AN INTEGRATED APPROACH FOR PLAGIARISM DETECTION IN MALAYALAM DOCUMENTS

Thesis submitted by

Sindhu.L

In partial fulfilment of the requirements
for the award of the degree of

DOCTOR OF PHILOSOPHY

Under the Faculty of Technology

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI-22, INDIA

December 2017

An Integrated Approach for Plagiarism Detection in Malayalam Documents

PhD thesis in the field of Natural Language Processing

Author:

Sindhu. L
Department of Computer Science and Engineering
College of Engineering
Poonjar,Kerala,India
Email :sindhul.cep@gmail.com

Supervisor:

Dr. Sumam Mary Idicula
Professor
Department of Computer Science
Cochin University of Science and Technology
Kochi-682022, Kerala, India Email:sumam@cusat.ac.in

December 2017

Department of Computer Science
Cochin University of Science and Technology
Kochi-22, India

Certificate

This is to certify that the work presented in this thesis entitled **“*An Integrated approach for Plagiarism Detection in Malayalam documents*”** submitted to Cochin University of Science and Technology, in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Computer Science is a bonafide record of research work done by Ms. Sindhu L in the Department of Computer Science, Cochin University of Science and Technology, under my supervision and the work has not been included in any other thesis submitted previously for the award of any degree.

Kochi
December 2017

Dr. Sumam Mary Idicula
(Supervising Guide)
Professor in Department of Computer Science
Cochin University of Science and Technology
Kochi-682022

Department of Computer Science
Cochin University of Science and Technology
Kochi-22, India

Certificate

This is to certify that the work presented in this thesis entitled **"*An Integrated approach for Plagiarism Detection in Malayalam documents*"** submitted to Cochin University of Science and Technology, in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Computer Science is a bonafide record of research work done by Ms. Sindhu L in the Department of Computer Science, Cochin University of Science and Technology, under my supervision and the work has not been included in any other thesis submitted previously for the award of any degree.

Kochi
December 2017

Dr. Sumam Mary Idicula
(Supervising Guide)
Professor in Department of Computer Science
Cochin University of Science and Technology
Kochi-682022

Department of Computer Science
Cochin University of Science and Technology
Kochi-22, India

Certificate

This is to certify that all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee of the candidate have been incorporated in the thesis entitled **"An Integrated approach for Plagiarism Detection in Malayalam documents"**

Kochi
December 2017

Dr. Sumam Mary Idicula
(Supervising Guide)
Professor in Department of Computer Science
Cochin University of Science and Technology
Kochi-682022

Declaration

I hereby declare that the work presented in this thesis entitled **"An Integrated approach for Plagiarism Detection in Malayalam documents"** is based on the original research work done by me under the supervision and guidance of Dr. Sumam Mary Idicula Professor, Department of Computer Science, Cochin University of Science and Technology, Kochi-682022 and has not been included in any other thesis submitted previously for the award of any degree.

Kochi-22 **Sindhu.L**
December 2017

Acknowledgement

This thesis would not have been possible without the help and support of many people:

First of all I thank God for strengthening me physically and mentally at the times of difficulties I faced during this PhD work.

I express my sincere gratitude and indebtedness to my guide ***Dr. Sumam Mary Idicula,*** *Professor, Department of Computer Science, Cochin University of Science and Technology for her excellent guidance, competent advice, keen observations and persistent encouragement as well as personal attention given to me during the entire course of work, without which the successful completion of this work would not have been possible.*

I extend my sincere gratitude to Dr. G. Santhosh Kumar, Head, Department of Computer Science for allowing me to use the facilities of the Department and for the support and guidance during the entire course of my work.

I express my sincere gratitude to ***Dr. David Peter S.,*** *Registrar, Cochin University of Science and Technology for his help and support.*

I specially thank the technical staff, Mr. Renjith, Mr. Shibu, & Mrs. Manju for providing me all the technical support required for carrying out my research work. I am grateful to all the staff of the department for their encouragement and support.

I am highly obliged to all my friends and colleagues for their encouragement and support.

I record my sincere and utmost gratitude to my family for their patience and tolerance during the entire period of my work

Sindhu.L

Abstract

Ever since we entered the digital communication era, the ease of information sharing through the internet has encouraged online literature searching. With this comes the potential risk of a rise in academic misconduct and intellectual property theft. Plagiarism is the process of creating new documents using existing ones.

As concerns over plagiarism grow, more attention has been directed towards automatic plagiarism detection. This is a computational approach which assists humans in judging whether documents are plagiarised. However, most existing plagiarism detection approaches are limited to simple string matching techniques. If the text has undergone substantial semantic and syntactic changes, string-matching approaches do not perform well. In order to identify such changes, linguistic techniques which are able to perform a deeper analysis of the text are needed. To date, very limited research has been conducted on the topic of utilising linguistic techniques in plagiarism detection especially in Malayalam language.

The main goal of this research is to develop methods for detecting monolingual extrinsic

Plagiarism, with a particular emphasis on Malayalam documents. Here both the source and plagiarised texts are in the same language and the aim is to identify whether a given document is plagiarised or not based on their similarity. Cases of plagiarism created by paraphrasing the source document is considered in this work because detecting them is very challenging.

This thesis focuses on the two phases related to the detection of text plagiarism. The first is candidate document selection, where the given document

is used against a document collection to identify a small set of relevant source documents called the candidate documents. The second problem is pairwise document comparison, where a pair of documents are compared with each other to determine whether one document has been plagiarised from the other.

An IR-based framework is proposed for candidate document selection. This thesis presents four models for plagiarism detection in Malayalam documents.

First, a general framework for plagiarism detection is proposed. It involves the use of Natural Language Processing techniques along with the traditional string-matching approaches. The objective is to investigate and evaluate the influence of text pre-processing and linguistic techniques in Malayalam. This is achieved by evaluating the framework using N-grams model, fingerprinting based model, Semantic role labelling based model and Probabilistic network based model.

Experiments reveal that N-gram model is the simplest to implement and gives best results for direct copy plagiarism. Fingerprinting based model can be used while comparing large files because the fingerprinting algorithm is a procedure that maps large to a much shorter fingerprint, that represents the original data. Semantic role labelling model identifies plagiarism based on the semantic roles and so matching of irrelevant sentences can be reduced. The PNN based model combines different individual similarity measures to classify a text as plagiarised or not.

Contents

List of Tables

List of Figures

Abbreviations

COPS	COpy Protection System
HTML	Hypertext Markup Language
IR	Information Retrieval
LCS	Longest Common subsequence
MD5	Message-Digest 5
ML-SOM	Multilayer self-organizing map
NLP	Natural Language Processsing
PAN	Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection
PDF	Probabilistic Density Function
PDS	Plagiarism Detection System
PNN	Probabilistic Neural Network
POS	Part of Speech
SOM	Self-organizing map
SVM	Support Vector Machine
TF-IDF	Term Frequency- Inverse Document Frequency
VSM	Vector Space Model

Chapter

1 INTRODUCTION

Contents

From the start of the digital communication age, the internet has encouraged information sharing. With this, online literature searching and the problem of plagiarism have risen drastically. As a result more research has been focused towards automatic plagiarism detection. Automatic plagiarism detection is a computational approach which identifies whether texts or documents are plagiarized. Most of the existing plagiarism detection approaches are inadequate due to the fact that only direct string matching between the texts is done. String-matching techniques do not identify plagiarism committed by intelligently changing the syntax and semantics of the text. Therefore, linguistic techniques are needed, to analyze the text. To date, very limited research has been conducted on the topic of Malayalam plagiarism detection, particularly by utilizing linguistic techniques. In the light of this, the purpose of this research work is to design and develop a novel integrated framework for plagiarism detection in Malayalam documents. A text can be plagiarized by copying and pasting word by word, changing parts of the text, or by summarizing the whole text.

Different strategies are to be used to detect different kinds of plagiarism. In this work, a system that can detect verbatim and obfuscated plagiarism with synonym replacement is presented. The goal is to show that texts that appear quite differently on the surface can be in fact copied or reused text. The objective is to investigate and evaluate the effect of linguistic techniques combined with similarity measures for plagiarism detection in Malayalam documents.

1.1 Background

Plagiarism, which is the act of using original words and ideas of others and presenting them as one's own, is known as intellectual theft. It is both a moral and also a legal offence. Now-a-days, the internet which provides fast, vast, and easy access of information has increased the plagiarism phenomenon. Plagiarism exists in many different forms, and is often difficult to identify. It is a major concern in education institutions. Students use the internet to compose a new document by copying sections from different sources extracted online. The large volume of information available online, makes it impossible to manually check for plagiarism. Also, copied text is usually modified with the aim of disguising the plagiarism. Hence, computational methods are needed to help in plagiarism detection.

This is where automatic plagiarism detection started to gain attention, as it may be able to offer an effective and efficient solution. Plagiarism detection is the task of finding out plagiarized portions within a piece of text.

This detection method can be classified into two: Intrinsic detection and extrinsic plagiarism detection. An intrinsic approach refers to cases where plagiarism is to be detected based on a single piece of text, which may contain both non-plagiarized and plagiarized passages. An extrinsic approach refers to cases where sets of suspicious plagiarized texts and their potential original source texts are both available. The detection task aims to identify pairs of matching suspicious source cases, by analyzing the similarity of each suspicious case against collection of potential original cases.

The process of extrinsic plagiarism detection can be separated into two steps: source retrieval and text alignment. Source retrieval involves pointing out all the source documents, parts of which might have been reused in a given suspicious document. After source documents have been recovered by the source retrieval step, the next step is text alignment. The purpose of text alignment is to locate plagiarized content in the suspicious document along with the corresponding original text in the source document. If however, there are no detections, the suspicious document will be categorized as being non-plagiarized. The focus of the source retrieval task is on efficiently labeling a small number of documents as source from a huge pool of documents, whereas the focus of text alignment is on accurately finding out the plagiarized content.

One of the earliest plagiarism detection systems introduced by Bird (1927) examines the application of statistical methods in detecting plagiarism of multiple-choice answers. Other methods developed later also studied detecting plagiarism in multiple-choice tests. Study of plagiarism detection systems for written texts started around the 1990s. The tools developed, used statistical methods to calculate similarity between texts. Computer source code plagiarism and written-text plagiarism were the focus of study during that period.

In the last decade, commercial systems have increased because of the increase in the fraudulent act of plagiarism. In early 2000, there were only very few established systems for identifying written-text plagiarism. Later more systems were developed but the problem of plagiarism has not yet been dealt with effectively. The use of plagiarism detection systems has become the standard practice in higher education institutions and research organizations due to issues associated with copyrights and patents.

1.2 Motivation

The biggest challenge in the plagiarism detection is that most approaches are inadequate in dealing with texts containing substantial semantic and syntactic changes. For a human it is easy to understand texts which carry similar meaning even when they are rewritten using different words and structures. However, computers are unable to understand texts in a similar manner, especially when automatic detection relies on exact text matching. Fairly good plagiarism detection systems exist for English and other Latin based languages. But, no such system exists for Indian languages, particularly Malayalam.

In this context the main research question is

"How can we effectively detect plagiarism in Malayalam text taking into account the linguistic features of the language?"

The main research question brings forth the following sub questions.

1) What are the linguistic features for Malayalam to be considered for plagiarism detection?
2) What are the different forms of plagiarism?
3) What are the Malayalam language resources available and their limitations?
4) What are the plagiarism detection techniques available for English and other languages?
5) What are the difficulties in using the available methods when it comes to Malayalam language?
6) How to develop plagiarism detection models for Malayalam documents and do the performance evaluation of those models.

1.3 Objectives

Considering the above mentioned issues and research questions, the main objective of this research is as given below,

- Development of a Morphological Analyzer for Malayalam.

- Identification of semantic roles of various words in the sentence with respect to the main verb.
- Use of Malayalam word net for synonym identification.
- Identification of cleverly masked and restructured words or sentence segments using linguistic techniques.
- Use of similarity measures for finding the level of plagiarism.
- Development of a classifier using the various similarity measures for classifying whether a document is plagiarised or not.
- Application of fingerprint matching for plagiarism detection in Malayalam.
- Application of semantic role labelling in plagiarism detection in Malayalam.
- Development of a framework for plagiarism detection in Malayalam text.
- Performance evaluation of the framework.

1.4 Scope

The scope of the research is limited to monolingual, external and offline detection at document level in Malayalam. The goal is to generalise text comparison to include morphological and lexical variations.

1.5 Organization of the Thesis

The layout of the thesis is as follows:

Chapter 1 provides an introduction about the plagiarism detection systems and its need in the present digital world. Significance of the present study, objectives and contributions of this research work are also summarized.

Chapter 2 defines the important concepts related to plagiarism. It then describes the various types and characteristics of plagiarism and the main types of methodologies used in automatic plagiarism detection. The chapter also gives a description of evaluation approaches used in automatic plagiarism detection.

Chapter 3 covers the state-of-the-art approaches in plagiarism detection. The chapter also describes the role of NLP in plagiarism detection, the limitations of existing approaches, and other related work.

Chapter 4 provides a detailed description of our plagiarism detection framework. It first outlines a general framework, then describes the text pre-processing and NLP techniques used in the experiments listed in Chapters 5 and 6. The rest of the chapter describes the similarity metrics and evaluation metrics used.

Chapter 5 describes the different experiments performed with Malayalam documents based on n-gram string matching, fingerprinting , semantic role labelling model and Probabilistic Neural Network based model .

Chapter 6 summarizes the thesis by recapitulating its objectives, highlighting how successfully these objectives were addressed, summarizes the contributions of the research work and also suggesting a few further research directions.

1.6 Chapter Summary

This chapter gives an introduction about the research work. The motivation behind the research work, the major objectives of the research and the tasks involved in the work are clearly described. Finally, the organization of the thesis is explained.

Chapter 2

PLAGIARISM CONCEPTS

Contents

The important concepts related to plagiarism are discussed in this chapter. An overview of the various types and characteristics of plagiarism and the current methodologies used in automatic plagiarism detection are *explained here. The chapter also gives a description of evaluation approaches used in automatic plagiarism detection.*

2.1 Definition

Several definitions for plagiarism are available in literature.

University of Cambridge defines plagiarism as ”submitting one's own work, irrespective of intent to deceive, that which derives in part or in its entirety from the work of others without due acknowledgement. It is both poor scholarship and a breach of academic integrity.”

University of Oxford defines plagiarism as “presenting someone else’s work or ideas as your own, with or without their consent, by incorporating it into your work without full acknowledgement. All published and unpublished material, whether in manuscript, printed or electronic form, is covered under this definition. Plagiarism may be intentional or reckless, or unintentional. Under the regulations for examinations, intentional or reckless plagiarism is a disciplinary offence.”

The Health Informatics department of the University of Illinois identifies academic plagiarism in the following forms.

1. Submitting another person's work as their own.
2. Copy fragments of own earlier work without citing.
3. Paraphrasing other's work without properly citing sources.
4. Using quotations without citing the source.
5. Mix various sources together in the work without citing.
6. Incomplete citing.
7. Joining cited and non-cited sections together.
8. Providing proper citations, but using the same words and syntax.
9. Incorrectly citing the source.

According to the Oxford English Dictionary, plagiarism is: “The action or practice of taking someone else's work, idea, etc., and passing it off as one's own; literary theft. (Oxford English Dictionary)”

Plagiarism is also defined in literature as copying an original text and claiming its authorship (Potthast et al., 2012), the “unauthorised use or close imitation” of an original text and claiming its authorship (Hannabuss, 2001), or the “unacknowledged copying of documents” (Joy and Luck, 1999). Nevertheless, all the definitions focus on the act of unacknowledged, unauthorized reuse or copy of an original text in different forms, like verbatim copies, the paraphrasing, or the omission of citations on referenced text parts (Clough, 2003).

In our research context, we define a plagiarism case as follows:

A plagiarism has a sequence of words of various lengths, either directly copied or paraphrased from one source to another and can exist in an entire document, or within segments of a document.

2.2 Plagiarism Characteristics

Studies in the field of plagiarism detection in the past have shown that text plagiarism is mainly due to lexical, syntactic and semantic changes created in the text.

Lexical changes: Lexical changes involve the inclusion, removal or substitution of words in the text. Direct copy and paste plagiarism can be easily detected whereas word substitution with synonyms would require the analysis of lexical information in the text.

Syntactic changes: Syntactic changes involve the rearrangement of the structure of the text . This is generally done by re-ordering the words or phrases, changing active and passive voice etc. These changes are not easily identifiable using the traditional methods , and detection would require the analysis of the syntactical structure of the text.

Semantic changes: Semantic changes comprise of major changes in the text, involving paraphrasing that can be either lexical changes or syntactic changes or both. . These changes are the hardest to identify since this involves identifying the meaning of the text.

Example:

Original source text:

Deforestation should not be a big issue in the United states; forest cover is increasing across most of the country. Clear cutting scars the landscape and leads to soil erosion and water pollution. Cutting down "old growth" forest destroys precious habitat and often inspires uproar of protests.

Lexical changes in the original text:

Deforestation should not be a big ***concern*** *in* ***North America*****;** *forest cover is increasing across most of the* ***U.S*****.** *Clear-cutting* ***damages*** *the landscape and leads to soil erosion and water pollution. Cutting down "old growth" forest destroys* ***valuable*** *habitat and often inspires many protests.*

Syntactic and semantic changes in the the original text*:*

There should not be much concern over deforestation in the U.S., as we actually are seeing an increase in forest cover over much of the country. The countryside can be damaged by clear-cutting, which results in erosion of the soil and pollution of the water. People often protest when old growth forests are cut down, because valuable habitat is destroyed.

2.3 Classification of Plagiarism

Plagiarism can occur in written text, computer source code, art, image, and music pieces. Previous researches for source code plagiarism detection used tools and metrics to capture statistical features and thus determine the similarities between the codes. Plagiarism detection in written text is more challenging because of the different ways of representing the copied text. This thesis focuses on written text only.

2.3.1 Classification based on the cause

Different types of plagiarism which have been identified by previous researchers are: Direct plagiarism, Self-plagiarism, Mosaic Plagiarism and Accidental plagiarism.

Direct plagiarism is the intentional word-for-word copy of parts or whole of someone else’s work, without credit and without reference to the author.

Mosaic plagiarism occurs when a person uses phrases from someone else’s work or replaces words with synonyms without using quotation marks. (“patch writing”).

It is called accidental plagiarism when a person forgets to cite or wrongly quotes the sources.

