP
[88.134] (NP [11.176] (DT [1.419] a) (NN [7.985] shift)) (PP [76.552] (IN [3.524]
from) (NP [71.281] (NP [30.508] (`` [0.039] ``) (JJ [12.883] instructivist) (‘’
[0.042] ‘’) (NNS [9.021] approaches)) (PP [40.232] (IN [8.966] towards) (NP
[30.864] (`` [0.039] ``) (JJ [12.883] constructivist) (‘’ [0.042] ‘’) (NNS [9.021]
approaches)))))) (PP [37.238] (IN [1.552] in) (NP [34.030] (NP [9.980] (DT [0.650]
the) (NN [7.558] field)) (PP [23.509] (IN [0.666] of) (NP [22.441] (JJ [11.058]
instructional) (NN [7.810] design))))))) (. [0.002] .)))(S
```

```
(ROOT (S [119.253] (NP [25.324] (JJ [12.548] Instructivist) (NNS [9.021]
approaches)) (VP [92.787] (VBP [6.333] keep) (NP [11.584] (DT [0.650] the) (NN
[8.951] belief)) (SBAR [66.943] (IN [0.637] that) (S [65.980] (NP [22.294] (NP
[9.251] (DT [0.650] the) (NN [6.829] role)) (PP [12.677] (IN [0.666] of) (NP
[11.610] (NN [9.280] knowledge)))) (VP [41.084] (VBZ [0.144] is) (ADJP [35.780] (RB
[7.060] basically) (S [23.430] (VP [23.412] (TO [0.011] to) (VP [23.382] (VB
[6.621] represent) (NP [14.727] (DT [0.650] the) (JJ [5.216] real) (NN [6.090]
world)))))))))) (. [0.002] .)))(S
```

```
(ROOT (S [98.574] (NP [17.419] (NN [13.718] Meaning)) (VP [80.013] (VP [33.449]
(VBZ [0.144] is) (VP [28.445] (VBN [5.886] determined) (PP [21.097] (IN [2.277] by)
(NP [18.146] (DT [3.859] this) (JJ [5.216] real) (NN [6.090] world))))) (CC [0.106]
and) (VP [42.291] (VBZ [0.144] is) (ADVP [6.186] (RB [5.859] thus)) (ADJP [28.427]
(JJ [9.391] external) (PP [16.880] (TO [0.003] to) (NP [14.988] (DT [0.650] the)
(NN [12.151] understander)))))) (. [0.002] .)))(S
```

Following predicate argument structures are generated by the model for the above parse trees.

[1] https://www.indiana.edu/~istd/example2paraphrasing.html

```
Predicate: Verb:[has been]Subject:[the last decade]Object:[a shift from ``
`` instructivist ’’ ’’ approaches towards `` `` constructivist ’’ ’’
approaches].
```

```
Predicate:    Verb:[keep]Subject:[Instructivist    approaches]Object:[the
belief].
```

```
Predicate: Verb:[is determined]Subject:[Meaning]Object:[this real world].
```

Pares trees of the second paragraph are shown in the results pane as follows

```
 (ROOT (S [188.638] (PP [33.193] (IN [5.110] Over) (NP [27.751] (DT [0.650] the)
(JJ [3.925] last) (NN [12.224] ten) (NNS [3.963] years))) (, [-0.000] ,) (NP
[4.519] (EX [0.433] there)) (VP [147.797] (VBZ [0.028] has) (VP [144.246] (VBN
[0.001] been) (NP [17.789] (DT [1.419] a) (VBN [7.440] marked) (NN [6.709] change))
(PP [42.251] (IN [2.449] from) (NP [38.146] (NP [27.058] (`` [0.039] ``) (JJ
[12.883] instructivist) (’’ [0.042] ’’) (NNS [5.571] points)) (PP [10.547] (IN
[0.666] of) (NP [9.480] (NN [7.151] view))))) (PP [73.445] (TO [0.003] to) (NP
[69.877] (NP [27.058] (`` [0.039] ``) (JJ [12.883] constructivist) (’’ [0.042] ’’)
(NNS [5.571] points)) (PP [42.278] (IN [0.666] of) (NP [39.866] (NP [9.756] (NN
[7.151] view)) (PP [29.569] (IN [5.948] among) (NP [23.219] (JJ [11.058]
instructional) (NNS [9.164] designers)))))))))) (. [0.002] .)))(S
```

```
(ROOT (S [127.406] (NP [32.215] (NP [21.301] (JJ [12.548] Instructivist) (NNS
[5.571] points)) (PP [10.547] (IN [0.666] of) (NP [9.480] (NN [7.151] view)))) (VP
[92.362] (VBP [5.486] hold) (NP [11.584] (DT [0.650] the) (NN [8.951] belief))
(SBAR [67.365] (IN [0.637] that) (S [66.402] (NP [22.294] (NP [9.251] (DT [0.650]
the) (NN [6.829] role)) (PP [12.677] (IN [0.666] of) (NP [11.610] (NN [9.280]
knowledge)))) (VP [41.505] (VBZ [0.144] is) (ADJP [36.202] (RB [7.481]
fundamentally) (S [23.430] (VP [23.412] (TO [0.011] to) (VP [23.382] (VB [6.621]
represent) (NP [14.727] (DT [0.650] the) (JJ [5.216] real) (NN [6.090]
world)))))))))) (. [0.002] .)))(S
```

```
(ROOT (S [115.048] (PP [14.780] (IN [1.250] In) (NP [13.197] (DT [3.859] this) (NN
[7.151] view))) (, [-0.000] ,) (NP [14.832] (NN [11.131] meaning)) (VP [82.307] (VP
[35.228] (VBZ [0.144] is) (VP [30.224] (VBN [10.874] ascertained) (PP [17.887] (IN
[2.277] by) (NP [14.937] (DT [0.650] the) (JJ [5.216] real) (NN [6.090] world)))))
(CC [0.106] and) (VP [42.806] (VBZ [0.144] is) (ADVP [6.701] (RB [6.374]
therefore)) (ADJP [28.427] (JJ [9.391] external) (PP [16.880] (TO [0.003] to) (NP
[14.988] (DT [0.650] the) (NN [12.151] learner)))))) (. [0.002] .)))(S
```

The generated predicate argument structures of the second paragraph are displayed in the following lines.

```
Predicate:  Verb:[has  been]Subject:[the  last  ten  years]Object:[a  marked
change].
```

```
Predicate:  Verb:[hold]Subject:[Instructivist  points  of  view]Object:[the
belief].
```

```
Predicate: Verb:[is ascertained]Subject:[meaning]Object:[the real world].
```

The processing steps which were used to compare the verbs, subjects and objects are displayed next in the results pane.

```
Verb Similarity Ratio is being processed…

Q File    has been    be        In File       has been      be        1

Q File    hold        have      In File       has been      has       2

Q File   ascertained  determine In File       is determined determine 3

Subject Similarity Ratio is being processed…

last ten years        last decade                            ➔        1.0

Instructivist points of view    Instructivist approaches     ➔     0.25

meaning       Meaning                                        ➔        1.0
```

```
Object Similarity Ratio is being processed…
marked change           shift from         ➔         0.0
belief        belief                       ➔         1.0
real world        real world               ➔         1.0
```

Finally the similarity ratios of the two paragraphs are displayed by the model in the results pane in the following lines.

