Plagiarism Detection using Enhanced Relative Frequency Model

Kotha Dinesh Reddy

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Plagiarism Detection using Enhanced Relative Frequency Model

Thesis submitted in

May 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Computer Science and Engineering

Specialization : Software Engineering

by

Kotha Dinesh Reddy

[Roll No. 211cs3297]

under the guidance of

Prof. Korra Sathya Babu

**Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India**

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Mr. Korra Sathya Babu
Assistant Professor

Certificate

This is to certify that the work in the thesis entitled *Plagiarism Detection using Enhanced Relative Frequency Model* by *Kotha Dinesh Reddy* is a record of an original work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Korra Sathya Babu

Acknowledgment

I would like to express my gratitude to my thesis guide Prof. Korra Sathya Babu for the useful comments, remarks and engagement through the learning process of this master thesis. The flexibility of work he has offered me has deeply encouraged me producing the research.

Furthermore I would like to thank Prof. Durga Prasad Mohapatra, Prof. Santanu Kumar Rath, Prof. Ashok Kumar Turuk, Prof. Banshidhar Majhi for for being a source of support and motivation for carrying out quality work.

My hearty thanks goes to Mr. Suresh for consistently showing me innovative research directions for the entire period of carrying out the research and helping me in shaping up the thesis. Also, I like to thank the participants in my survey, who have willingly shared their precious time during the process of interviewing. I would like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love.

Last but not the least,i like to thank my family and the one above all of us, the omnipresent God, for answering my prayers for giving me the strength to plod on despite my constitution wanting to give up, thank you so much Dear Lord.

Kotha Dinesh Reddy

Abstract

As the world is running towards greater heights of technology, it's becoming more complex to secure data from being copied. So it's better to detect the copied contents rather than securing the contents. Here, contents cover digital documents of scientific research, articles in newspapers, journals and assignments submitted by students. There are so many tools and algorithms to detect plagiarism, but the time complexity of the algorithm really matters where document comparison is against giant data set. Vector based methods are quite frequently used in the detection process of plagiarism. There are so many vector based methods, but having some drawbacks. In SCAM approach, selection of ε value is a drawback as ε value decides the closeness set and daniel approach fails to identify plagiarism when there were repeated terms in a sentence. Here we are proposing a new algorithm, which is developed using the concepts of the Relative Frequency Model overcomes the drawbacks involved in existing methods. In the implementation of our proposed method, we employed sentence splitter, stop-word removal process, and stemming of words.

Keywords: Information Retrieval, Plagiarism Detection, Copy Detection, Relative Frequency Model, Enhanced Relative Frequency Model.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter of the thesis, we are going to give a brief introduction about what is plagiarism and how it can be detected. After that we are going to give you a taxonomy of plagiarism, so that will help us in understanding the various types of plagiarism and how it can be performed. In the later sections, we give various technical terms and formulas used in plagiarism detection process, motivation and objectives of our research, and finally, we give an outline for the organization of the thesis.

In this thesis, we have done a brief survey on plagiarism types and plagiarism detection methods, which help us to have an idea of what kind of plagiarism it is about and how it is performed and how it can be detected. Here we also propose a new approach for the detection of plagiarism in suspected document w.r.t. to possible source documents.

1.1 Introduction to Plagiarism Detection

The process of presenting one's creative ideas, images, sounds, thoughts or any kind of literal property as our own creation is called "Plagiarism."Some times, plagiarism may occur unintentionally. Here Plagiarism doesn't state that, one cannot use other's thoughts or works, but it states that we cannot use them in our work without giving proper credit to the original author. Even though performing of plagiarism is not limited to text, here we mainly concentrated on plagiarism in text documents, because text documents play a vital role in academics, research and even in news articles.

The research works, invention of new methodologies, and innovative thoughts of researchers are also considered as IPR (Intellectual Property Rights) like the other and should be protected from other whom could misuse them.

Plagiarism has become a world-wide problem and is increasing day by day. This problem is getting worse mainly because of the high availability of original documents over the web in various forms [1]. Because of this problem, it's becoming easy for authors to reuse the data in their documents.

There are mainly two types of plagiarism.

1. plagiarism in programming languages
2. plagiarism in natural languages

The methods and algorithms of plagiarism detection in programming languages and natural languages are significantly different in nature.

Plagiarism detection in programming languages mainly focuses on keeping track of metrics, such as the number of lines, variables used, statements, subprograms, and other parameters, where plagiarism detection in natural languages focused on various kinds of textual features and different type of exhaustive methods.

There are several types of plagiarism depending on the level of reuse of original content.

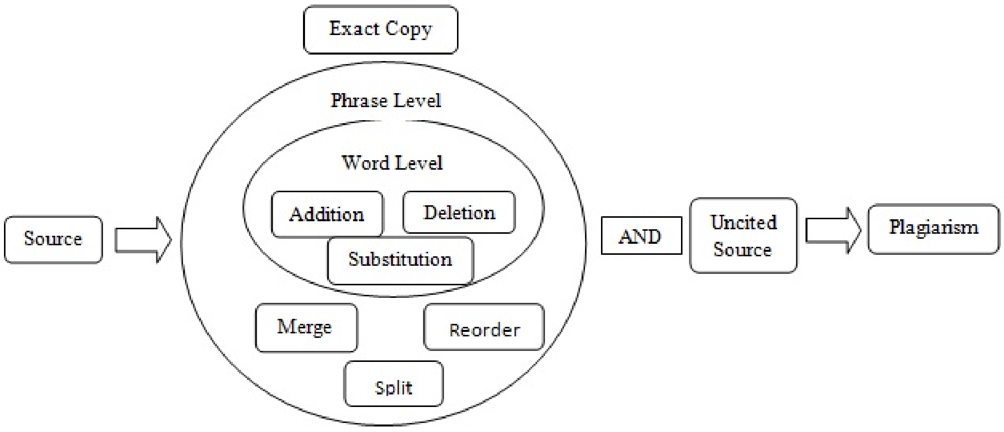


Figure 1.1: simple ways to perform Plagiarism

From the Figure 1.1, we can observe how plagiarism takes place at phrase level and word level. At word level, persons who are performing plagiarism can delete one or more words from a sentence (or) can add one or more words to a sentence. Some times, people with more vocabulary power can substitute one or more words with their relatively meaningful words in-order to avoid the problem of exact copy match. At the phrase level, a person can merge two or more phrases into one phrase (or) he can split one phrase into two or more phrases (or) he can simply reorder the word positions in that phrase.

Relying only on exact-word or phrase matching for plagiarism detection is not sufficient now [1]. People have started paraphrasing or rearranging words to give a new look to their sentences and thus declare themselves as authors of the material. Plagiarism is a complex problem and considered one of the biggest violations in publishing of scientific, engineering and other related types of documents or papers.

Plagiarism detection is a process of finding similar documents for a doubtful document by extracting different features like structural, semantic, syntactic and lexical, from that document and analyzing those features. Even a human can detect the similarity between the plagiarized and original documents, but "it requires much effort to be aware of all potential sources on a given topic and to provide strong evidence against an offender"[2].

An On-line Repository of documents provides access to all digitized documents in that repository for all kinds of users. Most of the times or sometimes the user may not have access permission to all the documents, but irrespective of this issue, users can access the documents and can use those documents in their work. So until this problem goes off some of the authors may be reluctant to upload their works into these repositories. And this could lead to duplication of work. So we should handle this problem with on-line repositories.

Handling this problem can be categorized into two groups [3].

- Copy Prevention
- Copy Detection

Copy prevention schemes include physical isolation of information, use of special-purpose hardware for authorization, and active documents those are essential documents encapsulated by programs. But prevention techniques may be cumbersome, may get into the way of the honest user, and may make it difficult to share information. However, prevention schemes are not always tool-proof since documents may be recorded by using software emulators.

The other approach is not to place restrictions on the distribution of documents, but to detect illegal copies. Detection schemes fall in two categories, signature based and registration based. In signature based schemes, a “signature”is added to the document, and this signature can be used to trace the origins of the document.

Signature schemes have two weaknesses [3]:

1. The signatures often can be removed automatically, leading to untraceable documents.

2. They are not useful for detecting partial overlap.

For these reasons, we use registration based copy detection schemes. In the usage of these scheme original documents are registered and stored in a database. Later, the new documents will be compared against the preregistered documents for partial or complete overlap.

In the plagiarism detection process, we compare the contents of test document with the previous documents, and these previous documents generally reside in the document database known as “Data Set.”And there are many methods for comparing the documents and finding the relativeness between them. These methods involve selections of appropriate frame work for their proposed method, textual features they are going to use, and type of model they are using.

1.2 Taxonomy of Plagiarism

No two persons can write exactly identical content even though they are having same kind of ideas to present and using same natural language. So the textual content on the same topic from two different person must be different at least by some extent. We can divide the plagiarism into various groups depending on the level of expertise and knowledge of the suspected author.

According to the taxonomy of plagiarism[2], we can mainly divide the plagiarism into literal and intelligent plagiarism, we can observe this from Figure 1.2. Several researchers worked on different type of plagiarisms and available software for it’s detection[4, 5, 6, 7].

- **Literal Plagiarism**

  Literal Plagiarism is a common most types of plagiarism. Suspected author need not to have any extra amount of knowledge in that particular natural language in which the original document is existing. We can divide this literal type of plagiarism into three types.

  1. **Exact Copy**

     In this kind of plagiarism suspected author can copy a whole document or some part of original document and uses it in his own document.

  2. **Near Copy:**

     In this kind of plagiarism suspected author inserts or deletes or a set of words from original document or substitutes a set of words with another set of words in original document and used the modified textual contents in his own document.

  3. **Modified Copy**

     In this kind of plagiarism suspected author modifies the structure of sentences in original document by reordering the phrases in those sentences.

- **Intelligent Plagiarism**

  Intelligent Plagiarism is a serious threat to the academics and scientific researches where suspected author changes the original textual content in various intelligent ways, which includes text manipulation, translation, and idea adoption [8].

  1. **Text manipulation**

     Plagiarism can be done by manipulating the text and changing the most of its appearance. It can be done by doing paraphrasing the original content or summarizing it.

  2. **Translation**

     Plagiarism can also be done by translating the text from one natural language to another natural language. It includes automatic translation and manual translation.

  3. **Idea adoption**

Idea adoption is the most serious type of plagiarism that refers to the use of original author ideas, such as results, contributions, and conclusions, without giving proper credit to the original author.