Self-plagiarism occurs when a person submits his or her own prior work in part or in whole as a new work.

2.3.2 Classification based on the level of copy

Another classification scheme for plagiarism is identified as: Direct literal copy and Intelligent modified copy.

Direct literal copy involves all cases of copy committed by the insertion, deletion, substitution and reordering of words. It also involves changing the syntax of the text and splitting a long sentence into small sentences or joining small sentences to form long sentences.

Intelligent modified copy involves paraphrasing, summarising, translation and performing other semantic changes to the text. Figure 2.1 shows plagiarism methods involved in the classification based on the level of copy.

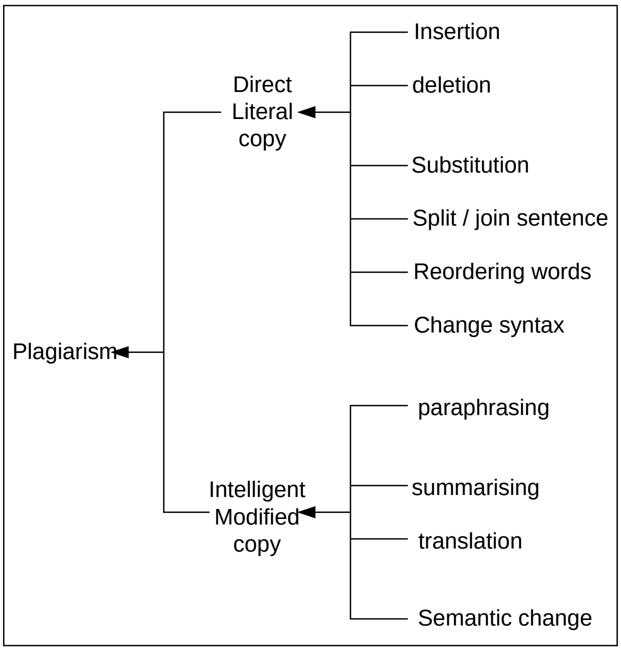


Figure 2.1: Plagiarism methods

2.4 Automated Plagiarism Detection

This section discusses what a plagiarism case is, the different plagiarism detection approaches, plagiarism corpora, and evaluation methodologies.

The plagiarism problem can be tackled from two different perspectives, prevention and detection. Plagiarism detection methods can be applied only after the plagiarism has been committed whereas the prevention methods can educate people not to do it. Implementation of prevention methods is not easy since it needs the participation of the entire society involved. Copy plagiarism detection methods, are easier to implement, and can tackle the problem at different levels of complexity (Potthast et al., 2010, 2010a). Both prevention and detection can be combined to effectively reduce plagiarism(Schleimer et al., 2003).

Automatic plagiarism detection techniques are generally based on a comparison of the contents of the documents and assigns a degree of similarity between the plagiarized and source documents, which is quantified by a similarity score.

Given a document *d* and a potential source of plagiarism *D*, automatic plagiarism detection consists of identifying pairs of sentences (*s* , *s'*) from *p* and *d'* (*d'* ϵ *D*) respectively, such that *s* and *s'* are highly similar. This similarity could be: *s* is exact copy of *s'*, *s* is obtained by obfuscating *s'* or *s* is semantically similar to *s'* but uses different words or language. This problem has been dealt with by many researchers in the last decade using many techniques related to information retrieval and copy detection. These techniques basically involve two steps : to retrieve the source *d'* from *D*, and the extensive comparison between *d* and *d'*.

2.4.1 Types of plagiarism detection approaches

Plagiarism detection systems can be classified based on the corpus used, the location of the corpus used and also the languages used in the documents under consideration.

Based on the corpus used, the task of plagiarism detection can be divided into two main types: intrinsic and external (Meyer zu Eissen and Stein, 2006; and Potthast et al., 2010).

External detection systems compare a suspicious document with a reference collection, which is a set of documents assumed to be the original source. Based on a chosen algorithm and predefined similarity criteria, the detection task is to retrieve all documents that contain text that is similar to a degree above a chosen threshold to text in the suspicious document (Stein et al., 2007). Intrinsic detection systems only analyze the suspicious text without performing comparisons to other documents. This approach tries to identify changes in the unique writing style of an author as an indicator for potential plagiarism (Meyer et al., 2006).

Based on whether the source documents used for the comparison are from the internet or from the local system, detection approaches are further classified as online or offline. Online plagiarism detection systems compare a suspicious document with a reference collection, which is a set of documents available on the web whereas an offline plagiarism detection systems compares a suspicious document with documents from the local system or database.

Detection approaches can be classified based on the languages used in the documents under consideration, as Monolingual detection and Cross-lingual detection. A monolingual detection approach detects suspicious cases that are derived from the source cases without any change in the language. Here, both suspicious and source documents contain text in the same language. A crosslingual detection approach detects the suspicious cases that are derived from source cases of different languages. Here, both suspicious and source documents contain text written in different languages.

In this thesis, the focus is on external offline detection of monolingual texts in Malayalam.

2.4.2 Textual features for document characterization

Several textual features can be used to detect and quantify plagiarism. This section discusses textual features used in different extrinsic frameworks:

Textual features to represent documents in extrinsic plagiarism detection include: lexical features such as character n-gram and word n-gram; syntactic features, such as chunks, sentences, phrases, and parts of-speech (POS); semantic features such as synonyms and also structural features that take contextual information into account. A detailed description of textual features for extrinsic plagiarism detection is given below.

1) Lexical Features: Lexical features are found at the character or word level. In character-based n-gram representation, a document is represented as a sequence of characters whereas in word-based n-gram representation, a document is represented as a collection of words. Simple word-based n-grams may be constructed by using bigrams (word-2-grams), trigrams (word-3-grams) or larger.

2) Syntactic Features: Syntactical features are evident in POS of the words in the sentences. Nouns, verbs, pronouns, adjectives, adverbs, prepositions, conjunctions, and interjections are the basic POS tags. POS tagging is the task of assigning a particular POS tag to corresponding words in a text. In sentence-based representations, the text is first split into statements by identifying the end-of-sentence delimiters, and then POS and phrase structures can be constructed by using POS taggers. On the other hand, chunks are generated by sliding windows and then POS could be used to generate POS chunks. Word order, in a sentence or a chunk, could further be combined as a feature, and used as a comparison scheme between sentences.

3) Semantic Features: Semantic features of a word include word class, synonyms, antonyms, hypernyms, and hyponyms. The use of thesaurus dictionaries and lexical databases like the Word-Net, helps in recognizing the semantic meaning of the text. Together with POS tagging, the semantic dependencies can be identified which will be helpful in plagiarism detection.

4) Structural Features: Most plagiarism detection algorithms use flat document features, such as lexical, syntactic, and semantic features and only a very few algorithms can handle structural or tree features. Structural features reveal text organization. Documents can be described as a collection of paragraphs or passages, which can be considered as topical blocks. Paragraphs that are topically related (discuss the same subject) can be grouped into sections. Structural features might characterize documents as headers, sections, subsections, paragraphs, sentences, etc. Structural features are mostly stored as XML trees for easier processing.

Structural features can be further divided into block-specific and content-specific. Block-specific tree structured features can be used to describe a collection of web documents as blocks, namely, document-page-paragraph. Then, paragraphs can be grouped into pages, whereby a new paragraph is added to each page until a maximum threshold of word count is reached; otherwise, a new page is created.

The document features can be encoded as content-specific tree-structured features by using semantically related blocks, such as document-section-paragraph or class-concept-chunk. The use of content-specific tree-structured features in combination with some flat features can be very useful in capturing the document's semantics and allow for the detection of idea plagiarism. Table 2.1 summarizes each type together with computational tools and resources required for their implementation.

Table 2.1 Summary of text features and tools required for their implementation

Text features	Works	Feature examples	Tools required
Lexical	Yerra, R. and Ng, Y.K., (2005).	Character n-grams	Feature selector
	Kasprzak, J. Et al., (2009) Koberstein, J. and Ng, Y.K., (2006). Alzahrani,S. And Salim,N., (2010)	Word n-grams	Tokenizer
Syntactic	Scherbinin, V. and Butakov, S., (2009)	chunks	Tokenizer, POS tagger, text chunker
	Elhadi, M. and Al-Tobi, A., (2009) Ceska, Z. and Fox, C., 2011.	POS and phrase structure	Sentence splitter, Tokenizer, POS tagger
	Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D. and Crockett, K., (2006).	Word order	Sentence splitter, Tokenizer, compressor
	Yerra, R. and Ng, Y.K., (2005) Alzahrani, S.M. and Salim, N., (2008)	sentence	Sentence splitter, Tokenizer, POS tagger, text chunker , partial parser
Semantic	Yerra, R. and Ng, Y.K., (2005) Alzahrani,S, (2008)	Synonyms	Tokeniser, POS tagger, Thesaurus
	Y. Li,D.McLean, Z.A. Bandar, J. D. O'Shea, andK. Crockett, (2006)	Semantic dependencies	Sentence splitter, Tokenizer, POS tagger, text chunker , partial parser, semantic parser
Structural	H. Zhang and T. W. S. Chow, (2011)	Block specific	HTML parser, specialised parsers
	S. Alzahrani, (2012)	Content specific	Tokeniser , specialised dictionaries.

2.5 General Framework

The external plagiarism detection task follows a general framework that involves three main stages of processing. The three stages are: text pre-processing, source retrieval (filtering) and text alignment. Figure 2.2 shows the generic architecture to detect plagiarism in a given suspicious document when a large collection of potential source documents is also available.

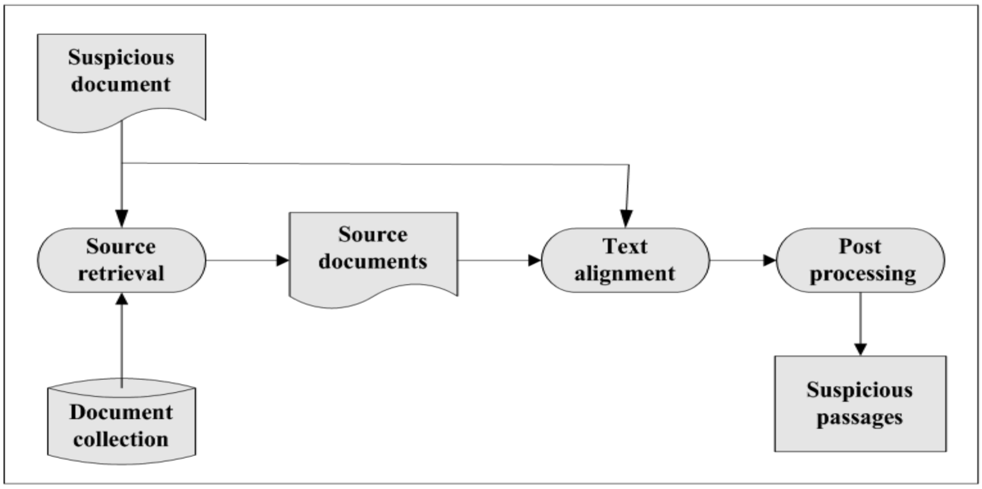


Figure 2.2 General architecture to detect plagiarism

The steps in the architecture is discussed below

i. Source retrieval

Source retrieval process identify a small set of candidate documents from a large collection of potential source documents. The approaches found in literature can be broadly classified as IR models and Clustering techniques.

IR models include fingerprinting and hash-based models, vector space model, latent semantic indexing, histogram based multilevel matching, signature-based multilevel matching and fuzzy models for candidate document retrieval. Clustering techniques include self-organizing maps (SOM), and multi-layer SOM (ML-SOM).

ii. Text alignment

Text alignment process compares each candidate source document to the suspicious document, and extract segments of text that are highly similar. The detailed analysis methods include character-based methods, vector-based methods, syntax-based methods, semantic-based methods, fuzzy-based methods and structural-based methods (Alzahrani et al., 2011).

iii. Post-processing

Post-processing involves cleaning, filtering and merging the extracted text segments obtained from the previous step.

2.6 Evaluation Approaches

Detecting plagiarism is a very difficult task when the writer has intentionally performed the plagiarism. The common method to evaluate a detection system is to use a corpus of previously annotated texts cases.

2.6.1 Evaluation corpora

A general approach of evaluating plagiarism detection systems is corpus-based evaluation. This normally involves providing a set of texts (both plagiarised and non-plagiarised) to the system to determine whether a particular case is plagiarised or not. In the early period, METER corpus (Gaizauskas et al., 2001; Clough et al., 2002b), was used, which contains manually annotated news articles. (Gaizauskas et al., 2001; Clough et al., 2002b).

Later, in their work Weber-Wulf (2008), used a corpus of 31 essays, which were created manually with original, translated and paraphrased German text. As the size of the corpus is not adequate for detailed linguistic analysis, this corpus is not used for plagiarism research.

Clough and Stevenson (2010) developed a corpus with 95 short answers of length ranging from 200 to 300 words, which incorporates original, verbatim copy, shallow paraphrasing and structural changes.

Imene Bensalem et al. (2014) developed an Arabic corpus with 1024 documents for intrinsic plagiarism detection. 46% of the documents were of very short length(1 to 3 pages), 37% of the documents were of short length(3 to 5 pages), 12% of the documents were of medium length(15 to 100 pages) and 5% of the documents were long(more than 100 pages).

Siddiqui et al. (2014) developed an Arabic corpus with 1665 documents, of which 1156 documents were taken as suspicious ones and 509 documents were taken as source documents. The suspicious documents contains 264 documents with no obfuscation and 892 documents with obfuscation (718 documents created from the Web and 174 documents from other sources).

Khoshnavataher et al. (2015) developed a Persian corpus with 1057 documents, of which 529 are designated as plagiarised documents, as part of the task of text alignment corpus construction at PAN 2015 competition. Here only 50% of suspicious documents contain plagiarism cases. Five obfuscation strategies such as no obfuscation, random change of order, POS preserving change of order, synonym substitution and random addition and deletion of words were used. Moreover, the percentage of plagiarism in each suspicious document is distributed between 5% and 100% of its length.

Mashhadirajab et al. (2016) developed a Persian corpus with 11089 documents. 48% of the documents were taken as source documents while 28% of the documents were taken as plagiarised documents with obfuscation and 24% of the documents were taken as plagiarised documents without obfuscation. According to the length of documents they have classified documents as short (less than 10 pages), medium (10 to 100 pages) and long (more than 100 pages).

A good corpus for evaluating Malayalam plagiarism detection is not yet available.

2.6.2 Evaluation metrics

The performance of a plagiarism detection system can be evaluated by standard evaluation metrics such as precision, recall, and F-score. Two additional metrics have been proposed in the context of the PAN competitions (Potthast et al., 2010b): granularity and plagdet.

Precision is defined as the fraction of retrieved documents that are relevant against query (Equation 2.1) and recall is defined as the fraction of relevant documents that are retrieved against query (see Equation 2.2).

For an information request, if $|R_r|$ is the number of relevant documents retrieved, $|R_j|$ is the number of irrelevant documents retrieved and $|N_r|$ is number of relevant documents not retrieved, then the precision and recall can be depicted as a set diagram as shown in figure2.3.

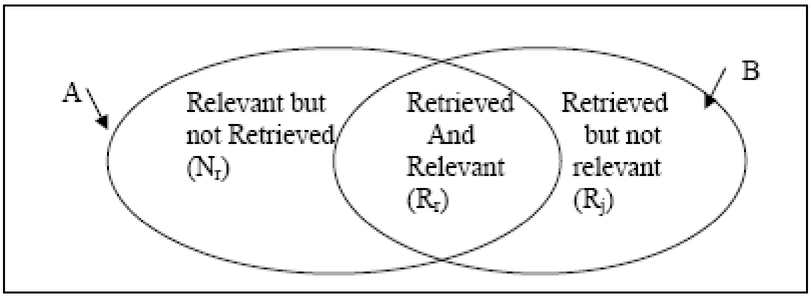

A
B
Relevant but not Retrieved (N_r)
Retrieved And Relevant (R_r)
Retrieved but not relevant (R_j)

Figure 2.3: Precision and recall

$$precision = \frac{|retrieved \cap relevant|}{|retrived|} \qquad (2.1)$$

$$recall = \frac{|retrieved \cap relevant|}{|relevant|} \qquad (2.2)$$

Precision and recall can be represented in a contingency table as shown in table 2.2 below.

Table 2.2 Contingency Table for precision and recall

Document set (D)		**Actual Class**	
		Relevant	**Not-Relevant**
Predicted Class	Retrieved	True Positive (correct result)	False Positive (irrelevant result found)
	Not-Retrieved	False Negative (Missing result)	True Negative (irrelevant result not found)

TP = Number of true positives

True positives are documents deemed relevant by both the human expert and the information retrieval system.

FP = Number of false positives

False positives are returned by the IR system, but are considered irrelevant to the query by the human expert

TN = Number of true negatives

True negatives are not returned by the system and are considered irrelevant by the human expert.

FN = Number of false negatives

False negatives are documents relevant to the query which are not found by the system. The value of precision ranges from 0 to 1, where 0 means that no relevant document is retrieved and 1 means that all the retrieved documents are relevant. The value of recall also ranges from 0 to 1, where 0 means no relevant documents have been retrieved and 1 means all relevant documents have been retrieved. Very high recall often results in low precision , which is not desirable. Combining precision and recall is proposed to overcome this problem .

F measure is the combination of precision and recall and is computed as [Baeza-Yates and Ribeiro-Neto, 2011]:

$$F_\alpha = \frac{(1+\alpha^2).p.r}{\alpha^2.p+r} \tag{2.3}$$

where p is precision, r is recall and α is the weight assigned to precision or recall.

F1 measure the harmonic mean of precision and recall is obtained when equal weights (α = 1) are assigned to precision and recall. F1 measure is computed as:

$$F_1 = \frac{2.p.r}{p+r} \tag{2.4}$$

The value of F-Measure ranges from 0 to 1, where 0 means no relevant documents have been retrieved and is 1 means all ranked documents are relevant. The harmonic mean F assumes a high value only when both recall and precision are high (Baeza-Yates and Ribeiro-Neto, 2011).

Granularity measures the accuracy of the approach in finding the correct segmentation for plagiarism cases, and it is only appropriate for passage level detection. Plagdet represents the overall score which combines granularity with F-score.