```
Verb Similarity Ratio :    100.0
Subject Similarity Ratio : 75.0
Object Similarity Ratio :  66.66667
BUILD SUCCESSFUL (total time: 32 seconds)
```

Although one verb is totally different in the lexical arrangement such as “ascertained” with “determined” the meaning is identified by the model and it gives 100% verb similarity ratio. Similarly, comparing the subject similarity “ten” is identified as “decade” with the help of the WordNet and gives the 75% similarity ratio. The object similarity ratio is obtained by direct checking and it is 66.67%. Finally, 80.5 % of semantic similarity has been provided by the model between these two paragraphs.

4.8 Experiments and the Results of Intrinsic Plagiarism Detection Model (IPDM)

The final model of the proposed plagiarism detection framework is IPDM. Basically, the experiments are conducted to identify the correct Stylometry features and the performance of the proposed model with Self Organizing Maps. Four documents are created by mixing the 5000 word segments mentioned in table 3.1 of chapter 03. Two experiments are designed by using these four documents and table 4.33 shows the mixing structure of the four documents.

Experiment No	Document Name	Authorization
First Experiment	Walden with Conduct	Different Author
	Walden with Concord	Same Author
Second Experiment	English Traits with Concord	Different Author
	English Traits with Conduct	Same Author

Table 4.33: Mixed structure of the four documents in two experiments

According to section 3.6.1 of chapter 03 the documents are converted into text files and are preprocessed. While the preprocessing is in on progress the segmentation of the document also is performed. The segmentation is based on number of words such as 100, 150, 200, 250, 300, 350, 400, 450, 500 and nine segmented files per each document are created by the java implementation of the feature extraction system. The feature set which is explained in section

3.6.2 of chapter 03 is extracted in each segmented text file. Figure 4.6 of Appendix I illustrates the sample feature extraction of the segmented text file. The segments are named as Sec1, Sec2 and so on. The feature values are in column vectors.

After the feature file is created by the system for each segment the feature selection procedure is run on each feature file. Genetic Algorithm is used as the feature selection method. The mutation rate of the genetic algorithm may vary in each segmentation between 0.1 to 1.0 and finally identifies the good clustering performance which can be obtained at 1.0 mutation rate. The files including selected features are stored under each segment. Finally, these selected feature files in the text format are run with SOM for clustering. AutoSOME tool is used to run the files with SOM. Since the identification of most appropriate SOM parameter set as mentioned in section 3.6.3 of chapter 03 one feature file may run several times with the AutoSOME tool. Finally, the most suitable parameter set of SOM is used by all the selected feature files of each segmentation. A sample running of AutoSOME tool is illustrated by Figure 4.7 in Appendix I.

4.8.1 The First Experiment and the Results

The experiment is mainly divided into two as same author and different author. The two documents which are already mixed together called "*Walden and Conduct*" and "*Walden and Concord*" are used in this experiment. The former is used for testing 'different author' and later is used for testing 'same author'. Only one cluster should appear in the former and in the two clusters for the latter. Nine successful SOM runs are included in each file and finally there are eighteen files. Parameters set up in both testing are equal. Since there is no opportunity that represents all SOM runs as one sample of the result pane is shown by figure 4.8 in Appendix I. Only two clusters appear in the figure and it is generated by the first file.

Table 4.34 shows the number of clusters of experiments obtained on "*Walden and Conduct"* (different author). There should be two clusters for each segmentation and the results obtained by the model should be correctly in all segmentations.

Segment	**100 Seg**	**150 Seg**	**200 Seg**	**250 Seg**	**300 Seg**	**350 Seg**	**400 Seg**	**450 Seg**	**500 Seg**
No of Clusters	2	2	2	2	2	2	2	2	2

Table 4.34: Clusters obtained by the models in each segmentation for two different authors

The clustering performance obtained by the model for this experiment is shown in Table 4.35. The segmentation 200 reveals the optimum clustering performance. However, all segmentations give correct results in this experiment. Hence, the SOM can be used to detect the styles of authors.

	100 Seg	150 Seg	200 Seg	250 Seg	300 Seg	350 Seg	400 Seg	450 Seg	500 Seg
Precision	0.90	0.88	1.00	1.00	0.94	1.00	1.00	1.00	1.00
Recall	0.53	0.52	0.74	0.56	0.54	0.56	0.52	0.5	0.59
F measure	0.67	0.65	0.85	0.72	0.69	0.72	0.68	0.67	0.74

Table 4.35: Clustering performance of the model in each segmentation on two different authors

The second part of the experiment one is testing the clustering capability of mixed documents of the same author. Only one cluster should be given for this experiment. The number of clusters given by the model in each segmentation is shown in table 4.36.

Segment	**100 Seg**	**150 Seg**	**200 Seg**	**250 Seg**	**300 Seg**	**350 Seg**	**400 Seg**	**450 Seg**	**500 Seg**
No of Clusters	1	2	1	1	2	1	1	2	2

Table 4.36: Clusters obtained by the model in each segmentation for same author

Inaccurate results are given by 150 and 300 segments and also 450 and 500. The reason for having two clusters in the 450 and 500 segmentations is, when the number of words are increased in the segment the model attempts to discriminate the styles of the same author according to the topic or the theme. It is clearly shown by the experiment. The clustering performance obtained is shown in table 4.37. Optimum clustering performance is given in the segmentations of 200, 250 and 350.

	100 Seg	150 Seg	200 Seg	250 Seg	300 Seg	350 Seg	400 Seg	450 Seg	500 Seg
Precision	0.91	0.97	1.00	1.00	0.94	1.00	1.00	1.00	1.00
Recall	0.78	0.94	1.00	1.00	0.54	1.00	0.5	0.5	0.53
F Measure	0.84	0.95	1.00	1.00	0.69	1.00	0.67	0.67	0.69

Table 4.37: Clustering performance of the model in each segmentation on two different authors

It is identified that very few outliers have affected to obtain more than two clusters in the segments 450 and 500. For example, figure 4.9 of Appendix I shows the clustering results and the signal plot of features. It clearly shows that only 3 segments are included out of 64 segments in the inaccurate cluster. Those are outliers and an outlier detection method can be used to identify them. However, 200 segmentation gives the most significant results in both detections.

All the 49 features are not used by each clustering and the significant feature set is selected by the genetic algorithm for each segmentation. The experiment is focused to find out the most suitable features for detecting the different clusters in both testing's. Table 4.1and 4.2, of Appendix II analyzes all the selected and unselected features of this experiment and table 4.3 of Appendix II shows the selected features for the most significant result of the discriminating authors.

The remarkable fact is that most of the traditional features are not selected and the newly introduced features are selected by the genetic algorithm according to Appendix II. For example, some traditional features are used in authorship attribution and verification like Hapax legomena, Hapax dislegomena, Simpson's D measure, Brunets W measure, Average Word Frequency Class (AWFC) etc. are not selected. The adverbial features which have been newly introduced in the intrinsic plagiarism detection get priority and seven features out of nine are selected. Conversely, both numbers of clauses and phrases extracted by using NLP are also significant in this experiment. Finally, the punctuation measures which are not in the literature of intrinsic plagiarism detection play a good role in this experiment and five measures out of eight are selected as significant for the experiment.

4.8.2 The Second Experiment and the Results

This experiment is done in the same way as the first experiment is done except for the documents used. The two documents which are already mixed together called "*English and Concord"* and "*English and Conduct*" are used for this experiment and the former is used for testing different authors and the latter is used for testing for the same author. Only two clusters are expected from the former and one cluster from the latter. Nine successful SOM processing are included in each file and finally, there are eighteen files. Both tests use equal parameter setup of SOM.

The number of clusters of experiments obtained on "*English and Concord"* (different authors) are shown in table 4.38. Although the expected number of clusters is two an incorrect result is given by two segmentations only.

Segment	**100 Seg**	**150 Seg**	**200 Seg**	**250 Seg**	**300 Seg**	**350 Seg**	**400 Seg**	**450 Seg**	**500 Seg**
No of Clusters	3	2	2	3	2	2	2	2	2

Table 4.38: Clusters obtained by the model in each segmentation for two different authors in the second experiment

It is also as same as the above and the four outliers out of one hundred are affected in the 100 segmentation and 03 outliers out of thirty nine affected in the 250 segmentation.

The clustering performance obtained by the model for this experiment is shown in Table 4.39. The segmentation 450 obtains the optimum clustering performance.

	100 Seg	**150 Seg**	**200 Seg**	**250 Seg**	**300 Seg**	**350 Seg**	**400 Seg**	**450 Seg**	**500 Seg**
Precision	0.71	0.91	0.92	0.85	1.00	0.92	1.00	1.00	0.8
Recall	0.58	0.53	0.52	0.47	0.5	0.52	0.52	0.55	0.5
F Measure	0.64	0.67	0.66	0.61	0.67	0.66	0.68	0.71	0.62

Table 4.39: Clustering performance of the model in each segmentation on two different authors in the second experiment

The mixed document of the same author is experimented in the second part of the experiment. The expected clusters are one in this experiment. The number of clusters given by the model in each segmentation is shown in table 4.40.

Segment	**100 Seg**	**150 Seg**	**200 Seg**	**250 Seg**	**300 Seg**	**350 Seg**	**400 Seg**	**450 Seg**	**500 Seg**
No of Clusters	1	2	1	2	2	1	1	1	1

Table 4.40: Clusters obtained by the model in each segmentation for the same author

Inaccurate clustering is given by 150, 250 and 300 segmentations. Even though more than 300 word segments give significant results on this experiment 200 word segmentation also gives the optimum performance. It is clearly shown in table 4.41.

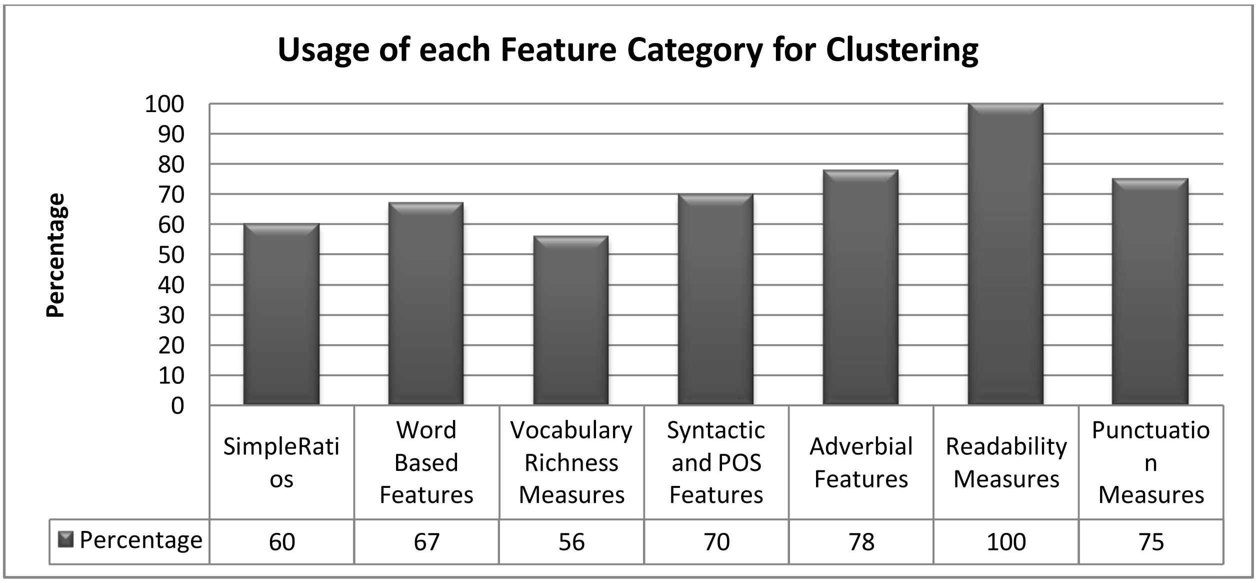
	100 Seg	**150 Seg**	**200 Seg**	**250 Seg**	**300 Seg**	**350 Seg**	**400 Seg**	**450 Seg**	**500 Seg**
Precision	1.00	0.97	1.00	1.00	0.94	1.00	1.00	1.00	1.00
Recall	1.00	0.91	1.00	0.83	0.6	1.00	1.00	1.00	1.00
F Measure	1.00	0.94	1.00	0.91	0.73	1.00	1.00	1.00	1.00

Table 4.41: Clustering performance of the model in each segmentation on two different authors

The selected features for the second experiment are also analyzed in Tables 4.4 and 4.5 in Appendix II. The features selected in the most significant result are analyzed in Table 4.6 in Appendix II. The same nature can be seen in these experiments. For example, some traditional features like Hapax legomena, Brunets W measure, Sichel's S measure, Harden's V measure, Honore's R measure, Flesh Index, Average Word Frequency Class (AWFC) as well as the features based on a number of words including the average length per word, Total number of short words per words are analyzed. The adverbial features do not involve significantly in the second experiment and only three features out of seven are selected. . Both the number of clauses and the number of phrases extracted by using NLP are also significant in this experiment like the first experiment. Finally, the majority of punctuation measures are also used to identify the clusters.

Further, it is important to analyze the features which are involved in both experiments. Appendix 02 analyzes the features and the summary of the analysis is sown in table 4.7 of Appendix II. In the section 3.6.3 of chapter 03 the Stylometry features used by the model are typically categorized into six simple ratios- vocabulary richness measures, Syntactic and POS features, adverbial features, readability measures, and punctuation measures. As shown in table 4.7 of Appendix II, some features of documents have a zero frequency including the total number of words (NW), the average length per word (ALCW), Syllables per words (SPW), the No. of complex words (More than 3 Syllables) (NCW), Hapax legomena/N (HL), Sichel's S measure (SS), and so on.

Table 4.8 of Appendix II and figure 4.5 present the summarization of categorized features on participation of the clustering the four major documents in both experiments. The usage of readability measures get 100% and the newly introduced adverbial and punctuation measures also have played major roll of the clustering according to the figure 4.5. The minimum usage of 56% is presented by the vocabulary richness measures. The selection of features vary on the nature of the document However, the four experiments cited above give more evidence for using 35 features out of 49 for further experiments in intrinsic plagiarism detection and the newly introduced features are more effective than the traditional features.



	SimpleRatios	Word Based Features	Vocabulary Richness Measures	Syntactic and POS Features	Adverbial Features	Readability Measures	Punctuation Measures
■ Percentage	60	67	56	70	78	100	75

Figure 4.5: The percentage of usage of features in each category

4.9 Chapter Summary

In this chapter the main study of this research is experimented and the results are outlined. Especially, the framework proposed as the methodology in chapter 03 is implemented and tested. The three actual document sets submitted by the university students are utilized as benchmark for testing the proposed algorithms of the four components such as Boolean plagiarism Detection Model, Normalized Vector Space Model for Plagiarism Detection, Fingerprinting Model for Plagiarism Detection and Document Property Analyzer of the framework. The three document sets are manually examined carefully and the plagiarized segments are detected and ranks are given for each document according to the number of plagiarized segments. This manual result is used to compare the detection performance of each of the four algorithms and the basic matrices of Receiver Operator Characteristics are used for this purpose.

Syntactic and Semantic Analysis for Plagiarism Detection are tested with the manually created sentences and finally, the original paraphrased document from the Internet is used. The actual results obtained by the java application are used in the experiment for clarifications.

Finally, the proposed Intrinsic Plagiarism Detection Model is experimented with four freely downloadable genuine 19th Century English Books. The obtained clustering performance of verifying the documents according to the referred authors is analyzed by using the basic metrics of Receiver Operator Characteristics again. The other main objective of this part of the research is proposing a new Stylometry feature set and the usage of these new features for proposed clustering the document according to the authors is also discussed in this chapter.

Chapter 5 – Conclusions and Future Work

5.1. Introduction

This chapter is mainly adopted for a conclusion on the experimental results of the chapter 4 and presents some future work pertaining to this research. Basically, the major consideration of the research which is based on the plagiarism detection in the higher education sector through the e-learning systems and how the proposed framework can fulfill such task is concluded by this chapter.

Actual document sets which are used for testing algorithms show that the university students are plagiarizing each other. Although learning management systems are utilized in the higher education sector identification of plagiarism has not yet been implemented in the sake of e-learning in the higher and university education sector in Sri Lanka. Actually, this research is identified the real situation which is going on among the university students in Sri Lanka and given the solution for such a vital problem.

5.2. Summarization and Conclusions

Mainly, prevention of plagiarism and detection of plagiarism can be used for avoiding this crucial problem from the academic community. Prevention of plagiarism rather depends on more pedagogical approach and detection of plagiarism concerns more technical approach. However, scope of this research concedes to make conclusions on detection of plagiarism and it will be more appropriate for establishing and maintaining rules and regulations to govern the problem. Hence, the pedagogical approach is not considered according to the scope of this research.

The framework has been implemented as modules and the first module mainly to be on the text similarity analysis. BPDM is given significant result for verbatim coping which is similar to the manual benchmark of verbatim coping among other methods. It is clearly proven by F measure and the paired t-test which is done on the obtained result.

BPDM analyzes the given documents according to the availability of the similar words in the original and the suspicious document. It is exact matching of words without applying any weighting factor. Such results obtained by the experiments in section 4.2 of chapter 04 highly depend on the number of words pertaining to each document pair. If number of words of the query document becomes larger than the original document, then the incorrect result is given by the algorithm. Although plagiarized sections are not in the query document the excess words will also be compared with the words of original document and similar words will be count again. This nature will give higher rank of incorrect Plag % for the non plagiarized large query document. This draw back mainly identified with the experiment 01 of the BPDM. Although the actual documents are used in the second experiment the minimum word difference is maintained. The second experiment has given promising result on this change and it shows the performance tables. F measure goes 0.8 to 0.97 after minimizing the word difference of the documents. Minimize the variance between the original document and the query document is affected directly to the result. Hence, the variance should be considered as a parameter when obtaining the result from the BPDM.

Since a weighting factor directly affected with applying ranks for the documents in the second model called NVSM is not shown the above limitation significantly. The vector space model which is mostly used in the information retrieval methods implements with some short queries and thus the needed processing time is very low. However, large number of words is included in the documents which are used as the query documents in the proposed normalized vector space algorithm. Hence, more processing time for preprocessing and post processing is needed.

Several weighting methods for reducing dimensionality have been used in experiments. The method mentioned in the chapter 3 section 1.2 called *tf-idf* was given highest appropriate result from the number of experiments were done with several other weighting techniques. However, NVSM has given similar accurate result relative to the BPDM.

The optimum results from the three experiments are made by the FPDM from three of information retrieval algorithms used in this framework. The setup of granularity and the fingerprinting resolution is important and the too small granularity is made incorrect detection because of the large number of similar chunks are made by the program incorrectly and give as the similar pairs without considering the actual partitions of the words . The same incident is happened in the large granularity too. According to the results good optimum detection capability is obtained in the four granularity in three experiments. Hence, it is suitable to conclude that the granularity value must be four and the fingerprinting resolution must also be four.

The BPDM model of the framework is more suitable for the detection of verbatim copying of the documents. Several reasons are pertaining to the idea. In the verbatim copying no changes were made by the plagiarizer and the words may appear same as the original document and hence, the Boolean algorithm will show the highest rank. The actual documents which were used in the experiments shows this phenomenon clearly and the highest ranks were obtained by the verbatim copied documents. Conversely, fingerprinting algorithm is more suitable for moving the text into different positions rather than the verbatim coping. However, actual document which were totally plagiarized from other document was detected by these three algorithms in 100% of higher rank.

At present, very sophisticated word processing software available in the market which are behaved like not only the word processing but also desktop publishing. The proposed algorithm uses these implemented functionalities of word processing software to detect the plagiarism. Although the model extracts more than fifty such properties typically all the properties are not used by the most of the students. The model tries to use the rarely used properties which are common for the document set to discriminate the documents. The used selection method is not given optimum separation and hence, the method does not perform the significant result like the above BPDM, NVSM and FPDM models.

However, the proposed concept of the framework has been proved that the one of the major technological advancement of word processing can be utilized with the plagiarism detection.

The Semantic Analyzing Model for plagiarism Detection model in the framework is basically proposed to detect illegal paraphrasing which is very difficult to detect in string similarity methods like the above algorithms. Although the proposed techniques have not been applied in actual plagiarized documents the proposed methods of SSAMPD can be used to detect such plagiarism with the proper implementation.

The implementation used in this research has some drawbacks especially processing time is drastically high in the tested documents. Creating the parse tree of each sentence by using the Stanford parser is performed very low efficiency and some time it consumed three second for one sentence. The next main drawback is when applying the technique in long sentences and long documents the memory consumption of the parse trees is very high. It is also inversely affected to the processing time. Searching senses from the WordNet database is also considerably influenced to increase the processing time.