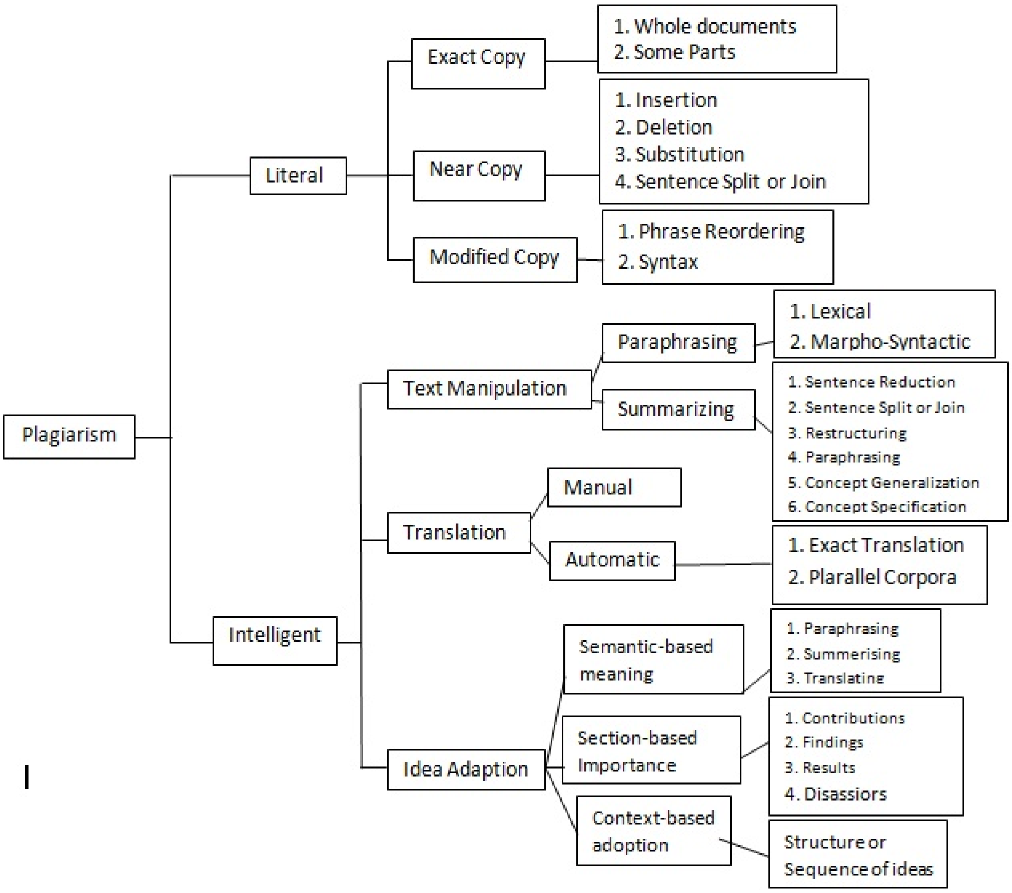


Figure 1.2: Taxonomy of Plagiarism

1.3 Terminology and General Formulas

- Precision: In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the search [9].

$$\text{Precision} = \frac{|\{relevant \ \ documents\} \cap \{retrieved \ \ documents\}|}{|\{retrieved \ \ documents\}|}$$

- Recall: Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved [9].

$$\text{Recall} = \frac{|\{relevent\ \ documents\} \cap \{retrieved\ \ documents\}|}{|\{relevant\ \ documents\}|}$$

- Stop words : In information retrieval, stop words are the words that frequently occur in documents. These words do not give any hints or meanings to the content of their documents such as (the, a, and), hence they are eliminated from the set of index terms. Salton and McGill [10] reported that such words comprise around 40 to 50% of a collection of documents text words. Eliminating the stop words in automatic indexing will speed the system processing, saves a huge amount of space in index, and does not damage the retrieval effectiveness.

- Words stemming: One of the problems involved in information retrieval is variation in word forms. The most common types of variation are spelling errors, alternative spelling, multi-word construction, transliteration, affixes, and abbreviations. One way to overcome such problem is to use stemming. Stemming is a process to remove the affixes (prefixes and suffixes) in a word in order to generate its root word. Using root word in pattern matching provides a much better effectiveness in information retrieval.
  Ex: learning will become learn after stemming, as well as learned.

- Word sense disambiguation: Word sense disambiguation is the process of identifying which sense of a word is used in a sentence, when the word has multiple meanings.
  Ex: Word “bank”has different meaning depending on the scenario its used.

  1. river bank
  2. financial bank
  3. bank of books, etc...

1.4 Motivation

As we have seen, a lot of research work is going on these days on various streams of sciences it is necessary to protect the originality from duplication. And for digital documents, plagiarism detection is the only way to protect the originality. And if we implement plagiarism detection more deeply, then we can improve the genuine research also. If we don't implement this detection process more cleanly, some of the plagiarized contents may slip away from our matching process.

It has been observed from the literature survey that quite a good number of schemes and approaches have been proposed until the date. Furthermore, so many researchers are still actively working in this domain. This is because the proposed schemes for plagiarism detection system are computationally high cost as the input data set for source documents are increasing extensively and mainly there is still lack of efficient enough systems to understand the semantic meanings of natural languages, i.e., no system can understand a document as a human brain understands it.

From the literature survey on various plagiarism detection methods, it is observed that vector based methods are having fewer computational costs when compared with other methods and uses most simple and effective textual features like lexical and syntactic and semantic features.

However, it is observed that the some approaches in vector based methods are having some pitfalls. In SCAM approach, there is no fixed value for ε, which is deciding factor in selection of closeness set. And it has been observed that depending on ε similarity values also changing which is not acceptable. In Daniel's approach, they used common words shared between the suspected and source document sentences in-order to find the similarity between them. This approach works well when there are no repeated words within a sentence, i.e. word frequency in a sentence, but if the word frequencies of individual words are not same between the sentences under consideration will lead to wrong results.

Keeping the research directions in view, it has been realized that there exists enough scope to improve the efficiency of vector based methods. In the thesis, concentration has been made to enhance the concepts of relative frequencies of individual words within the sentences under consideration and to improve the efficiency of finding similarity by using different textual features like lexical, semantic and syntactic features.

1.5 Objective

In particular, the objectives are to

1. Enhance the Relative Frequency Model used in SCAM Approach.

2. Use most of textual features to get better results.

3. Utilization of word net in-order to identify the word replacement plagiarism.

4. Implementation of stop word removal in-order to make the computational costs less

1.6 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2, we discuss about various kinds of frame works used in plagiarism detection systems. And then we give a list of various types of textual features used in the evaluation of documents to find similarity values in the plagiarism detection process. And finally, we stated various kinds of exhaustive methods used for evaluation of similarity between documents using the textual features.

Chapter 3 gives brief overview of various methods introduced and implemented by different researchers which uses different types of textual features, frame work, and exhaustive methods.

Chapter 4, we list the drawbacks of existing methods in two vector based methods, i.e. SCAM approach and Daniel approach, and discuss about our proposed methods which utilizes the concepts of relative frequencies and over comes the drawbacks of above approaches. And we discussed about the frame work used for our proposed method.

Chapter 5, presents different softwares and packages we used in order to implement our proposed approach along with existing approaches. We give the results we got for various kinds of input data sets and we made the comparison between the results we got for different vector based approach.

Chapter 6, provides the concluding remarks with an emphasis on achievements and limitations of the proposed scheme when compared with existing vector based approaches. The scopes for further research are outlined at the end.

Chapter 2

Literature Review

Plagiarism detection is a process of identifying and extracting various features in documents, analyzing those features, and using those features on various exhaustive methods for detecting similar contents between the documents, if there any.

In this chapter, we are going to discuss about various frame works used in the design of plagiarism detection system, various textual features used in the detection of plagiarism, and various exhaustive methods used to detect the similarity between suspicious and source documents as well as the level of plagiarism involved between them.

2.1 Frame Works used in Plagiarism Detection System (PDS)

Depending upon the visibility nature of the framework used for the design of plagiarism detection system, we can mainly divide into two types

2.1.1 Black Box Frame Work

In this type of frame work, we can't observe what are the various operations going on/performed in a plagiarism detection system, in order to get the related/similar work for suspicious work. Figure 3, shows us the procedure of a black box frame work for plagiarism detection.

In Fig. 2.1 Query Document d_s is cross checked over Document Set D, in order to find possible plagiarised sentences s_p in d_s.

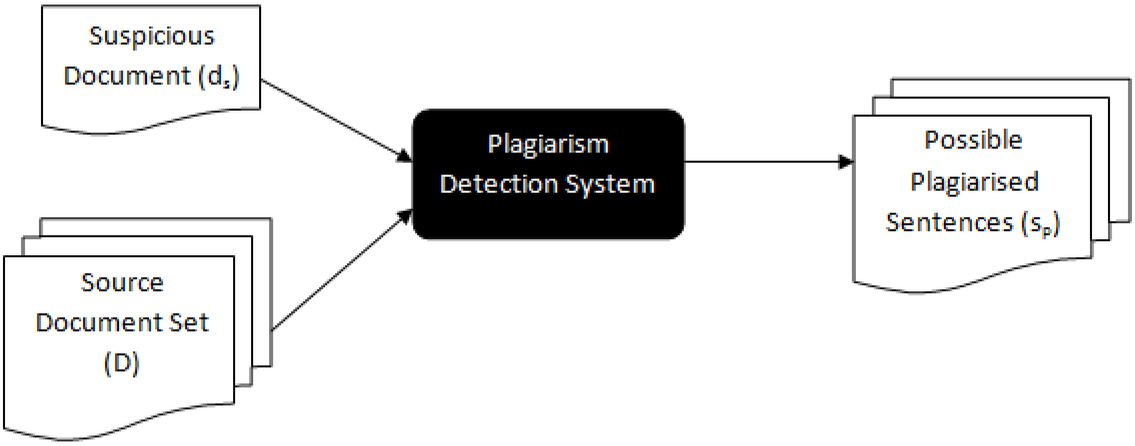

Suspicious Document (d_s)
Source Document Set (D)
Plagiarism Detection System
Possible Plagiarised Sentences (s_p)

Figure 2.1: Black box frame work

2.1.2 White Box Frame Work:

In simple terms, we can say White box frame work is an opposite model for Black Box frame work. It means here we can observe what are the various processes and procedures taken care in order to find suspicious document and its related similar documents in a particular plagiarism detection system.

This type of frame work will give us the information about various procedures taken place in a plagiarism detection system along with inputs and outputs. Fig. 2.2 shows us the procedure of a white box frame work for plagiarism detection.

Design of Plagiarism Detection System can be further classified into Intrinsic and Extrinsic depending on the situation where both suspicious and source document actively participated in detection process or suspicious document alone.

Intrinsic Plagiarism Detection

In intrinsic plagiarism detection, we try to detect the suspicious contents or plagiarized contents by looking into that document alone. It conveys that we are not using any kind of

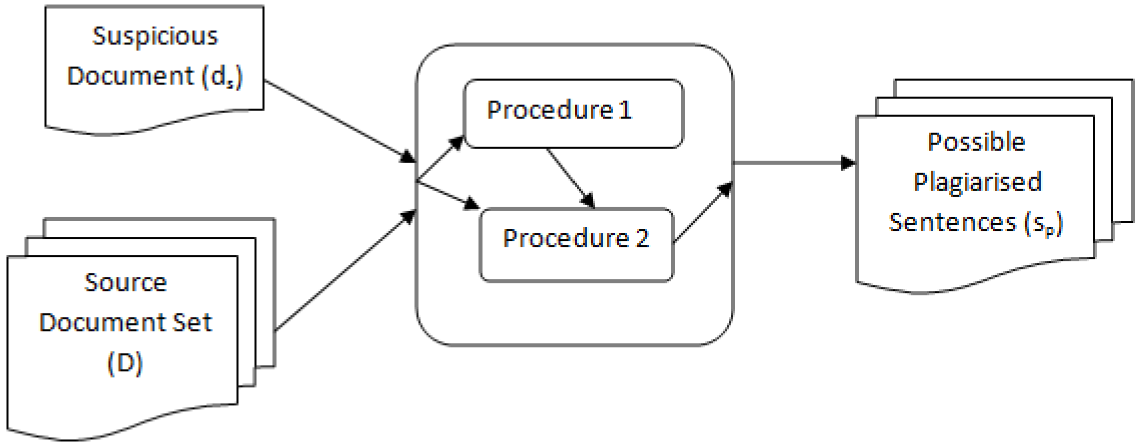

Suspicious Document (d_s)
Source Document Set (D)
Procedure 1
Procedure 2
Possible Plagiarised Sentences (s_p)

Figure 2.2: White box frame work

document collection to compare the suspicious document.