2.7 Chapter Summary

This chapter reviewed different definitions for plagiarism and described the important concepts related to plagiarism. The different types of plagiarism, and the characteristics of plagiarism are discussed. Methods for the mono-lingual extrinsic plagiarism detection including character based, vector based, syntax based, semantic based, fuzzy based and structural based are briefly described. Finally, evaluation measures used to evaluate the performance of plagiarism detection systems such as precision, recall and F measure are described.

…….ꙮ…….

Chapter

3 LITERATURE SURVEY

Contents

This chapter covers the state-of-the-art approaches and reviews international competitions and shared tasks on plagiarism detection. The chapter also describes some of the existing tools that are commonly used for plagiarism detection.

3.1 History of PDS

Early PDS systems were developed for multiple choice tests and source code. Until the year 2000, most research focussed on detecting software code plagiarism only. A prototype, COPS (Brin et al, 1995) to detect full or partial copies of documents and SCAM (Shivakumar and Gracia-Molina, 1995, 1996) were the early works for detecting text plagiarism.

From 2000, more research was directed to address the problem of text plagiarism. A lot of commercial tools emerged during this period. A major disadvantage of these systems is that they do not incorporate language processing techniques.

3.2 Recent Works

3.2.1 Content-based detection

Content-based detection is the most widely used technique for identifying plagiarism. It usually consists of comparing words, word n-grams, character n-grams, sentences or paragraphs from the suspicious documents against possible source documents. Content-based detection can be classified as belonging to three methods namely: bag-of-words model, n-grams model, and fingerprint model.

i. Bag-of-words model

Bag-of-words model does not consider word order and is commonly used for the candidate retrieval step. It involves applying an information retrieval system to retrieve documents that contain the same words with the suspicious document. Retrieving candidates with the bag-of-words approach was used in several works. (Kong et al., 2012; Kong, Qi, Du, Wang, & Han, 2013; Sanchez-Perez, Sidorov, & Gelbukh, 2014; Torrejón & Ramos, 2013).

ii. N-gram matching

N-gram matching is the most common approach in content based detection. A word n-gram is a sequence of n consecutive words. The suspicion is that the similarity of a pair of documents id directly propotional to the number of n-grams they have in common.

Barron-Cedeno and Rosso (2009) used word n-grams and found that 2-grams achieved the best recall while 3-grams yielded the best precision.

Valles Balaguer (2009) used word 4-grams,5-grams and 6-grams found the use of 6-grams the best , as it achieved the best precision. Gupta and Rosso (2012) also experimented with word 6-grams.

Grozea and Popescu (2011) used character 256-grams in their system, named Encoplot.

Torrejón and Ramos (2010) in their system named CoReMo used skip n-grams to detect plagiarized cases. A skip n-gram allows words to be skipped, and this way more matches can be selected. This system also uses translation to detect cross-language plagiarism.

Stamatatos (2011) used stopword n-grams where documents are represented based on the presence of a predefined list of stopwords in the text. The aim is to find common n-grams of stopwords between the source and suspicious documents. In the candidate retrieval step pairs of documents that share only very frequent stopwords are discarded. This eliminates those pairs of documents that share only very frequent stopwords.

iii. Fingerprints

Fingerprints are compact representations of documents. For generating fingerprints documents are partitioned into chunks and a function is applied to each of them producing an integer value. The probability of generating the same representation for different documents should be negligible.

Hoad and Zobel (2003) proposed an identity measure and their experiments showed that the proposed identity measures outperforms other fingerprinting methods.

Stein and Meyer (2006) proposed an improvement to the MD5 algorithm based on fuzzy fingerprints. They found that fuzzy-fingerprints resembled the cosine similarity.

Kasprzak and Brandejs (2010) computes MD5 hashes fingerprints to documents with overlapping word 5-grams. A chunk is represented by the most significant 30 bits of the hash. The similarity between document pairs is calculated from their common chunks. Document pairs that contain 20 or more common chunks are selected as candidates. For each pair, the method analyzes whether the common chunks form one or more valid intervals, in which the gap between two neighboring common chunks is not bigger than 50 chunks. Common chunks that satisfy this condition are reported as plagiarism cases.

HaCohen-Kerner, Tayeb, and Ben-Dror (2010) applied a variety of methods to identify similar papers on a collection of 10,100 published academic papers from the ACL Anthology. Compared methods include several variations of fingerprinting and anchor-based strategies and combinations. The paper reports how many pairs of papers were considered similar by each method. They concluded that full- fingerprints of length 3 (i.e., considering all chunks of length 3) was the best method. Dealing with paraphrases is an important feature for identifying cases of plagiarism in which the offender has tried to disguise the duplication. Systems that handle paraphrases have performed well in evaluation campaigns.

The system by Grman and Ravas (2011) was the winner of the PAN-11 competition. The method is based on calculating the number of matching words for a pair of passages from the source and the suspicious documents. First, pairs of passages in which the number of matching words exceeds a certain threshold are selected. WordNet is used as a source of synonyms. The use of synonyms and the disregard for word order aid the detection of paraphrased and translated plagiarism.

Hoad and Zobel (2003) provide a detailed review of fingerprinting and compare it with an identity measure that they propose. Their experiments showed that the proposed identity measures outperform fingerprinting methods. They have also noted that anchor-based methods achieved good results. An anchor is a string (or an n-gram) in the text of the document. Anchors should be chosen so that there is at least one in each document but not so common that the fingerprint becomes too large.

Stein and Meyer (2006) proposed an algorithm based on fuzzy fingerprints with the goal of generating the same hash code to similar fragments. The fuzzy fingerprint is the union of the hash values for a word *n*-gram. The goal is to investigate the runtime performance and the difference between the scores for fuzzy-fingerprint similarity and cosine similarity under the vector space model. The authors concluded that fuzzy-fingerprints resembled the cosine similarity better than the MD5 fingerprint.

The system by Kasprzak and Brandejs (2010) computes MD5 hashes fingerprints to documents with overlapping word 5-grams. A chunk is represented by the most significant 30 bits of the hash. The similarity between document pairs is calculated from their common chunks. Document pairs that contain 20 or more common chunks are selected as candidates. For each pair, the method analyzes whether the common chunks form one or more valid intervals, in which the gap between two neighboring common chunks is not bigger than 50 chunks. Common chunks that satisfy this condition are reported as plagiarism cases.

HaCohen-Kerner, Tayeb, and Ben-Dror (2010) applied a variety of methods such as several variations of fingerprinting and anchor-based strategies to identify similar papers on a collection of 10,100 published academic papers from the ACL Anthology. Their finding is that full fingerprints with chunks of length three was the best method.

Grman and Ravas (2011) proposed a method for dealing with paraphrased text. WordNet is used to replace words with their synonyms and passages in which the number of matching words exceeds a certain threshold are selected. Word order is not considered .

3.2.2 Detection based on structural information

Structural information of a document is represented by headers, sections, paragraphs, references etc.

In the work by Chow and Rahman (2009), documents are represented as trees in which paragraphs are at the bottom layer, pages at the middle layer, and whole documents at the top. Candidate retrieval is performed at the top layer, while the other two layers are used for detailed analysis. The nodes contain word histograms which then go through principal component analysis to reduce dimensionality.

H. Zhang and Chow (2011) propose to enhance the above work by adding a weight parameter to the word histograms of the nodes of the tree to generate a signature representation. Documents are sorted in ascending order of distance and the top matches are selected for the lower level analysis. They reported good results from experiments on a collection of 10K HTML documents.

El Bachir Menai and Bagais (2011) represented Arabic documents in a tree structure with different levels for document, paragraph, and sentence and comparison was done level by level from root to leaf. To extract the fingerprints of a document, the method uses the hash function of Kernighan and Ritchie (1988). At document level if two documents have a common number of hashes above a fixed threshold, then those pair of documents are considered for analysis at the paragraph level. The Longest Common Substring (LCS) algorithm is used to compute the similarity between sentences. A pair of sentences are considered similar if the length of the LCS is greater than a fixed threshold.

Alzahrani et al. (2012) partition scientific publications into components, and a plagiarism case in the “Introduction” and “Definitions” is considered less important compared to a case of plagiarism in the

“Evaluation” or “Discussion” components. Different functions are used to measure component factor-weight based on their distinct terms. When a component from the suspicious document gets a high similarity score in relation to the corresponding component from a source document a case of plagiarism is identified. The experiments were performed on an artificial collection, and the results showed structural information can be used for candidate retrieval as well as text alignment.

3.2.3 Detection based on References and citation

PDS that analyze citations and references can be classified as

- Systems that check whether the citation gives credit to the original source (as a filter).
- Systems that examine the similarities in citations or references across documents (as a source of similarity).

Sorokina et al. (2006) and Alzahrani et al. (2012) used citation analysis as a filter to discard false positives. If the author(s) of a document D1 appears in the reference list of document D2, then they consider it as a case of “mild plagiarism” since plagiarists usually do not cite their source.

Gipp and Beel (2010), Gipp and Meuschke (2011), Meuschke et al. (2012), and Gipp et al. (2014) used citation analysis as a source of similarity between documents and also proposed several similarity functions based on shared references and citations. Longest Common Citation Sequence and Greedy Citation Tiling are similarity functions take the order of the citations into consideration whereas bibliographic coupling and citation chunking are similarity functions that ignore the order of the citations. Their experimental results showed that citation-based detection is better than content-based detection in cases of strongly disguised plagiarism.

Pertile et al. (2013) compared the similarity of scientific papers based on the analysis of co-occurrences in citations. Their experimental results showed that most of the cases with co-occurrences in citations correspond to plagiarism. They conducted experiments only on artificially created cases of plagiarism and not on real scientific documents.

The above classifications of PDS for candidate retrieval and text alignment are shown in table 3.1.

Table 3.1: Candidate retrieval and Text alignment

		Work	Candidate retrieval	Text alignment
Content based	Bag of Words	Kong et al.(2012)	ChatNoir API, TF-IDF, VSM Ranking	cosine and Dice similarity
		Kong et al. (2013)	ChatNoirAPI, TF-IDF, PatTree and Weighted TF-IDF	cosine similarity, Bilateral Alternating Merging
		Sanchez-Perez et al. (2014)	TF-IDF, VSM and Dice	cosine + thresholds for allowed gaps
		Tarrejon & Ramos (2013)	IR system & Reference Monotony Pruning	Surrounding Context N-grams & Odd-Even N-grams
	N-grams	Barron-Cedeno & Rosso (2009)	n-grams (*n* = 2 and 3)	classification based on a containment measure
		Grozea & Popescu (2011)	similarity matrix with the number of common n-grams within a window	common n-grams and clustering

		Gupta and Rosso (2012)	word n-grams in common (*n* = 6)	word n-grams in common (*n* = 6)
		Stamatatos (2011)	stopword n-grams (n = 11)	common sequence word n-grams of stopwords (*n* = 8)
		Kasprzak & Brandejs (2010)	n-grams (*n* = 5)	gap between two neighbouring common n-grams
		Grman & Ravas (2011)	matching words between a pair of passages and WordNet for synonyms	matching words between a pair of passages and WordNet for synonyms
	Fingerprinting	Stein & Meyer (2006)	fuzzy-fingerprints	Not available
		Menai & Bagais (2011)	fingerprints	Longest Common substring
		HaCohen-Kerner et al. (2010)	Not available	Fingerprints, Anchor-based methods, Titles of references
Content & structure based		Zhang & Chow (2011)	Multilayer Self Organizing Map (SOM), word histograms, PCA, Earth Mover's Distance	Multilayer Self Organizing Map, word histograms, PCA, Earth Mover's Distance
		Chow & Rahman (2009)	Multilayer SOM, word histograms, PCA	Multilayer SOM, word histograms, PCA
		Gipp & Bela (2010)	references/citations in common	references/citations in common
Citation based		Gipp & Meuschke (2011) Meuschke et al.(2012) Gipp et al. (2014)	the absolute number of references in common between them	rarely cited documents are more important and should receive a higher score
		Pertile, Rosso & Moreira (2013)	a high rate of inter-document co-occurrences could be an indication of plagiarism	Co-occurrence in citations
		Alzahrani et al. (2012)	TF-IDF weighting for different sections, Cosine similarity	Jaccard similarity

3.2.4 NLP based plagiarism detection:

Clough (2003) suggested applying NLP techniques for plagiarism and that this could give better accuracies through the detection of paraphrased texts. No experiments were performed to prove his suggestion but it has inspired the use of NLP in the plagiarism detection field.

The NLP techniques can be applied at various stages of plagiarism detection. Plagiarism detection systems performs pre-processing and candidate filtering tasks prior to the text alignment stage. Pre-processing generalises the texts, and candidate filtering reduces the search space for text alignment. The common method found in literature is to apply shallow NLP techniques such as tokenisation, lowercasing, stop word removal, lemmatisation and stemming as part of the pre-processing stage.

Uzuner et al. (2005) used shallow semantic and syntactic rules to detect plagiarism. The semantic class of each verb is determined by a part-of-speech (POS) tagger and the syntactic structures are extracted for each sentence. A semantic class represents a group of verbs which are similar in meaning. The similarity matching is not based on words, but on the verb classes. Their results showed that syntactic features can achieve better performance than tf-idf, and that linguistic techniques can be used to identify paraphrases better than statistical methods.

Mozgovoy et al. (2006) proposed to apply NLP techniques for text pre-processing for the Russian language. The techniques include tokenisation, generalisation of words into their hierarchical classes and extraction of functional words for matching. Mozgovoy (2007) proposed incorporating tokenisation and syntactic parsing for improving the string matching algorithms used for plagiarism detection. However, it only resulted in the development of a fast string-matching algorithm.

Chuda and Navrat (2010) proposed the use of tokenisation, stop word removal and stemming in the text pre-processing stage to Slovak texts, but the effect of using them is not reported.

Ceska and Fox (2009) and Ceska (2007, 2009) proposed the inclusion of latent semantic analysis (LSA) along with text pre-processing techniques such as removal of punctuation, removal of irrelevant words, replacement of numbers with a dummy symbol and lemmatisation for plagiarism detection. They also integrated a thesaurus to generalise the words in the texts. The comparison is done by n-gram matching using singular value decomposition (SVD), which involves the retrieval of truncated singular values and vectors from an original term-document matrix.

Leung and Chan (2007) suggested incorporating both shallow and deep NLP in automatic plagiarism detection, involving the application of synonym generalisation and extraction of syntactic structure. Semantic processing identifies the deep structure of a sentence by converting parse trees into case grammar structure. This approach compares sentences at semantic level. The method was not implemented due to the lack of a semantic analysis tool and a suitable corpus.

Mozgovoy et al. (2007) also suggested using deep NLP techniques like parse trees to find the structural relations between documents . They proposed a two-stage approach to plagiarism detection. In the first stage all documents in the dataset are parsed using Stanford Parser, and the grammatical relations so generated are post-processed into groups of words. In the second stage the amount of similar grammatical relations between documents is computed. Parsing was found to be useful for detecting sentence re-ordering, but it results in a loss of the original word order in every sentence. Their system ,therefore cannot highlight similar blocks of text. The problem of paraphrasing is also not tackled using this approach.

Ceska (2009) and Alzahrani and Salim (2010) performed experiments using a Czech thesaurus and an English thesaurus respectively. For English, the authors used WordNet a well-developed thesaurus which is semantically structured. It provides information on relationships between words, which allows the matching of synonyms and hyponyms. Since the WordNet has one or more synsets for a word the matching of WordNet synsets with the correct sense (word dense disambiguation) is the main challenge.

In Chong et al. (2010) the combination of shallow and deep NLP techniques was employed in an experiment using a small-scale corpus of short plagiarised texts. Techniques such as chunking and parsing are used to generate features which are compared against an overlapping 3-gram word baseline. Language models are used to generate probabilities for word n-grams, perplexities and out-of-vocabulary rates. The Jaccard coefficient similarity metric, is applied to the extracted features to generate similarity scores. Experimental results report that a combination of word 3-grams, lemmatisation, language model perplexities and parsing were the best performing features.

Chong and Specia (2011) studied lexical generalisation for word-level matching with the aim of tackling paraphrased text in plagiarism detection. Lexical generalisation substitutes each content word with the set of all its synsets. Word order is not taken into consideration. Similarity comparison is carried out at the word level and the results were compared against an overlapping 5-gram word metric. The experiment was tested on a large-scale corpus and the results showed that lexical generalisation improves recall by reducing the false negatives.

3.3 Tools for Detecting Plagiarism

It is quite obvious from the severity of the problem of plagiarism , that academia requires tools to automate and improve plagiarism detection (Maurer et al**.,** 2006)**.** Plagiarism detection tools are programs that compare documents with possible sources in order to identify similarity and hence discover submissions that might be plagiarized(Lancaster and Culwin, 2005). According to the type of text that the tool operates on, the tools can be divided into tools that operate on non-structured or free text and tools that operate on source code.

There are a large number of detection tools that have been developed for automated plagiarism detection of text. Examples are Turnitin, PlagAware, PlagScan, Check for Plagiarism, iThenticate, PlagiarismDetection.org, Academic Plagiarism, The Plagiarism Checker, Urkund, Docoloc etc.

3.3.1 Turnitin

This is a product from iParadigms [iParadigm, 2006]. It is a web based service. Detection and processing is done remotely. The user uploads the suspected document to the system database. The system creates a complete fingerprint of the document and stores it. Proprietary algorithms are used to query the three main sources: indexed archive of Internet, books and journals in the ProQuest database and 10 million documents already submitted to the Turnitin database. Turnitin offers different account types. They include consortium, institute, department and individual instructor. At instructor account level, teachers can create classes and generate class enrolment passwords. Such passwords are distributed among students when joining the class and for the submission of assignments.

3.3.2 PlagAware

PlagAware is an online-service offering services around the topics searching, finding, analyzing and tracing of plagiarisms. The central element of PlagAware is a search engine specialized in detecting identical contents of given texts. The two primary application fields of the plagiarism search engine PlagAware are

- the plagiarism assessment of texts transmitted to PlagAware and
- the continuous monitoring of texts and web pages for eventual content theft.

The plagiarism assessment analyzes to what extent a given text is a plagiarism of already published texts and contents. The document to be checked is assumed to be plagiarised and the utilized sources are found from the web. The function text monitoring regards a given text (text contents of a web page) as original and regards all places of finding in conformity with it as potential plagiarisms of the monitored text or web page.