However, the proposed method covers most of the obfuscations and it accomplishes the idea which is very important area of the plagiarism detection in the proposed framework.

Forming proper methods of detection the plagiarism externally was one of the major objectives of the framework and the above discussed algorithms has been used for achieve the objectives successfully. Conversely, internal plagiarism detection has been utilized in the framework with the authorship verification and unsupervised learning has only been limited to show the perfect methodology. It is proved that the clustering a document on the proposed feature set for identifying basic semblance of author(s) can be done by using SOM learning algorithm.

The segmentation of the analyzed document is the basic preprocessing and conventional attribute of the intrinsic plagiarism detection. Sentence segmentation has been used by most of the previous studies in the literature discussed in chapter 2. Hence, why does word segmentation use in these experiments and why does not use segmentation as sentences? These two questions should be discussed simply. Most of the used features are depended on the number of words. Generally, sentences are not contained same number of words and finally, the obtained values give incorrect result and especially, the variations may get high among the fracture values. Normalization techniques may not enough to eliminate the higher variation and simply word segmentation is the most suitable and successful solution.

The proposed method of the intrinsic plagiarism detection has other two important areas. Former is the SOM parameters and latter is the selected features. Large number of experiments has been done for identifying the SOM parameters which can be utilized to obtain the optimum clustering performance on the proposed methodology. The input feature set was adjusted and normalized on the several factors such as normalized according to the unit variance, each segment value is subjected to median center and finally both feature values and the segments values are normalized as sum of squares = 1. The maximum efficient parameters of the algorithm such as the grid length training iterations, the topological arrangement of the self organizing and cartogram resolution of the grid have been found during these numbers of experiments. The maximum grid length should be 25 and 1000 training iterations with 100 ensemble runs were given significant clustering performance. 64 x 64 cartogram resolution should also be added for this result. Although such number of normalizing techniques were been used on the input feature set several outliers have been detected in some experiments.

Although forty nine features were proposed to cluster the document in several dimensions most of them were not selected by the used genetic algorithm. Different numbers of features were selected in all experiments. The significance of each feature varies according to the style. But some features were not varied and they were 100% used by all discriminations. On the other hand zero percent on participations were made by some features which are not considered as significant for these experiments. Can those non significant features discard from the feature extracting list? Neither discards such features from the list nor manipulates the feature values differently will not be a good decision. Because of the importance depends on the author's writing style.

All the measures of the proposed MAPDetect framework cover the most important characteristics of the plagiarizer. The user who uses the framework has more evidence on various paths and hence, can give an assurance as at least one of the model of the framework capable to detect the available behaviors of plagiarism especially in higher education and universities. Although the coded algorithms are in testing mode the open source community can obtain the codes and can improve it as a plagiarism detection system.

5.3. Future Work

Since, the Plag% mainly depends on the number of words of the document the word difference of the documents should be considered as a parameter when obtaining the result from the BPDM, NVSM, and FPDM. This research manually tested this phenomenon and it should be embedded with the algorithm and should be automated.

The weighting function of the *ff-idf* of DPA algorithm must be improved to obtain the proper discrimination is another future work pertaining to the plagiarism detection on formatting properties.

Three information retrieval algorithms and the DPA algorithm have another main disadvantage. The original text is broken into necessary chunks by the preprocessor and, reproduced vectors of terms or vectors of combination of characters. Lexical arrangement of these text files is totally different from the original documents. Returning the process of preprocessing to identify the plagiarized text segments has to be done for actual identification of the plagiarized segments of the documents.

The proposed methods of syntactic and semantic analysis with natural language processing must be implemented efficiently in order to utilize with the actual documents.

User is directed to identify the plagiarism is the main task of the framework. However, combining the results of all models of the framework and give the final decision is one of the future work. The final result should be given according to the importance of type of plagiarism.

The Implementation of the MAPDetect framework in UCSC LMS and minimize the plagiarism among the students is one of the objectives of this research. Although the initial steps of this objective has been done and further testing of the system and using it as a tool is allotted as another future work.

References

1. Aleksi A., Sami S., & Mikko R. (2006). Plaggie: Gnu-licensed source code plagiarism detection engine for java exercises. In Baltic Sea '06: Proceedings of the 6th Baltic Sea conference on Computing education research, pp 141-142, New York, NY, USA, ACM.
2. Alfred, U. (2003). U-Matrix: a Tool to visualize Clusters in high dimensional Data, University of Marburg, Department of Computer Science, Technical Report, Nr.36.
3. Anonymous (1995). Defining plagiarism, Science Communication, 16 (4), pp 459–461.
4. Antonio, S., Hong V.L., & Rynson W.H.L. (1997). CHECK: A document plagiarism detection system, In Proceedings of ACM Symposium for Applied Computing, pp 70-77, February.
5. Argamon, S., & Levitan, S. (2005). Measuring the usefulness of function words for authorship attribution. In Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing.
6. Argamon, S., Whitelaw, C., Chase, P., Dhawle, S., Hota, S., Carg, N., & Levitan, S. (2007). Stylistic text classification using functional lexical features. Journal of the American Society of Information Science and Technology, 58(6), pp 802–822.
7. Ashworth, P., Bannister, P. & Thorne, P. (1997). Guilty in whose eyes? University students' perceptions of cheating and plagiarism in academic work and assessment, Studies in Higher Education, 22 (2), pp. 187–203.
8. Baayen, R., Halteren V., & Tweedie, F. (1996). Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. Literary and Linguistic Computing, 11(3), pp 121–131.
9. Barlow, H.B. (1989). Unsupervised learning. Neural Computation, MIT Press, Vol.1, No. 3, pp 295-311.
10. Barnhart, R. K. (Ed.) (1988). Chambers Dictionary of Etymology (Edinburgh, Chambers).
11. Barret, R., & Cox, A.L. (2005). At least they're learning something': the hazy line between collusion and collaboration. Journal of assessment and evaluation in Higher Education, 30 (2), 0260-2938.
12. Bresnan, J. (1995). Lexicality and Argument Structure, Paris Syntax and Semantics Conference.
13. Burrows, J.F. (1987). Word patterns and story shapes: The statistical analysis of narrative style. Literary and Linguistic Computing, 2, pp 61–70.

14. Burrows, J.F. (1992). Not unless you ask nicely: The interpretative nexus between analysis and information. Literary and Linguistic Computing, 7(2), pp 91–109.
15. Carroll, J. (2002). A Handbook for Deterring of Plagiarism in Higher Education, Oxford Center for Staff and Learning Development.
16. Carroll, J. (2004). Institutional Issues in deterring, detecting and dealing with plagiarism. http://www.jisc.ac.uk/uploaded_documents/plagFinal.pdf [01/08/2008]
17. Chris, P. (2003). In Other (People's) Words: plagiarism by university students—literature and lessons, Assessment & Evaluation in Higher Education, Vol. 28, No. 5.
18. Chi-Hong, L., & Yuen-Yan, C. (2007). Natural Language Processing Approach to Automatic Plagiarism Detection, SIGITE' 07, October.
19. Chomsky, N. (1954). Three models for the Description of Language, IEEE Transaction , 2, 3, pp 113-124.
20. Clough, P. (2000). Analyzing style - Readability. Technical report, University of Sheffield. Available from: http://ir.shef.ac.uk/cloughie/papers/readability.pdf [Accessed 2010-11-02].
21. Coulthard, R. M. (1993). Beginning the study of forensic texts: corpus, concordance, collocation, in M Hoey (ed.), Data Description Discourse,London: HarperCollins, PP 86-97.
22. Daniel, P. Luiz, O.S. Leonardo, E. J. & Batista,V. (2008). Using Conjunctions and Adverbs for Author Verification, Journal of Universal Computer Science, vol. 14, no. 18.
23. de Vel, O. Anderson, A., Corney, M. & Mohay, G. (2001). Mining e-mail content for author identification forensics.SIGMODRecord, 30(4), pp 55–64.
24. De Marneffe, M.C., MacCartney, B., & Manning, C. (2006). Generating Typed Dependency Parses from Phrase Structure Parses, Proceedings of the 5th International Conference on Language Resources and Evaluation.
25. Farringdon, J. M. (1996). Analyzing for Authorship: A Guide to the Cusum Technique. Cardiff: University of Wales Press.
26. Fei, B.K.L., Eloff, J.H.P., Olivier, M.S., & Venter, H.S. (2005). Computer Forensic Tools with self-organizing maps. IFIP WG 11.9 First International Digital Forensics Conference, Orlando Florida.
27. Fucks, W. (1952). On the mathematical analysis of style, Biometrika, vol. 39, pp 122–129.

28. Hart, M.C. & Friesner, T. (2004). Plagiarism and Poor Academic Practice - A Threat to the Extension of e-Learning in Higher Education?, Electronic Journal of eLearning, Vol. 2(1), March.

29. Heintze, N. (2000). Scalable document finger printing, USENIX Workshop of electronic commerce p 9.

30. Hiary, K.H. (2005). Water Mark: From paper texture to digital media, proceeding 1st international conference on automated production of cross media content for multi channel distribution, pp 261 – 264.

31. Hirst, G. & Feiguina, Ol'ga. (2007). Bigrams of Syntactic Labels of Authorship Discrimination of Short Texts, Literary and Linguistic Computing, To Appear.

32. Holmes, D. I., & Forsyth, R. S. (1995). The Federalist Revisited: New Directions in Authorship Attribution. Literary and Linguistic Computing 10 (2), pp 111–127.

33. Honor´e, T. (1979). Some simple measures of richness of vocabulary. Association for literary and linguistic computing bulletin 7 (2), pp 172–177.

34. Howard, R. M. (2000). Sexuality, textuality: the cultural work of plagiarism, College English, 62, pp 473–492.

35. Jones, D. (2006), Authorship gets lost on Web, USA TODAY http://www.usatoday.com/tech/news/2006-07-31-net-Plagiarism_x.htm?POE=TECISVA [01/08/2010].

36. Johnson, K. (1998). Readability [online]. Available from: http://www.timetabler.com/readable.pdf [2010-10-12].

37. Johnson, E. (1996). Lexical Change and Variation in the Southeastern United States 1930–1990. Tuscaloosa, AL: University of Alabama Press.

38. Juola, P. (2006). Authorship attribution for electronic documents. In M. Olivier & S. Shenoi (Eds.), Advances in digital forensics II pp 119–130, Boston: Springer.

39. Juola, P. (2008). Author attribution. Foundations and Trends in Information Retrieval, 1(3), pp 233 – 334.

40. Jurriaan, H., Peter, R., & Nike van V. (2010). A comparison of plagiarism detection tools. Technical Report UU-CS-2010-015, ISSN: 0924-3275, Department of Information and Computing Sciences, Utrecht University.

41. Karlgren, J., & Eriksson, G. (2007). Authors, genre, and linguistic convention. In Proceedings of the SIGIRWorkshop on Plagiarism Analysis, Authorship Attribution, and Near-Duplicate Detection, pp 23–28.

42. Koppel, M., Schler, J., & Argamon, S. (2009). Computational methods in authorship attribution. Journal of the American Society for Information Science and Technology, 60(1), pp 9 – 26.

43. Kjell, B. (1994). Authorship determination using letter pair frequencies with neural network classifiers. Literary and Linguistic Computing, 9(2), pp 119-124.

44. Kjell, B., Woods, W.A., & Frieder, O. (1995). Information retrieval using letter tuples with neural network and nearest neighbor classifiers. In IEEE International Conference on Systems, Man and Cybernetics, volume 2, pp 1222-1225, Vancouver, BC.

45. Keselj, V., Peng, F., Cercone, N., & Thomas, C. (2003). N-gram-based author profiles for authorship attribution. In Proceedings of the Pacific Association for Computational Linguistics, pp 255–264.

46. Kim, L., & Walter, D. (2008). Authorship Attribution and Verification with Many Authors and Limited Data, Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008), pp 513-520, Manchester, UK.

47. Kim, L. & Walter, D. (2008). Personae: A Corpus for Author and Personality Prediction from Text. In: Proceedings of the 6th Language Resources and Evaluation Conference (LREC), Marrakech, Morocco.

48. Klein, D., & Manning, C. (2003). Accurate Unlexicalized Parsing, Proceedings of the 41st Meeting of the Association forComputational Linguistics, pp 423-430.

49. Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE, vol. 78, no. 9, pp 1464-1480.

50. Kohonen, T. (2001). Self-organizing maps. Springer-Verlag.

51. Koppel, M., Argamon, S., & Shimoni, A. (2003). Automatically Categorizing Written Texts by Author Gender, Literary and Linguistic Computing, 17(4), pp 401-412.

52. Koppel, M., & Schler, J. (2003). Exploiting stylistic idiosyncrasies for authorship attribution. In Proceedings of IJCAI’03 Workshop on Computational Approaches to Style Analysis and Synthesis, pp 69–72.

53. Kruh, L. (1988). The Beale cipher as a bamboozlement: Part II, Cryptologia, vol. 12, no. 4, pp 241–246.

54. Manuel, Z. Marco, F. Massimo, M. & Alessandro, P. (2006). Plagiarism Detection with Multi Level Text Compression, Proceeding of the Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution(AXMEDIS’06) ,IEEE Mendenhall, T.C. (1887). The characteristic curves of composition. Science, IX, pp 237–249.

55. Mario, Z. Markus, M. Roman, K. & Michael, G. (2009). External and Intrinsic Plagiarism Detection Using Vector Space Models Stein, Rosso, Stamatatos, Koppel, Agirre (Eds.): PAN'09, pp 47-55.

56. Matsuura, T., & Kanada, Y. (2000). Extraction of authors' characteristics from Japanese modern sentences via n-gram distribution. In Proceedings of the 3rd International Conference on iscovery Science, pp 315–319. Berlin, Germany: Springer.

57. Meyer zu Eissen, S., & Stein, B. (2006). Intrinsic plagiarism detection. In M. Lalmas, A. MacFarlane, S. M. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, Proceedings of the European Conference.

58. McCabe, D. L. (2005). Cheating among college and university students: A North American perspective, *International Journal for Educational Integrity, 1*(1).

59. McEnery, T., & Oakes M. (2000). Authorship Identification and Computational Stylometry. In R. Dale, H. Moisl, and H. Somers (Eds.), Handbook of Natural Language Processing, Chapter Authorship Identification and Computational Stylometry, pp 545–562. New York: Marcel Dekker.

60. Meyer zu Eissen, S., Stein, B., & Kulig, M. (2007). Plagiarism detection without reference collections. Advances in data analysis, pp 359–366. Berlin, Germany: Springer.

61. Miguel, R. (2006). Avoiding plagiarism, self-plagiarism, and other questionable writing practices: A guide to ethical writing, Revised on-line version. http://facpub.stjohns.edu/~roigm/plagiarism/Index.html [28/09/2010].

62. Mosteller, F. & Wallace, D. L. (1964). Inference and Disputed Authorship: The Federalist. Reading, MA: Addison-Wesley.

63. Mozgovoy, M. Kakkonen, T. Sutinen, E. (2007a). Using Natural Language Parsers in Plagiarism Detection. Proceedings of the SLaTE Workshop on Speech and Language Technology in Education. Farmington, Pennsylvania, USA.

64. Mozgovoy, M. Karakovskiy, S. Klyuev, V. (2007b). Fast and Reliable Plagiarism Detection System 37th ASEE/IEEE Frontiers in Education Conference, S4H-11, October 10 – 13.

65. Ol'ga, F. & Graeme, H. (2007). Authorship attribution for small texts: Literary and forensic experiments SIGIR '07, Amsterdam, Workshop on Plagiarism Detection, Authorship Identification and Near Duplicate Detection.

66. On Information Retrieval (ECIR 2006), volume 3936 of Lecture Notes in Computer Science, pp 565-569. Springer.

67. Patrick, J., 2008. Authorship Attribution (Foundations and Trends in Information Retrieval), (Especially introductory chapters).

68. Pienemann, M. (Ed.). (2005). Cross-linguistic aspects of process ability theory. Amsterdam: John Benjamins.

69. Prechelt, L. Malpohl, G. & Philippsen, M. (2000). JPlag: Finding plagiarisms among a set of programs. Technical report, University of Karlsruhe, Department of Informatics.

70. Ranatunga, R.V.S.P.K., Ajantha, S.A., & Hewagamage, K.P. (2009). An Integrated Framework for Detecting Plagiarism in e-Learning Systems, eAsia conference, Colombo, Sri Lanka.

71. Richard, O. D., Peter E. H., & David G. S. (2001). Pattern Classification, by Johan Welys & Sons Inc.

72. Seppanen, R. (2002). Finns target master plagiarists, Times Higher Education Supplement, 1st February, pp 11.

73. Shivakumaran, N. (2003). A real life instance of plagiarism detection by Scan, http://www.db.stanford.edu/shiva/scan/plag.html.

74. Schleimer, A.A.S., & Wilkerson, D.S. (2003). Winnowing: Local algorithm for document finger printing, Proceeding of the 2003 ACM SIGMOD international conference on management of data.

75. Salton, G. (1983). Introduction to Modern Information Retrieval. McGraw-Hill.

76. Salton G., & Buckly, C. (1988). Term Weighting Approaches in Automatic Text Retrievaln Cornell University Ithaca, NY, USA.

77. Sebastian, N. & Thomas, P. W. (2006). SNITCH: A Software Tool for Detecting Cut and Paste Plagiarism, SIGCSE'06 March 1-5, Houston, Texas, USA.

78. Singhal, A. (2001). Modern Information Retrieval: A Brief Overview, IEEE Computer Society Technical Committee on Data Engineering.

79. Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. Computational Linguistics, 26(4), pp 471–495.

80. Stamatatos, E., 2006. Ensemble-based author identification using character n-grams. In Proceedings of the 3rd International workshop on Text-Based Information Retrieval pp 41– 46.

81. Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods, Journal of the American Society for Information Science and Technology, 60(3), pp 538-556.

82. Simpson, E. H. (1949). Measurement of diversity, Nature, vol. 163, pp 688.

83. Somers, H. H. (1972). Statistical methods in literary analysis, in The Computer and Literary Style, (J. Leed, ed.), Kent, OH: Kent State University Press.

84. Tallentire, D. R. (1976). Towards an archive of lexical norms — a proposal in The Computer and Literary Studies, Cardiff: Unversity of Wales Press.

85. Tweedie, F. J., Singh, S., & Holmes, D. I. (1996). Neural Network Applications in Stylometry: The Federalist Papers. Computers and the Humanities. 30, pp 1 - 10.

86. Tweedie, F., & Baayen, R. (1998). How variable may a constant be? Measures of lexical richness in perspective. Computers and the Humanities, 32(5), pp 323–352.

87. Van Halteren, H. (2007). Author verification by linguistic profiling:An exploration of the arameter space.ACMTransactions on Speech and Language Processing, 4(1), pp 1–17.

88. Vesanto, J., Himberg, J., Alhoniemi, E., & Parhankangas, J. (2000). SOM tool Box for Matlab 5, Helsinki University of Technology, Finland, ISBN 951-22-4951-0, ISSN 1456-2243.

89. White, E. M. (1993). Too many campuses want to sweep student plagiarism under the rug, Chronicle of Higher Education, 24 February, 39 (25), pp A44.

90. Weeks, S. (2001). Plagiarism: think before pointing finger of blame, Times Higher Education Supplement, 15 May, pp 24.

91. Witten, I. H., Moat, A., & Bell, T. C. (1999). Managing Gigabytes: Compressing and indexing documents and images. Morgan Kaufmann, Second Edition.

92. Yule, G. U. (1938). On sentence-length as a statistical characteristic of style in prose, with application to two cases of disputed authorship, Biometrika, vol. 30, pp 363–90.

93. Yule, G. U. (1944). The Statistical Study of Literary Vocabulary. Cambridge: Cambridge University Press.

94. Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: Writing style features and classification techniques. Journal of the American Society of Information Science and Technology, 57(3), pp 378–393.

95. [WWW 1] http://moodle.org/ [09/01/08].

96. [WWW 2] http://www.webex.com/ft/index.php [10/01/2008].

97. [WWW 3] http://www.usatoday.com/tech/news/2006-07-31-net-plagiarism_x.htm?POE=TECISVA [05/02/2008].

98. [WWW4] www.turnitin.com [05/09/2009].

99. [WWW 5] www.mydropbox.com [05/02/2008].

100. [WWW 6] http://plagiarism.phys.virginia.edu/Wsoftware.html [06/09/2008].

101. [WWW 7] http://www.copycatchgold.com/index.html [13/01/2008].

102. [WWW 8] www.plagiarism.com [10/01/2008].

103. [WWW 9] http://www.paperbin.com [05/02/2008].

104. [WWW 10] http://cise.sbu.ac.uk [07/09/2009].

105. [WWW 11] http://www.findsame.com [07/09/2009].

106. [WWW 12] http://www.howoriginal.com [07/09/2009].

107. [WWW 13] http://www.plagiserve.com [07/09/2009].

108. [WWW 14] http://en.wikipedia.org/wiki/Lorenzo_Valla [06/04/2010].

109. [WWW 15] http://web.archive.org/web/20060630113842/chronicle.com/temp/reprint.php?id=4fvlt82gn640d1rp48srbpjsvlzhmyrs [28/10/2010].

110. [www 16] http://nlp.stanford.edu/software/lex-parser.shtml [28/12/2009].

111. [WWW 17] http://www.gutenberg.org/ebooks/ [05/06/2010].

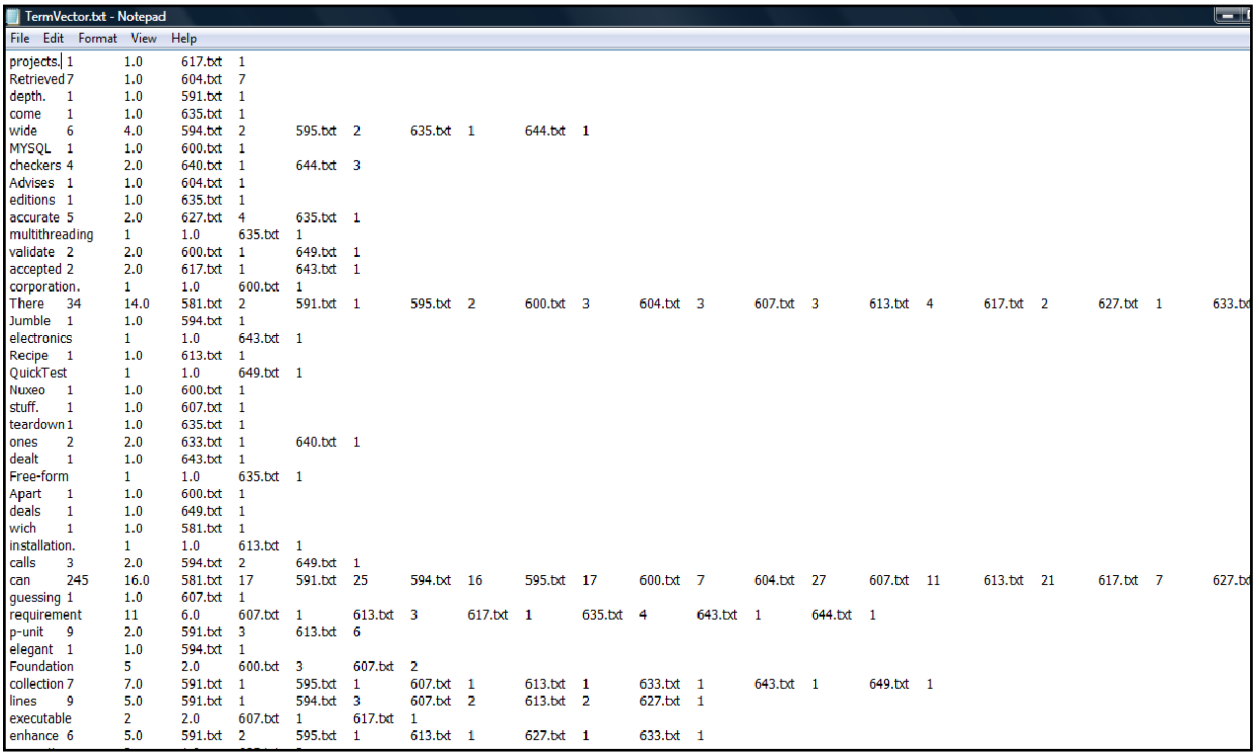
112. [WWW 18] http://jimcooperlab.mcdb.ucsb.edu/autosome/ [05/09/2010].

113. [WWW 19] https://www.ipd.uni-karlsruhe.de/jplag/ [06/04/2010]

114. [WWW 20] http://theory.stanford.edu/~aiken/moss/ [06/04/2010]

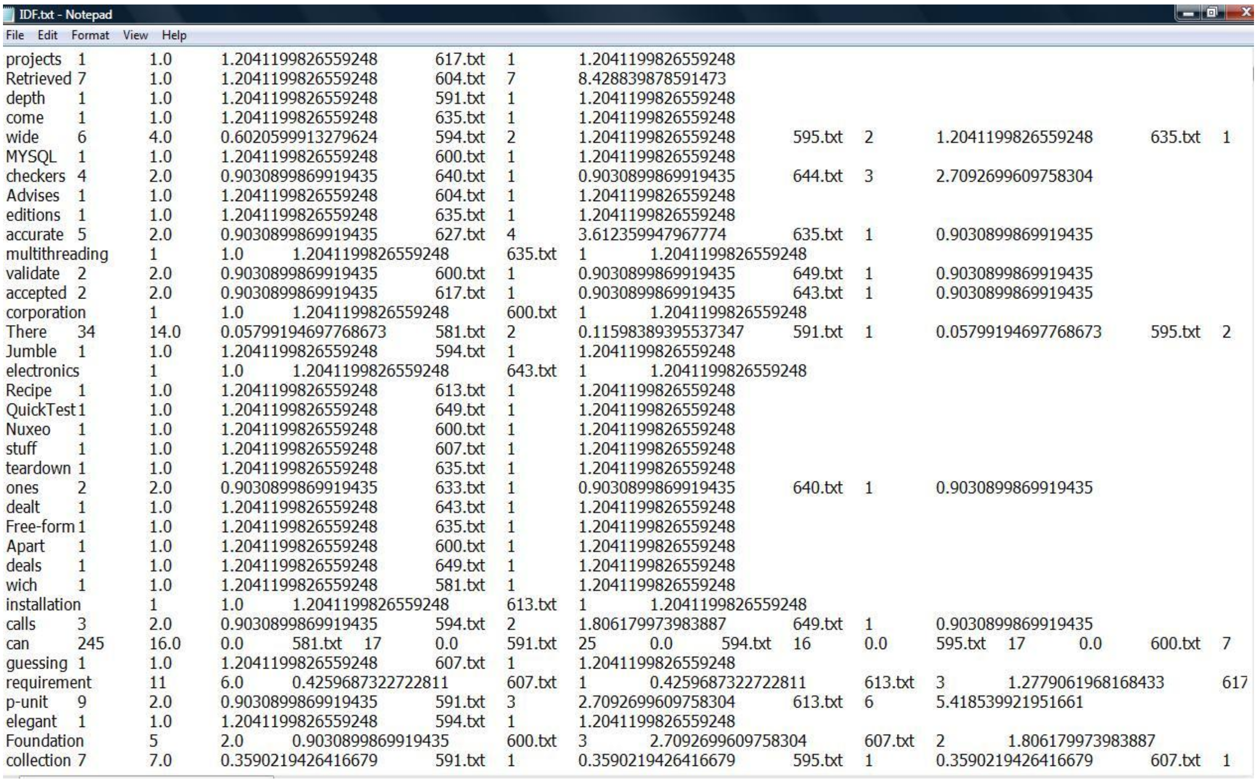
115. [WWW 21] http://www.scanmyessay.com/plagiarism-free-software.php [28/10/2010].

Appendix I