Intrinsic plagiarism can also be considered as a combination of authorship verification and attribution. Because intrinsic plagiarism detection analyzes the query document in alone. Here all three intrinsic plagiarism, authorship verification, and authorship attribution are similar tasks but having different approaches. A few researchers worked on intrinsic plagiarism detection[11, 12, 13].

"Intrinsic plagiarism aims at identifying potential plagiarism by analyzing a document with respect to undeclared changes in writing style. Authorship verification aims at determining whether or not a text with doubtful authorship is from an author A, given some writing examples of A, while authorship attribution aims at attributing a documents d of unknown authorship, given a set D of candidate authors with writing examples"[14].

Here, in Intrinsic plagiarism detection, the input is nothing but a suspicious document d_s. This frame work has three main steps.

- Step 1: d_s is segmented into smaller parts, such as sections, paragraphs, or sentences. Text segmentation can be done on word level as well.
- Step 2: Stylometric features are extracted from the segmented parts.
- Step 3: Stylometric - based measurements and generalization functions are employed to analyze the variance of different style features. Parts with style which are inconsistent with the remaining document style are marked as possible plagiarized.

Thus the final output is fragments or sections $s_p : s_p \in d_s$ such that s_p has quantified writing style features different from other sections s in d_s. Fig. 2.3 shows White Box representation of Intrinsic Plagiarism Detection.

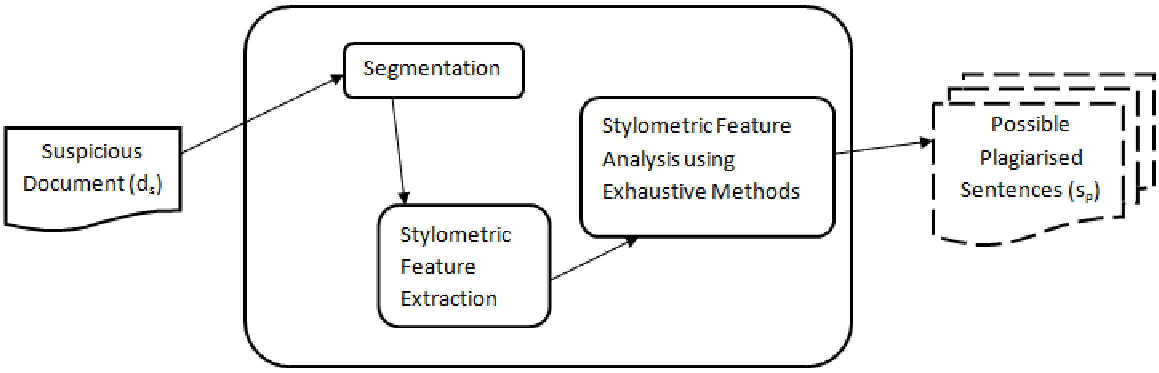

Suspicious Document (d_s)
Segmentation
Stylometric Feature Extraction
Stylometric Feature Analysis using Exhaustive Methods
Possible Plagiarised Sentences (s_p)

Figure 2.3: White box representation of Intrinsic plagiarism detection

Extrinsic plagiarism detection:

Extrinsic plagiarism detection is a method of finding similarity for a suspicious document against a numerous set of collection of data, where we get different types of textual features to find similarity[15]. Major part of research about plagiarism detection has been done using extrinsic plagiarism detection only.

The inputs for this type of plagiarism detection system are a suspicious query document d_s and a reference dataset collection D which may contain the sources of plagiarism. In the extrinsic model there are three main stages in which we perform different operations to detect plagiarism.

- Step 1: A set of Candidate Documents d_c will be retrieved from input set of documents, which may contain sources for plagiarism. This procedure done by IR model.

- Step 2: A pair wise feature based exhaustive analysis is performed to compare d_s with its candidates by using some Exhaustive method.

- Step 3: A knowledge based post processing step is performed to merge small detected units into passages or sections and to present the result to a human, who may decide whether or not a plagiarism offense is given.

Thus, the final output is pairs of sections (s_c, s_p), where $s_c \in d_c$, $s_p \in d_s$, and $d_c \in D$, such that s_p is pattern of plagiarism from s_c. Fig. 2.4 shows White Box representation of Extrinsic Plagiarism Detection.

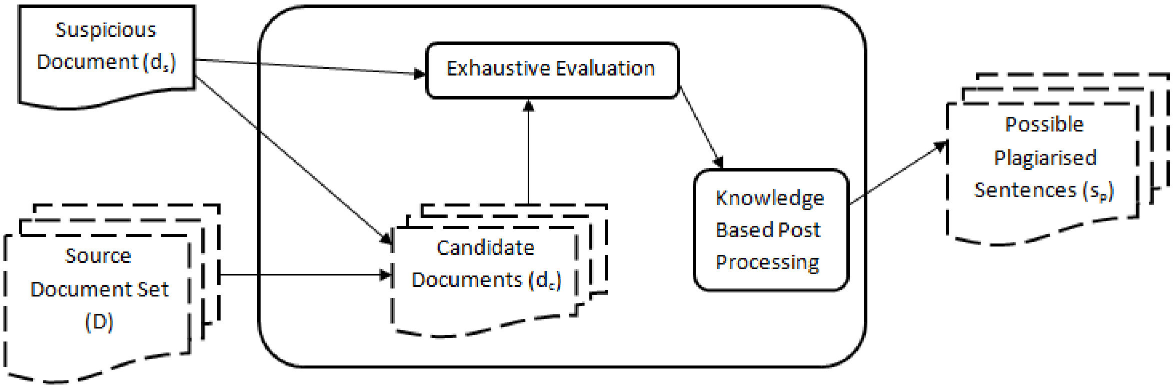

Suspicious Document (d_s)
Exhaustive Evaluation
Source Document Set (D)
Candidate Documents (d_c)
Knowledge Based Post Processing
Possible Plagiarised Sentences (s_p)

Figure 2.4: White Box representation of Extrinsic plagiarism detection

As the focus of research on Natural Language Processing is more since two decades, extrinsic plagiarism detection process further divided into monolingual and cross lingual plagiarism detection.

- **Monolingual plagiarism detection:**

  Monolingual plagiarism detection deals with the automated identification of plagiarism in homogeneous languages.

  Ex: English - English plagiarism detection system.

  Most of the plagiarism detection systems are developed to find plagiarism in Monolingual only.

- **Cross language plagiarism detection:**

  Cross language plagiarism detection deals with identification of similarity/plagiarism between the documents, which belongs to different languages.

  Ex: English - Telugu Plagiarism Detection.

  Research on Cross language plagiarism detection stressed more in recent few year [16, 17, 18, 19], where research on Monolingual plagiarism detection is carrying on since more than 20 years.

It focuses on text similarity computation across languages. The inputs to the system are a query or suspicious document d_s in a language L_s, and a corpus collection D, this corpus collection could be the World Wide Web, which is expressed in multiple languages L_1, L_2,...L_n. there are three steps to implement this procedure.

- Step 1: A list of most promising documents d_c, where $d_c \in D$ is retrieved based on some CLIR model. Or d_c can be retrieved based on some IR model, if d_s is translated by using a machine translation approach.
- Step 2: A pair wise feature based detailed analysis is performed to find all suspicious parts s_p from d_s in language L_s that are similar to parts s_c from d_c, $d_c \in D$ in language L_c, where $L_s \neq L_c$.
- Step 3: Post Processing Operations are used to obtain the results in human readable format. Cross Language plagiarism detection task, therefore contrasts extrinsic plagiarism detection task by bringing candidate set of documents Dx of different languages to be compared with the suspicious document d_s. Fig. 2.5 Shows White box representation of Cross language plagiarism detection.

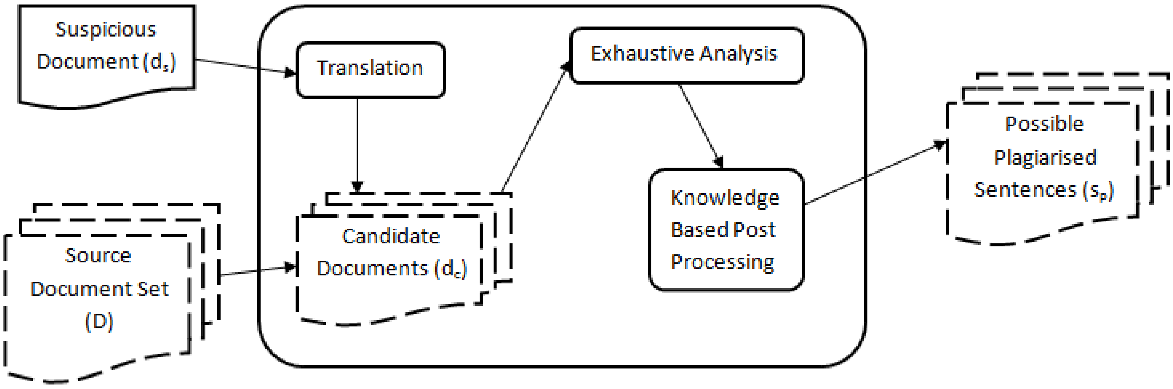

Suspicious Document (d_s)
Source Document Set (D)
Translation
Candidate Documents (d_c)
Exhaustive Analysis
Knowledge Based Post Processing
Possible Plagiarised Sentences (s_p)

Figure 2.5: White Box representation of Cross Language Plagiarism Detection

2.2 Textual features used in Plagiarism detection system

There are various types of textual features available in order to quantify and characterize a document, for applying an exhaustive method to find similar contents.

2.2.1 Textual features for Extrinsic plagiarism detection

Textual features which represents a documents in Extrinsic Plagiarism Detection includes

- Lexical features: includes character-n-grams and word-n-grams.
- Syntactic features: like chunks, sentences, phrases and POS.
- Semantic features: like synonyms, antonyms and hypernyms.
- Structural features: it takes contextual information into account.

1. **Lexical features:**

   Lexical features basically work on base level like character level and word level. Where character n-gram is the simplest form where by a document s is represented as a sequence of characters $s = (c_1, s), (c_2, s)...(c_n, s)$, where (c_i, s) refers to the i^{th} character in s and n = s is the length of s in characters. On the other hand, word-based n-gram (WNG) represents s as a collection of words $s = (w_1, s), (w_2, s), ...(w_n, s)$, where (w_i, s) refers to the i^{th} word in s, and n = s is the length of s (in words) with ignoring sentence and structural bounds.

   Simple WNG may be constructed by using bi-grams (word-2-grams), tri-grams (word-3-grams) or larger. These Character n-grams and Word n-grams are usually known as fingerprints as they can distinguish two documents. And we can simulate these fingerprints with human fingerprints as human fingerprints distinguish two humans based on the concept "no two human have same pattern of fingerprints."The process of generating fingerprints (n-grams) is called as fingerprinting of a document.

2. **Syntactic features:**

   Syntactical features are extracted for document by generating parts of speech (POS) tags for phrases and words in that document. Basic POS tags include verbs, nouns, pronouns, adjectives, adverbs, prepositions, conjunctions, and interjections [8]. POS tagging is the task of marking words in a text or more precisely in a statement as corresponding to a particular POS tag based on both its definition as well as its context.

Sentence-based representation can be achieved by splitting the textual contents into statements/sentences with the use of end-of-sentences delimiters, such as full stops, exclamation, and question marks. After splitting the text into sentences, POS and phrase structures can be constructed by using POS taggers.

3. **Semantic features:**
   Semantic features are useful in quantifying the document by taking into account the use of word classes, synonyms, antonyms, hypernyms, and hyponyms in that document [8]. The usage of dictionaries, POS Tagger, Tokenizer, Sentence Splitter and Parsers would significantly help in extracting semantic features from a document. Usage of Semantic features is more helpful in detection of intelligent plagiarism. One of the good examples is Word-Net Dictionary.