- The main features of PlagAware are:
    - Database Checking: PlagAware does not have local database but it offers checking other database that are available over the internet.
    - Internet Checking: PlagAware is an online application and it considered as one of search engine, allows the student or webmaster to upload and check their academic documents, homework, manuscript and articles to be searched against plagiarism over world wide web.ans also provides a webmaster to have capability to do automatic observation of their own page against possible contents theft.
    - Publications Checking: PlagAware: support mainly used in academic filed so it provides checking of most types of submitted publication like homework, manuscript, documents, including, books, articles, magazines, journals, editorial and PDFs etc.
    - Synonym and Sentence Structure Checking: PlagAware does not support synonym and sentence structure checking.
    - Multiple Document Comparison: PlagAware offers comparison of multiple documents.

  - Supported Languages: PlagAware supports German as primary language, English and Japanese as secondary languages.

3.3.3 PlagScan

PlagScan is online software used for textual plagiarism checker. PlagScan is often used by school and provides different types of account with different features. PlagScan use complex algorithms for checking and analyzing uploaded document for plagiarism detection, based on up-to-date linguistic research. Unique signature extracted from the document's structure that is then compared with PlagScan database and millions of online documents. So PlagScan is able to detect most of plagiarism types either directs copy and paste or words switching, which provides an accurate measurement of the level of plagiarized content in any given documents.

- The Main features of PlagScan are:
  - Database Checking: PlagScan it has own database that include millions documents like (paper, articles and assignments), and articles over World Wide Web. So it offers database checking whether locally or others database over the internet.
  - Internet checking: PlagScan is an online checker so it provides internet checking to all submitted documents. Whether that the document available on the internet or available in the local database or cached.
  - Publications Checking: PlagScan: is mainly used in academic filed so it provides checking most types of submitted publication

like documents, including, books, articles, magazines, journals, newspapers, PDFs etc. online only.

- Synonym and Sentence Structure Checking: PlagScan does not support synonym and sentence structure checking but provides integration via application programming interface in your existing content management system or learning management system possible.
- Multiple Document Comparison: CheckForPlagiarism.net offers comparison of multiple documents in parallel.
- Supported Languages: PlagScan supports all the language that use the international UTF-8 encoding and all language with Latin or Arabic characters can be checked for plagiarism.

3.3.4 CheckForPlagiarism.net

CheckForPlagiarism.net was developed by a team of academic people and became one of the best online plagiarism checkers. In order to maximize the accuracy CheckForPlagiarism.net has used methods like document fingerprint and document source analysis to protect document against plagiarism. The fingerprint-based approach is used to analyze and summarize collection of document and create a kind of fingerprint for it. So by creating fingerprint for each document with some of numerical attributes for each document in the collection, the matching or the similarity between documents can be easily found across billions of articles.

- The main features of CheckForPlagiarism.net are:
    - Database Checking: CheckForPlagiarism.net uses its own database that include millions documents like (paper, articles and assignments), and articles over World Wide Web. So it offers fast and reliable depth database checking, also provides checking through all other databases in different fields like medical database, law- related database and other specialty and generalized databases.
    - Internet Checking: CheckForPlagiarism.net: live(online) and cached links to websites used for extensive internet checking to all submitted documents. One more advantage is that it can still check your documents against if a website that is no longer online, this include all contents of website like forums, message boards, bulletin boards, blogs, and PDFs etc., all this check is done automatically and in (almost) real-time.
    - Publications Checking: CheckForPlagiarism.net offers detailed and deep checking of most types of submitted publication documents, including, books, articles, magazines, journals, newspapers, PDFs etc. this is done whether the publications is available online (active on the internet) or not available on the internet offline (store paper based).
    - Synonym & Sentence Structure Checking: CheckForPlagiarism.net is said to have a sole advantage, that other soft-wares do not

support, which is the fact that it uses a ”patented” plagiarism checking approach. In which the sentence structure of a document is checked to ensure improper paragraphing and thus is susceptible to plagiarism. Also a synonym check is done to words and phrases to identify any attempt of plagiarism.

- Multiple Document Comparison: CheckForPlagiarism.net can compare a set of different documents simultaneously with other documents and can diagnose different type of plagiarisms.
- Supported Languages: CheckForPlagiarism.net supports English languages, Spanish, German, Portuguese, French, Italian, Arabic, Korean, and Chinese languages.

3.3.5 iThenticate

iThenticate one of the application or services designed especially for the researchers, authors’ publisher and other. It provided by iParadigms that have introduced Turnitin in 1996 to become the online plagiarism detection. It is designed to be used by institutions rather than personal, but lastly they provided a limit service for single plagiarism detection user like master and doctoral students and this allows them to check a single document of up to 25,000 words. So they can use this service to insure or to check their draft thesis whether containing correct citation and content originality.

- The main features of iThenticate are:
  - Database Checking: iThenticate used its own database that contain millions of documents like (books, paper, essays, articles and assignments), with a large number of this documents that have been stored in iThenticate database locally, allowing the users who have account to do either online and offline comparison of submitted documents against it and to identify plagiarized content.
  - Internet Checking : iThenticate, is considered as the first online plagiarism checker that provides live and cached links to websites and database to have extensive internet checking to all submitted documents. This Provides deep internet checking. One more advantage is that it can still check your documents even if a website is no longer online, this include all contents of website like forums, message boards, bulletin boards, blogs, and PDFs etc., all this check is done automatically and in (almost) real-time.
  - Publications Checking: iThenticate offers an online and offline detailed and depth checking most types of publication like documents, including, books, articles, magazines, journals, newspapers, website and PDFs etc.
  - Synonym & Sentence Structure Checking: Not supported by iThenticate.
  - Multiple Document Comparison: iThenticate offers two types of document comparison document to document and multiple

documents checking against database and also direct source comparison word to word also.

- Supported Languages: iThenticate supports more than 30 languages, it mean that it supports most of languages likes ”English, Arabic, Chinese, Japanese, Thai, Korean, Catalan, Croatian, Czech, Danish, Dutch, Finnish, French, German, Hungarian, Italian, Norwegian, Polish, Portuguese, Romanian, Serbian, Slovak, Slovenian, Spanish, Swedish, Greek, Hebrew, Farsi, Russian, and Turkish.”.

3.3.6 Plagiarism Detection.org

PlagiarismDetection.org: an online service provides high level of accuracy result in plagiarism detection. Mainly designed to help the teachers and student to maintain and to ensure or prevent and detect plagiarism against their academic documents. It provides quickly detect plagiarism with high level of accuracy.

The main features of PlagiarismDetection.org:

- Database Checking: PlagiarismDetection.org used it own database that contains millions of documents like (books, paper, essays, articles and assignments).
- Internet Checking: PlagiarismDetection.org is an online plagiarism detector, so it is mainly based on the internet checking and is faster in plagiarism detection, it does not support offline detection.

- Publications Checking: PlagiarismDetection.org offers the students and teachers to check their publication against the published document and support most types of publication.
- Synonym & Sentence Structure Checking: PlagiarismDetection.org not supports Synonym & Sentence Structure Checking.
- Multiple Document Comparison: PlagiarismDetection.org does not support multiple document comparison but it takes long time to return the result.
- Supported Languages: PlagiarismDetection.org supports English languages and all languages that using Latin characters.

3.3.7 URKUND

URKUND is a practical tool for plagiarism control and for certifying the authenticity of professional texts. URKUND automatically checks texts against the Internet, archives and databases, reporting any similarities, and offers source track-back in an easy to operate analysis. URKUND is available via the web, via e-mail and through integrated with a number of common Learning Management Systems, such as Moodle, Blackboard, Fronter, SharePoint, PingPong, Vklass, It’s Learning and others.

Table 3.2 gives the comparison of the tools according to their features.

Table 3.2: Summarization of the comparison of tools according to their features:

Tools	Online and Offline check	Internet check	Publications check	Multiple document comparison	Multiple language support	Sentence structure and synonym checking
Turnitin	Excellent	Excellent	Excellent	Excellent	Excellent	Good
PlagAware	Excellent	Excellent	Excellent	Excellent	Excellent	Very Good
PlagScan	Excellent	Excellent	Excellent	Excellent	Excellent	Acceptable
iThenticate	Excellent	Excellent	Excellent	Excellent	Excellent	Very Good
URKUND	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent
CheckForPlagiarism.net	Very Good	Excellent	Good	Very Good	Excellent	Good
PlagiarismDetection.org	Very Good	Excellent	Good	Very Good	Excellent	Acceptable

Operation of plagiarism detection tools is based on statistical or semantical methods or both to get better results. Information about methods and algorithms which are applied in each particular tool is a business secret that is not revealed. From available descriptions of some detection tools it may be concluded that the great part of tools uses statistical methods to detect plagiarism, because these methods are well understood and they are easier to implement in software.

3.4 Competitions and Shared Tasks on Plagiarism Detection

This section presents an overview of the International Competitions on Plagiarism Detection. The first workshop of “Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection" (PAN) was held in conjunction with the 30th Annual International ACM SIGIR conference (Stein et al., 2007). The workshop focused on three tasks: 1) Plagiarism analysis, 2) Authorship identification and 3) Near-duplicate detection. The workshop acted as the pilot of the first PAN plagiarism detection competition in 2009. The findings of the workshop was that it is necessary to segment long texts in a document to chunks, and identified two main issues: 1) the lack of a benchmark corpus to evaluate plagiarism detection systems, and 2) the lack of an effective plagiarism detection tool that does not trade off computational cost with performance.

Therefore the main goal was to develop standard resources and evaluation measures to enable a direct comparison of different methods for plagiarism detection.

The two main tasks of each competition were: (1) extrinsic plagiarism detection and (2) intrinsic plagiarism detection. Since the focus of this work is on mono-lingual extrinsic plagiarism detection, only this task will be discussed in detail. The extrinsic plagiarism detection task included both mono-lingual and cross-lingual plagiarism. In the case of mono-lingual plagiarism, both the plagiarised and source texts were in English, whereas in the case of cross-lingual plagiarism, the source text was in either German or Spanish and the plagiarised text in English.

Each corpus was set up as (D_{susp};D_{src}; S), where D_{susp} represents the suspicious collection, D_{src} represents the source collection and S represents the annotations for plagiarism cases between D_{susp} and D_{src}. The plagiarism detection tasks were defined as (Stein et al., 2009).

- Extrinsic Plagiarism Detection

Given D_{susp} and D_{src} the task is to identify the sections in D_{susp} which are plagiarised, and their source sections in D_{src}.

- Intrinsic Plagiarism Detection

Given only D_{susp} the task is to identify the plagiarised sections.

In the 2010 competition, the intrinsic and extrinsic tasks were merged into a single task. (Barron-Cedeno and Rosso, 2008), described a preliminary experiment on external plagiarism detection using statistical language models on three aspects: word, POS and stem. Statistical language models trained on original words, part-of-speech of words and stemmed words provided a platform to analyse sequences of tokens. The result suggested that further experiments should combine the three aspects instead of analysing them separately. Creswick et al., (2008) described an indexing approach for information retrieval used for plagiarism detection. It pointed out the need to find the trade-off between precision and recall to suit various tasks. Lavergne et al., (2008) presented two approaches to distinguish natural texts from artificially generated ones, which can be applied in tasks such as detecting spam emails. The first approach used language models and the second focused on using relative entropy scoring, which gives higher weight to n-grams which exist in the Google's n-grams model.

The third PAN workshop on “Uncovering Plagiarism, Authorship and Social Software Misuse" was held in conjunction with the 25th Annual Conference of the Spanish Society for Natural Language Processing (Stein et al., 2009). The aims of the workshop remained the same as the 2008 workshop. Different from previous years, the workshop was co-organised with the first International Competition on Plagiarism Detection. The focus was shifted from bringing together theoretical research in the field to a more competitive development workshop. The competition consisted of two subtasks: external plagiarism detection and intrinsic plagiarism detection. There were a total of 13 groups participating in the competition. The competition was based on a large-scale artificially created plagiarism corpus and provided an evaluation framework for plagiarism detection. Nine groups entered in the external plagiarism detection task and three groups entered in the intrinsic plagiarism detection task, with one group entering in both tasks.

There has been a further increase in plagiarism detection research between 2010 and 2013. With the increased interest in plagiarism detection, plagiarism detection competitions have been continually organised to encourage development and evaluation of detection systems. The corpora used in the competitions were created with automatic insertion of texts from source texts to suspicious texts. Some of the cases involve translated plagiarism and some cases contain various levels of obfuscation, which are either artificial or manual text operations aiming to imitate paraphrasing. The evaluation is based on the standard metrics of precision, recall and F-score, and two specific metrics: granularity and overall score. Granularity measures the accuracy of the system in finding the exact plagiarised segments, and the overall score is combination of F-score and granularity. No baseline was set for the external detection task

Most of the participants in the competitions focused on external plagiarism. In the second competition (PAN-PC-10), there were only three systems which explored intrinsic plagiarism detection, with one system developed solely for intrinsic detection, and two systems developed for both external and intrinsic detection, compared to 17 external plagiarism detection systems. Although some levels of intrinsic detection using techniques from authorship identification and stylometry emerged in the competition, their accuracies are yet to reach a satisfactory level, as only one system performed better than the baseline, where the baseline assumed everything belongs to the plagiarised class.

In Nawab et al. (2010) attempt in the PAN-PC-10 competition, n-gram matching is used as the filtering metric and the Running-Karp-Rabin Greedy String Tiling algorithm is used in detailed analysis. The use of n-gram filtering is similar to the proposed framework in this study. One of the biggest challenges is the difficulty of finding a parameter that can specifically identify various levels of obfuscation in the text alignment stage, and also make sure that source documents are not overlooked in the filtering stage.

In the third competition (PAN-PC-11), seven systems participated in external detection, two systems participated in intrinsic detection, and two systems participated in both tasks. According to the organisers, the PAN-PC-11 corpus features plagiarism cases which are more difficult to detect, as it is clear that verbatim plagiarism does not pose enough of a challenge. Therefore, the PAN-PC-11 corpus features more manually or artificially obfuscated cases. The results from the competition show that there is a drop in performance, which indicates that obfuscation does pose a better challenge to plagiarism detection systems and that there are no good enough techniques that can tackle paraphrasing.

In the 2012 PAN workshop (Potthast et al., 2012), 15 teams participated in the external plagiarism detection task. Two sub-tasks are introduced, which include candidate document retrieval and detailed document comparsion. Seven teams re-used their systems from previous PAN competitions. New approaches to detect similarities in the detailed comparison stage include sequence alignment algorithms which are applied in the bioinformatics field. Other developments suggest that a one-fits-all approach is not ideal, but an adjustable approach poses a challenge to current research.

In general, the external plagiarism detection task participants in the workshop series can be summarised as taking a three-stage approach: pre-processing, detailed analysis and classification. The first stage, pre-processing, is done by processing the document collection using stopword removal, synonym replacement and stemming, then transforming the document into hashed word n-grams. The source documents are processed as an inverted index and compared with the suspicious documents by using a metric similar to the Jaccard coefficient. This filtering stage is essentially narrowing down the search span of suspicious-source document pairs. The second stage, detailed analysis, investigates the candidate suspicious source document pairs. This is usually done by using heuristic sequence alignment algorithms or similarity scores from n-gram overlap counts. The third stage, classification, aims to reduce the number of false positive detections. This is done by applying heuristics such as setting a minimum length of passage detected, or a threshold on the similarity score.

To conclude the approaches used in the PAN competition, it is found that most approaches employ brute-force pair-wise matching, and that the use of word 5-grams contributed to the winning approach in 2010 . The participants do not apply any deep natural language processing techniques to search for the deeper linguistic information, which is needed for handling paraphrases. Although some approaches employed shallow language processing techniques, the benefit of NLP in plagiarism detection is not used much.

Although precision of the PAN systems is very high, recall is generally low, with the exception of recall on verbatim copies which is higher. The competition indicated that manual obfuscation which includes paraphrases poses a far greater challenge than artificially obfuscated texts.

AraPlagDet.

AraPlagDet is the first international competition on detecting plagiarism in Arabic documents. The competition was held as a PAN shared task at FIRE 2015 and included two sub-tasks corresponding to the first shared tasks at PAN: external plagiarism detection and intrinsic plagiarism detection. The competition followed the formats used at PAN. One of the main motivations of organizers for this shared task was to raise awareness in the Arab world on the seriousness of plagiarism, and, to promote the development of plagiarism detection approaches that deal with the peculiarities of the Arabic language, providing for an evaluation corpus that allows for proper performance comparison between Arabic plagiarism detectors.

AraPlagDet, the first shared task for the evaluation of Arabic text plagiarism detection method comprises two subtasks, namely external plagiarism detection and intrinsic plagiarism detection. A total of 8 runs have been submitted and tested on the standardized corpora developed for the track. The methods used by the participants for extrinsic plagiarism detection and the evaluation corpora is discussed here.

The corpus for external plagiarism detection sub-task (ExAra-2015 corpus), was constructed using documents from the Corpus of Contemporary Arabic (CCA) and Arabic Wikipedia. The CCA involves hundreds of documents tagged with its topic. Two kinds of plagiarism cases, artificial and simulated were created. Phrase shuffling and word shuffling were used for creating the artificial or automatically created cases, and the simulated or manually created plagiarism was done using synonym substitution and paraphrasing.

The methods of 2 participants (Magooda, A., Mahgoub, A.Y., Rashwan, M., Fayek, M.B. and Raafat, H. 2015.) and (Alzahrani, S. 2015) are discussed here.

Magooda et al. in their three methods used the Lucene search engine and sentence based and keyword based indexing approaches. This enables their methods to be used with a large corpus , and also can be used online with a commercial search engine. They use two language dependent processing in the candidate retrieval phase: queries and stemmed and submitted to the search engine and named entities extraction. In the text alignment or detailed analysis phase, words are stemmed in the skip-gram approach. Pre-processing of the text is also done to remove diacritics and normalize letters.

Alzahrani's method is based on fingerprinting all the source documents, and requires a complete comparison between the n-grams of the suspicious document and each source document. Their method is feasible when the source documents are local to the system and the corpus is small as in the case of students' assignments. Since the only language-specific process done was stop words removal in the pre-processing step on both the source and suspicious documents, this approach is almost language independent.

The evaluation was done by detecting common word 5-grams between the source and suspicious documents and adjacent chunks with less than 800 characters between them are merged. Also, passages with less than 1000 characters are filtered out. Plagiarism cases that are not obfuscated are primarily detected.