```
projects. 1       1.0     617.txt 1
Retrieved 7       1.0     604.txt 7
depth.    1       1.0     591.txt 1
come      1       1.0     635.txt 1
wide      6       4.0     594.txt 2       595.txt 2       635.txt 1       644.txt 1
MYSQL     1       1.0     600.txt 1
checkers  4       2.0     640.txt 1       644.txt 3
Advises   1       1.0     604.txt 1
editions  1       1.0     635.txt 1
accurate  5       2.0     627.txt 4       635.txt 1
multithreading    1       1.0     635.txt 1
validate  2       2.0     600.txt 1       649.txt 1
accepted  2       2.0     617.txt 1       643.txt 1
corporation.      1       1.0     600.txt 1
There     34      14.0    581.txt 2       591.txt 1       595.txt 2       600.txt 3       604.txt 3       607.txt 3       613.txt 4       617.txt 2       627.txt 1       633.tx
Jumble    1       1.0     594.txt 1
electronics       1       1.0     643.txt 1
Recipe    1       1.0     613.txt 1
QuickTest         1       1.0     649.txt 1
Nuxeo     1       1.0     600.txt 1
stuff.    1       1.0     607.txt 1
teardown  1       1.0     635.txt 1
ones      2       2.0     633.txt 1       640.txt 1
dealt     1       1.0     643.txt 1
Free-form         1       1.0     635.txt 1
Apart     1       1.0     600.txt 1
deals     1       1.0     649.txt 1
wich      1       1.0     581.txt 1
installation.     1       1.0     613.txt 1
calls     3       2.0     594.txt 2       649.txt 1
can       245     16.0    581.txt 17      591.txt 25      594.txt 16      595.txt 17      600.txt 7       604.txt 27      607.txt 11      613.txt 21      617.txt 7       627.tx
guessing  1       1.0     607.txt 1
requirement       11      6.0     607.txt 1       613.txt 3       617.txt 1       635.txt 4       643.txt 1       644.txt 1
p-unit    9       2.0     591.txt 3       613.txt 6
elegant   1       1.0     594.txt 1
Foundation        5       2.0     600.txt 3       607.txt 2
collection 7      7.0     591.txt 1       595.txt 1       607.txt 1       613.txt 1       633.txt 1       643.txt 1       649.txt 1
lines     9       5.0     591.txt 1       594.txt 3       607.txt 2       613.txt 2       627.txt 1
executable        2       2.0     607.txt 1       617.txt 1
enhance   6       5.0     591.txt 2       595.txt 1       613.txt 1       627.txt 1       633.txt 1
```

Figure 4.1: A representation of calculated term vectors of each word of the first document set.