4. **Structural features:** Most plagiarism detection algorithms use document features, such as lexical, syntactic, and semantic features as they are simple and having fewer computational costs when compared with structural features. Well a very few algorithms and methods have been developed to handle structural features. Structural features indicates how data in a document is organized in a document which gives more information regarding semantic meaning of document.

   Documents can be described as a collection of paragraphs or passages, which can be considered as blocks. This type of features can be used in structured documents, such as HTML Web-pages and XML files, and semi structured documents, such as books, thesis, and academic journal papers [8]. Note that structural features are most likely to be stored as XML trees for easier processing.

2.2.2 Textual features for Intrinsic plagiarism detection

Stylometric features are based on the fact that each author develops an individual and unique writing style. For example, authors employ, consciously or subconsciously, patterns to construct sentences, and use an individual vocabulary. These Stylometric features includes different features of extrinsic plagiarism detection system, which includes [8]

1. Lexical features, which genarally operate at character level or word level, what type of set of words the author using to represent a particular situation each time when in comes up.

2. Syntactic features, which work at the sentence level, to consider the usage of word classes and parts of speech in order to quantify a document.

3. Semantic features, which quantify the use of synonyms, functional words, and/or semantic dependencies between the words in a document. It means what kind of words the author is using when a particular situation comes up.

4. Structural features, which reflect textual data organization in the document, quantifies how a author organizing his document when he writes about a particular topic.

2.2.3 Textual features for Cross language plagiarism detection

It includes features like Syntactic , Semantic and statistical where statistical features are language specific [8, 18]. Semantic features includes synonyms, hypernyms, tokenizers, translation word dictionaries. For cross-lingual text relatedness and plagiarism detection, syntactic features are usually combined with semantic or statistical features.

2.3 Exhaustive analysis methods used in Plagiarism detection system:

Methods to compare, manipulate, and evaluate textual features in order to find plagiarism can be categorized into eight types [8].

2.3.1 Character based methods:

Most of the plagiarism detection methods uses character-based lexical features, word-based lexical features, and syntax features, such as sentences, to compare the suspicious document

d_s with each candidate document $d_c \rightarrow D_c$ [8]. The words under comparision in this scenario can be exactly same or approximate. Exact string matching between two strings x and y means that they have exactly same characters in the same order.
Ex: The character 9-gram string x = “aabbbcccc”is exactly the same as “aabbbcccc”but differs from y = “aaabbbbcd”.

2.3.2 Vector based method:

We can represent a sentence in a document as a vector of words. For example, a sentence having 5 distinct words can be represented as a vector with 5 terms. So lexical features along with syntax features of a document can be used as vectors rather than strings. Similarity between the sentences can be calculated by using vector similarity metrics like Jaccard coefficient, Dice's coefficient, Overlap coefficient, Cosine coefficient, and Euclidean distance.

Most of the research works on plagiarism detection have mainly focused on usage of Cosine similarity and Jaccard coefficient. Barron [20] estimated the similarity between the sentences using Jaccard coefficient and Zang and Chow used exponential Cosine similarity as a document dissimilarity. Daniel and Mike [21] used the matching coefficient with a threshold to the scores of similar sentences.

2.3.3 Syntax based methods:

A very few researches are carried out using syntactical features extracted from the documents under comparison. These methods generally uses syntactical features along with other textual features like semantic features.

Elhadi and Al-Tobi [22, 23] used Parts Of Speech (POS) tags followed by other string similarity metrics. This method assumes that similar documents uses same set of POS tags. Cebrian used Lempel-Ziv algorithm to compress the syntax and morphology of two texts based on a normalized distance measure.

2.3.4 Semantic based method:

A sentence can be considered as a group of words arranged in a particular order. Two sentences can be semantically same but differ in their structure, e.g., using the active versus passive voice, or differ in their word choice. There is a very few research in this area due to the fact that involved in the representation of semantics of sentences under consideration and complexity of representative algorithms.

Li and Bao [24] used semantic features for similarity analysis and similarity analysis.

2.3.5 Fuzzy based methods:

The fuzzy based methods evaluates the similarity between textual contents under comparison ranges the value of similarity on a scale of 0 to 1, where 0 represents completely dissimilar and 1 represents exactly same. The concept “fuzzy”in plagiarism detection can be modeled by considering that each word in a document is associated with a fuzzy set that contains words with same meaning, and there is a degree of similarity between words in a document and the fuzzy set.

In a term-to-term correlation matrix [25] is constructed using the individual words and their corresponding correlation values in documents under comparison. Using this matrix a similarity value is calculated.

2.3.6 Structural based methods:

The methods we have been discussed yet uses flat features of documents. Flat features use mainly includes the usage of representation of lexical, syntactic and semantic features of the text in the document [8]. But these flat features do not consider the contextual similarity based on the organization of words, sentences, paragraphs. It means flat features do not consider the structure of document, where structural based methods takes these structural features into consideration.

Tree-structured features representation is a rich data characterization, and multi-layer self organizing maps model (ML-SOM) is very effective in handling such contextual information Chow et al. used block-specific tree-structured representation and utilized ML-SOM for plagiarism detection where the top layer executes document clustering and candidate retrieval, and the bottom layer detects similar, potentially plagiarized, paragraphs using Cosine similarity coefficient.

2.3.7 Stylometric based method:

In these methods we formulate some formulas based on stylometric features in-order to find style plagiarism[12, 26, 27].

Intrinsic plagiarism detection focused more on this type of style based plagiarism, and the formulas which are constructed to quantify the stylometric plagiarism are mainly divided into two categories [8]:

1. Writer specific: It aims to quantify the author's vocabulary standards and style patterns.

2. Reader specific: It aims to quantify the minimum level of standard, a reader must have in order to understand the text. Recent research includes outliers analysis, meta learning, and symbolic knowledge processing.

2.3.8 Cross language based methods:

Cross language methods are mainly based on the measurement of the similarity between textual contents of suspicious document and source documents, where the documents under comparison belong to different natural languages [8, 18]. Methods include

1. Cross language syntax based methods

2. Language dictionary based methods

3. Cross language semantic based methods

4. Statical based methods

Chapter 3

Survey On Plagiarism Detection Methods

3.1 Character based methods

3.1.1 ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Syntactic Features, and Lexical Features
- Exhaustive method used: Character Based Methods
- Description:
  - Stage 1: Matrix of string kernel values is computed which gives similarity between each source and each suspicious documents. Then for each source the possible suspicious documents are ranked based on their similarity level in decreasing order [28].
  - Stage 2: Each similar pair is further investigated, in order to get the precise positions and lengths of the subtexts that have been copied and may be obfuscated by the random plagiarist.

- Formula used:

  Kernal Function(x,y)

$$\text{Liner} = \textstyle\sum_{ng \in A_n} \phi_{ng}(x).\phi_{ng}(y)$$
$$\text{RBF Kernal} = exp(-\tfrac{\sum_{ng \in A_n} \|\phi_{ng}(x)-\phi_{ng}(y)\|^2}{2\sigma^2})$$

  n-gram extraction set $A_n = \sum^n$ where $\sum$ is alphabet set.

  $\phi(x)$ is n-gram embedding function for a byte sequence x.

  Dimension of vector $\phi(x)$ is $|A_n|$.

3.1.2 A Plagiarism Detection Procedure in Three Steps: Selection, Matches and Squares

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical Features
- Exhaustive method used: Character Based Methods
- Description:
  - Stage 1: A First Selection

    For each suspicious document, first identify a small subset of source documents for a second and deeper analysis. This can be done by calculating distance between each suspicious document and every source document. After finding all the values we choose the 10 closest source documents to a suspicious documents for further analysis [29].
  - Stage 2: T9 & Matches

    After finding the top 10 closest neighbor source documents for a suspicious document, look for common subsequence (matches) longer than a fixed threshold. For converting a normal text document into a coded document they used T9 encoding.
  - Stage 3: Looking for Squares

Here they plotted a bi-dimensional plot with a suspicious document text on the x and source document text on y axis. For Each match of length l, starting at (x,y) to (x+l,y+l) draw a line on the plot. Now try to find the abnormal distribution on lines on the plot, by this we can find the matching portions.

- Formula used:

  Distance will be calculated using n-gram distance method.

$$d_n(x,y) = \frac{1}{|D_n(x)|+|D_n(y)|} \sum_{w \in D_n(x) \cup D_n(y)} \left(\frac{f_y(w)-f_x(w)}{f_y(w)+f_x(w)}\right)$$

3.1.3 Finding Plagiarism by Evaluating Document Similarities

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical Features
- Exhaustive method used: Character Based Methods
- Description:
  - Stage 1: Tokenization

    In this method words are used as the base units for finding similarity. So split the documents into set of words [30].
  - Stage 2: Creating Index

    After Tokenization process all tokens are grouped together to make chunks. These chunks are then hashed by a hash function. The mapping of document ID to the sequence of hash values is constructed. After constructing index, an inverted index is constructed by mapping the hash values to the sequence of document IDs.
  - Stage 3: Computing similarities

    Finally similarity values will be calculated using the inverted index of different documents.

3.1.4 Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical Features
- Exhaustive method used: Character Based (word n-gram) Methods
- Description:
  - Step 1:

    Represent the lexical entities by integers, and initiate the table by comparing the integers. If they are the same, the node is red [31].
  - Step 2:

    Connect the red nodes which can made up a diagonal line, mark the rectangles for which the diagonal lines belongs to, and compute the number of nodes as similar degree.
  - Step 3:

    Divide the table into grids and make the rows and columns concluding rectangles as pointer that need not compare.
- Smith-Waterman:

$$S_{i,j} = S_{i-1,j-1} + 1 \text{ if } x(i) = y(j)$$

$$S_{i,j} = max(0, S_{i-1,j} - 1, S_{i,j-1} - 1, S_{i-1,j-1} - 1) \text{ if } x(i) \neq y(i)$$

- Levenshtein Distance algorithm:

Algorithm: Levenshtein Distance.

Input: character arrays representing the source and suspicious text lexical entities respectively.

Output: distance matrix.

```
Begin
ld(char s[1...m], char t[1...n])
// d is a table with m+1 rows and n+1 columns.
declare int d[0...m, 0...n]
for i from 0 to m do
   d[i,0] = i
end for
for j from 0 to n do
   d[0,j] = j
end for
for i from 1 to m do
   for j from 1 to n do
      if s[i] = t[j] then
         cost = 0
      else
         cost = 1
      end if
      d[i,j] = mind[i-1,j]+1, d[i,j-1]+1, d[i-1,j-1]+cost
      return d[m,n]
   end for
end for
```

3.2 Syntax Based Methods

3.2.1 Duplicate detection in documents and web pages using improved longest common subsequence and documents syntactical structures

- Frame work used : Monolingual extrinsic white box frame work
- Textual features used: Lexical features and Syntactical features
- Exhaustive method used: Syntax based method
- Description:
  - Step 1: Syntactical processing phase

    A given text document will be processed to produce a representation for further processing. Tree Tagger is used to tag the text to produce a representation strings. Tags are mapped into single characters to save space and reduce computation costs [23].
  - Step 2: String optimization phase

    Used to adjust the accuracy of the system by changing the size and type of the tags-set is used to annotate a given text.
  - Step 3: Matching and Ranking phase

    Do further process and analysis of strings using the refined and improved LCS algorithm.