Table 3.3 provides the precision, recall, granularity and plagdet score of the participants' methods as well as the baseline on the test corpus of AraPlagDet

Table 3.3 Performance of the AraPlagDet external plagiarism detection methods

method	precision	recall	granularity	plagdet
Magooda_2	0.852	0.831	1.069	0.802
Magooda_3	0.854	0.759	1.058	0.772
Magooda_1	0.805	0.786	1.052	0.767
Baseline	0.990	0.535	1.209	0.608
Alzahrani	0.831	0.530	1.186	0.574

The performance based on cases length, type of plagiarism and obfuscation are discussed here. All the three methods of Magooda et al. can detect cases with word shuffling obfuscation. Magooda_1 and Magooda_2 methods perform better than Magooda_3 in this aspect because the common words approach can identify similar passages irrespective of the order of the words. The fingerprinting approach used by Alzahrani is unable to detect shuffled words. Considering the case length factor, with medium length cases, all methods achieved good results. Although detection of manual paraphrasing and word shuffling cases remains a challenging task, high recall was achieved by all the methods in detecting cases without obfuscation.

PlagDet Task at AAIC.

The first competition on Persian plagiarism detection was held as the third AmirKabir Artificial Intelligence Competition (AAIC) in 2015. The competition was the first to plagiarism detection in the Persian language and led to the release of the first plagiarism detection corpus in Persian . The PAN standard framework on evaluation and corpus annotation has been used in this competition.

The following section describes the approaches of nine teams who participated in the Persian plagiarism detection competition, the Persian PlagDet shared task at PAN 2016. The task was to identify the similarity of text fragments between the pairs of suspicious document and the sources documents.

Mashhadirajab et al. (2016) build sentence vectors from both the source and suspicious documents, applying the vector space model (VSM) along with TF-IDF weighting scheme. SVM neural net is used to calculate the obfuscation type and adjust the required parameters. The synsets of terms are extracted from FarsNet to calculate the semantic similarity between sentences. Later, similar sentences are merged whereas overlapping passages or passages that are too short are removed. Their approach achieved the highest PlagDet score of 0.9220 on the complete corpus.

Gharavi et al. (2016) represent sentences of both the suspicious and source documents as vectors employing a deep learning approach. The vectors of words are extracted using Word2Vec and sentence vectors are computed as average of word vectors. The similarity between sentences are calculated using the cosine similarity and the Jaccard coefficient to identify plagiarism cases. Their approach achieved the highest PlagDet score of 0.9793 for corpus with no obfuscation. The runtime, to process the entire corpus is only 1:03 minutes.

Momtaz et al. (2016) split both the source and suspicious documents using sentence boundaries. Preprocessing steps like text normalization, removal of stop words and punctuations are also done. Next the sentences are converted to graphs, where nodes represent words and an edges correspond to its four surrounding words. The graphs thus obtained for both the source and suspicious documents are compared and their similarity computed If the similarities between two nodes is greater than the threshold, then that node is selected as the similar node. A sentence having similar nodes greater than the threshold is labelled as plagiarism. In the final step, the sentences close to each other are merged to create contiguous cases of detected plagiarism, thereby improving granularity.

Minaei et al. (2016) use n-grams to locate matches between suspicious and source documents. Direct cases of plagiarism and also cases of paraphrased plagiarism can be found this way. To identify the plagiarized passages, matches lesser than a specified threshold are merged. False positive cases are minimised by eliminating detected cases smaller than a pre-defined threshold. The runtime, to process the entire corpus is only 1:33 minutes.

Esteki et al. (2016) split documents into sentences and applied pre-processing steps like normalization, stemming and stop words removal. Similarity is computed using the Levenshtein distance, the Jaccard coefficient, and the Longest Common Subsequence (LCS). The use of synonyms is also identified to detect paraphrased sentences. A Support Vector Machine (SVM) is used to classify sentences as similar or not.

Talebpour et al. (2016) use trie trees to index the source documents after preprocessing. The steps involved are text tokenization, POS tagging, text normalization, removal of stop words and frequent words, and stemming. FarsNet is used to find the synonyms and synsets of the compared words to detect paraphrased plagiarism. After preprocessing, all the words of a source document and their exact positions are inserted into a trie. The same is repeated for all the source documents. Next, the suspicious document is iteratively analyzed, word by word against the trie to identify probable sources. Their approach achieved the highest precision of 0.638.

Ehsan et al. (2016) split both the source and suspicious documents using sentence boundaries. Each sentence is represented as a vector using the vector space model (VSM) along with TF-IDF weighting scheme. Sentence vectors with cosine similarity greater than a pre-defined threshold are marked as cases of plagiarism. Later matched sentences are merged whereas overlapping passages and extremely short passages are removed to improve performance.

Gillam et al. (2016) use an approach based on their previous PAN efforts. Similar text is identified without analysing the textual content directly. The content words and auxiliary words are identified and binary patterns are produced directly from these dependent words. This approach is similar to hashing functions, and uses the number of concurrent matches to measure similarity.

Mansoorizadeh et al. (2016) use the same method as and Ehsan et al. (2016). The difference is the lack of the merging stage . This method performs well in terms of granularity but the PlagDet score is very low.

Table 3.4 Persian PlagDet score of the approaches of nine participants based on obfuscation type.

Table 3.4: Persian PlagDet score of the nine approaches based on obfuscation type

Team	No obfuscation	Artificial obfuscation	Simulated obfuscation	Overall corpus
Mashhadirajab	0.9663	0.9440	0.8613	0.9220
Gharavi	0.9793	0.9301	0.8054	0.9059
Momtaz	0.9240	0.8999	0.7613	0.8710
Minaei	0.9060	0.8750	0.6422	0.8301
Esteki	0.9735	0.8530	0.5224	0.8008
Talebpour	0.9765	0.8149	0.5788	0.7749
Ehsan	0.7682	0.7557	0.6225	0.7266
Gillam	0.5221	0.4080	0.2876	0.3996
Mansoorizadeh	0.4080	0.4091	0.3082	0.3899

3.5 Chapter Summary

This chapter described the existing approaches to plagiarism detection. The works are classified as approaches based on the content of the text document, based on the structural information of the document or based on the referencing and citations. Some approaches have also incorporated NLP techniques in the detection process. Some of the commonly used automated tools available are also discussed. Finally competitions held for the task of plagiarism detection in English , Arabic and Persian languages are described.

Chapter 4

A FRAMEWORK FOR MALAYALAM NATURAL LANGUAGE PROCESSING IN PLAGIARISM DETECTION

• Contents •

This chapter describes the general framework used for this proposed plagiarism detection approach. The text pre-processing, NLP techniques used, similarity metrics used to measure the similarity between texts and the evaluation metrics which are used in this analysis are also described.

4.1 General Framework

The framework for our external plagiarism detection is shown in figure 4.1 which comprises of the following main stages.

(i) Candidate retrieval: Identify a small set of candidate documents from a large collection of potential source documents.

(ii) Pre-processing: This stage prepares both suspicious and source texts. Text pre-processing includes tokenisation, stop word removal, lemmatisation and other NLP techniques applied to the texts.

(iii) Text alignment: This stage compares each selected candidate source document to the suspicious document.

(iv) Similarity computation: Different similarity measures are applied in the text alignment stage to quantify the identified similarity.

(v) Classification: This stage uses the similarity scores from the previous stage to assign each text pair a classification as plagiarised or non-plagiarised. Classifications are verified by applying standard evaluation metrics which include precision, recall, f-score and accuracy.

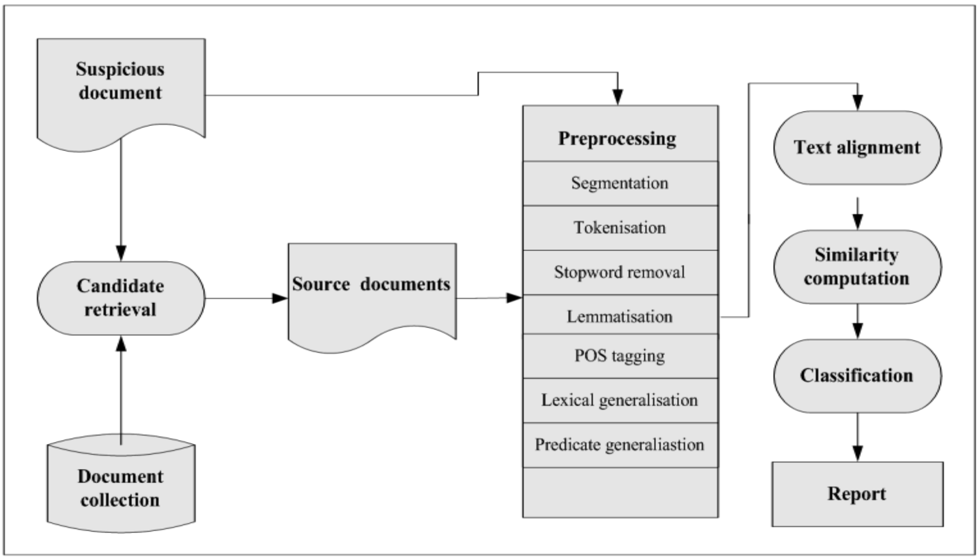


Figure 4.1: Proposed general architecture

This five-stage framework has been applied in the experiments described in Chapter 5.

4.2 Candidate Retrieval

A good candidate retrieval algorithm can lessen the document comparisons thereby decreasing the time complexity in the in-depth detection stage. In the Candidate Document Retrieval phase, an IR-based approach is proposed for retrieving candidate source documents.

The candidate retrieval process can be divided into the following steps: (i) pre-processing (ii) query formulation and (iii) retrieval.

(i) Pre-processing: The suspicious documents are split into sentences. The stop words are removed from each sentence and the remaining words in a sentence are lemmatised and synonym substitution is done.

(ii) Query Creation: Sentences from the suspicious document are used to make a query. The length of a query can vary from a single sentence to the entire document, because text copied for plagiarism can be obtained from one or more documents and the amount of text copied for plagiarism can vary from a single sentence to an entire document. A long query can perform well when large portions of text are copied for plagiarism. Similarly, a short query is likely to perform well for small portions of plagiarised text. Therefore, the length of the query is very significant to get good results.

(iii) Retrieval: Terms are weighted using the tf.idf weighting scheme.

$$tfidf_{i,d} = tf_{i,d}.idf_i = \frac{ni,d}{\sum_k n_{k,d}}.\log\frac{|D|}{|D_i|} \qquad (4.1)$$

Each query is used to retrieve relevant source documents from the source collection. The top N documents from result returned are merged to generate the list of source documents. It is likely that portions of portions of text from a single source document can be copied to different places in the same plagiarised document. Therefore, selecting the top N documents will lead to the original source documents appearing at the top of the final list of the documents.

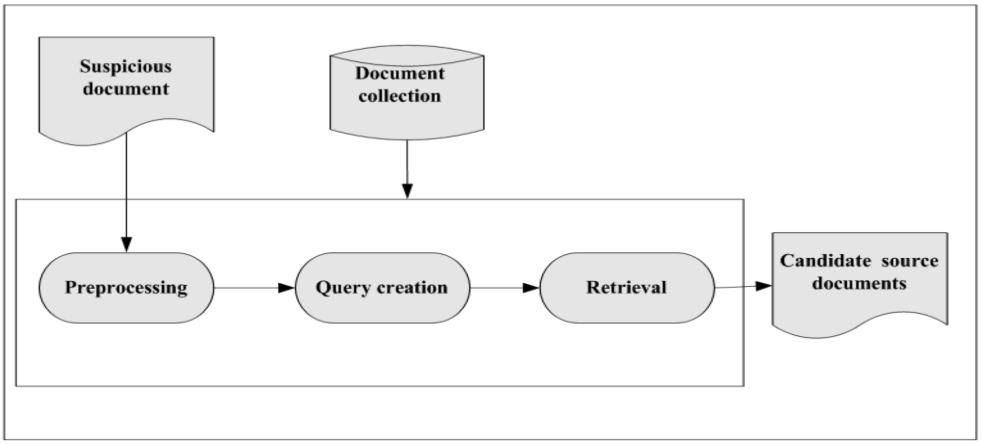


Figure 4.2: Candidate retrieval process

The final list of documents is obtained by combining the similarity scores of source documents retrieved against multiple queries. The final similarity score, $S_{\text{finalscore}}$, is obtained by adding the similarity scores of source documents obtained from each query (Fox and Shaw, 1994).

$$S_{finalscore} = \sum_{q=1}^{N_q} S_q(\text{d}) \tag{4.2}$$

Where N_q is the total number of queries to be combined and

$S_q(d)$ is the similarity score of a source document d for a query q. The top K documents in the ranked list generated by the $S_{finalscore}$, are selected as potential candidate source documents.

4.3 Natural Language Processing Techniques used for Text Pre-processing

This section describes the text pre-processing techniques and the NLP techniques used in our experiments.

4.3.1 Sentence segmentation:

This technique splits the text in the document into sentences, which allows sentence-by-sentence processing in the subsequent stages. For example:

Input text: സെമി കൊതിച്ചിറങ്ങിയ ഓസ്‌ട്രേലിയയെ വിരാട് കൊഹ്‌ലി എന്ന ബാറ്റ്സ്‌മാൻ ഒറ്റയ്ക്ക് കശാപ്പ് ചെയ്തു. വിരാട് കൊഹ്‌ലിയുടെ മാത്രം ബലത്തിൽ ഇന്ത്യ ഇരുന്നൂറ്റി അമ്പതു റൺസെടുത്തു. ഈ കളിയിൽ കൊഹ്‌ലിയുടെ റൺ റേറ്റ് എൺപത്തി രണ്ട് ആയിരുന്നു .

Text after sentence segmentation:

Sentence 1: സെമി കൊതിച്ചിറങ്ങിയ ഓസ്‌ട്രേലിയയെ വിരാട് കൊഹ്‌ലി എന്ന ബാറ്റ്സ്‌മാൻ ഒറ്റയ്ക്ക് കശാപ്പ് ചെയ്തു.

Sentence 2: വിരാട് കൊഹ്‌ലിയുടെ മാത്രം ബലത്തിൽ ഇന്ത്യ ഇരുന്നൂറ്റി അമ്പതു റൺസെടുത്തു.

Sentence 3: ഈ കളിയിൽ കൊഹ്‌ലിയുടെ റൺ റേറ്റ് എൺപത്തി രണ്ട് ആയിരുന്നു .

4.3.2 Tokenisation:

This technique determines token boundaries, such as words and punctuation symbols in the sentences. Example: For the above input text, tokenisation produces tokens as given in table 4.1.

Table 4.1: Example of tokenisation

Tokens	Tokens	Tokens
സെമി	വിരാട്	ഈ
കൊതിച്ചിറങ്ങിയ	കൊഹ്‌ലിയുടെ	കളിയിൽ
ഓസ്ട്രേലിയയെ	മാത്രം	കൊഹ്‌ലിയുടെ
വിരാട്	ബലത്തിൽ	റൺ
കൊഹ്‌ലി	ഇന്ത്യ	റേറ്റ്
എന്ന	ഇരുന്നൂറ്റി	എൺപത്തി
ബാറ്റ്സ്മാൻ	അമ്പതു	രണ്ട്
ഒറ്റയ്ക്ക്	റൺസെടുത്തു	ആയിരുന്നു
കശാപ്പ്		
ചെയ്തു		

4.3.3 Stopword removal:

This technique removes function words, which include articles, pronouns, prepositions, complementisers, and determiners, such as

ഈ, ഒരു, ആയ, അതേ, മറ്റ്, എന്നിവ, കൂടി, എന്നും, കൂടെ etc.

Input text after stopword removal:

സെമി കൊതിച്ചിറങ്ങിയ ഓസ്ട്രേലിയയെ വിരാട് കൊഹ്‌ലി ~~എന്ന~~ ബാറ്റ്സ്മാൻ ഒറ്റയ്ക്ക് കശാപ്പ് ചെയ്തു. വിരാട് കൊഹ്‌ലിയുടെ

~~മാത്രം~~ ബലത്തിൽ ഇന്ത്യ ഇരുന്നൂറ്റി അമ്പതു റൺസെടുത്തു. ~~ഈ~~ കളിയിൽ കൊഹ്ലിയുടെ റൺ റേറ്റ് എൺപത്തി രണ്ട് ~~ആയിരുന്നു~~ .

4.3.4 Parts of Speech (POS) tagging:

This technique assigns grammatical tags to each word, such as ”noun", “verb", etc., for detecting cases where words are replaced, but the style in terms of grammatical categories remains similar.

Input text:

(source) : രാമൻ കാട്ടിൽ പോയി

(plagiarised) : രാമൻ വനത്തിൽ പോയി

After POS-tagging:

(source) : രാമൻ [NOUN-NOM] കാട്ടിൽ [NOUN-LOC] പോയി [VERB]

(plagiarised) : രാമൻ [NOUN-NOM] വനത്തിൽ [NOUN-LOC] പോയി [VERB]

4.3.5 Lemmatisation

This technique transforms words into their dictionary base forms, which generalises the texts for similarity analysis.

Words in Malayalam have a strong inflectional component. For verbs these inflections are based on tense, mood, aspect etc. For nouns and pronouns inflections distinguish the categories of gender, number, and case. These inflections are called suffixes.

ALGORITHM (To find root form of a word):

Input : A word

Output : Root and grammatical features of input word

Uses : suffix table and root dictionary

Steps:

1. If input word is in root dictionary then return word with grammatical features, Stop.

   Else
2. Parse input word from right to left and find the longest suffix present in the suffix table.
3. Remove the identified suffix from the input word, concatenate the replacement string, if any, to form the root word.
4. If this new word is found in the root dictionary, return it along with its grammatical features.
5. Return.

Example:

Input word	word after lemmatisation
വനത്തിൽ	വനം
വനത്തിലേക്ക്	വനം

4.3.6 Lexical generalisation (synonym replacement):

Generalising words for word-level matching is done at this stage. Here, all synonyms of a word are retrieved and compared, making it possible to achieve a matching even if the plagiarised word has been substituted with another word of similar meaning. This approach was described in Chong et al. (2010); Chong and Specia (2011) where all synsets were selected. In this work synonyms are retrieved from the Malayalam WordNet which provides related groups of synonym words.

For the experiments with lexical generalisation, stopwords are removed and all remaining words are generalised using WordNet. WordNet lemmatises words and generates synsets for each content word. So, this technique expands the source and suspicious texts by replacing each content word by the words (synonyns) from WordNet. For each word in the source and suspicious documents, all the synsets are extracted.