```
projects   1      1.0     1.2041199826559248      617.txt  1      1.2041199826559248
Retrieved  7      1.0     1.2041199826559248      604.txt  7      8.428839878591473
depth      1      1.0     1.2041199826559248      591.txt  1      1.2041199826559248
come       1      1.0     1.2041199826559248      635.txt  1      1.2041199826559248
wide       6      4.0     0.6020599913279624      594.txt  2      1.2041199826559248      595.txt  2      1.2041199826559248      635.txt  1
MYSQL      1      1.0     1.2041199826559248      600.txt  1      1.2041199826559248
checkers   4      2.0     0.9030899869919435      640.txt  1      0.9030899869919435      644.txt  3      2.7092699609758304
Advises    1      1.0     1.2041199826559248      604.txt  1      1.2041199826559248
editions   1      1.0     1.2041199826559248      635.txt  1      1.2041199826559248
accurate   5      2.0     0.9030899869919435      627.txt  4      3.612359947967774       635.txt  1      0.9030899869919435
multithreading    1       1.0     1.2041199826559248      635.txt  1      1.2041199826559248
validate   2      2.0     0.9030899869919435      600.txt  1      0.9030899869919435      649.txt  1      0.9030899869919435
accepted   2      2.0     0.9030899869919435      617.txt  1      0.9030899869919435      643.txt  1      0.9030899869919435
corporation       1       1.0     1.2041199826559248      600.txt  1      1.2041199826559248
There      34     14.0    0.05799194697768673     581.txt  2      0.11598389395537347     591.txt  1      0.05799194697768673     595.txt  2
Jumble     1      1.0     1.2041199826559248      594.txt  1      1.2041199826559248
electronics       1       1.0     1.2041199826559248      643.txt  1      1.2041199826559248
Recipe     1      1.0     1.2041199826559248      613.txt  1      1.2041199826559248
QuickTest  1      1.0     1.2041199826559248      649.txt  1      1.2041199826559248
Nuxeo      1      1.0     1.2041199826559248      600.txt  1      1.2041199826559248
stuff      1      1.0     1.2041199826559248      607.txt  1      1.2041199826559248
teardown   1      1.0     1.2041199826559248      635.txt  1      1.2041199826559248
ones       2      2.0     0.9030899869919435      633.txt  1      0.9030899869919435      640.txt  1      0.9030899869919435
dealt      1      1.0     1.2041199826559248      643.txt  1      1.2041199826559248
Free-form  1      1.0     1.2041199826559248      635.txt  1      1.2041199826559248
Apart      1      1.0     1.2041199826559248      600.txt  1      1.2041199826559248
deals      1      1.0     1.2041199826559248      649.txt  1      1.2041199826559248
wich       1      1.0     1.2041199826559248      581.txt  1      1.2041199826559248
installation      1       1.0     1.2041199826559248      613.txt  1      1.2041199826559248
calls      3      2.0     0.9030899869919435      594.txt  2      1.806179973983887       649.txt  1      0.9030899869919435
can        245    16.0    0.0     581.txt 17      0.0     591.txt  25     0.0     594.txt 16      0.0     595.txt 17      0.0     600.txt  7
guessing   1      1.0     1.2041199826559248      607.txt  1      1.2041199826559248
requirement       11      6.0     0.4259687322722811      607.txt  1      0.4259687322722811      613.txt  3      1.2779061968168433      617
p-unit     9      2.0     0.9030899869919435      591.txt  3      2.7092699609758304      613.txt  6      5.418539921951661
elegant    1      1.0     1.2041199826559248      594.txt  1      1.2041199826559248
Foundation        5       2.0     0.9030899869919435      600.txt  3      2.7092699609758304      607.txt  2      1.806179973983887
collection 7      7.0     0.3590219426416679      591.txt  1      0.3590219426416679      595.txt  1      0.3590219426416679      607.txt  1
```

Figure 4.2: A representation of calculated tfidf of each word of the first document set.

	581.txt	591.txt	594.txt	595.txt	600.txt	604.txt	607.txt	613.txt	617.txt	627.txt	633.txt	635.txt	640.txt	643.txt	644.txt	649.txt:
581.txt	700	264	253	246	218	297	271	304	250	196	191	311	221	239	230	156
591.txt	264	985	313	305	280	361	345	408	345	240	251	386	279	284	285	197
594.txt	253	313	952	306	293	373	306	353	294	248	239	369	250	306	277	187
595.txt	246	305	306	804	273	309	298	484	291	237	229	367	239	274	249	177
600.txt	218	280	293	273	859	331	291	321	283	261	231	381	266	273	254	188
604.txt	297	361	373	309	331	1165	359	383	378	290	272	466	355	370	310	211
607.txt	271	345	306	298	291	359	859	387	331	250	242	380	273	312	292	189
613.txt	304	408	353	484	321	383	387	1182	338	274	288	459	284	355	348	207
617.txt	250	345	294	291	283	378	331	338	920	267	277	367	302	314	255	185
627.txt	196	240	248	237	261	290	250	274	267	607	234	299	234	253	207	151
633.txt	191	251	239	229	231	272	242	288	277	234	578	293	224	262	196	159
635.txt	311	386	369	367	381	466	380	459	367	299	293	1328	336	384	319	218
640.txt	221	279	250	239	266	355	273	284	302	234	224	336	696	285	230	174
643.txt	239	284	306	274	273	370	312	355	314	253	262	384	285	885	257	195
644.txt	230	285	277	249	254	310	292	348	255	207	196	319	230	257	715	179
649.txt	156	197	187	177	188	211	189	207	185	151	159	218	174	195	179	543

Table 4.1 illustrates the 16x16 matrix which shows the highest similarity file according to the number of exact matches on words.

OD	SD	No of Words in OD	No of Words in SD	No of Matched Word	Manual Plag%
10MS 004	10MS 099	628	579	533	84.87
10MS 007	10MS 004	579	628	450	77.72
10MS 008	10MS 064	843	855	341	40.45
10MS 011	10MS 050	893	903	841	94.18
10MS 013	10MS 020	962	841	822	85.45
10MS 017		338		0	0.00
10MS 020	10MS 013	841	962	822	97.74
10MS 028	10MS 063	879	891	829	94.31
10MS 041	10MS 064	877	855	538	61.35
10MS 048	10MS 028	816	879	406	49.75
10MS 050	10MS 011	903	893	841	93.13
10MS 052	10MS 056	823	771	473	57.47
10MS 054	10MS 050	769	903	741	96.36
10MS 056	10MS 052	771	823	473	61.35
10MS 057	10MS 200	787	920	674	85.64
10MS 063	10MS 028	891	879	829	93.04
10MS 064	10MS 041	855	877	538	62.92
10MS 068		979		0	0.00
10MS 074		710		0	0.00
10MS 075	10MS 028	748	879	239	31.95
10MS 077	10MS 064	737	855	158	21.44
10MS 081	10MS 011	908	893	692	76.21
10MS 087	10MS 096	715	697	293	40.98
10MS 091	10MS 099	676	579	355	52.51
10MS 092	10MS 050	754	903	651	86.34
10MS 094	10MS 096	797	697	102	12.80
10MS 096	10MS 041	697	877	329	47.20
10MS 099	10MS 004	579	628	533	92.06
10MS 118	10MS 020	747	841	39	5.22
10MS 200	10MS 057	920	787	674	73.26

Table 4.2: Manual Detection of Plag% of the third experiment

OD	SD	Manual Plag%	NVSM Plag%
10MS 004	10MS 099	84.87	68
10MS 007	10MS 004	77.72	68
10MS 008	10MS 064	40.45	25
10MS 011	10MS 050	94.17	68
10MS 013	10MS 020	85.45	55
10MS 017	10MS 011	0	6
10MS 020	10MS 013	97.74	55
10MS 028	10MS 063	94.31	35
10MS 041	10MS 064	61.35	36
10MS 048	10MS 028	49.75	34
10MS 050	10MS 054	93.13	69
10MS 052	10MS 056	57.47	36
10MS 054	10MS 050	96.36	69
10MS 056	10MS 091	61.35	38
10MS 057	10MS 200	85.64	97
10MS 063	10MS 028	93.04	35
10MS 064	10MS 041	62.92	36
10MS 068	10MS 091	0	13
10MS 074	10MS 020	0	7
10MS 075	10MS 028	31.95	25
10MS 077	10MS 064	21.44	11
10MS 081	10MS 011	76.21	64
10MS 087	10MS 096	40.98	32
10MS 091	10MS 099	52.51	31
10MS 092	10MS 050	86.34	60
10MS 094	10MS 096	12.79	12
10MS 096	10MS 041	47.20	30
10MS 099	10MS 004	92.05	57
10MS 118	10MS 020	5.22	12
10MS 200	10MS 057	73.26	97

Table 4.3: Manual Plag% and NVSM Plag% of the third experiment

OD	**SD**	**Manual Plag%**	**FPDM Plag%**
10MS 004	10MS 099	85	39.82
10MS 007	10MS 004	78	39.74
10MS 008	10MS 064	40	16.75
10MS 011	10MS 081	94	32.84
10MS 013	10MS 020	85	60.08
10MS 017	10MS 052	0	13.42
10MS 020	10MS 013	98	59.82
10MS 028	10MS 063	94	58.18
10MS 041	10MS 064	61	44.92
10MS 048	10MS 028	50	18.45
10MS 050	10MS 011	93	30.54
10MS 052	10MS 056	57	26.56
10MS 054	10MS 050	96	51.9
10MS 056	10MS 052	61	32.09
10MS 057	10MS 200	86	93.2
10MS 063	10MS 028	93	58.71
10MS 064	10MS 041	63	44.51
10MS 068	10MS 056	0	3.51
10MS 074	10MS 028	0	3.01
10MS 075	10MS 028	32	15.1
10MS 077	10MS 064	21	13.76
10MS 081	10MS 011	76	33.79
10MS 087	10MS 096	41	24.66
10MS 091	10MS 099	53	14.94
10MS 092	10MS 050	86	27.99
10MS 094	10MS 096	13	11.96
10MS 096	10MS 041	47	26.93
10MS 099	10MS 004	92	39.75
10MS 118	10MS 052	5	13.49
10MS 200	10MS 057	73	92.95

Table 4.4: Manual Plag% and FMPD Plag% of third experiment

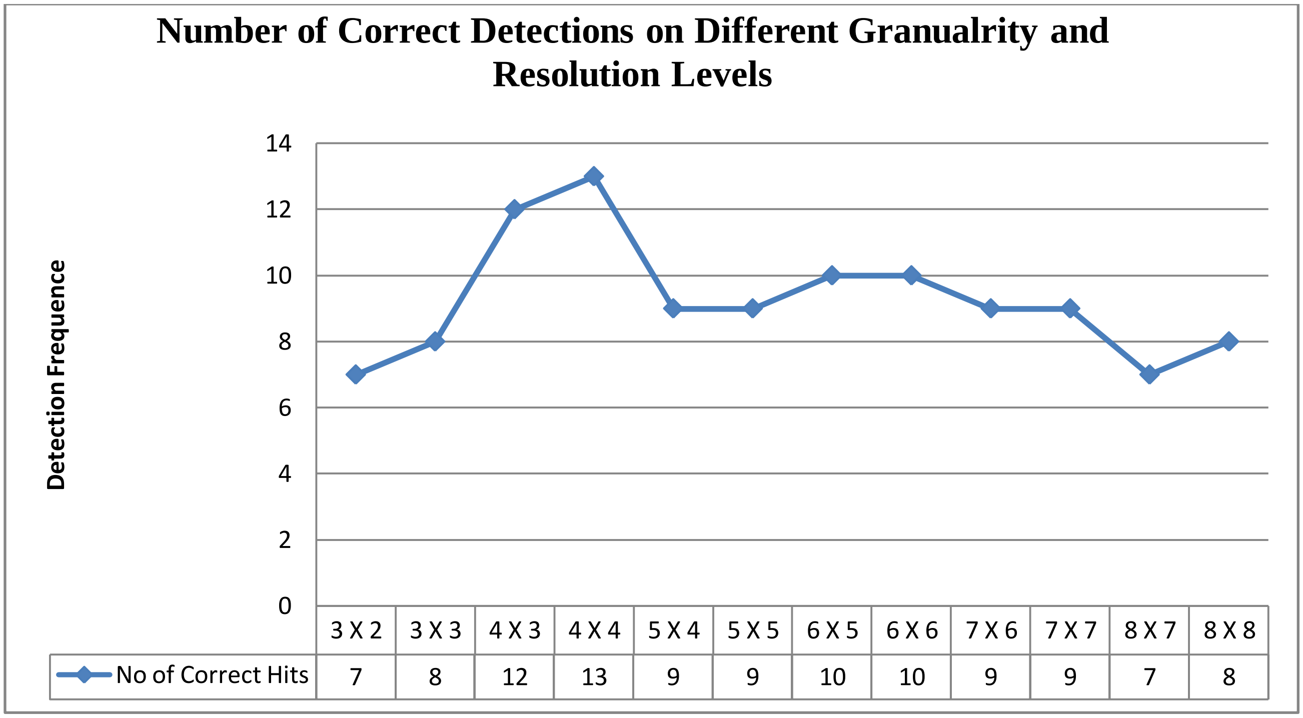


Figure 4.3: Number of detections on different granularity and fingerprinting resolutions of the first dataset.

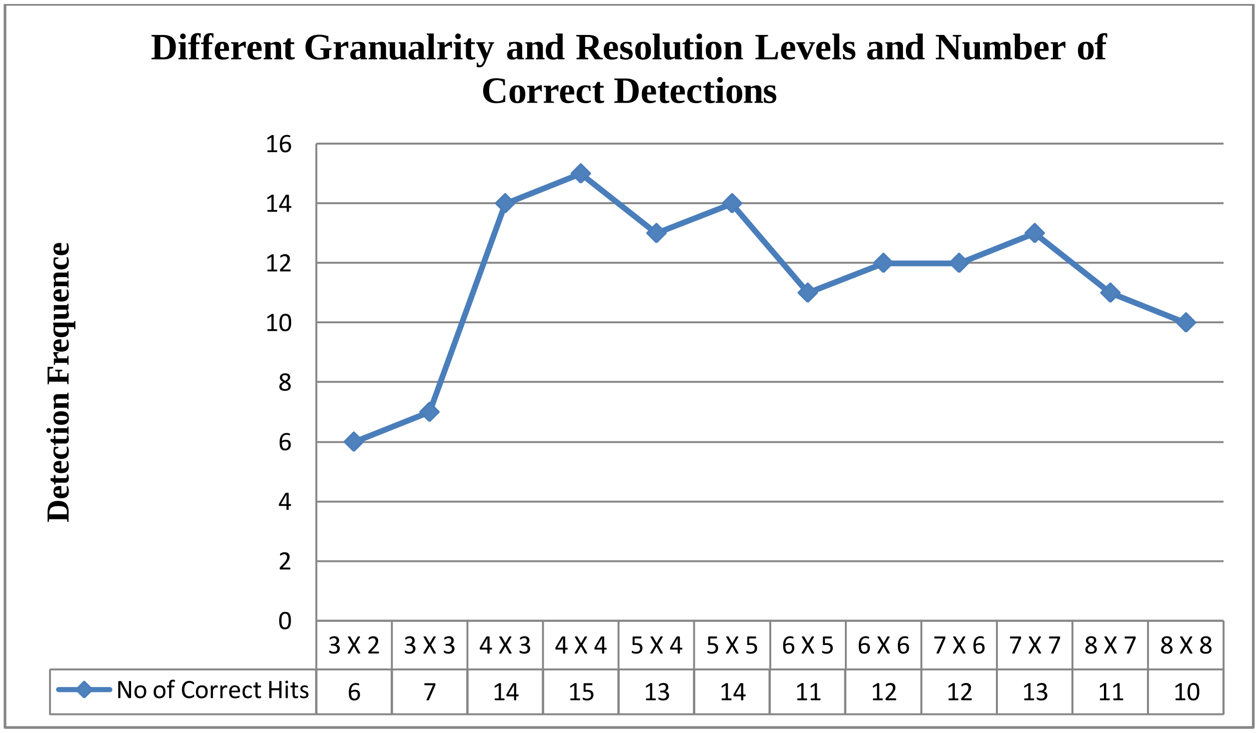
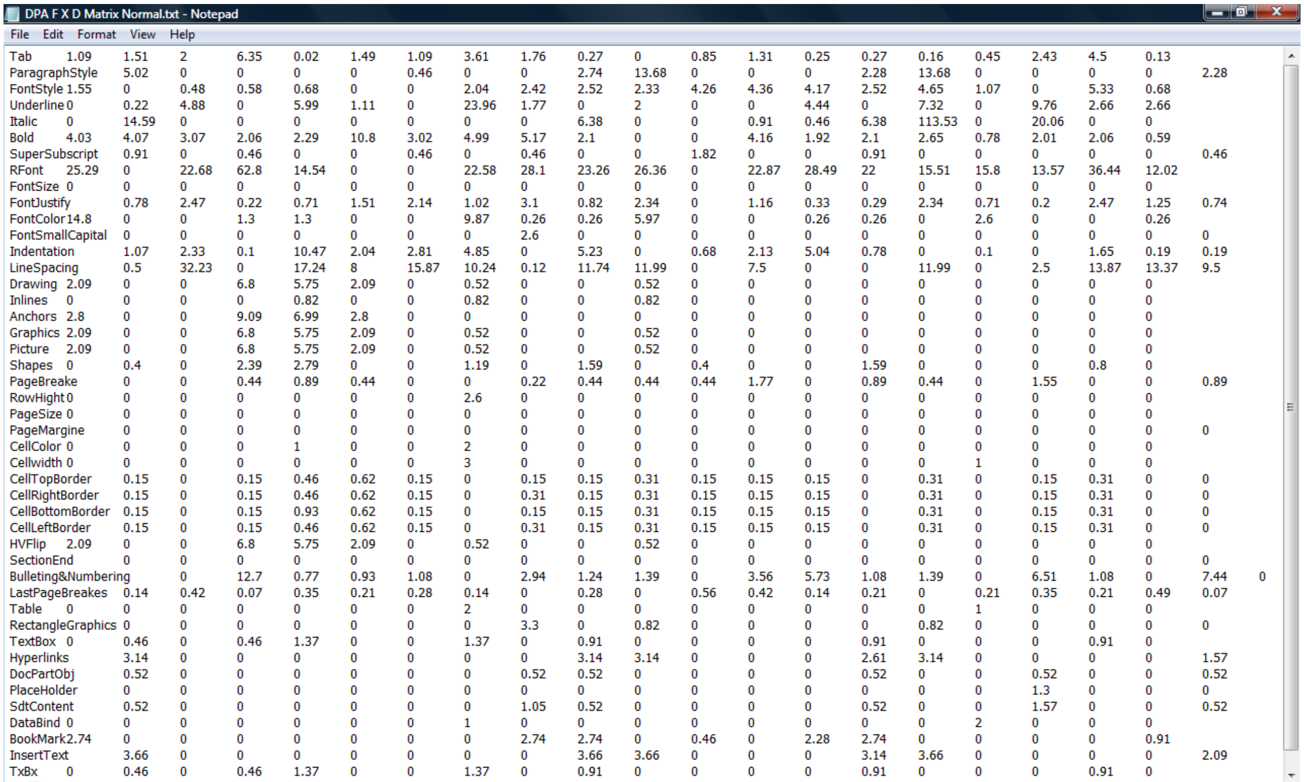
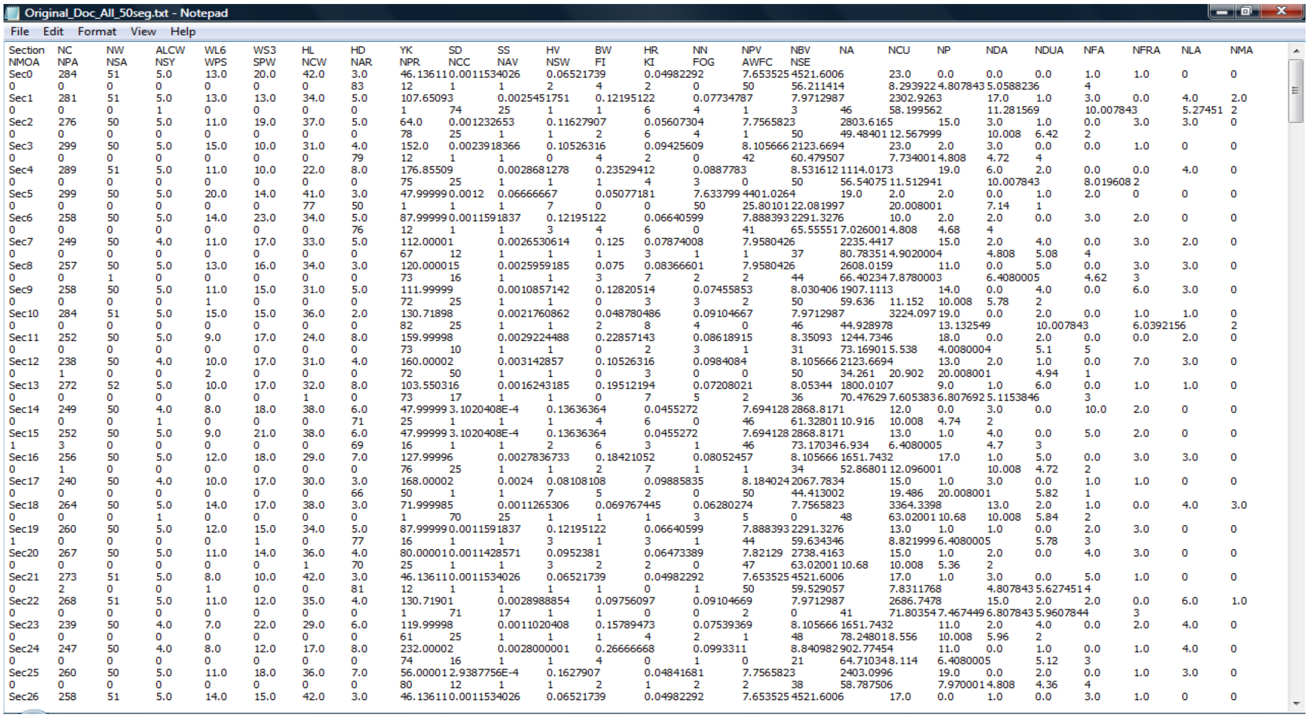


Figure 4.4: Number of detections on different granularity and fingerprinting resolutions of the second dataset.


DPA F X D Matrix Normal.txt - Notepad
File Edit Format View Help

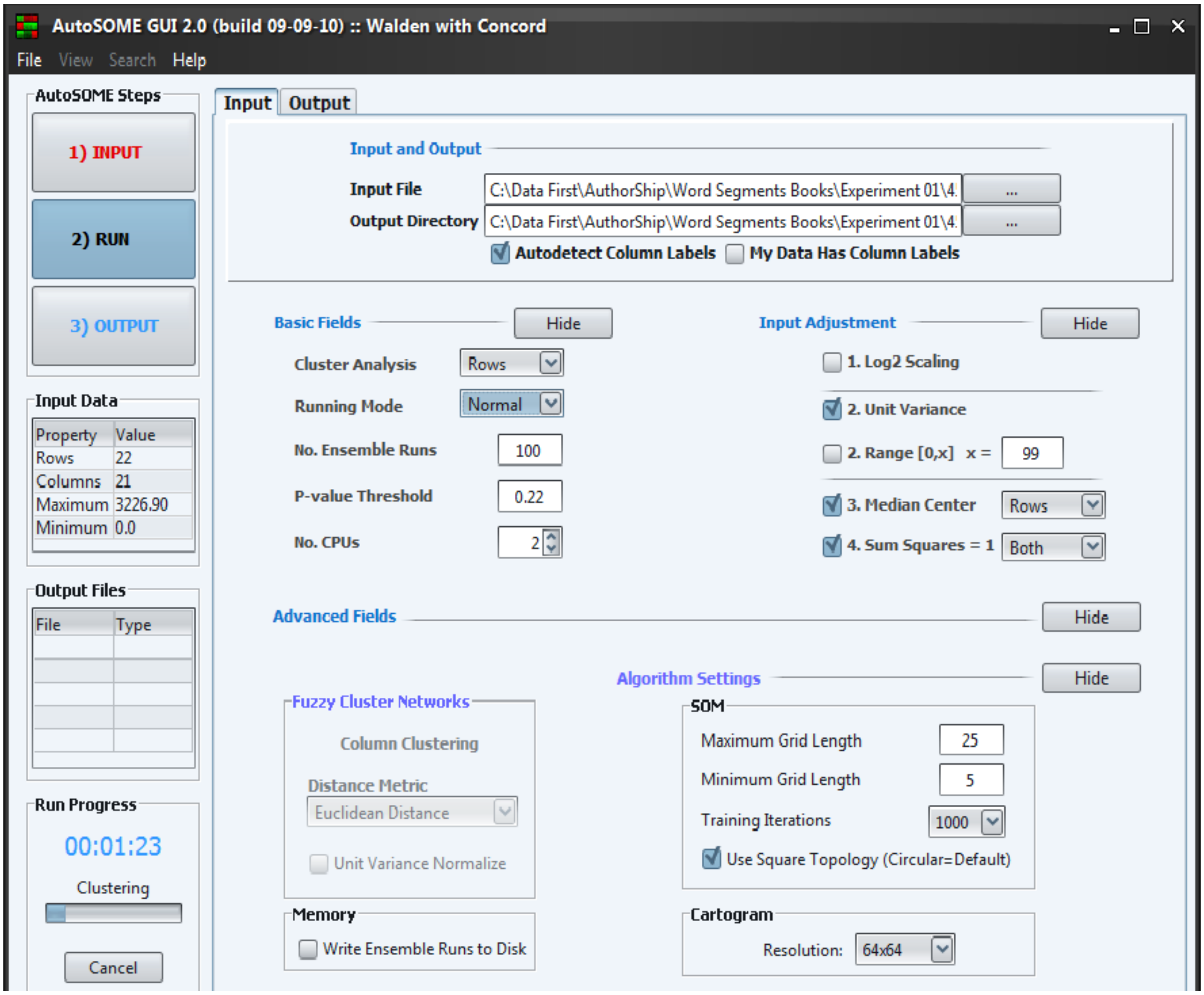
```
Tab     1.09    1.51    2       6.35    0.02    1.49    1.09    3.61    1.76    0.27    0       0.85    1.31    0.25    0.27    0.16    0.45    2.43    4.5     0.13
ParagraphStyle  5.02    0       0       0       0       0.46    0       0       2.74    13.68   0       0       0       2.28    13.68   0       0       0       0       2.28
FontStyle 1.55  0       0.48    0.58    0.68    0       0       2.04    2.42    2.52    2.33    4.26    4.36    4.17    2.52    4.65    1.07    0       5.33    0.68
Underline0      0.22    4.88    0       5.99    1.11    0       23.96   1.77    0       2       0       0       4.44    0       7.32    0       9.76    2.66    2.66
Italic  0       14.59   0       0       0       0       0       0       0       6.38    0       0       0.91    0.46    6.38    113.53  0       20.06   0       0
Bold    4.03    4.07    3.07    2.06    2.29    10.8    3.02    4.99    5.17    2.1     0       0       4.16    1.92    2.1     2.65    0.78    2.01    2.06    0.59
SuperSubscript  0.91    0       0.46    0       0       0.46    0       0.46    0       0       1.82    0       0       0.91    0       0       0       0       0       0.46
RFont   25.29   0       22.68   62.8    14.54   0       0       22.58   28.1    23.26   26.36   0       22.87   28.49   22      15.51   15.8    13.57   36.44   12.02
FontSize 0      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
FontJustify     0.78    2.47    0.22    0.71    1.51    2.14    1.02    3.1     0.82    2.34    0       1.16    0.33    0.29    2.34    0.71    0.2     2.47    1.25    0.74
FontColor14.8   0       0       1.3     1.3     0       0       9.87    0.26    0.26    5.97    0       0       0.26    0.26    0       2.6     0       0       0.26
FontSmallCapital 0      0       0       0       0       0       0       2.6     0       0       0       0       0       0       0       0       0       0       0       0
Indentation     1.07    2.33    0.1     10.47   2.04    2.81    4.85    0       5.23    0       0.68    2.13    5.04    0.78    0       0.1     0       1.65    0.19    0.19
LineSpacing     0.5     32.23   0       17.24   8       15.87   10.24   0.12    11.74   11.99   0       7.5     0       0       11.99   0       2.5     13.87   13.37   9.5
Drawing 2.09    0       0       6.8     5.75    2.09    0       0.52    0       0       0.52    0       0       0       0       0       0       0       0       0
Inlines 0       0       0       0       0.82    0       0       0.82    0       0       0.82    0       0       0       0       0       0       0       0       0
Anchors 2.8     0       0       9.09    6.99    2.8     0       0       0       0       0       0       0       0       0       0       0       0       0       0
Graphics 2.09   0       0       6.8     5.75    2.09    0       0.52    0       0       0.52    0       0       0       0       0       0       0       0       0
Picture 2.09    0       0       6.8     5.75    2.09    0       0.52    0       0       0.52    0       0       0       0       0       0       0       0       0
Shapes  0       0.4     0       2.39    2.79    0       0       1.19    0       1.59    0       0.4     0       0       1.59    0       0       0       0.8     0
PageBreake      0       0       0.44    0.89    0.44    0       0       0.22    0.44    0.44    0.44    1.77    0       0.89    0.44    0       1.55    0       0       0.89
RowHight0       0       0       0       0       0       0       2.6     0       0       0       0       0       0       0       0       0       0       0       0
PageSize 0      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
PageMargine     0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
CellColor 0     0       0       0       1       0       0       2       0       0       0       0       0       0       0       0       0       0       0       0
Cellwidth 0     0       0       0       0       0       0       3       0       0       0       0       0       0       0       0       1       0       0       0
CellTopBorder   0.15    0       0.15    0.46    0.62    0.15    0       0.15    0.15    0.31    0.15    0.15    0.15    0       0.31    0       0.15    0.31    0       0
CellRightBorder 0.15    0       0.15    0.46    0.62    0.15    0       0.31    0.15    0.31    0.15    0.15    0.15    0       0.31    0       0.15    0.31    0       0
CellBottomBorder 0.15   0       0.15    0.93    0.62    0.15    0       0.15    0.15    0.31    0.15    0.15    0.15    0       0.31    0       0.15    0.31    0       0
CellLeftBorder  0.15    0       0.15    0.46    0.62    0.15    0       0.31    0.15    0.31    0.15    0.15    0.15    0       0.31    0       0.15    0.31    0       0
HVFlip  2.09    0       0       6.8     5.75    2.09    0       0.52    0       0       0.52    0       0       0       0       0       0       0       0       0
SectionEnd      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Bulleting&Numbering     0       12.7    0.77    0.93    1.08    0       2.94    1.24    1.39    0       3.56    5.73    1.08    1.39    0       6.51    1.08    0       7.44    0
LastPageBreakes 0.14    0.42    0.07    0.35    0.21    0.28    0.14    0       0.28    0       0.56    0.42    0.14    0.21    0       0.21    0.35    0.21    0.49    0.07
Table   0       0       0       0       0       0       0       2       0       0       0       0       0       0       0       0       1       0       0       0
RectangleGraphics 0     0       0       0       0       0       0       3.3     0       0.82    0       0       0       0       0.82    0       0       0       0       0
TextBox 0       0.46    0       0.46    1.37    0       0       1.37    0       0.91    0       0       0       0       0.91    0       0       0       0.91    0
Hyperlinks      3.14    0       0       0       0       0       0       0       3.14    3.14    0       0       0       2.61    3.14    0       0       0       0       1.57
DocPartObj      0.52    0       0       0       0       0       0       0.52    0.52    0       0       0       0       0.52    0       0       0.52    0       0       0.52
PlaceHolder     0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       1.3     0       0       0
SdtContent      0.52    0       0       0       0       0       0       1.05    0.52    0       0       0       0       0.52    0       0       1.57    0       0       0.52
DataBind 0      0       0       0       0       0       0       1       0       0       0       0       0       0       0       0       2       0       0       0
BookMark2.74    0       0       0       0       0       0       0       2.74    2.74    0       0.46    0       2.28    2.74    0       0       0       0       0.91
InsertText      3.66    0       0       0       0       0       0       0       3.66    3.66    0       0       0       3.14    3.66    0       0       0       0       2.09
TxBx    0       0.46    0       0.46    1.37    0       0       1.37    0       0.91    0       0       0       0       0.91    0       0       0       0.91    0
```

Figure 4.5: $F \times D$ matrix of the second dataset generated by the DPA


Original_Doc_All_50seg.txt - Notepad
File Edit Format View Help