3.2.2 Use of text syntactical structures in detection of document duplicates

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Syntactical features
- Exhaustive method used: Syntax based method

- Description:
  - Syntactical processing phase:

    In this method we reduce the text into a smaller set of syntactical (POS) tags making use of most of the document's content. Here the selection of tagger and tag-set have an impact on the accuracy [22].
  - String optimization phase:

    The size of the tag-set is an important accuracy factor. The bigger the tag-set the more detailed the produced string and thus the more accurate. Here in this method tree tagger is used.
  - Matching and Ranking phase:

    The result of tagging and tag optimization is a set of strings representing the ordered tags corresponding to each word in the original document. The procedure string can be considered for further processing.

3.3 Semantic based methods

3.3.1 Sentence similarity based on semantic nets and corpus statistics

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Syntactical features and Semantic features
- Exhaustive method used: Semantic based method
- Description:
  - This method computes the textual similarity between the source and suspicious texts from semantic and syntactic information contained in those texts [24].
  - A text is considered to be a sequence of words each of which carries use full information. Here in this method first we form a joint word set using all the distinct words in the pair of sentences. For each sentence a raw semantic vector is derived with the assistance of a lexical data base.

  - A word order vector is formed for each sentence, again using information from the lexical database.
  - Since each word contributes differently to the meaning of the whole sentence, the significance of the word is weighted by using information content derived from a corpus.
  - By combining row semantic vector with information content from corpus, a semantic vector is obtained for each of the two sentences. Semantic similarity is computed based on the two semantic vectors.
  - Finally, the sentence similarity is derived by combining semantic similarity and order similarity.

3.4 Fuzzy Based Methods

3.4.1 A sentence-based copy detection approach for web documents

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical features and Semantic features
- Exhaustive method used: Fuzzy based method
- Description:
  - Step 1:

    Let us assume there are two sentences. First we need to find the correlation factor between word *i* from sentence 1 and word *j* from sentence 2 [25].

    This correlation factor, $C_{i,j}$, defines the extent of similarity between any two words *i* and *j* in the term-term correlation matrix.

    $$C_{i,j} = n_{i,j}/(n_i + n_j - n_{i,j})$$

    where $n_{i,j}$ = number of documents in a collection(Data Set) with both words *i* and *j*,

    n_i = number of documents in a collection with word *i*,

    $C_{i,j}$ = correlation factor between word *i* and *j*.

- Step 2:

  The degree of similarity of the two sentences is the extent to which the sentence match. To obtain the degree of similarity between two sentences S_i & S_j, first need to compute the word-sentence correlation factor $\mu_{i,j}$ of word i in S_i & all words in S_j.

  $$\mu_{i,j} = 1 - \prod_{k \in S_j}(1 - C_{i,k})$$

  where k is a word in sentence S_j,

  $C_{i,k}$ is word-word correlation factor between word i and k.

- Step 3:

  After finding word-sentence correlation factor for all words in S_i. we will calculate similarity between S_i & S_j.

  $$\text{sim}(S_i,S_j) = \frac{\mu_{w_1,j}+\mu_{w_2,j}+\ldots+\mu_{w_n,j}}{n}$$

  where w_k is a word in S_i

- Step 4:

  calculate the equality of S_i and S_j

  EQ(S_i,S_j) = 1 if min(sim(S_i,S_j),sim(S_j,S_i))$\geq$ 0.825 $\wedge$ —sim(S_i,S_j)-sim(S_j,S_i)— $\leq$ 0.15

  EQ(S_i,S_j) = 0 Otherwise

3.5 Structural Based Methods

3.5.1 Conceptual similarity and Graph based method for Plagiarism detection

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Structural features
- Exhaustive method used: Structural based method

- Description: Ahmed Hamza Osman et al. used graph based representation of text documents to get the underlying semantic meaning. And finally compare two documents using the graphs of respective documents [32].

- Algorithms:
  Similarity between(S1,S2) = Overlap(S1,S2) = $\frac{(|CS1| \cap |CS2|)}{|CS1| \cup |CS2|}$
  Where CS1 = Concepts of Sentence1; CS2 = Concepts of Sentence2;

  Similarity between(D1,D2) =$\sum_{t=1,n;f=1,m}(S1_t, S2_f)$
  D1 = document 1; D2 = document 2; S1 = sentence in Doc 1; S2 = Sentence in Doc 2.

3.6 Vector based methods

3.6.1 SCAM: A Copy Detection mechanism for Digital documents

- Description: Narayanan Shivakumar [3], and Hector proposed a new model which is based on frequency relative model, which uses the concepts of vector based methods. Here this relative frequency model overcomes the drawbacks like range of frequency for keyword, of vector based method.

  Relative Frequency Model: Cosine similarity measure works well when word frequencies are of similar magnitudes, but it doesn't when the magnitudes differ significantly. To use the properties of the cosine similarity measure and remove its insensitivity to word frequency magnitudes, they used relative frequencies of words as indicators of similar word usage and combines it with cosine similarity measure.

  In order to use relative frequencies, they first selected set of terms which are able to clear epsilon condition. And named the set as Closeness Set.

- Frame work used : Monolingual Extrinsic White Box Frame Work

- Textual features used: Syntactic features, and Semantic features

- Exhaustive method used: Relative Frequency Model which uses Vector based methods

- Algorithm:

  Closeness set = $\forall i \in W_i$ select word i if

$$\varepsilon - (\frac{F_i(R)}{F_i(S)} + \frac{F_i(S)}{F_i(R)}) > 0$$

where $\varepsilon = (2+, \infty)$ is a user tunable parameter. If either$F_i(R)$ or $F_i(S)$are zero, we say that the closeness condition is not satisfied.

$$\text{subset}(D_1,D_2)=\frac{\sum_{w_i \in c(D_1,D_2)} \alpha_i^2 * F_i(D_1) * F_i(D_2)}{\sum_{i=1}^{N} \alpha_i^2 F_i^2(D_1)}$$

$F_i(D_1)$ = frequency of word i in document 1

$F_i(D_2)$ = frequency of word i in document 2

and finally

$$\text{Sim(R,S) = maxsubset(R,S),subset(S,R)}$$

3.6.2 Detecting short passages of similar text in large document collections

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical features
- Exhaustive method used: Vector based method
- Description:

  Comparison can take place in two modes.

  1. The documents under comparison are having similar lengths, and in this case we will be looking for "resemblance"between texts [33].
  2. A small part of document text can be compared against a large document, which could have been used as a source document. Here in this case the documents under comparison are having different lengths. In this case we will be looking for "containment". If a significant portion of the smaller text is contained within the larger, then it indicates that material has been lifted from the suspected source.

Resemblance: The number of matches between the elements of the sets of trigrams, scaled by a joint set of combination of both sets. Here sets represents both source and suspicious documents of similar lengths.

Ex: Let S(A), S(B) be set of trigrams from documents A and B respectively. Let R(A,B) be the resemblance between A and B

$$\mathrm{R} = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|}$$

Containment: Let us assume that document A is derived from concatenated potential source material from the web, a large set. Document B is a derived from a single student easy, a small set.

Containment is the number of matches between the elements of trigram sets from A and B, scaled by the size of set B.

$$\mathrm{C} = \frac{|S(A) \cap S(B)|}{|S(B)|}$$

3.6.3 On Automatic Plagiarism Detection based on n-grams comparison

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical features and Syntactical features
- Exhaustive Method Used: Vector based method
- Description:

  There are four prior steps in this method.

  1. The suspicious document S is split into sentences(S_i) [20].
  2. S_i is split into word n-grams. The set of n-grams represents the sentence i.
  3. A document d, which belonging to document set D, is not split into a sentence, but simply into word n-grams.
  4. Each sentence $S_i \in S$ is searched singleton over the reference documents.

In order to determine if S_i is a candidate of being plagiarized from $d \in D$. Compare the corresponding sets of n-grams. Due to the different size of these sets here we use containment measure

$$C(S_i/d) = \frac{|N(S_i) \cap N(d))|}{|N(S_i)|}$$

where N(S_i) = the set of n-grams in sentence i.

N(d) = the set of n-grams in document d.

If the maximum value of C for different document d ∈ D, is greater than a given threshold, S_i becomes a suspected part of plagiarism.

3.6.4 Sentence Based Natural Language Plagiarism Detection

- Frame work used : Monolingual Extrinsic White Box Frame Work
- Textual features used: Lexical features and Syntactical features
- Exhaustive method used: Vector based method
- Description:
  - Stage 1: Preprocessing the document

    The preprocessing step of method is to read in documents that are being compared and split them into documents objects, which contains sentence objects, and this sentence objects contains a list of words that were found in the original sentence of document text. After splitting a sentence into a list of words, implement three filters [21].

    1. Convert all words in lower case in order to reduce the comparison costs.
    2. Remove most commonly used words (stop words).
    3. Remove repeated words so that they appear only once in a sentence.
  - Stage 2: Comparing Documents

    In this stage both the documents are compared pairwise by computing every sentence in the one document to every sentence in the other. The comparisons are done by computing a similarity score, based on the word count metric. The score is the average similarity between the sentences, computed as a function of words in common and the lengths of the sentences being compared.

- Pseudo Code

```
Documents[] docs = read_Docs_from_Disk();
for each Document, i, in docs do
   for each Document, j, following i in docs do
      compare_sentence(docs[i],docs[j]);
   end for
end for
compare_sentence(Documents doc1, Document doc2)
for each sentence, i, in doc1 do
   for each sentence, j, in doc2 do
      int common = number of shared words;
      int score = similarity_score(i,j,common);
      if score > SIM_THRESHOLD —— common > common_THRESHOLD
      then
         Store_Link(sent1,sent2,score);
      end if
   end for
end for
```

Chapter 4

Proposed Approach

4.1 Drawbacks in existing methods

4.1.1 Drawbacks in SCAM approach

In SCAM algorithm, a word w_i is included in the set C(Closeness set), if the following condition is true:

$$\varepsilon - (\frac{F_i(R)}{F_i(S)} + \frac{F_i(S)}{F_i(R)}) > 0$$

But there is a ambiguity in selecting an appropriate value for 'ε', a larger value of 'ε'expands the above set including words sometimes not relevant, a lower value of 'ε'reduces the ability to detect minor overlap, since some words can be excluded. Even though researchers at Stanford empirically found a value (2.5) of 'ε'that works well for a set of conventional text documents [3].

Let us consider an example, consisting of two input sentences 'R'and 'S', where

R = "aaaaabbcddd"
S = "aabbbbcccddd"

where a, b, c, d represent distinct terms.
After executing these sentences on SCAM approach using different ε values ranging [2.1,3.5] observed that for different ε values the number of distinct words in closeness set C are varying. And as well as the similarity value between the sentences.

From Table 4.1 we can observe that how the elements in set C are changing as well as the similarity values. Here higher value of ε includes all distinct terms in the sentences into closeness set C which leads to inclusion of irrelevant words like ‘a’and ‘c’into set C, where lower values of ε discards the minor overlapping terms like ‘b’in adding into set C.

Table 4.1: Similarity value between S and R for different ε values

ε value	words in C	similarity value
2.1	d	9/38
2.5	d	9/38
3	a,b,d	27/38
3.5	a,b,c,d	30/38

4.1.2 Drawbacks in Daniel approach

Daniel approach uses the common words shared between the both sentences under comparison. After finding all common words, we will evaluate the similarity between sentences w.r.t sentence 1 using the number of distinct words in sentence 1 and number of common words shared between the sentence 1 and sentence 2 [21].

Let us consider an example.