WORD	:	സമുദ്രം
SYNONYMS	:	സമുദ്രം , കടല് , ആഴി , അകൂപാരം , അപാം പതി , അപ്പതി , അബ്ധി , അര്ണ്ണവം , ഉദധി, ജലനിധി , പാരാവാരം , സാഗരം
POS	:	noun

WORD	:	സാഗരം
SYNONYMS	:	സാഗരം , കടല് , സമുദ്രം
POS	:	noun

In the examples, although the words carry the same meaning, word-for-word matching metrics will not identify them as similar. By comparing the synsets of words, they are found to match with each other.

4.3.7 Predicate generalisation:

To analyse the grammatical components of a sentence, the predicate of a sentence, which can be represented by the verbs within it is analyzed. First extract the verbs in the document, then lemmatise to generalise the verbs to their base forms, and finally matching is done just as was done in the lexical generalisation using WordNet.

For example, for the verbs

WORD	: കൊടുക്കുക
SYNONYMS	: ഏല്പ്പിക്കുക , പിടിപ്പിക്കുക , കൊടുക്കുക
POS	: verb

WORD	: നല്കുക
SYNONYMS	: ഏല്പ്പിക്കുക , കൊടുക്കുക, പ്രദാനം ചെയ്യുക , കൈമാറ്റം ചെയ്യുക, സംഭാവന ചെയ്യുക
POS	: verb

In the examples, both the verbs carry the same meaning but word-for-word matching metrics will not identify them as similar. By substituting the synsets of words, they are found to match with each other.

4.3.8 Challenges faced during text preprocessing

Malayalam is a morphologically rich agglutinative language and is relatively of free order. Also Malayalam has a productive morphology that allows the creation of complex words which are often highly ambiguous. Due to its complexity, development of an NLP system for Malayalam is a tedious and time consuming task. No tagged corpus or tag set is available for this language. NLP systems developed in other languages are not suitable for Malayalam language due to its differences in morphology, syntax, and lexical semantics.

Malayalam being a less resource language, the above mentioned tools are not readily available as in English. Tools like morphological analyser, tagger , lemmatiser are to be developed. A list of Malayalam stop-words are also to be prepared.

4.4 Text Alignment

This stage compares each selected candidate source document to the suspicious document. Four models have been developed for this purpose namely: N-grams model, Fingerprinting based model, Semantic labelling based model and Probabilistic Neural Network (PNN) based model. These models are explained in chapter 5.

4.5 Plagiarism Corpus Construction

This section discusses the construction of plagiarism corpora. A plagiarism case occurs as a result of copying portions of text from a source document into another document. Due to the fact that word for word (exact or direct) copies can be detected easily, plagiarists often rewrite the source to obfuscate or conceal their fraudulent act. This behaviour of the plagiarist must be modelled when constructing a training corpus for plagiarism detection.

Potthast et al. (2010) introduce three levels of plagiarism authenticity, namely, real plagiarism, simulated plagiarism, and artificial plagiarism. Creating and using a corpus with real plagiarism is not feasible due to following reasons:

- Real plagiarism is often distributed in the document and hence it is difficult to detect.
- Approval from the original author and the plagiarist is required before using the real cases in constructing the corpus.
- A corpus with real cases is ethically and legally questionable.

Hence it is more practical to create simulated plagiarism cases. This is done by rewriting the original text with a different wording and phrasing, so that the rewritten version has the same meaning as the original.

The other possibility is to generate artificial plagiarism using some obfuscation strategies. Generating artificial plagiarism cases with semantic equivalence is a difficult task. Artificial plagiarism can be generated using three obfuscation strategies namely :

- Random text operations: Shuffling, removing, inserting, or replacing words or short phrases at random. Table 4.2 shows examples of these operations.

Table 4.2 Examples of random Text operations

Original text	രാമൻ കാട്ടിലേക്ക് പോയി
Shuffling words	കാട്ടിലേക്ക് രാമൻ പോയി
Deleting words	രാമൻ പോയി
Insertng words	രാമൻ ഇന്നലെ കാട്ടിലേക്ക് പോയി
Replacing words	രാമൻ നാട്ടിലേക്ക് പോയി

- Semantic word variation: Words are replaced by one of their synonyms chosen at random. A word is retained unchanged if none is available. Table 4.3 shows examples of this.

Table 4.3 Example of Semantic word variation

Original text	രാമൻ ഇന്നലെ കാട്ടിലേക്ക് പോയി
Synonym replacement	രാമൻ ഇന്നലെ വനത്തിലേക്ക് പോയി

- POS-preserving word shuffling where the sequence of parts of speech in original document is identified and plagiarism is created by shuffling words at random whereas the original POS sequence is retained. Table 4.4 shows examples of this.

Table 4.4 Example of POS-preserving word shuffling

Original text	രാമൻ [NOUN] കാട്ടിലേക്ക് [NOUN] പോയി [VERB]
POS-preserving word shuffling	കാട്ടിലേക്ക്[NOUN] രാമൻ [NOUN] പോയി[VERB]

4.6 Similarity Metrics

N-gram string This section describes the similarity metrics that are applied after the text has been pre-processed. Different similarity metrics are computed depending on the type and level of processing performed. The application of similarity metrics is essential to feature generation, as each feature consists of similarity scores generated by comparing processed text pairs, and the level of similarity for each suspicious-source text pair is determined by the similarity score.

The calculation of overlapping n-grams, either 2-grams or 3-grams, is a common approach to measuring similarity between texts. An n-gram represents n number of consecutive words. Similarity scores can be computed by counting the matching n-grams between the suspicious and source documents. Table 4.5 shows an example of overlapping 3-grams for a source and suspicious text.

Table 4.5 Example of overlapping 3-grams for a source and suspicious text

Source Text A :	എല്ലാവർക്കും സംഭവിക്കുന്നത് എനിക്കും സംഭവിക്കും.
3-grams in Text A	[എല്ലാവർക്കും സംഭവിക്കുന്നത് എനിക്കും] [സംഭവിക്കുന്നത് എനിക്കും സംഭവിക്കും]
Suspicious Text B :	എല്ലാ മനുഷ്യനും സംഭവിക്കുന്നത് എനിക്കും സംഭവിക്കും
3-grams in Text B:	[എല്ലാ മനുഷ്യനും സംഭവിക്കുന്നത്] [മനുഷ്യനും സംഭവിക്കുന്നത് എനിക്കും] [സംഭവിക്കുന്നത് എനിക്കും സംഭവിക്കും]

The example in table 4.5 contains two 3-grams in the source text and three 3-grams in the suspicious text. N-grams alone do not provide an indication of the level of similarity between two texts. Hence, similarity metrics are needed to calculate the similarity scores between the texts. A similarity metric basically counts the number of overlapping n-grams between texts, and the count is normalised according to the settings of the experiment.

Commonly used similarity measures are:

4.6.1 Jaccard similarity:

The Jaccard similarity co-effcient is a symmetric measure which treats the document pair to be compared as sets of n-grams. Jaccard similarity between two sentences is the ratio of the number of matches to the total number of unique words in both sentences.

If S_x and S_p are the sets of n-grams in the source and plagiarized documents then the Jaccard similarity co-efficient is given in equation (4.3).

$$S_{jaccard}(S_x, S_p) = \frac{S_x \bigcap S_p}{S_x \cup S_p} \text{......................................}(4.3)$$

4.6.2 Cosine Similarity:

The cosine similarity measure is commonly used for similarity calculation in text. Cosine similarity between the two sentences S_x and S_p is the ratio of the dot product of the sentence vectors to the product of their lengths as in equation (4.4).

$$S_{\cos ine}(S_x, S_p) = \frac{S_x \cdot S_p}{|S_x||S_p|} \text{......................................}(4.4)$$

4.6.3 Dice Similarity:

The Dice similarity between the two sentences S_x and S_p is the ratio of twice the number of shared/common tokens between the sentences to the sum of their lengths as in equation (4.5).

$$S_{dice}(S_x, S_p) = 2 \text{ x } \frac{|S_x \bigcap S_p|}{|S_x| + |S_p|} \text{...........................}(4.5)$$

4.7 Machine learning classifier

Once a set of features is selected, a machine learning model is then built to predict a class for each text pair as a binary classification as plagiarised or non-plagiarised. Machine learning allows the classification of text pairs based on a combination of features generated by more than one similarity metric, which enables a more flexible approach and is much more beneficial than classifying a text pair based on only one similarity metric with a predetermined threshold. The Probabilistic neural network (PNN) is used for the classification.

A PNN is primarily a classifier that maps any input pattern to a number of classifications. The probabilistic neural network is composed of many interconnected processing units or neurons organized in four successive layers. They are Input layer, two hidden layers called pattern layer and summation layer and an output layer. The input layer does not perform any computation and simply distributes the input to the neurons in the pattern layer It is an implementation of a statistical algorithm called kernel discriminant analysis.

4.8 Evaluation Metrics

The standard metrics of precision, recall, F-score and accuracy over the classification results are used for evaluation. The correctly classified plagiarised texts (True Positives: TP), correctly classified non-plagiarised texts (True Negatives: TN), non-plagiarised texts incorrectly classified as plagiarised (False Positives: FP), plagiarised texts incorrectly classified as non-plagiarised (False Negatives: FN) are used in the standard calculation of precision, recall, F-score, and accuracy as shown in the equations given in chapter 2.

4.9 Chapter Summary

This chapter described the general framework for the proposed plagiarism detection approach. The text pre-processing, and NLP techniques used were explained. The description of the techniques was followed by a list of similarity metrics that measure the similarity between texts and generate features to be used in the machine learning classifier. The chapter concluded with a list of the conventional evaluation metrics which are used in this analysis.

…….ജ്ഞ…….

Chapter

5 PLAGIARISM DETECTION MODELS FOR MALAYALAM

Contents

This chapter describes the different experiments performed with Malayalam documents based on n-gram string matching, fingerprinting, semantic role labelling method and the PNN model. Candidate retrieval process common to the four models have been described in chapter 4.

5.1 N-gram model for plagiarism detection in Malayalam

A plagiarism detection system for Malayalam text passages based on the n-gram Model is proposed. This model uses n-grams for representing the text. N-gram model was first used in text categorization based on the statistical information gathered from the usage of sequence of characters (Cavnar et al., 1994). N grams are consecutive overlapping characters formed from an input stream. N-gram means that token of n words are used for extracting the words from the passages and these n-grams are matched. Then the resemblance measures are computed for text categorization.

Comparison of word n-grams and character n-grams has been used by Lyon et al., (2001); Potthast et al., (2010, 2011); Stein et al., (2009) for plagiarism detection. However, this approach is not useful when the plagiarised text has been greatly paraphrased. It is found that insertion, deletion or substitution of even a single character (character n-grams) or word(word n-grams) in a text results in difference of at least one n-gram Ceska (2009). Based on this finding , if every nth token (character/word) in a text is changed by using any of the above mentioned edit operations then text copy will not be detected by the n-grams comparison. N-gram overlap method was used by Chen et al. (2010) for plagiarism detection where synonym-based and relationship-based measures were used to identify semantic similarity between a pair of words.

These experiments compute the degree of overlap between a pair of documents using the containment similarity measure (Broder, 1997), which is computed as:

$$C_n(A,B) = \frac{|S(A,n) \cap S(B,n)|}{|S(A,n)|} \qquad (5.1)$$

where S (A, n) and S (B, n) are the sets of word n-grams of length n in source and suspicious documents respectively. The containment similarity was also used by (Chong et al., 2010) for measuring plagiarism and obtained good results.

5.1.1 Modified n-grams

When the original text has been modified, it is difficult to detect text copy using the simple n-gram overlap approach. Consider the following example:

Source : രാമൻ കാട്ടിൽ പോയി

Plagiarised : രാമൻ വനത്തിൽ പോയി

The set of n-grams generated for the source and plagiarised texts do not match because the editing of the source text has reduced the similarity between the texts. Consequently, the plagiarism is not likely to be detected.

To detect text reuse created with paraphrasing, a modified n-gram approach is proposed. Two common text editing operations are deletions and substitutions (Bell, 1991). Hence in this approach, n-grams are created in two ways: (1) Deletions and (2) Substitutions. In the first approach, words in an n-gram are deleted, whereas in the second approach, words in an n-gram are substituted with synonymous words from a lexical resource. The methods used to generate modified n-grams are described below:

- Substitutions using WordNet

Modified n-grams are created by substituting one of the words in an n-gram with one of its synonyms from the WordNet. The modified n-grams created by substitutions are likely to identify semantic similarity between suspicious-source sets of n-grams. Consequently, the overall similarity score will increase and help in detecting text copy particularly when the original text has been paraphrased.

All the synsets of the word are used to generate modified n-grams. Synonymous words are selected from all senses because it will generate more modified n-grams as compared to choosing the first sense only. Each word in an n-gram is checked in WordNet. If found, all the synonyms from all senses are extracted as shown in table 5.1.

Table 5.1 Synonym substitution

Original text:	ഇന്ത്യ നമ്മുടെ നാട് ആകുന്നു
Text substituted with synonyms from word-net:	ഭാരതം നമ്മുടെ രാജ്യം ആകുന്നു ഭാരതം നമ്മുടെ മാതൃഭൂമി ആകുന്നു ഇന്ത്യ നമ്മുടെ രാഷ്ട്രം ആകുന്നു ഇന്ത്യ നമ്മുടെ ദേശം ആകുന്നു

5.1.2 Compare Modified N-grams

The modified n-grams are generated for the document that is suspected to be plagiarised. These n-grams are then compared with the original document to determine the similarity. Comparison between the documents is carried out by determining the proportion of n-grams in B which also occur as n-grams in A or as modified n-grams generated from A.

For each n-gram in A, the set of possible modified n-grams is created, denoted as *mod* (*ngram*). The original n-gram denoted as *ngram* is also included in *mod* (*ngram*). The modified count for the number of occurrences of an n-gram in A, *mod_count* (*ngram*, A), is then computed as the number of times it appears in *mod* (*ngrams*) as given in equation 5.2.

$$mod\_count(ngram, A) = \sum_{ngram' \in mod(ngram)} count(ngram', A) \quad (5.2)$$

The deletion and substitution operations generate large numbers of modified n-grams which can cause the number of shared n-grams to exceed the total number of n-grams in B and generating a score greater than 1. To take care of this, the overlap counts are limited by the number of times that n-gram appears in B. Therefore, the plagiarism detection score, $score_n$ (A, B), is computed as:

$$score_n(A, B) = \frac{\sum_{ngram \in B} \min(mod\_count(ngram, A), count(ngram, B))}{\sum_{ngram \in B} count(ngram, B)} \quad (5.3)$$

where mod_count(ngram, A) is the number of times an n-gram (ngram) in the set of modified n-grams mod(ngram) occurs in A and count(ngram,B) is the number of times ngram occurs in B.

5.1.3 Experimental Setup

This section describes the datasets and the evaluation methodology used to evaluate the proposed approach.

- Datasets

We have used passages from the standard Malayalam online newspaper articles as source documents and the rephrased ones of them as the suspicious documents. The documents used in experiments had average size of 200 words. N-grams for both the documents are calculated. To get the n-gram from the text, we process the text by following strategies:

Firstly, we divide the text into sentences. Secondly, the non-malayalam characters in the sentences were deleted. Finally, all the extracted sentences were divided into n-grams (n=2, 3, 4). Table 5.2 shows the bigrams(n=2), trigrams(n=3) and fourgrams(n=4) for the source text and table 5.3 shows the bigrams, trigrams and fourgrams for the suspicious text.

Table 5.2: Examples of n-grams with n=2,3,4 for source text

Source document	ഈ ദശാബ്ദത്തില് ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയിൽ വർധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും റിപ്പോര്ട്ട്.
Source bi-grams	(ഈ ദശാബ്ദത്തില്) , (ദശാബ്ദത്തില് ലോകത്തെ), (ലോകത്തെ മനുഷ്യരുടെ), (മനുഷ്യരുടെ ക്ഷണലഭ്യതയില്), (ഭക്ഷണലഭ്യതയില് വര്ധനവുണ്ടാകുമെന്നും), (വര്ധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ), (ദാരിദ്ര്യത്തിന്റെ തോത്), (തോത് കുറയമെന്നും), (കുറയമെന്നും റിപ്പോര്ട്ട്)
Source tri-grams	(ഈ ദശാബ്ദത്തില് ലോകത്തെ), (ദശാബ്ദത്തില് ലോകത്തെ മനുഷ്യരുടെ), (ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയില്), (മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയില് വര്ധനവുണ്ടാകുമെന്നും), (ഭക്ഷണലഭ്യതയില് വര്ധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ), (വര്ധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത്), (ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും), (തോത് കുറയമെന്നും റിപ്പോര്ട്ട്)
Source four-grams	(ഈ ദശാബ്ദത്തില് ലോകത്തെ മനുഷ്യരുടെ), (ദശാബ്ദത്തില് ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയില്), (ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയില് വര്ധനവുണ്ടാകുമെന്നും), (മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയില് വര്ധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ), (ഭക്ഷണലഭ്യതയില് വര്ധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത്), (വര്ധനവുണ്ടാകുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും), (ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും റിപ്പോര്ട്ട്)

Table 5.3: examples of n-grams with n=2,3,4 for suspicious text

Suspicious document	ഈ നൂറ്റാണ്ടിൽ ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയുടെ അളവ് കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും റിപ്പോര്ട്ട്.
Suspicious bi-grams	(ഈ നൂറ്റാണ്ടിൽ) , (നൂറ്റാണ്ടിൽ ലോകത്തെ), (ലോകത്തെ മനുഷ്യരുടെ), (മനുഷ്യരുടെ ക്ഷണലഭ്യതയുടെ), (ഭക്ഷണലഭ്യതയുടെ അളവ്), (അളവ് കൂടുമെന്നും), (കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ), (ദാരിദ്ര്യത്തിന്റെ തോത്), (തോത് കുറയമെന്നും), (കുറയമെന്നും റിപ്പോര്ട്ട്)
Suspicious tri-grams	(ഈ നൂറ്റാണ്ടിൽ ലോകത്തെ), (നൂറ്റാണ്ടിൽ ലോകത്തെ മനുഷ്യരുടെ), (ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയുടെ), (മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയുടെ അളവ്), (ഭക്ഷണലഭ്യതയുടെ അളവ് കൂടുമെന്നും), (അളവ് കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ), (കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത്), (ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും), (തോത് കുറയമെന്നും റിപ്പോര്ട്ട്)
Suspicious four-grams	(ഈ നൂറ്റാണ്ടിൽ ലോകത്തെ മനുഷ്യരുടെ), (നൂറ്റാണ്ടിൽ ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയുടെ), (ലോകത്തെ മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയുടെ അളവ്), (മനുഷ്യരുടെ ഭക്ഷണലഭ്യതയുടെ അളവ് കൂടുമെന്നും), (ഭക്ഷണലഭ്യതയുടെ അളവ് കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ), (അളവ് കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത്), (കൂടുമെന്നും ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും), (ദാരിദ്ര്യത്തിന്റെ തോത് കുറയമെന്നും റിപ്പോര്ട്ട്)

5.1.4 Results

In order to categorize the suspicious document as plagiarised or non-plagiarised, containment similarity scores for each suspicious-source document pair are computed for word bigrams, trigrams, and fourgrams. We have set a threshold of 50% resemblance as the threshold for classifying text as plagiarized. It is found that if a document has w words, then the number of bigrams, trigrams and fourgrams will be (w-1), (w-2) and (w-3) respectively. Hence for a trigram search there will be a maximum of (w1-2)*(w2-2) comparisons where w1 and w2 are the number of words generated from the original and the plagiarized document respectively.