```
Section   NC        NW        ALCW      WL6       WS3       HL        HD        YK        SD        SS        HV        BW        HR        NN        NPV       NBV       NA        NCU       NP        NDA       NDUA      NFA       NFRA      NLA       NMA
NMOA      NPA       NSA       NSY       WPS       SPW       NCW       NAR       NPR       NCC       NAV       NSW       FI        KI        FOG       AWFC      NSE
Sec0      284       51        5.0       13.0      20.0      42.0      3.0       46.13611  0.0011534026        0.06521739          0.04982292          7.653525  4521.6006           23.0      0.0       0.0       0.0       1.0       1.0       0         0
0         0         0         0         0         0         0         83        12        1         1         2         4         2         0         50        56.211414           8.293922  4.807843  5.0588236           4
Sec1      281       51        5.0       13.0      13.0      34.0      5.0       107.65093           0.0025451751        0.12195122          0.07734787          7.9712987           2302.9263           17.0      1.0       3.0       0.0       4.0       2.0
0         0         0         1         0         0         0         0         1         74        25        1         1         6         4         1         3         46        58.199562           11.281569           10.007843           5.27451   2
Sec2      276       50        5.0       11.0      19.0      37.0      5.0       64.0      0.001232653         0.11627907          0.05607304          7.7565823           2803.6165           15.0      3.0       1.0       0.0       3.0       3.0       0
0         0         0         0         0         0         0         0         78        25        1         1         2         6         4         1         50        49.48401  12.567999           10.008    6.42      2
Sec3      299       50        5.0       15.0      10.0      31.0      4.0       152.0     0.0023918366        0.10526316          0.09425609          8.105666  2123.6694           23.0      2.0       3.0       0.0       0.0       1.0       0         0
0         0         0         0         0         0         0         79        12        1         1         0         4         2         0         42        60.479507           7.734001  4.808     4.72      4
Sec4      289       51        5.0       11.0      10.0      22.0      8.0       176.85509           0.0028681278        0.23529412          0.0887783           8.531612  1114.0173           19.0      6.0       2.0       0.0       0.0       4.0       0
0         0         0         0         0         0         0         0         75        25        1         1         1         4         3         0         50        56.54075  11.512941           10.007843           8.019608  2
Sec5      299       50        5.0       20.0      14.0      41.0      3.0       47.99999  0.0012    0.06666667          0.05077181                    7.633799  4401.0264 19.0      2.0       2.0       0.0       1.0       2.0       0         0         0
0         0         0         0         0         0         77        50        1         1         1         7         0         0         50        25.80101  22.081997           20.008001           7.14      1
Sec6      258       50        5.0       14.0      23.0      34.0      5.0       87.99999  0.0011591837        0.12195122          0.06640599          7.888393  2291.3276           10.0      2.0       2.0       0.0       3.0       2.0       0         0
0         0         0         0         0         0         0         76        12        1         1         3         4         6         0         41        65.55551  7.026001  4.808     4.68      4
Sec7      249       50        4.0       11.0      17.0      33.0      5.0       112.00001           0.0026530614        0.125     0.07874008          7.9580426           2235.4417           15.0      2.0       4.0       0.0       3.0       2.0       0
0         0         0         0         0         0         0         0         67        12        1         1         1         3         1         1         37        80.78351  4.9020004           4.808     5.08      4
Sec8      257       50        5.0       13.0      16.0      34.0      3.0       120.000015          0.0025959185        0.075     0.08366601          7.9580426           2608.0159           11.0      0.0       5.0       0.0       3.0       3.0       0
0         0         1         0         0         0         0         0         73        16        1         1         3         7         2         2         44        66.40234  7.8780003           6.4080005           4.62      3
Sec9      258       50        5.0       11.0      15.0      31.0      5.0       111.99999           0.0010857142        0.12820514          0.07455853          8.030406  1907.1113           14.0      0.0       4.0       0.0       6.0       3.0       0
0         0         0         0         1         0         0         0         72        25        1         1         0         3         3         2         50        59.636    11.152    10.008    5.78      2
Sec10     284       51        5.0       15.0      15.0      36.0      2.0       130.71898           0.0021760862        0.048780486         0.09104667          7.9712987           3224.097  19.0      0.0       2.0       0.0       1.0       1.0       0
0         0         0         0         0         0         0         0         82        25        1         1         2         8         4         0         46        44.928978           13.132549           10.007843           6.0392156           2
Sec11     252       50        5.0       9.0       17.0      24.0      8.0       159.99998           0.0029224488        0.22857143          0.08618915          8.35093   1244.7346           18.0      0.0       2.0       0.0       0.0       2.0       0
0         0         0         0         0         0         0         0         73        10        1         1         0         2         3         1         31        73.16901  5.538     4.0080004           5.1       5
Sec12     238       50        4.0       10.0      17.0      31.0      4.0       160.00002           0.003142857         0.10526316          0.0984084           8.105666  2123.6694           13.0      2.0       1.0       0.0       7.0       3.0       0
0         1         0         0         2         0         0         0         72        50        1         1         0         3         0         0         50        34.261    20.902    20.008001           4.94      1
Sec13     272       52        5.0       10.0      17.0      32.0      8.0       103.550316          0.0016243185        0.19512194          0.07208021          8.05344   1800.0107           9.0       1.0       6.0       0.0       1.0       1.0       0
0         0         0         0         0         0         1         0         73        17        1         1         0         7         5         2         36        70.47629  7.605383  6.807692  5.1153846           3
Sec14     249       50        4.0       8.0       18.0      38.0      6.0       47.99999  3.1020408E-4        0.13636364          0.0455272           7.694128  2868.8171           12.0      0.0       3.0       0.0       10.0      2.0       0         0
0         0         0         1         0         0         0         71        25        1         1         1         4         6         0         46        61.32801  10.916    10.008    4.74      2
Sec15     252       50        5.0       9.0       21.0      38.0      6.0       47.99999  3.1020408E-4        0.13636364          0.0455272           7.694128  2868.8171           13.0      1.0       4.0       0.0       5.0       2.0       0         0
1         3         0         0         0         0         0         69        16        1         1         2         6         3         1         46        73.17034  6.934     6.4080005           4.7       3
Sec16     256       50        5.0       12.0      18.0      29.0      7.0       127.99996           0.0027836733        0.18421052          0.08052457          8.105666  1651.7432           17.0      1.0       5.0       0.0       3.0       3.0       0
0         1         0         0         0         0         0         0         76        25        1         1         2         7         1         1         34        52.86801  12.096001           10.008    4.72      2
Sec17     240       50        4.0       10.0      17.0      30.0      3.0       168.00002           0.0024    0.08108108          0.09885835          8.184024  2067.7834           15.0      1.0       3.0       0.0       1.0       1.0       0         0
0         0         0         0         0         0         0         66        50        1         1         7         5         2         0         50        44.413002           19.486    20.008001           5.82      1
Sec18     264       50        5.0       14.0      17.0      38.0      3.0       71.999985           0.0011265306        0.069767445         0.06280274          7.7565823           3364.3398           13.0      2.0       1.0       0.0       4.0       3.0
0         0         0         1         0         0         0         0         1         70        25        1         1         1         3         5         0         48        63.02001  10.68     10.008    5.84      2
Sec19     260       50        5.0       12.0      15.0      34.0      5.0       87.99999  0.0011591837        0.12195122          0.06640599          7.888393  2291.3276           13.0      1.0       1.0       0.0       2.0       3.0       0         0
1         0         0         0         0         1         0         77        16        1         1         3         1         3         1         44        59.634346           8.821999  6.4080005           5.78      3
Sec20     267       50        5.0       11.0      14.0      36.0      4.0       80.00001  0.0011428571        0.0952381           0.06473389          7.82129   2738.4163           15.0      1.0       2.0       0.0       4.0       3.0       0         0
0         0         0         0         0         0         1         70        25        1         1         3         2         2         0         47        63.02001  10.68     10.008    5.36      2
Sec21     273       51        5.0       8.0       10.0      42.0      3.0       46.13611  0.0011534026        0.06521739          0.04982292          7.653525  4521.6006           17.0      1.0       3.0       0.0       5.0       1.0       0         0
0         2         0         0         1         0         0         81        12        1         1         1         1         0         1         50        59.529057           7.8311768           4.807843  5.627451  4
Sec22     268       51        5.0       11.0      12.0      35.0      4.0       130.71901           0.0028988854        0.09756097          0.09104669          7.9712987           2686.7478           15.0      2.0       2.0       0.0       6.0       1.0
0         0         0         0         0         0         0         0         1         71        17        1         1         0         0         2         0         41        71.80354  7.467449  6.807843  5.9607844           3
Sec23     239       50        4.0       7.0       22.0      29.0      6.0       119.99998           0.0011020408        0.15789473          0.07539369          8.105666  1651.7432           11.0      2.0       4.0       0.0       2.0       4.0       0
0         0         0         0         0         0         0         0         61        25        1         1         1         4         2         1         48        78.24801  8.556     10.008    5.96      2
Sec24     247       50        4.0       8.0       12.0      17.0      8.0       232.00002           0.0028000001        0.26666668          0.0993311           8.840982  902.77454           11.0      0.0       1.0       0.0       1.0       4.0       0
0         0         0         0         0         0         0         0         74        16        1         1         4         0         1         0         21        64.71034  8.114     6.4080005           5.12      3
Sec25     260       50        5.0       11.0      18.0      36.0      7.0       56.00001  2.9387756E-4        0.1627907           0.04841681          7.7565823           2403.0996           19.0      0.0       2.0       0.0       1.0       3.0       0
0         0         0         0         0         0         0         0         80        12        1         1         2         1         2         2         38        58.787506           7.970001  4.808     4.36      4
Sec26     258       51        5.0       14.0      15.0      42.0      3.0       46.13611  0.0011534026        0.06521739          0.04982292          7.653525  4521.6006           17.0      0.0       1.0       0.0       3.0       1.0       0         0
```

Figure 4.6: Sample of the extracted features from segmented file



AutoSOME GUI 2.0 (build 09-09-10) :: Walden with Concord

Input Data

Property	Value
Rows	22
Columns	21
Maximum	3226.90
Minimum	0.0

Output Files

File	Type

Run Progress

00:01:23

Clustering

Input and Output

Input File: C:\Data First\AuthorShip\Word Segments Books\Experiment 01\4.

Output Directory: C:\Data First\AuthorShip\Word Segments Books\Experiment 01\4.

☑ Autodetect Column Labels ☐ My Data Has Column Labels

Basic Fields

Cluster Analysis: Rows

Running Mode: Normal

No. Ensemble Runs: 100

P-value Threshold: 0.22

No. CPUs: 2

Input Adjustment

☐ 1. Log2 Scaling

☑ 2. Unit Variance

☐ 2. Range [0,x] x = 99

☑ 3. Median Center: Rows

☑ 4. Sum Squares = 1: Both

Advanced Fields

Fuzzy Cluster Networks

Column Clustering

Distance Metric: Euclidean Distance

☐ Unit Variance Normalize

Memory

☐ Write Ensemble Runs to Disk

Algorithm Settings

SOM

Maximum Grid Length: 25

Minimum Grid Length: 5

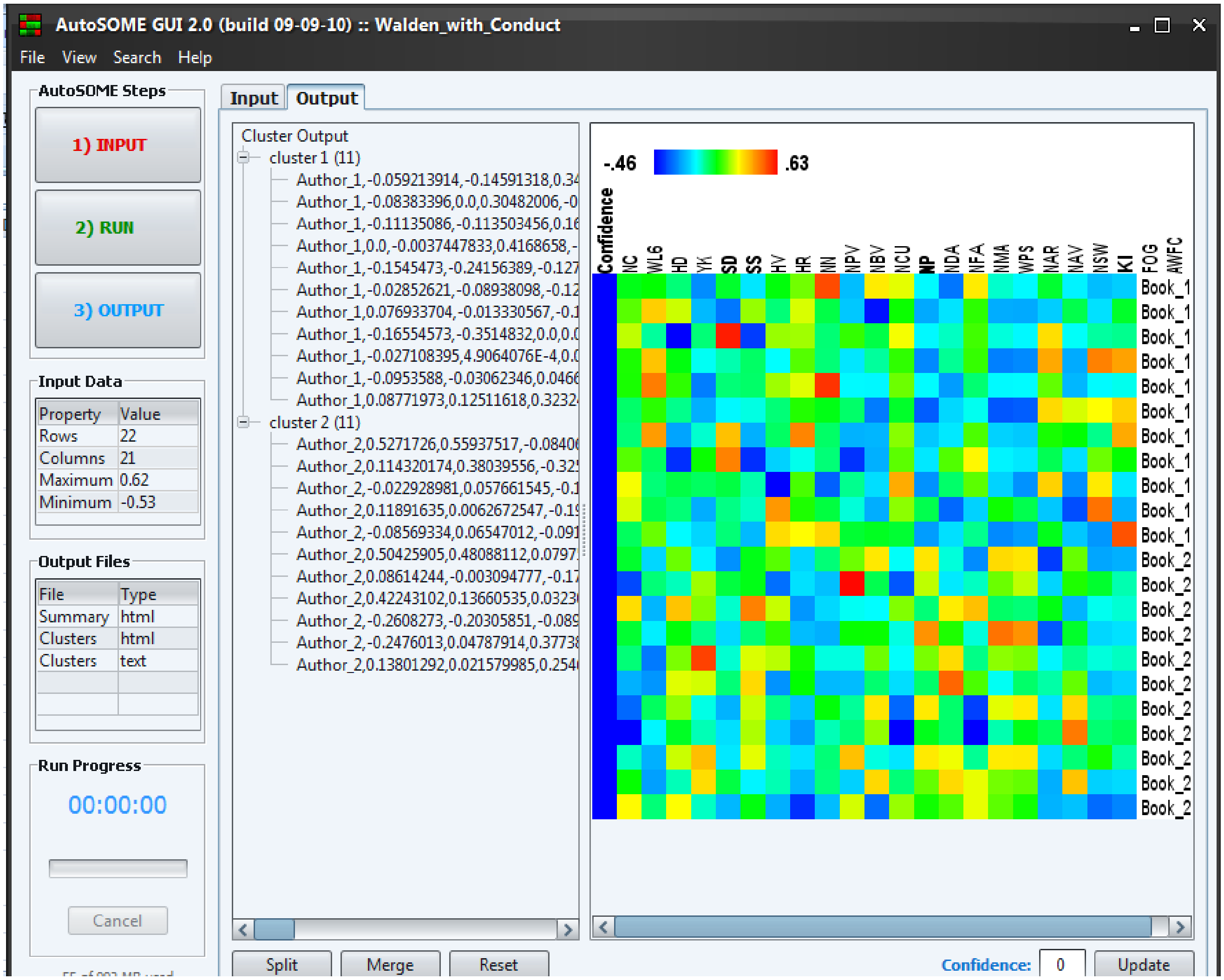
Training Iterations: 1000

☑ Use Square Topology (Circular=Default)

Cartogram

Resolution: 64x64

Figure 4.7: A setup of parameters and running example of AutoSOME tool on same author and 450 segmentation.



Input Data

Property	Value
Rows	22
Columns	21
Maximum	0.62
Minimum	-0.53

Output Files

File	Type
Summary	html
Clusters	html
Clusters	text

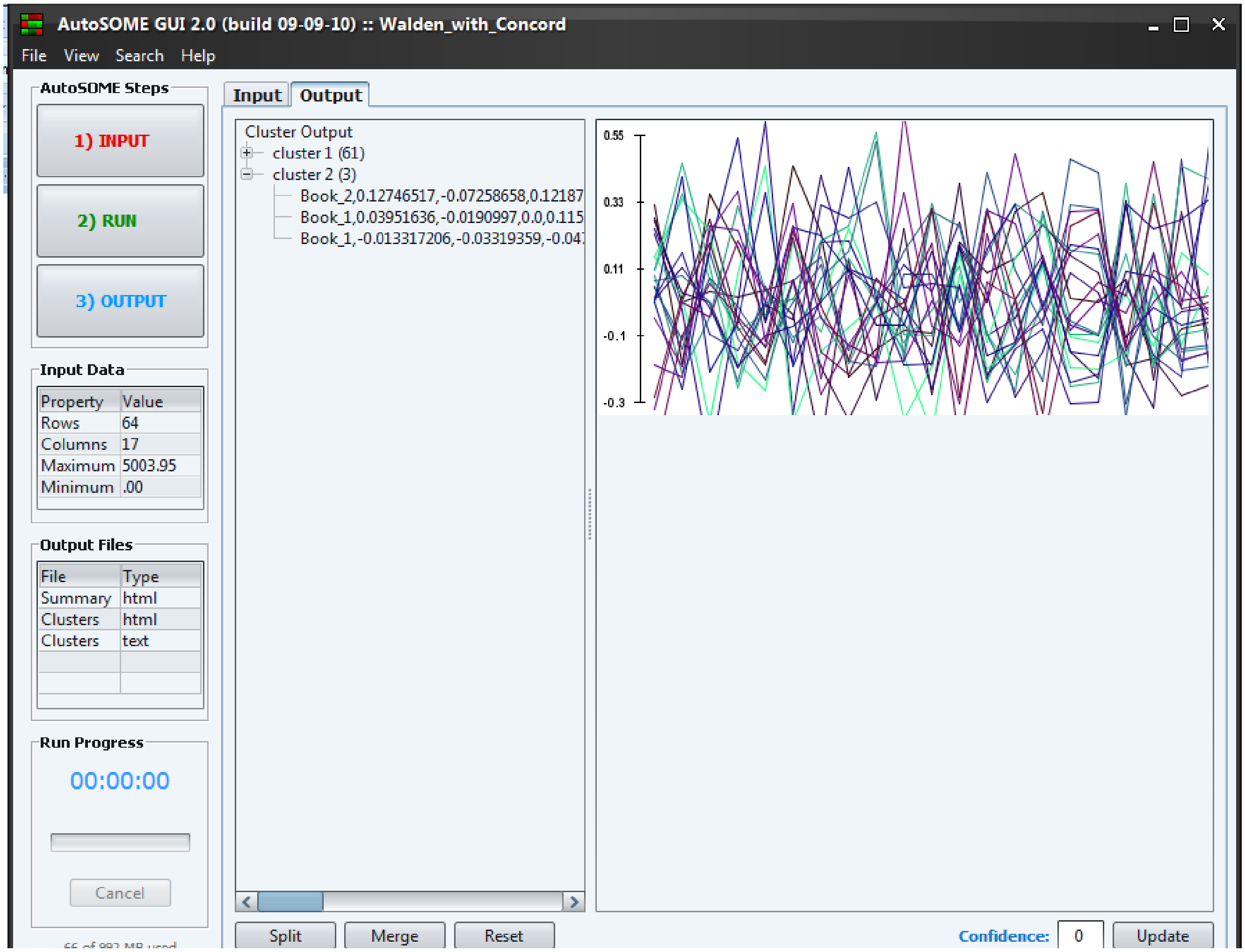
Run Progress

00:00:00