- **sentence 1:** “The brown dog was feeling very tired”
- **sentence 2:** “While the grey cat was full of energy, the brown dog was feeling very tired”

Distinct words in sentence 1 = brown, dog, was, feeling, very, tired
Distinct words in sentence 2 = while, grey, cat, was, full, energy, brown, dog, feeling, very, tired
Distinct common words between sentences 1 and 2 = brown, dog, was, feeling, very, tired

- **Similarity w.r.t sentence 1** = $\frac{6}{6} \times 100 = 100\%$
- **Similarity w.r.t sentence 2** = $\frac{6}{11} \times 100 = 55\%$
- **Average similarity** = $\frac{(100+55)}{2} = 77.5\%$

But here sentence 1 is completely reused in sentence 2 which falls under Exact Plagiarism. Another problem with daniel approach is, this method will not give any priority to the repeated words in the sentences under comparison. Let us take the example used in section 4.1.1.

- R = “aaaaabbcddd”
- S = “aabbbbcccddd”

Distinct words in sentence R = a, b, c, d
Distinct words in sentence S = a, b, c, d
Distinct common words between sentences 1 and 2 = a, b, c, d

- **Similarity w.r.t Sentence 1** = $\frac{4}{4} \times 100 = 100\%$
- **Similarity w.r.t Sentence 2** = $\frac{4}{4} \times 100 = 100\%$
- **Average Similarity** = $\frac{(100+100)}{2} = 100\%$

It means this method will return both sentences R and S are exactly same, which is not a desired result.

4.2 Proposed Algorithm

In order to overcome the drawbacks faced in Vector Based methods like SCAM and Daniel approach, here we are proposing ERFM algorithm.

Algorithm: Enhanced Relative Frequency Model.

```
Input: Documents D1 and D2
Output: similarity between documents D1 and D2.
Begin
for each sentence j, $S_j \in D1$ do
  for each sentence n, $S_n \in D2$ do
    for each distinct term i$\in S_j$ do
      z = z + 1;
      for each distinct term $m \in S_n$ do
        y = y + 1;
        if $S_{j,i}(D1) = S_{n,m}(D2)$ then
          if $(f(S_{j,i}(D1)) = f(S_{n,m}(D2)))$ then
            count = count + 1;
          else if $(f(S_{j,i}(D1)) < f(S_{n,m}(D2)))$ then
            count = count + $\frac{f(S_{j,i}(d1))}{f(S_{n,m}(d2))}$;
          else if $(f(S_{n,m}(D2) < f(S_{j,i}(D1))))$ then
            count = count + $\frac{f(S_{n,m}(d2))}{f(S_{j,i}(d1))}$;
          end if
        end if
      end for
    end for
    similarity between $S_j$ and $S_n$ w.r.t $S_j$ (sm)= count/z;
    similarity between $S_j$ and $S_n$ w.r.t $S_n$ (sm1)= count/y;
    if $sm1 \geq sm$ then
      sm = sm1;
    end if
    if $sm \geq 0.9$ then
      Sentences $S_j$ and $S_n$ are similar
    end if
  end for
end for
```

4.3 Working Procedure for ERFM

Our proposed Enhanced Relative Frequency Model takes two text documents as input in order to calculate the similarity value between them.

First we will consider the 1^{st} sentences in both documents and compare them using below steps.

1. take first word in both sentences and compare them, if the words are same or similar then goto step 2, else step 4.
2. now consider how many times the word is repeated in sentence 1 of document 1 and sentence 1 of document 2. If the word repeated same number of time in both the sentences then increase the counter by 1, else increase the counter by ration of lower frequency to higher frequency and goto step 3.
3. take next word in sentence 1 of document 1 and goto step 1
4. take next word in sentence 1 of document 2 and goto step 1.

We repeat the above steps until we compare all the words in sentence 1 of document 1 with all the words in sentence 1 of document 2. Then we will find the similarity using counter value and total number of distinct words both the sentences.

- **Similarity w.r.t. s1 of d1** = $\frac{countervalue}{totalno.ofdistinctwordsins1ofd1}$
- **Similarity w.r.t. s1 of d2** = $\frac{countervalue}{totalno.ofdistinctwordsins1ofd2}$

And finally we will take the maximum value out of the both similarity values.

Similarity between the sentence = Max {sm, sm1}

And we will repeat this procedure for all sentence pairs between the documents under comparison.

Let us consider the example of 4.1.1.

- R = “aaaaabbcddd”
- S = “aabbbbcccddd”

Counter value = $(\frac{2}{5}) + (\frac{2}{4}) + (\frac{1}{3}) + (\frac{3}{3}) = \frac{134}{60} = 2.333$

- **Similarity w.r.t. s1 of d1** = $\frac{2.333}{4}$
- **Similarity w.r.t. s1 of d2** = $\frac{2.333}{4}$

And the Similarity Value is 0.5583 which is more promising value than earlier approaches used.

4.4 Frame work design for proposed approach

Here in the design of frame work for Enhanced Relative Frequency Model plagiarism detection method, we used Monolingual Extrinsic White Box Frame Work.

Figure 4.1 represents the frame work. This Frame work consists of four stages.

- Stage 1: In this stage we gather the input to the detection system like query document d_q and document collection D or the pair of documents (d_q, D_x) between which we need to check the similarity.
- Stage 2: In this stage we retrieve candidate set of documents from document collection (we can skip this stage by taking similar document set as input).
  If we are comparing two documents for their similarity then there is no need of this retrieval of candidate set of documents.
- Stage 3: In Stage 3 we perform indexing of both query document d_q and candidate set of document D_x. Before performing Indexing operation, we uses textual features, and syntactic features in order to split the document sentence wise. And we use semantic features like synonyms, stemming, stop-word removal to reduce the size of comparisons.
- Stage 4: This is the last stage of frame work where we match the terms and their respective frequencies of test and registered documents using Enhanced Relative Frequency Model algorithm in-order to compute the similarity between the sentences of both documents.

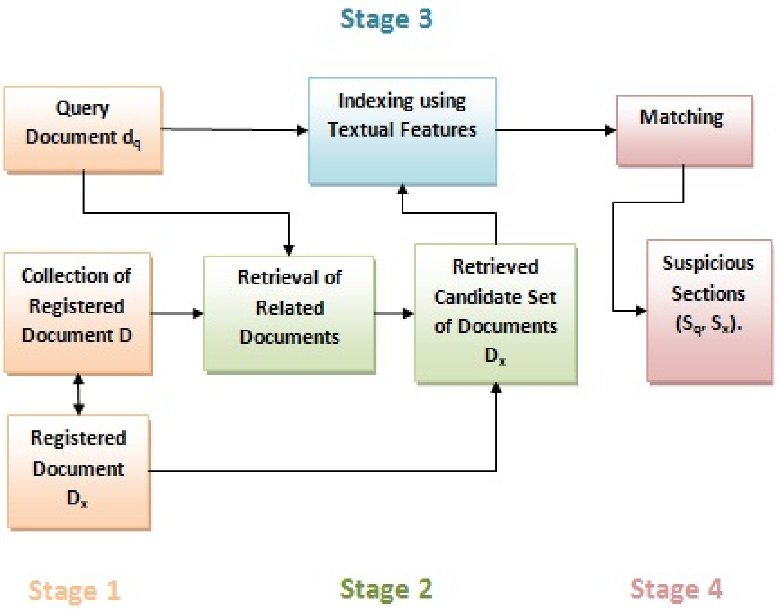

Stage 3
Query Document d_q
Indexing using Textual Features
Matching
Collection of Registered Document D
Retrieval of Related Documents
Retrieved Candidate Set of Documents D_x
Suspicious Sections (S_q, S_x).
Registered Document D_x
Stage 1
Stage 2
Stage 4

Figure 4.1: Monolingual Extrinsic White Box Frame Work for Enhanced Relative Frequency Model

Finally we are with the suspicious sections(sentences) (s_q, s_x). Where (s_q, s_x):$s_q \in d_q, s_x \in d_x, d_x \in D$

For the implementation purpose we have omitted the step of retrieving candidate set documents by considering the assumption of taking related documents as input for the plagiarism detection system.

Chapter 5

Implementation and Results

5.1 Techniques and concepts employed

The following concepts and techniques are employed in the proposed plagiarism detection algorithm.

Stop word removal and Indexing:

- Apache Lucene version 3.6.0 was used for indexing of words and stop word removal.

Sentence splitting:

- We used String Tokenizer for sentence splitting.

Word stemming:

- Porter stemmer was used for stemming process.

5.2 Design and working of GUI

The executed jar file, provided an User interface to upload test and registered documents. Figure 5.1 shows UI for uploading the documents.

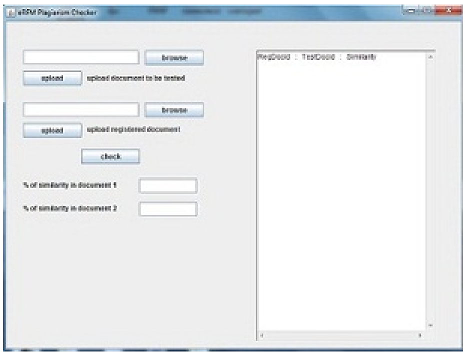
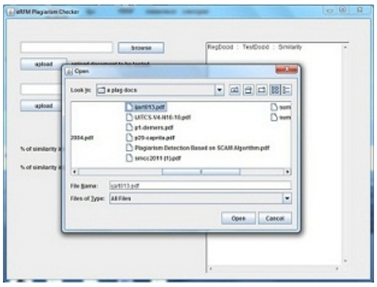

eRFM Plagiarism Checker
browse
upload
upload document to be tested
browse
upload
upload registered document
check
% of similarity in document 1
% of similarity in document 2
RegDocid : TestDocid : Similarity

Figure 5.1: ERFM interface

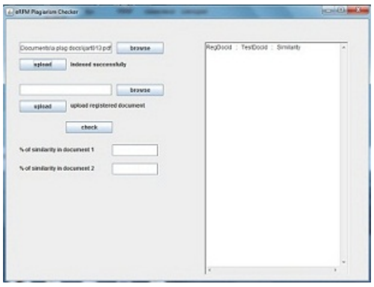
Then we will use a file chooser to select the respective document(Fig. 5.2). After that we need to perform indexing on both the selected documents(Fig. 5.3).



ijart013.pdf
IJITCS-V4-N10-10.pdf
p1-demers.pdf
2004.pdf
p29-caprila.pdf
Plagiarism Detection Based on SCAM Algorithm.pdf
smcc2011 (1).pdf
sum
sum

File Name: ijart013.pdf
Files of Type: All Files

Figure 5.2: File Chooser for selecting documents


eRFM Plagiarism Checker
browse
upload
browse
upload
upload registered document
check
% of similarity in document 1
% of similarity in document 2
RegDocid : TestDocid : Similarity

Documents\a plag docs\ijart013.pdf

indexed successfully

Figure 5.3: Indexed Successfully first document

5.3 Results

This section provides results of some of the existing techniques we have considered and our proposed technique. In the thesis, however, proposed schemes are in detail compared with existing methods.

For implementation purpose, we choose “METER”dataset.

- METER dataset

  The METER corpus was created by the Departments of Journalism and Computer Science at Sheffield University as part of the EPSRC-funded METER project.

In METER dataset, they have taken five individual task and given each task to a group of 19 people to write a file related to the task. So METER dataset contains 5 original documents and 19*5 possible plagiarised documents.