The tri-gram model was compared with other n-gram models to asses the suitability of using tri-grams as the extracting word model. Plagiarism detection with bi-gram model is the maximum but the complexity of extracting and comparing bi-gram is also the maximum. The copy detection rate of four-gram model is the smallest as it compare longer sequences. The trigram model gives the average acceptable performance with affordable cost in terms of complexity and false positives.

Table 5.4 depicts the percentage of plagiarism detected for bi-gram, tri-gram and four-gram models for a source document R1 against six suspicious documents.

Table 5.4. Percentage of plagiarism detected using Bi-gram, Tri-gram and Four-gram similarity and corresponding accuracy

Source document	Suspicious document	Bi-gram similarity	Bi-gram accuracy	Tri-gram similarity	Tri-gram accuracy	Four-gram similarity	Four-gram accuracy
R1	S1	38.26	0.96	19.29	0.94	9.42	0.90
	S2	94.5	0.94	88.80	0.92	85.2	0.90
	S3	73.82	0.98	57	0.95	40	0.92
	S4	63.29	0.94	43.80	0.91	33	0.90
	S5	72.4	0.97	55	0.95	44.6	0.94
	S6	76.42	0.96	60.4	0.94	48.2	0.92

Where similarity>50% is Plagiarised.

The limitation of this method is that they do not consider the meaning of words, phrases, or sentence. However they can provide significant speedup when compared to semantic-based methods especially for large data sets since the comparison does not involve deeper analysis of the structure or the semantics of terms.

5.2 Fingerprinting technique for plagiarism detection in Malayalam

Fingerprinting is one of the most widely used approaches to plagiarism detection.

Fingerprinting has been used by for the retrieval of similar documents by Pereira Jr and Ziviani (2003) and for the identification of versioned and plagiarized documents by (Hoad and Zobel (2003)) and Finkel et al. (2002).

The technique of fingerprinting for plagiarism detection is based on the work of Manber (1994) and subsequent work by Garcia-Molina and Shivakumar (1995). Fingerprinting produces a compact description, called fingerprint, for each document in the collection (Broder et al.,1997, Manber 1994) without using term occurrences or frequency information. The fingerprint represents the content of the document, and, by comparing these fingerprints, it is possible to determine whether the documents are copied or not (Manber 1994).

5.2.1 Basic concepts

A document fingerprint is a collection of integers called “minutia” that represent the content of the document (Hoad, Timothy; Zobel, Justin (2003), Stein, Benno (2005)). Generally a fingerprint (minutia) is generated by selecting substrings from the text and applying a mathematical function (hashing function (Brin et al.,1995, Manber 1994)) to each selected substring. The minutiae are then stored in an index for quick access when querying. When a query document is compared to the collection, the fingerprint for the query is generated. For each minutia in the fingerprint, the index is queried, and a list of matching fingerprints is retrieved. The number of minutiae in common between each fingerprint in the collection and the query fingerprint determines the score of the corresponding document (Broder et al., 1997, Heintze 1996, Shivakumar and H. Garcia-Molina 1998).

While designing a fingerprinting process, the following factors need to be considered:

- The function used to generate a minutia from a substring in the document.
- The granularity - size of the substrings that are extracted from the document.
- The resolution - number of minutiae used to build a document fingerprint.
- The selection strategy - choice of the algorithm used to select substrings from the document.

Several methods have been proposed for fingerprinting, based on variations in these design parameters.

The fingerprint generation process must be reproducible (Manber 1994)] Every time a given string is processed, the resulting integer must be the same. Any hashing function satisfies this condition. The fingerprint generation function should produce as close as possible to a uniform distribution of integers (Heintze 1996). The minutiae produced lie between the bounds - 0 and a randomly high number such as 2^{32}. With any fingerprint generation function or hashing function where the set of possible strings is unknown, it is expected that some pairs of different strings share the same integer representation. Also, the function selected must be fast. Very few hashing functions satisfy all the above mentioned conditions (M.V. Ramakrishna and J. Zobel.1997). Unlike many functions which use multiplication extensively, the following algorithm is efficient as well as reproducible and acceptably uniform (M.V. Ramakrishna and J. Zobel.1997).

The Fingerprint granularity has a significant impact on the accuracy of fingerprinting (N. Shivakumar and H. Garcia-Molina.1996). The granularity can be specified by the number of characters in the string (Manber 1994), or the number of sentences (N. Shivakumar and H. Garcia-Molina.1995), or the number of words in the string (Heintze 1996). Selecting a fine granularity makes the fingerprint prone to false matches, whereas selecting a coarse granularity makes the fingerprint too sensitive to change (N. Shivakumar and H. Garcia-Molina.1995).. As the granularity becomes larger, the likelihood that two documents sharing a minutia actually share the same phrase becomes smaller. With a coarse granularity, a large proportion of matches will occur. But, considering a granularity of one word and two documents that share a minutia, it is probable that these documents do actually share the substring used to generate the minutia.

Fingerprint resolution has a proportional impact on the processing required to evaluate the query, and the space required to store the index. The fingerprint resolution may be fixed or variable and may also be determined by the size of the document (Heintze 1996).

Substring selection strategy can affect both the accuracy and efficiency of the fingerprinting process (N. Shivakumar and H. Garcia-Molina.1996). There are different ways for substring selection for both variable and fixed fingerprint resolution. There are four classes of selection strategies: full fingerprinting, positional selection strategies, frequency-based strategies, and structure-based strategies.

Full fingerprinting is the simplest selection strategy, where every substring of size *g* in the document is selected (where *g* is the fingerprint granularity). This strategy produces the largest possible fingerprint resolution for the document. This method is suitable for the query, since only one fingerprint must be produced for the query, and this fingerprint does not need to be stored permanently. Because it uses every substring of the document, full fingerprinting is very effective.

Positional selection is a class of simple strategies that select phrases based on the offset from the beginning of the document.

Random substring selection is suited to fixed resolution fingerprinting. A fixed number of substrings are selected at random from the document. All substrings selection is similar to full fingerprinting, but selects all non-overlapping substrings of size g from the document rather than overlapping sub-strings. This strategy is suited to variable resolution. First-r selection is suited to a fixed resolution. This strategy selects the first r non-overlapping substrings of length g from the document, where r is the resolution and g is the granularity. First-r-sliding selection is similar to the first-r strategy, with the difference that it is based on overlapping phrases.

Frequency-based strategies. These select phrases based on their frequency. The intuition is that phrases that are less common are more effective discriminators when comparing documents for similarity.

Rarest-in-document selection chooses the substrings that produce the rarest minutiae in the document. This means that all of the minutiae must be calculated and sorted according to the frequency in the document, then the rarest r of them selected. This strategy suffers from the problem that many of the minutiae will appear only once in any one document, resulting in only the most common substrings being eliminated; the selection would then fall to the first r substrings that are not repeated in the document. We have not tested this strategy.

Rarest-in-collection selection requires generation of all the minutiae for all documents. They are then sorted according to the frequency of the minutia in the collection, rather than the frequency in the document. The rarest r minutiae are selected. This strategy is intended to reduce the number of coincidental matches caused by the matching of common phrases. We have not tested this strategy.

Rarest prefix selection begins by finding all distinct p-character strings that form the start of a word. For each of these strings, the number of occurrences in the document is counted. The r substrings beginning with the rarest prefixes in the document are selected. This approach is suited to a fixed fingerprint resolution.

Structure-based strategies. These strategies use the structure of the document. This allows detection of co-derivatives after changes in word positions that can affect the positional strategies, and changes in word or minutia frequencies which can affect the frequency-based strategies.

Anchor selection works by locating specific, predefined strings, or anchors, in the text of the document. The anchors are chosen to be common enough that there is at least one in almost every document, but not so common that the fingerprint becomes very large (Manber 1994).

K^{th}-sentence selection chooses phrases beginning at the start of every Kth sentence in the document. It can be used for both fixed and variable fingerprint resolution.

A text plagiarism detection process is a pair wise comparison. Given a pair of documents, the amount of text copied between the two documents is to be estimated. The amount of text of document A that is shared with document B can be represented as a ratio of the number of shared fingerprints to the number of fingerprints of document A. The containment of A in B is estimated as given below (A. Z. Broder. 1997):

$$C(A,B) = \frac{|\mathrm{F}_A \cap F_B|}{|F_A|} \qquad (5.4)$$

where F_A and F_B are sets of fingerprints of document A and B, respectively.

The main steps involved in plagiarism detection using fingerprinting based model is the candidate retrieval, preprocessing, fingerprint generation, similarity calculation and classification. Figure 5.1 shows the architecture of fingerprinting based model.

In the pre-processing phase the document is broken up into tokens or words, stop words are removed, words are converted to their base form and synonym replacement is done.

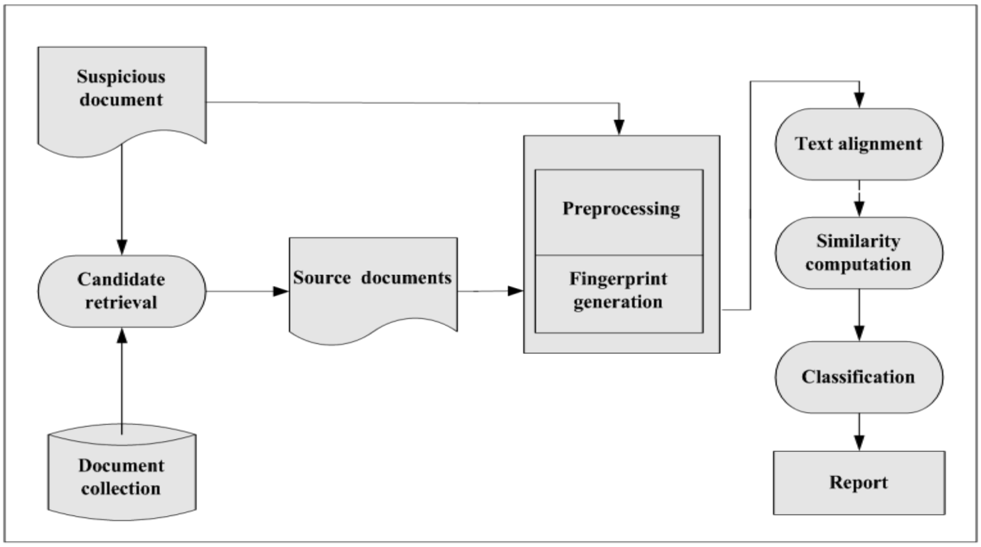


Figure 5.1 Architecture of fingerprinting based model

Fingerprinting: The fingerprints of a document are numerical representations for text reuse detection and, for local reuse detection, should represent as much as possible of the content of the document. For efficient plagiarism detection, an inverted index is built with fingerprints extracted from documents. To find all documents which have text copy with a document A, we first read all inverted lists of the fingerprints of document A, then merge the lists, and finally, find similarity. Since the maximum length of the inverted list is the number of documents in the collection, this is O(M^n) algorithm, where M and n are the number of the fingerprints of document A and the number of documents in the collection, respectively.

Good fingerprinting techniques for text reuse detection should satisfy the following properties.

- For accuracy, a fingerprinting technique should generate fingerprints that accurately represent documents.
- For efficiency, a fingerprinting technique should generate the smallest number of fingerprints possible.

5.2.2 Overlap Methods

Overlap methods use a sliding window. Basically, the window is shifted by a word, and a word sequence in the window or its hash value is handled as a chunk. If the size of the window is k, i.e. the ith window contains the ith word to the (i+ k −1)th word in the document, then the ith chunk in document D is computed as follows:

$$C(D,i) = h(t(D,i), t(D,i+1),...,t(D,i+k-1) \qquad (5.5)$$

where h is the the hash function and t(D, i) is the ith term in document D. Even though overlap methods show good performances, processing a large number of chunks as fingerprints is computationally expensive. Thus chunk selection techniques are used (N. Heintze.,1996, U. Manber.1994, N. Shivakumar and H. Garcıa-Molina, 1995).

- *k-gram*

k-gram is the simplest technique of the overlap methods. It uses all the chunks generated from each sliding window as fingerprints. Thus, the number of the fingerprints of document D is computed as follows:

$$M_{k\text{-gram}}(D) = L(D) - k + 1 \qquad (5.6)$$

where L(D) is the term count of document D. As k-gram uses all chunks, it generally shows good performance. However, it might be infeasible in big collections because of too many fingerprints.

- *0 mod p*

Instead of using all the chunks generated by the sliding window, 0 mod p tries to select some of them as fingerprints. A random selection of chunks would reduce the number but we cannot predict which chunks would be selected. If different chunks are selected each time, then two documents may be determined to be different even when they are identical. Therefore, all chunk selection methods have to satisfy the property that the same chunks should be selected for identical documents.

0 mod p selects only chunks such that C(D, i) mod p $\equiv$ 0. When two documents are identical, chunks in the documents are the same. Assuming that the chunk values are uniformly distributed, the expected number of selected chunks, i.e. the number of fingerprints of document D, is given by:

$$M_{0 \bmod p} = M_{k-gram}(D) / p \qquad (5.7)$$

That is, 0 mod p can reduce the number of the fingerprints by a factor p.

- *Winnowing*

Winnowing is another selection method based on k-gram. Winnowing adopts a winnowing window (window of fixed size) over the sequence of chunks generated by the original window, and it selects a chunk whose value is the minimum in each winnowing window. If there is more than one minimum value in the winnowing window, then the rightmost minimum value in the window is selected. Schleimer et al.() showed that winnowing performs better than 0 mod p in practice. Further, they showed that the expected number of fingerprints has a lower bound as follows:

$$M_{winnowing}(\mathrm{D}) = \frac{2}{w+1} M_{k-gram}(D) \qquad (5.8)$$

where w is the size of winnowing window.

*S*imilarity comparison:

The comparison step takes each pair of documents that needs comparison and looks for matching phrases. The comparison process basically looks at the two numerical-ordered lists of hash codes. The algorithm has a pair of counters, one for each document's numerical-ordered list of hash codes. The algorithm advances those counters through the numerical-ordered lists until it comes to a pair of identical hash codes—representing a matching pair of words. Since the hash code lists are in numerical order, the algorithm finds those matching pairs quickly and efficiently. The algorithm makes only one pass through each of those numerical-ordered lists.

Whenever the algorithm finds a pair of matching hash code in its numerical-ordered lists, the algorithm then looks the document-ordered list of hash codes. The document-order word number is attached to each hash code in the numerical-ordered list. The algorithm then searches through the document-order list of hash codes to find the matching phrases around those matching hash codes. When the longest available matching phrases are found, it records data about those matching phrases in the document-ordered lists of hash codes. It then removes the words in those matching phrases from further matching in this pair of documents.

When there are multiple copies of a certain word in one or both of documents, the algorithm checks for matching phrases around each possible pairing of those duplicate words. The algorithm checks all of the matching pairs of words in the numerical-ordered lists of hash codes, checking each matching word pair to see if it is part of a matching phrase pair. This is done until it reaches the ends of the two numerical-ordered lists of hash codes. The comparison of the document pair is then complete.

```
Algorithm  :  Document similarity
Input      :  DocA, DocB
Output     :  Similarity between DocA, DocB
   Step 1  :  MinDocSize = min (|DocA|, |DocB|)
   Step 2  :  IntersectionDocSize = |DocA ∩ DocB|
   Step3   :  If(IntersectionDocSize>=MinDocSize*DocThreshold)
              goto step 4  else goto step5
   Step4   :  similarity = true, goto step6
   Step5   :  similarity = false
   Step6   :  Stop
```

Based on a fixed threshold the two documents are compared and the number of hashes in the intersection subset exceeds the threshold, then both documents are found to have similar content.

5.2.3 Performance Evaluation of fingerprinting model

A. Dataset

20 documents were extracted from the online Malayalam newspaper Mathrubhumi. From the original documents, plagiarized documents were generated as follows:

- half of the total number of words in each document was replaced randomly with their similar words. Stop-words were avoided.
- by changing the sentence structure of some sentences. Such sentences amounts to half of the entire number of sentences.
- by copying randomly selected sentences, substituting some words with one of their synonyms, and changing the structure of selected sentences.

The different fingerprinting techniques- k-gram , 0 mod p, and winnowing were evaluated with the value of k was chosen as 3 for k-gram, the value of p was chosen as 6 for 0 mod p, and the value of w was chosen as 10 for winnowing. It is found that k-gram generated too many fingerprints, but gave most accurate results . So k-gram method can be considered best when the document collection is small. For detecting patial copies, winnowing gives better results. Table 5.5 and figure 5.3 depicts the precision, recall and F-measure while using the k-gram, 0 mod p and winnowing overlap methods.