```
Cluster Output
cluster 1 (11)
    Author_1,-0.059213914,-0.14591318,0.34
    Author_1,-0.08383396,0.0,0.30482006,-0
    Author_1,-0.11135086,-0.113503456,0.16
    Author_1,0.0,-0.0037447833,0.4168658,-
    Author_1,-0.1545473,-0.24156389,-0.127
    Author_1,-0.02852621,-0.08938098,-0.12
    Author_1,0.076933704,-0.013330567,-0.1
    Author_1,-0.16554573,-0.3514832,0.0,0.0
    Author_1,-0.027108395,4.9064076E-4,0.0
    Author_1,-0.0953588,-0.03062346,0.0466
    Author_1,0.08771973,0.12511618,0.3232
cluster 2 (11)
    Author_2,0.5271726,0.55937517,-0.0840
    Author_2,0.114320174,0.38039556,-0.32
    Author_2,-0.022928981,0.057661545,-0.1
    Author_2,0.11891635,0.0062672547,-0.1
    Author_2,-0.08569334,0.06547012,-0.091
    Author_2,0.50425905,0.48088112,0.0797
    Author_2,0.08614244,-0.003094777,-0.17
    Author_2,0.42243102,0.13660535,0.0323
    Author_2,-0.2608273,-0.20305851,-0.089
    Author_2,-0.2476013,0.04787914,0.3773
    Author_2,0.13801292,0.021579985,0.254
```

Confidence: 0

Figure 4.8: Clustering result and heat map of the SOM in AutoSOME GUI



Confidence: 0

Property	Value
Rows	64
Columns	17
Maximum	5003.95
Minimum	.00

File	Type
Summary	html
Clusters	html
Clusters	text

00:00:00