Figure 5.4 shows us the behavior of different vector based methods in finding

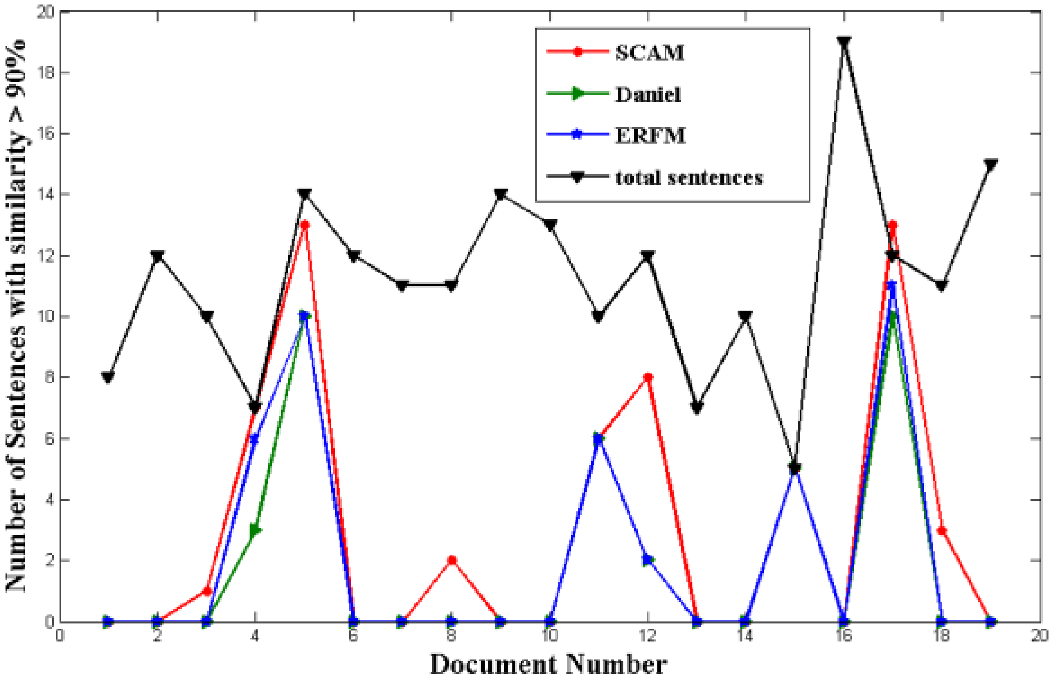


Figure 5.4: Document Number *vs.* No. of sentences with similarity >90% on task A

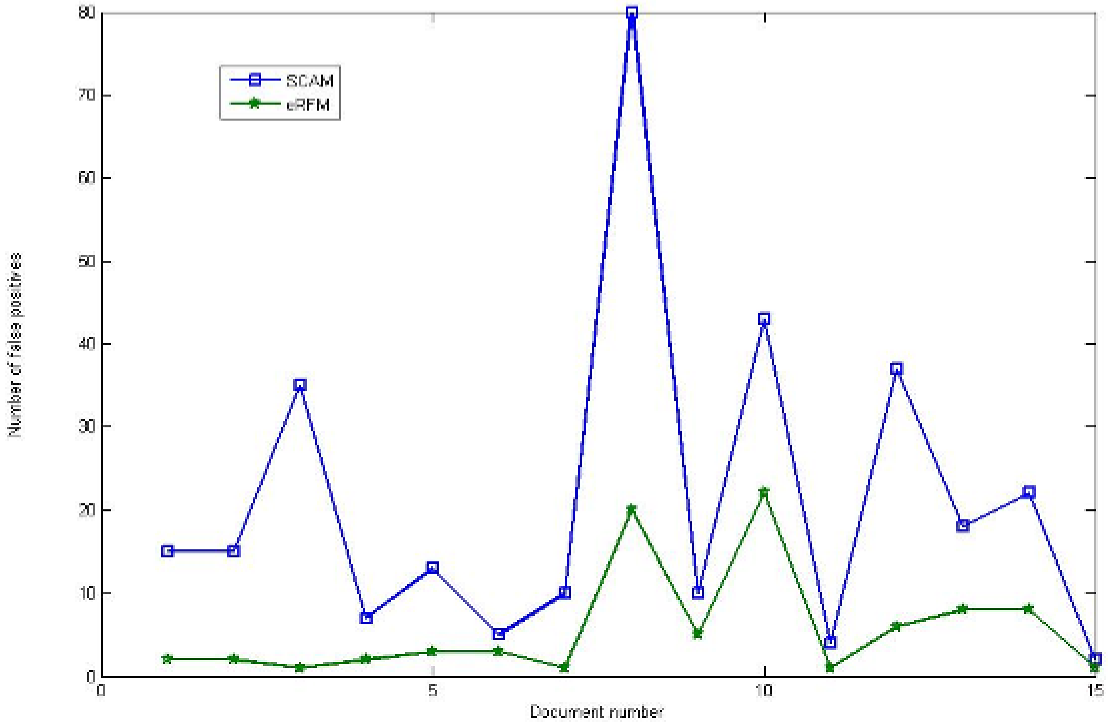


Figure 5.5: Document Number *vs* No. of false positives

similarity/similar sentences in documents written by group A. Here we can clearly observe that ERFM model works effectively than daniel method, and it looks like little bit less than SCAM Model but if we observe Figure 1 closely for document 17 SCAM approach reports 13 sentences are having similarity more than 90 % where total number of sentences in document 17 are 12 only. And we observed that there are no two similar sentences in document 17. It means SCAM approach reported a “false positive”.

And we have been observed that SCAM reported more number of false positives in sentence with similarity between 80-90 % . Then we have taken another input dataset (documents) chosen from: http://dspace.nitrkl.ac.in . In experimental work, 15 random documents of various sizes and number of sentences were chosen for plagiarism detection. For example, if a document ‘doc1’has 151 sentences, it means we run both the algorithms for all 151 * 151 possible cases and the results were obtained for respective cases. Figure 5.5 represents the total number of false positive occurred in our experiment.

Figure 5.7 represents the behavior of methods on documents written by group B and

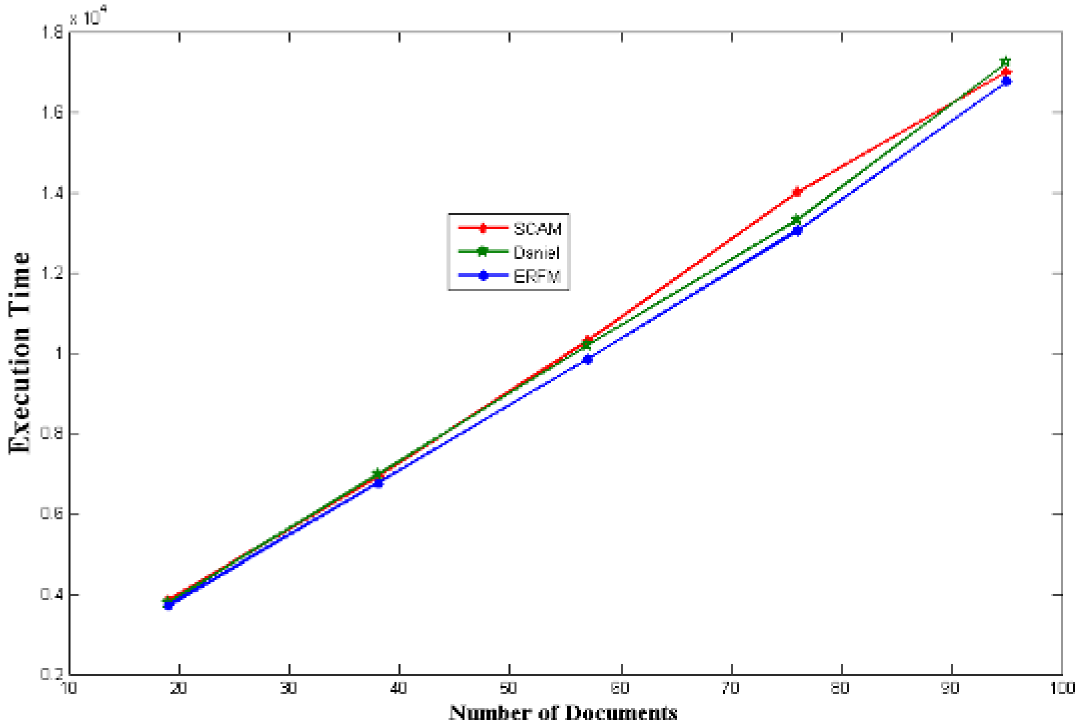


Figure 5.6: No of Documents *vs.* Execution Time taken (milli seconds)

like wise Figure 5.8 represents group C, Figure 5.9 represents group D, and Figure 5.10 represents group E.

We executed the all three vector based approaches by taking various number of documents as input dataset and calculated the execution time taken by three methods respectively. Figure 5.6 shows the execution time in milli seconds taken by three methods and from that figure we can observe ERFM model giving better execution time than SCAM and Daniel approach.

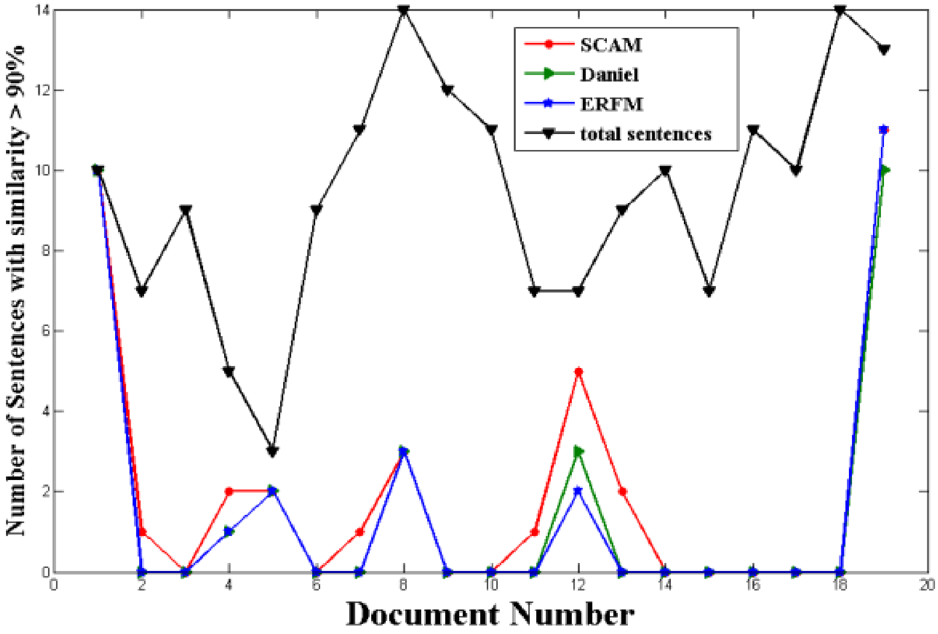


Figure 5.7: Document Number *vs.* No. of sentences with similarity >90% on task B

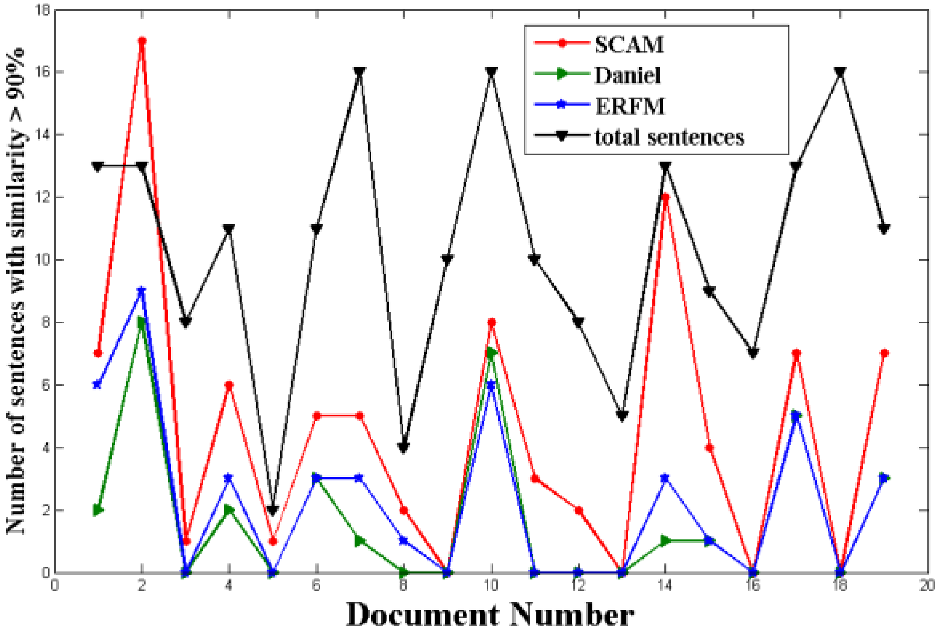


Figure 5.8: Document Number *vs.* No. of sentences with similarity >90% on task C

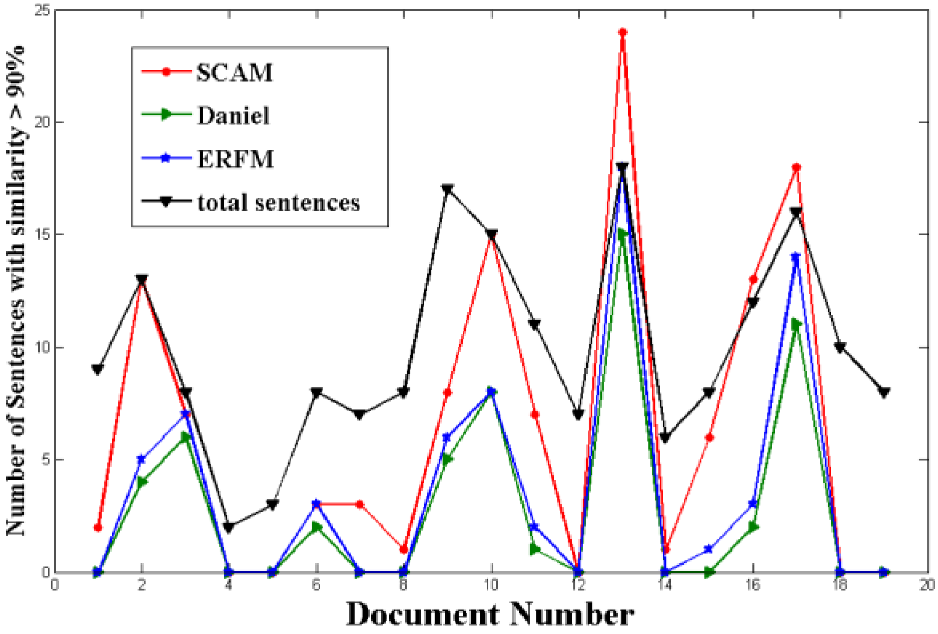


Figure 5.9: Document Number *vs.* No. of sentences with similarity >90% on task D

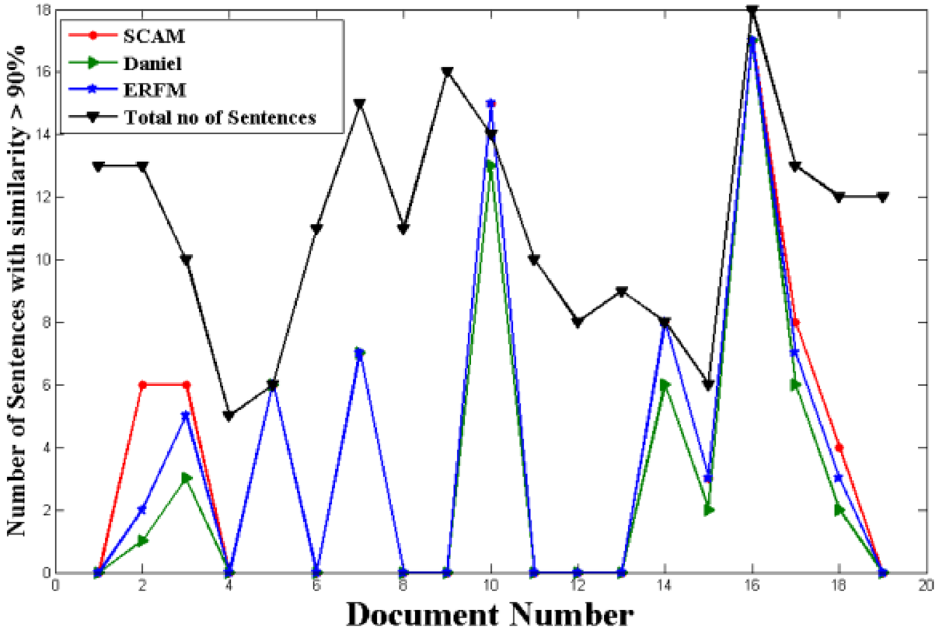


Figure 5.10: Document Number *vs.* No. of sentences with similarity >90% on task E

Chapter 6

Conclusion and Future work

In our research work we have attempted to improve the efficiency of finding plagiarism between source and suspicious documents by applying ERFM approach. And we achieved below objectives.

- The usage of word frequencies instead of words alone.
- Enhancing the relative frequency model by changing the approach for calculation of similarity values.
- The implementation of stop word removal process in-order to decrease the computational costs and to improve efficiency.

And we are planing to extend our work to cross language plagiarism detection by implementing natural language translation techniques.

Bibliography

[1] S Jamal. Plagiarism detection techniques. Master's thesis, COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY, 2010.

[2] M Z Eissen, B Stein, and M Kulig. Plagiarism detection without reference collections. In *Proceeding of 30th Annual Conference of the German Classification Society*, pages 359–366, Berlin, 2007.

[3] N Shivakumar and H G Molina. Scam: A copy detection mechanism for digital documents. In *Proceedings of 2nd International Conference in Theory and Practice of Digital Libraries*, pages 1–13, Texas, 1995.

[4] M Roig. *Avoiding Plagiarism, Self-Plagiarism, and Other Questionable Writing Practices: A Guide to EthicalWriting*, volume 2. St. Johns Univ. Press, 2006.

[5] J Bloch. Academic writing and plagiarism: A linguistic analysis. *In English for Specific Purposes*, 28:282–285, 2009.

[6] I Anderson. Avoiding plagiarism in academic writing. *In Nursing Standard*, 23(18):35–37, 2009.

[7] K R Rao. Plagiarism, a scourge. *In Current Sciences*, 94:581–586, 2008.

[8] S M Alzahrani, N Salim, and A Abraham. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, PartC*, 42(2):133–149, Feb 2012.

[9] Precision and recall. Website. http://en.wikipedia.org/wiki/Precision_and_recall.

[10] G Salton and M J McGill. *Introduction to modern information retrieval*, volume 2. McGraw-Hill, 1983.

[11] E Stamatatos. Intrinsic plagiarism detection using character n-gram profiles. In *Proceedings of 3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 38–46, Donostia, Spain, 2009.

[12] S Meyer, Z Eissen, and B Stein. Intrinsic plagiarism detection. In *Proceedings of the European Conference on Information Retrieval*, pages 565–569, London, UK, 2006.

[13] B Stein, M Koppel, and E Stamatatos. Plagiarism analysis, authorship identification, and near-duplicate detection. *In Special Interest Group of Information Retrieval*, 41(2):68–71, 2007.

[14] B Stein, L Nedim, and P Peter. Intrinsic plagiarism analysis. *In Journal of Language Resources and Evaluation*, 45(1):63–82, March 2011.

[15] M Potthast, B Stein, A Eiselt, W Bauhaus, A C Barron, and P Rosso. Overview of the 1st international competition on plagiarism detection. In *Proceedings of SEPLN Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse*, pages 1–9, Donostia, Spain, 2009.

[16] Z Ceska, M Toman, and J Karel. Multilingual plagiarism detection. In *Proceedings of the 13th international conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 83–92, Varna, Bulgaria, 2008.

[17] A Barron, P Rosso, D Pinto, and A Juan. On cross-lingual plagiarism analysis using a statistical model. In *Proceedings of the ECAI'08 PAN Workshop: Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 9–13, Patras, Greece, 2008.

[18] P Martin, B Alberto, B Stein, and P Rosso. Cross-language plagiarism detection. *In Journal of Language Resources and Evaluation*, 45(1):45–62, March 2011.

[19] P David, C Jorge, B C Alberto, J Alfons, and P Rosso. A statistical approach to crosslingual natural language tasks. *In Journal of Algorithms*, 64(1):51–60, January 2009.

[20] A Barron and P Rosso. On automatic plagiarism detection based on n-grams comparison. In *Proceedings of the 31st European Conference on Information Retrieval*, pages 696–700, Toulouse, France, 2009.

[21] R W Daniel and S J Mike. Sentence-based natural language plagiarism detection. *In ACM Journal of Educational Resources*, 4(4):1–20, December 2004.

[22] M Elhadi and A Al-Tobi. Use of text syntactical structures in detection of document duplicates. In *Proceedings of the 3rd International Conference on Digital Information Management*, pages 520–525, London, U.K., 2008.

[23] M Elhadi and A Al-Tobi. Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures. In *Proceedings of the Fourth International Conference on Computer Sciences and Convergence Information Technology*, pages 679–684, Seoul, Korea, 2009.

[24] Y Li, D McLean, Z A Bandar, J D OShea, and K Crockett. Sentence similarity based on semantic nets and corpus statistics. *In IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, Augest 2006.

[25] R Yerra and Y K Ng. A sentence-based copy detection approach for web documents. In *Proceedings of the Second international conference on Fuzzy Systems and Knowledge Discovery*, pages 557–570, Changsha, China, 2005.

[26] G Stefan and N Stuart. Tool support for plagiarism detection in text documents. In *Proceedings of the ACM symposium on Applied computing*, pages 776–781, Santa Fe, New Mexico, 2005.

[27] A B Ryul, K Heon, and K M Hyun. An application of detecting plagiarism using dynamic incremental comparison method. In *Proceedings of International Conference on Computational Intelligence and Security*, pages 864–867, Guangzhou, China, 2006.

[28] C Grozea, C Gehl, and M Popescu. Encoplot: Pairwise sequence matching in linear time applied to plagiarism detection. In *Proceedings of Conference of the Spanish Society for Natural Language Processing*, pages 10–18, Donostia, Spain, 2012.

[29] C Basile, D Benedetto, E Caglioti, G Cristadoro, and M D Esposti. A plagiarism detection procedure in three steps: Selection, matches and squares. In *Proceedings of Conference of the Spanish Society for Natural Language Processing*, pages 19–23, Donostia, Spain, 2012.

[30] J Kasprzak, M Brandejs, and M Kripac. Finding plagiarism by evaluating document similarities. In *Proceedings of Conference of the Spanish Society for Natural Language Processing*, pages 24–28, Donostia, Spain, 2012.

[31] Z Su, B R Ahn, K Y Eom, M K Kang, J P Kim, and M K Kim. Plagiarism detection using the levenshtein distance and smith-waterman algorithm. In *Proceedings of 3rd International Conference on Innovative Computing Information and Control*, pages 569–572, Liaoning, China, 2008.

[32] A H Osman, N Salim, M S Binwahlan, H Hentably, and A M Ali. Conceptual similarity and graph-based method for plagiarism detection. *In Journal of Theoritical and Applied Information Technology*, 32(2):135–143, October 2011.

[33] C Lyon, J A Malcolm, and R G Dickerson. Detecting short passages of similar text in large document collections. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 118–125, Pittsburgh, USA, 2001.