```
Cluster Output
⊞ cluster 1 (61)
⊟ cluster 2 (3)
    Book_2,0.12746517,-0.07258658,0.12187
    Book_1,0.03951636,-0.0190997,0.0,0.115
    Book_1,-0.013317206,-0.03319359,-0.04
```

Figure 4.9: Clustering result and signal plot of the SOM for 150 segmentation in AutoSOME GUI

Appendix II

Feature Analysis of “Walden and Conduct”

Feature	Section									Total Frequency
	100	150	200	250	300	350	400	450	500	
total number of characters in words (NC)	0				1		1	1		3
total number of words (NW)										0
Average length per word (ALCW)										0
Words longer than 6 characters per words (WL6)					1	1	1	1		4
Total number of short words per words (1-3 characters) (WS3)		1	1	1	1		1			5
Hapax legomena/N (HL)										0
Hapax dislegomena/N (HD)	1	1			1	1		1		5
Yule’s K measure (YK)		1	1	1	1	1	1	1	1	8
Simpson’s D measure (SD)	1	1			1	1		1	1	6
Sichel’s S measure (SS)	1	1		1		1		1		5
Harden’s V measure (HV)	1	1	1	1	1	1	1	1	1	9
Brunets W measure (BW)		1		1					1	3
Honore’s R measure (HR)	1				1	1		1		4
No of Nouns (NN)	1	1		1	1	1	1	1	1	8
No of Passive Verbs (NPV)	1	1	1	1		1		1	1	7
No of Base Verbs (NBV)	1	1		1	1	1	1	1	1	8
No of Adjectives (NA)										0
No of Clauses (NCU)	1	1	1	1	1	1	1	1	1	9
No of Phrases (NP)	1	1	1	1		1	1	1		7
No of Domain Adverbs (NDA)		1	1	1			1	1		5
No of Duration Adverbs (NDUA)		1								1
No of Focus Adverbs (NFA)	1	1	1	1	1	1	1	1	1	9
No of Frequency Adverbs (NFRA)				1		1	1	1	1	5
No of Locating Adverbs (NLA)										0
No of Manner Adverbs (NMA)	1	1	1				1	1	1	6
No of Model Adverbs (NMOA)		1	1				1			3
No of Promina Adverbs (NPA)			1				1	1		3
No of Sequence Adverbs (NSA)			1			1	1			3
No of Syllables (NSY)										0
Word per Sentence (WPS)	1	1	1	1				1	1	6
Syllables per Words (SPW)										0
No of Complex Words (More than 3 Syllables) (NCW)										0
No of Articles (NAR)	1	1	1	1	1	1	1	1	1	9

No of Prepositions (NPR)										0
No of Coordinate Conjunctions (NCC)										0
No of Auxiliary Verbs (NAV)		1				1	1	1	1	5
No of Specific Words (NSW)					1	1	1	1	1	5
Flesh Index (FI)	1	1	1	1			1	1	1	7
Kincaid Index (KI)	1	1	1	1			1		1	6
Fog Index (FOG)	1	1	1	1				1	1	6
Average Word Frequency Class (AWFC)	1		1					1	1	4
No of Sentences (NSE)	1		1	1		1		1	1	6
Number of Commas (,) (NCO)		1		1	1			1		4
Number of Single Quotes(’) (NSQ)	1				1					2
Number of Double Quotes(’) (NDQ)					1		1		1	3
Number of Colons(:) (NCL)	1			1	1		1	1	1	6
Number of Semi-Colons(;) (NSC)						1				1
Number of Question Marks(?) (NQ)	1	1	1	1			1	1	1	7
Number of Exclamation Marks(!) (NE)	1	1	1	1		1	1	1		7
Number of etc. (NETC)										0
Total Number of Features used	**23**	**26**	**21**	**23**	**18**	**21**	**25**	**30**	**23**	

Table 4.1: Total number of features selected in each segment and the frequency of each feature in “Walden and Conduct”

Feature Analysis of “Walden and Concord”

	Section									**Total Frequency**
Feature	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**	
total number of characters in words (NC)	1	0	1	0	0	1	1	1	1	6
total number of words (NW)	0	0	0	0	0	0	0	0	0	0
Average length per word (ALCW)	0	0	0	0	0	0	0	0	0	0
Words longer than 6 characters per words (WL6)	0	0	1	0	0	0	1	0	0	2
Total number of short words per words (1-3 characters) (WS3)	1	1	1	1	1	1	1	1	1	9
Hapax legomena/N (HL)	0	0	0	0	0	1	0	1	0	2
Hapax dislegomena/N (HD)	1		1	1	1	0	0	0	0	4
Yule’s K measure (YK)	0	0	0	0	0	1	1	0	1	3
Simpson’s D measure (SD)	0	0	0	0	0	0	0	0	0	0
Sichel’s S measure (SS)	1		0	1	0	0	0	0	0	2
Harden’s V measure (HV)	1	1	0	1	0	1	1	1	1	7
Brunets W measure (BW)	1	0	1	1	0	1	0	1	1	6
Honore’s R measure (HR)	1	1	1	1	1	1	1	1	1	9
No of Nouns (NN)	1	0	1	0	1	1	1	0	0	5
No of Passive Verbs (NPV)	1	0	0	1	1	1	0	1	1	6

No of Base Verbs (NBV)	0	1	1	1	1	0	1	1	1	7
No of Adjectives (NA)	0	0	0	0	0	0	0	0	0	0
No of Clauses (NCU)	1	1	1	0	1	1	1	0	1	7
No of Phrases (NP)	1	1	1	1	1	1	1	1	1	9
No of Domain Adverbs (NDA)	1	0	1	1	1	1	1	1	1	8
No of Duration Adverbs (NDUA)	0	1	0	1	1	0	0	0	1	4
No of Focus Adverbs (NFA)	1	1	1	1	0	1	1	1	1	8
No of Frequency Adverbs (NFRA)	0	0	0	0	0	0	1	1	0	2
No of Locating Adverbs (NLA)	0	0	0	0	0	1	0	0	0	1
No of Manner Adverbs (NMA)	0	0	1	1	1	1	1	1	0	6
No of Model Adverbs (NMOA)	0	0	1	1	1	1	1	1	1	7
No of Promina Adverbs (NPA)	0	0	0	0	1	0	1	0	0	2
No of Sequence Adverbs (NSA)	0	0	0	0	1	0	1	1	0	3
No of Syllables (NSY)	1	0	1	1	1	1	1	1	1	8
Word per Sentence (WPS)	1	1	0	1	1	1	1	1	1	8
Syllables per Words (SPW)	0	0	0	0	0	0	0	0	0	0
No of Complex Words (More than 3 Syllables) (NCW)	0	0	0	0	0	0	0	0	0	0
No of Articles (NAR)	0	1	1	1	1	1	1	1	1	8
No of Prepositions (NPR)	1	0	0	0	0	1	0	1	0	3
No of Coordinate Conjunctions (NCC)	0	0	1	1	1	1	0	0	0	4
No of Auxiliary Verbs (NAV)	1	1	1	1	0	1	1	1	1	8
No of Specific Words (NSW)	0	0	0	0	0	0	0	0	0	0
Flesh Index (FI)	0	0	0	0	0	1	1	1	1	4
Kincaid Index (KI)	1	1	0	1	1	1	1	1		7
Fog Index (FOG)	1	1	0	1	1	1	1	1	1	8
Average Word Frequency Class (AWFC)	0	1	0	0	0	1	1	0	0	3
No of Sentences (NSE)	1	1	1	1	1	1	0	1	1	8
Number of Commas (,) (NCO)	0	1	0	0	0	0	0	0	0	1
Number of Single Quotes(’) (NSQ)	0	0	0	0	1	1	1	1	1	5
Number of Double Quotes(’) (NDQ)	0	0	1	1	0	1	1	1	0	5
Number of Colons(:) (NCL)	0	0	1	1	0	0	1	1	0	4
Number of Semi-Colons(;) (NSC)	0	0	0	1	0	0	0	0	0	1
Number of Question Marks(?) (NQ)	1	1	1	1	1	1	1	1	0	8
Number of Exclamation Marks(!) (NE)	1	0	1	1	0	1	1	1	0	6
Number of etc. (NETC)	0	0	0	0	0	0	0	0	0	0
Total Number of Features used	**22**	**17**	**23**	**27**	**23**	**31**	**30**	**29**	**22**	

Table 4.2: Total number of features selected in each segment and the frequency of each feature in “Walden and Concord”

Feature name	Walden and Conduct	Walden and Concord
total number of characters in words (NC)	Yes	Yes
total number of words (NW)	No	No
Average length per word (ALCW)	No	No
Words longer than 6 characters per words (WL6)	Yes	Yes
Total number of short words per words (1-3 characters) (WS3)	Yes	Yes
Hapax legomena/N (HL)	No	No
Hapax dislegomena/N (HD)	No	No
Yule’s K measure (YK)	Yes	Yes
Simpson’s D measure (SD)	No	No
Sichel’s S measure (SS)	No	No
Harden’s V measure (HV)	Yes	Yes
Brunets W measure (BW)	No	No
Honore’s R measure (HR)	Yes	No
No of Nouns (NN)	Yes	Yes
No of Passive Verbs (NPV)	No	No
No of Base Verbs (NBV)	Yes	Yes
No of Adjectives (NA)	No	No
No of Clauses (NCU)	Yes	Yes
No of Phrases (NP)	Yes	Yes
No of Domain Adverbs (NDA)	Yes	Yes
No of Duration Adverbs (NDUA)	No	No
No of Focus Adverbs (NFA)	Yes	Yes
No of Frequency Adverbs (NFRA)	Yes	Yes
No of Locating Adverbs (NLA)	No	No
No of Manner Adverbs (NMA)	Yes	Yes
No of Model Adverbs (NMOA)	Yes	Yes
No of Promina Adverbs (NPA)	Yes	Yes
No of Sequence Adverbs (NSA)	Yes	Yes
No of Syllables (NSY)	Yes	No
Word per Sentence (WPS)	Yes	No
Syllables per Words (SPW)	No	No
No of Complex Words (More than 3 Syllables) (NCW)	No	No
No of Articles (NAR)	Yes	Yes
No of Prepositions (NPR)	No	No
No of Coordinate Conjunctions (NCC)	No	No
No of Auxiliary Verbs (NAV)	Yes	Yes
No of Specific Words (NSW)	No	Yes
Flesh Index (FI)	Yes	Yes
Kincaid Index (KI)	Yes	Yes
Fog Index (FOG)	Yes	No
Average Word Frequency Class (AWFC)	Yes	No
No of Sentences (NSE)	No	No
Number of Commas (,) (NCO)	No	No
Number of Single Quotes(’) (NSQ)	Yes	No
Number of Double Quotes(’) (NDQ)	Yes	Yes
Number of Colons(:) (NCL)	Yes	Yes
Number of Semi-Colons(;) (NSC)	No	No

Number of Question Marks(?) (NQ)	Yes	Yes
Number of Exclamation Marks(!) (NE)	Yes	Yes
Number of etc. (NETC)	No	No

Table 4.3: Selected and not selected features in the most significant segmentation for the discrimination

Feature Analysis of "English and Concord"

Feature	**Section**									**Total Frequency**
	100	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**	
total number of characters in words (NC)	1	1	1	1	1	1	1			7
total number of words (NW)										0
Average length per word (ALCW)										0
Words longer than 6 characters per words (WL6)	1	1			1	1	1	1	1	7
Total number of short words per words (1-3 characters) (WS3)	1	1	1	1	1	1	1	1	1	9
Hapax legomena/N (HL)						1				1
Hapax dislegomena/N (HD)						1	1		1	3
Yule's K measure (YK)				1	1	1	1	1	1	6
Simpson's D measure (SD)	1	1		1		1	1			5
Sichel's S measure (SS)									1	1
Harden's V measure (HV)	1	1	1		1	1		1		6
Brunets W measure (BW)	1	1	1		1	1		1		6
Honore's R measure (HR)	1					1				2
No of Nouns (NN)					1	1	1	1	1	5
No of Passive Verbs (NPV)				1	1	1	1	1		5
No of Base Verbs (NBV)	1		1			1		1		4
No of Adjectives (NA)										0
No of Clauses (NCU)			1	1			1	1		4
No of Phrases (NP)					1		1			2
No of Domain Adverbs (NDA)				1						1
No of Duration Adverbs (NDUA)								1		1
No of Focus Adverbs (NFA)		1					1	1		3
No of Frequency Adverbs (NFRA)						1	1	1		3
No of Locating Adverbs (NLA)										0
No of Manner Adverbs (NMA)										0
No of Model Adverbs (NMOA)										0
No of Promina Adverbs (NPA)			1			1	1	1		4
No of Sequence Adverbs (NSA)	1				1	1		1		4

No of Syllables (NSY)	1	1	1	1	1	1	1	1		8
Word per Sentence (WPS)	1	1	1	1	1	1	1	1	1	9
Syllables per Words (SPW)										0
No of Complex Words (More than 3 Syllables) (NCW)										0
No of Articles (NAR)		1	1	1	1	1	1	1	1	8
No of Prepositions (NPR)	1	1	1		1	1		1	1	7
No of Coordinate Conjunctions (NCC)			1	1				1		3
No of Auxiliary Verbs (NAV)	1	1	1	1		1	1	1		7
No of Specific Words (NSW)					1	1	1			3
Flesh Index (FI)	1	1	1	1	1	1	1		1	8
Kincaid Index (KI)	1	1	1	1	1	1	1	1	1	9
Fog Index (FOG)	1	1	1	1	1	1		1	1	8
Average Word Frequency Class (AWFC)	1	1	1	1	1	1	1	1	1	9
No of Sentences (NSE)	1	1	1	1	1	1	1	1	1	9
Number of Commas (,) (NCO)					1	1	1	1		4
Number of Single Quotes(') (NSQ)		1								1
Number of Double Quotes(') (NDQ)		1	1	1	1		1		1	6
Number of Colons(:) (NCL)					1	1	1	1	1	5
Number of Semi-Colons(;) (NSC)	1	1			1	1	1	1	1	7
Number of Question Marks(?) (NQ)										0
Number of Exclamation Marks(!) (NE)						1	1	1	1	4
Number of etc. (NETC)										0
Total Number of Features used	**19**	**20**	**19**	**18**	**24**	**31**	**27**	**28**	**18**	

Table 4.4: Total number of features selected in each segment and the frequency of each feature in "English and Concord"

Feature Analysis of "English and Conduct"

Feature	Section									Total Frequency
	100	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**	
total number of characters in words (NC)					1					1
total number of words (NW)										0
Average length per word (ALCW)		1								1
Words longer than 6 characters per words (WL6)	1				1	1	1	1	1	6
Total number of short words per words (1-3 characters) (WS3)								1		1
Hapax legomena/N (HL)	1					1		1	1	4
Hapax dislegomena/N (HD)				1		1	1	1		4

Yule’s K measure (YK)	1	1	1	1	1	1	1	1	1	9
Simpson’s D measure (SD)	1		1		1		1	1	1	6
Sichel’s S measure (SS)	1			1		1				3
Harden’s V measure (HV)	1	1	1	1	1	1	1	1	1	9
Brunets W measure (BW)					1	1			1	3
Honore’s R measure (HR)		1				1			1	3
No of Nouns (NN)	1	1	1	1	1		1	1		7
No of Passive Verbs (NPV)	1	1		1	1	1	1	1	1	8
No of Base Verbs (NBV)		1				1			1	3
No of Adjectives (NA)										0
No of Clauses (NCU)		1		1	1		1			4
No of Phrases (NP)	1	1	1	1	1	1	1	1	1	9
No of Domain Adverbs (NDA)										0
No of Duration Adverbs (NDUA)				1		1				2
No of Focus Adverbs (NFA)		1		1		1	1	1		5
No of Frequency Adverbs (NFRA)			1		1		1	1	1	5
No of Locating Adverbs (NLA)										0
No of Manner Adverbs (NMA)	1									1
No of Model Adverbs (NMOA)							1		1	2
No of Promina Adverbs (NPA)							1	1	1	3
No of Sequence Adverbs (NSA)						1	1	1	1	4
No of Syllables (NSY)				1	1	1	1	1		5
Word per Sentence (WPS)							1			1
Syllables per Words (SPW)										0
No of Complex Words (More than 3 Syllables) (NCW)										0
No of Articles (NAR)	1	1	1	1	1	1	1	1	1	9
No of Prepositions (NPR)			1	1						2
No of Coordinate Conjunctions (NCC)	1						1	1		3
No of Auxiliary Verbs (NAV)		1	1	1	1	1	1	1		7
No of Specific Words (NSW)			1			1			1	3
Flesh Index (FI)										0
Kincaid Index (KI)							1			1
Fog Index (FOG)							1			1
Average Word Frequency Class (AWFC)		1	1	1	1			1	1	6
No of Sentences (NSE)				1	1	1	1	1		5
Number of Commas (,) (NCO)	1	1	1	1	1	1	1	1	1	9
Number of Single Quotes(’) (NSQ)	1					1				2
Number of Double Quotes(’) (NDQ)		1		1		1	1	1	1	6
Number of Colons(:) (NCL)	1	1			1		1		1	5
Number of Semi-Colons(;) (NSC)				1	1		1	1	1	5
Number of Question Marks(?) (NQ)	1	1			1	1				4

Number of Exclamation Marks(!) (NE)						1	1	1		3
Number of etc. (NETC)							1			1
Total Number of Features used	**16**	**17**	**12**	**19**	**20**	**24**	**28**	**24**	**21**	

Table 4.5: Total number of features selected in each segment and the frequency of each feature in "English and Concord"

Feature name	Walden and Conduct	Walden and Concord
total number of characters in words (NC)	No	Yes
total number of words (NW)	No	No
Average length per word (ALCW)	No	No
Words longer than 6 characters per words (WL6)	Yes	Yes
Total number of short words per words (1-3 characters) (WS3)	No	Yes
Hapax legomena/N (HL)	No	No
Hapax dislegomena/N (HD)	Yes	Yes
Yule's K measure (YK)	Yes	Yes
Simpson's D measure (SD)	Yes	Yes
Sichel's S measure (SS)	No	No
Harden's V measure (HV)	Yes	No
Brunets W measure (BW)	No	No
Honore's R measure (HR)	No	No
No of Nouns (NN)	Yes	Yes
No of Passive Verbs (NPV)	Yes	Yes
No of Base Verbs (NBV)	No	No
No of Adjectives (NA)	No	No
No of Clauses (NCU)	Yes	Yes
No of Phrases (NP)	Yes	Yes
No of Domain Adverbs (NDA)	No	No
No of Duration Adverbs (NDUA)	No	No
No of Focus Adverbs (NFA)	Yes	Yes
No of Frequency Adverbs (NFRA)	Yes	Yes
No of Locating Adverbs (NLA)	No	No
No of Manner Adverbs (NMA)	No	No
No of Model Adverbs (NMOA)	Yes	No
No of Promina Adverbs (NPA)	Yes	Yes
No of Sequence Adverbs (NSA)	Yes	No
No of Syllables (NSY)	Yes	Yes
Word per Sentence (WPS)	Yes	Yes
Syllables per Words (SPW)	No	No
No of Complex Words (More than 3	No	No

Syllables) (NCW)		
No of Articles (NAR)	Yes	Yes
No of Prepositions (NPR)	No	No
No of Coordinate Conjunctions (NCC)	Yes	No
No of Auxiliary Verbs (NAV)	Yes	Yes
No of Specific Words (NSW)	No	Yes
Flesh Index (FI)	No	Yes
Kincaid Index (KI)	Yes	Yes
Fog Index (FOG)	Yes	No
Average Word Frequency Class (AWFC)	No	Yes
No of Sentences (NSE)	Yes	Yes
Number of Commas (,) (NCO)	Yes	Yes
Number of Single Quotes(’) (NSQ)	No	No
Number of Double Quotes(’) (NDQ)	Yes	Yes
Number of Colons(:) (NCL)	Yes	Yes
Number of Semi-Colons(;) (NSC)	Yes	Yes
Number of Question Marks(?) (NQ)	No	No
Number of Exclamation Marks(!) (NE)	Yes	Yes
Number of etc. (NETC)	Yes	No

Table 4.6: Selected and not selected features in most significant segmentation for the discrimination in the second experiment

Analysis of frequency of each feature selection of four experiments

Feature name	Frequency	Percentage
Simple Ratios		
total number of characters in words (NC)	3	**75%**
total number of words (NW)	0	**0%**
No of Sentences (NSE)	2	**50%**
Word per Sentence (WPS)	3	**75%**
Average length per word (ALCW)	0	**0%**
Word Based Features		
Words longer than 6 characters per words (WL6)	4	**100%**
Total number of short words per words (1-3 characters) (WS3)	3	**75%**
No of Syllables (NSY)	3	**75%**
Syllables per Words (SPW)	0	**0%**
No of Complex Words (More than 3 Syllables) (NCW)	0	**0%**
No of Specific Words (NSW)	2	**50%**
Vocabulary Richness Measures		
Hapax legomena/N (HL)	0	**0%**
Hapax dislegomena/N (HD)	2	**50%**
Yule’s K measure (YK)	4	**100%**
Simpson’s D measure (SD)	2	**50%**

Sichel’s S measure (SS)	0	**0%**
Harden’s V measure (HV)	3	**75%**
Brunets W measure (BW)	0	**0%**
Honore’s R measure (HR)	1	**25%**
Average Word Frequency Class (AWFC)	2	**50%**
Syntactic and POS Features		
No of Nouns (NN)	4	**100%**
No of Passive Verbs (NPV)	2	**50%**
No of Base Verbs (NBV)	2	**50%**
No of Adjectives (NA)	0	**0%**
No of Clauses (NCU)	4	**100%**
No of Phrases (NP)	4	**100%**
No of Articles (NAR)	4	**100%**
No of Prepositions (NPR)	0	**0%**
No of Coordinate Conjunctions (NCC)	1	**25%**
No of Auxiliary Verbs (NAV)	4	**100%**
Adverbial Features		
No of Domain Adverbs (NDA)	2	**50%**
No of Duration Adverbs (NDUA)	0	**0%**
No of Focus Adverbs (NFA)	4	**100%**
No of Frequency Adverbs (NFRA)	4	**100%**
No of Locating Adverbs (NLA)	0	**0%**
No of Manner Adverbs (NMA)	2	**50%**
No of Model Adverbs (NMOA)	3	**75%**
No of Promina Adverbs (NPA)	4	**100%**
No of Sequence Adverbs (NSA)	3	**75%**
Readability Measures		
Flesh Index (FI)	3	**75%**
Kincaid Index (KI)	4	**100%**
Fog Index (FOG)	2	**50%**
Punctuation Features		
Number of Commas (,) (NCO)	2	**50%**
Number of Single Quotes(’) (NSQ)	1	**25%**
Number of Double Quotes(’) (NDQ)	4	**100%**
Number of Colons(:) (NCL)	4	**100%**
Number of Semi-Colons(;) (NSC)	2	**50%**
Number of Question Marks(?) (NQ)	2	**50%**
Number of Exclamation Marks(!) (NE)	4	**100%**
Number of etc. (NETC)	1	**25%**

Table 4.7: Percentages of the usage of each feature in each category

Feature Category	Total Number of features	Total number of selected features	Percentage
Simple Ratios	5	3	60
Word Based Features	6	4	67
Vocabulary Richness Measures	9	5	56
Syntactic and POS Features	10	7	70
Adverbial Features	9	7	78
Readability Measures	3	3	100
Punctuation Measures	8	6	75

Table 4.8: Usage of features in each category