Table 5.5 Comparison of results of fingerprinting methods

Method	Precision	Recall	Fmeasure
k-gram	1.00	0.93	0.96
0 mod p	0.88	0.90	0.89
winnowing	0.92	0.90	0.90

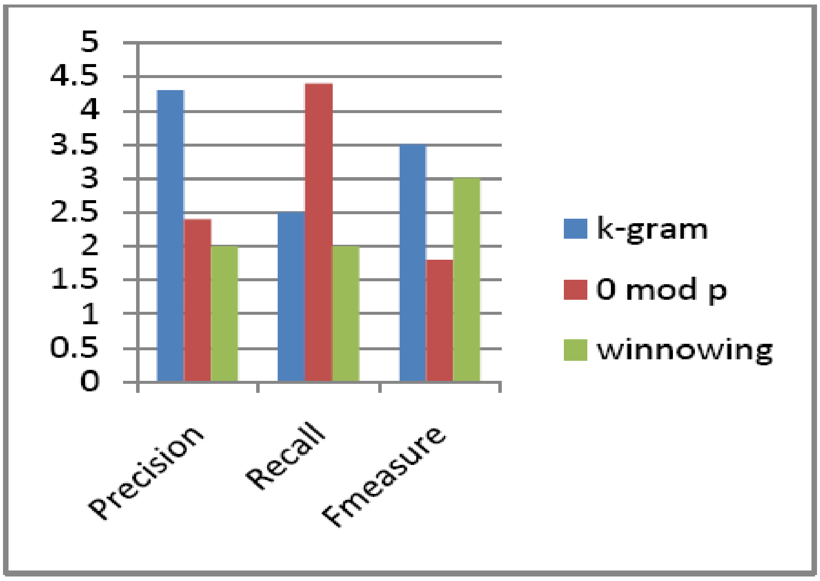


Figure 5.2 Comparative results of fingerprinting methods

5.3 Semantic role labeling based model

5.3.1 SRL for Malayalam language

Malayalam, is both an agglutinative as well as an inflectional language. Based on the tense, number, gender etc, the root word is inflected to produce new words. These features of inflection and agglutination makes computer based Malayalam language processing a challenging task. During the semantic analysis, verb is taken as the fundamental, required element of the sentence. Panini, the Sanskrit grammarian used this idea in his grammar. Accordingly, the relation of a noun to the verb in a Malayalam sentence is called kaaraka. The system implemented makes use of this relation between vibhakti and kaaraka roles in Malayalam sentences. Kaarakas provides the necessary information relative to a verb by giving the relations between the nouns and the verbal root. Kaaraka is a relation between a verb which denotes an action and nominals in the sentence. So, the verb determines the karaka of nominal words used in a sentence. Verbs are related to nominal words in different ways based on which the karaka differs. So, for any verb, different kaarakas may occur. Based on the semantic relation between the nouns and verbal root, the Kaaraka relations are identified. So, the syntactic-semantic relationship between the different words of the sentence is provided by the Kaaraka relation . Following Panini's theory ,six kaarakas are defined for Malayalam based on the noun's relation to the verb. The karakas are as follows:

k1 : kartaav (subject): actor of the verb

k2 : karma (object): the one most necessary for the Kartaav

k3 : Karanam (instrumental): instrument essential for the action to take place

k4 : swami (dative): recipient of the action

k5 : sakshi (sociative): movement away from a source

k6 : adhikaranam (locative): location where the action occurs

Any action can thus be represented as a function of verb(k1, k2, k3, k4, k5, k6) which means that a verb is related to nominal words on the basis of these six aspects. Syntactically noun phrases are can appear as subjects, direct or indirect objects and compliment of postpositional phrases.

Malayalam is a comparatively free word order language . It is a verb final language and normally all the noun phrases in the sentence appear to the left of the verb. The subject noun phrase may also appear in many different positions with relation to other noun phrases in the sentence. This can be easily illustrated with the example ‘Mother gave the child an umbrella.’

അമ്മ കുട്ടിയ്ക്ക് ഒരു കുട കൊടുത്തു

കുട്ടിയ്ക്ക് അമ്മ ഒരു കുട കൊടുത്തു

കുട്ടിയ്ക്ക് ഒരു കുട അമ്മ കൊടുത്തു

ഒരു കുട അമ്മ കുട്ടിയ്ക്ക് കൊടുത്തു

In all the cases, the subject is അമ്മ (Mother) , the object is കുട (umbrella) and the dative (indirect object) is കുട്ടി (child). From the above example, it is clear that word order does not determine the functional structure in Dravidian languages especially Malayalam and permits scrambling. This mapping between vibhakti and kaaraka roles in Malayalam sentences is made use of in this implementation.

5.3.2 Vibhakthi to Kaaraka mapping

Case endings differentiate the vibhakthis. This is illustrated in table 5.6. In the first step, obtain the vibhakthis from the tokens of the given text. In the second step, the corresponding kaarakas are obtained by mapping using Table 5.7.

Vibhakati in malayalam are of seven types nirdesika (nominative), prathigrahika(accusative), samyojika(sociative), uddesika (dative), prayojika (instrumental), sambandika (genitive) and , aadhaarika(locative).

Table 5.6 Vibhakthi-case endings (suffix)

Vibhakthi (Case)	Example	Suffix
Nirdesika (Nominative)	അമ്മ	nil
Prathigrahika (Accusative)	അമ്മയെ	എ
Samyojika (Sociative)	അമ്മയോട്	ഓട്
Udesika (Dative)	അമ്മയ്ക്ക്	ക്ക്
Prayojika (Instrumental)	അമ്മയാൽ	ആൽ
Sambhandika (Possesive)	അമ്മയുടെ	ഉടെ
Aadharika (Locative)	അമ്മയിൽ	ഇൽ

Table 5.7 Vibhakthi-Kaaraka relation

Kaaraka	Vibhakthi (Case)
Subject കർത്താവ്	Nirdesika (nominative) നിർദ്ദേശിക
Object കർമ്മം	Prathigrahika (accusative) / Nirdesika പ്രതിഗ്രാഹിക
Instrument കരണം	Prayojika (instrumental) പ്രയോജിക
Indirect Object സ്വാമി	Udesika (dative) ഉദ്ദേശിക
Agent സാക്ഷി	Samyojika (sociative) സംയോജിക
Location അധികരണം	Aadharika (locative) ആധാരിക

Subject (Karthaav):- Subject of the sentence has nirdesika (nominative) as Vibhakthi in active voice.

Eg. Ramu vannu. (Ramu came.)

Object (Karmam):- Object of the sentence has Prathigrahika (accusative) as Vibhakthi in active voice.

Eg. Avan Ramuvine adichu. (He beat Ramu)

Indirect Object (Saakshi):- It denotes the indirect object or somebody else who is participating in the action together with the subject. It has Samyojika (sociative) as vibhakthi Eg. Avan Ramuvinodu oru katha paranju. (He told Ramu a story)

Agent (Swaami):- If the verb is not intended for the subject, the other noun that get involved is the Swami Kaarakam. The vibhakthi of this noun will be Uddesika.

Eg. Avan oru pena Ramuvinu koduthu.(He gave a pen to Ramu)

5.3.3 Plagiarism Detection Using SRL

Two text units are found as similar if they share the same focus on a common idea, actor, object, or action. In addition, the common actor or object must perform or be subjected to the same action, or be the subject of the same description. In this Section, we discuss the architecture of our proposed method.

First the suspected documents and original documents are pre processed using text segmentation, eliminating commonly occurring words or stopwords and reducing words to their lemmas or lemmatization. Then, semantic role labelling transforms the sentences into arguments of the verb based on the kaaraka – vibhakthi relation. Such arguments obtained from the text were grouped according to the argument type as kartaav (subject), karma (object), Karanam (instrumental), swami (dative), sakshi (sociative), and adhikaranam (locative).

Figure 5.3 shows the architecture of the proposed system.

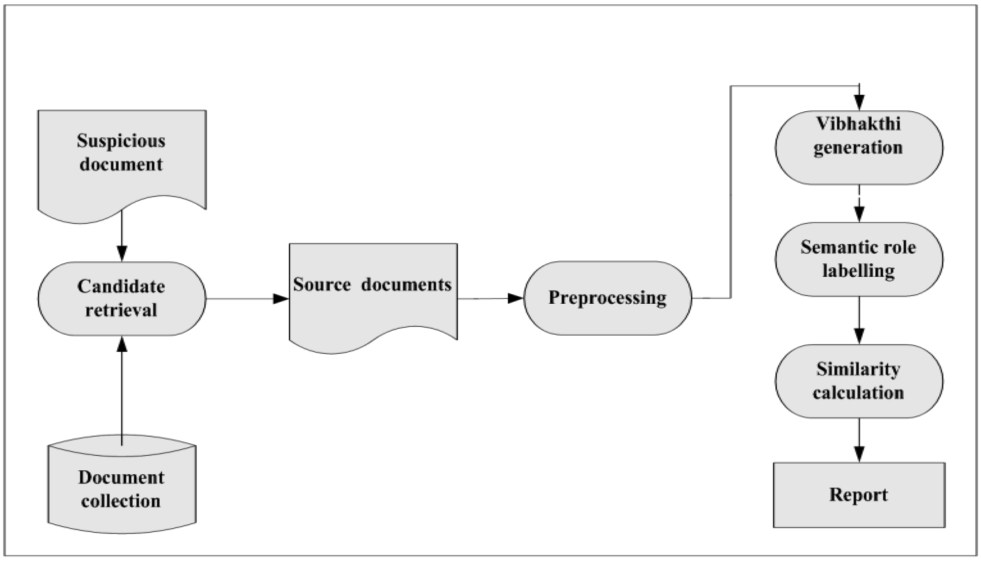


Figure 5.3: Architecture of SRL based model

Pre-processing is an essential step in Natural Language Processing tasks. Text segmentation, stop words removal, Lemmatization and POS tagging

(a) Vibhakti generation

This step classifies the words according to their vibhakthi(case). A noun may belong to one of the seven cases namely, nirdesika(nominative), prathigrahika(accusative), samyojika(sociative), uddesika(dative), prayojika (instrumental), sambandika (genitive) and , aadhaarika (locative).

(b) Semantic role labelling

Based on the vibhakthi – Kaaraka relation, the word is tagged as kartaav (subject), karma (object), Karanam (instrumental), swami (dative), sakshi (sociative), and adhikaranam (locative).

Malayalam has free word order and the case is determined based on its inflections and not the position of the word as in English.

(c) Similarity detection

In this step, sentence-based similarity analyses between the suspected and original documents are performed. Sentences in suspected documents are compared with each sentence in the candidate documents according to the verbs of the sentences. If verbs or their synonyms of the sentences match, then the corresponding arguments or their synonyms are compared. This leads to a decrease in the number of comparisons because each argument in suspected sentence will only be compared with a similar argument in original sentence.

```
Algorithm1 Algorithm for the similarity check:
Input: Source document and suspected document
Output: Plagiarism report
1.  Extract sentences from document
2.  For all sentences in the document do step3 to step8
3.  Tokenization of the sentences
4.  Stop words removal from the tokens
5.  Lemmatization to find root forms of the tokens
6.  Obtain the syntactic-semantic relation of the roots
7.  If the root verb or the synonym of the verb is found to match
    that of the source document, that sentence becomes a
    candidate for similarity checker.
8.  Calculate sentence similarity
9.  If similarity of sentence > threshold, tag sentence similarity
    as 1 otherwise as 0
10. Check all sentences and obtain text similarity
11. Classify document as plagiarized or not.
```

The similarity metrics used

i. Jaccard similarity measure

ii. Cosine similarity measure

iii. Dice similarity

5.3.4 Data Set and Experimental Results

Experiments were conducted to determine the amount of plagiarized sentences based on the sentences from the original document. A corpus for plagiarism detection is not available in Malayalam .A total of 80 plagiarised documents were used for the experiments. Each plagiarized document was created manually from 10 original documents collected from articles of online Malayalam newspapers. The plagiarised documents included different levels of plagiarism like direct copy and paste, modifying words with synonyms, inserting new words into the sentence, deleting words from the sentence, altering the structure of the sentences by reordering the words in the sentence and also changing the voice . (active to passive voice or vice-versa). The verbs of the corresponding sentences were compared first . If they are found to be matching, the corresponding arguments from the plagiarised and original documents are checked for similarity.

The evaluation is based on the standard metrics of precision, recall, accuracy and F-score.

Figure 5.4 gives the comparison between the proposed method with SRL-based similarity and semantics similarity

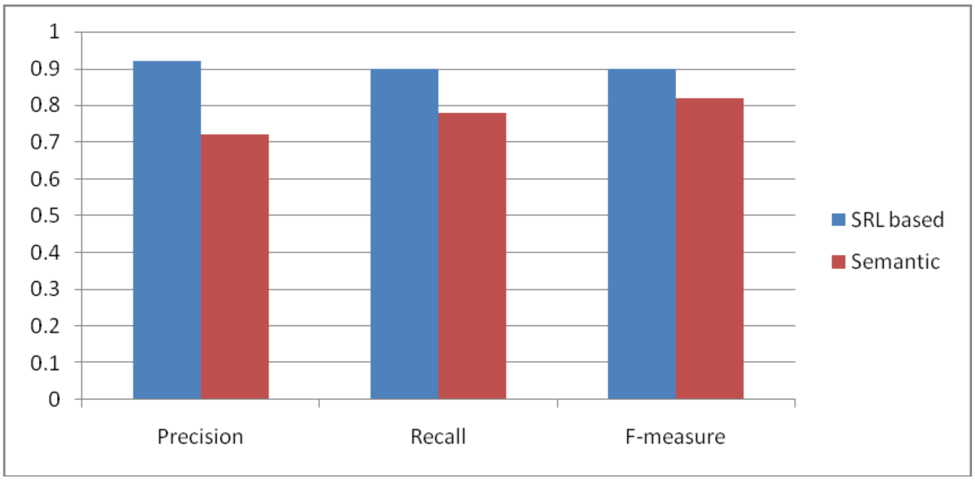


Figure 5.4: Comparison of SRL and semantic similarity based on precision, Recall, F-measure

5.4 PNN based model for plagiarism detection in Malayalam

This section analyses how different similarity metrics can be combined using a PNN for classifying Malayalam documents as plagiarised or not.

5.4.1 Basic concepts

Identifying plagiarised documents is basically a classification problem where a given document needs to be assigned to a class namely plagiarised or not plagiarised. Artificial neural networks, usually called neural networks, emerged as an important tool for classification. Neural networks are simplified models of the biological nervous system which consists of highly interconnected network of a large number of processing elements called neurons in an architecture inspired by the brain (Rajasekaran and Pai, 2009). Neural networks learn by examples. They can be trained with known examples of the problem. Once appropriately trained the network can be put in effective use in solving unknown or untrained instances of the problem. In this model classification is based on the values of the similarity metrics for a pair of documents. This model is used to classify new documents.

5.4.2 Architecture of PNN based model

The proposed work is divided into different phases as shown in figure 5.5. The first phase is candidate retrieval which involves identifying the possible source documents from which the suspicious document has copied text. The second phase comprises of different steps involved in preparing the texts to be compared. Shallow NLP techniques are used in this phase. The third phase is the similarity calculation using Jaccard, Dice and Cosine similarity co-efficient. Finally Probabilistic Neural Network is used to classify the text.

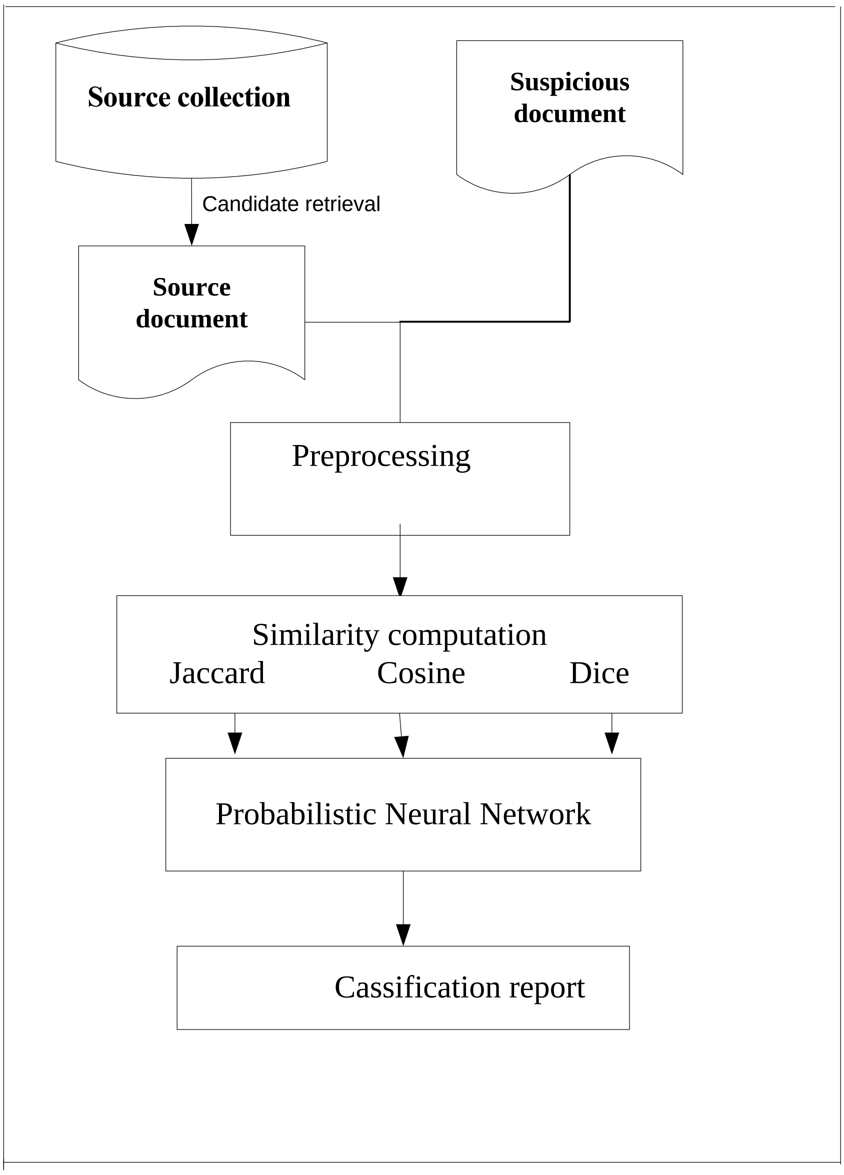


Figure 5.5: Architecture of PNN model

A. Pre-Processing of documents

Given the suspicious and source document sets, pre-processing steps are done to eliminate noise and present the documents in a form suitable for comparison. The pre-processing steps done here are tokenization and stop-word removal. Documents are tokenized in order to transform them into word n-grams. This facilitates the use of overlapping n-grams as features in the similarity measurement. Stop-words have low discerning power in the semantics of a sentence and hence are removed.

Malayalam is an inflectional language and nouns and verbs may appear in different forms due to the addition of suffixes. Hence the tokens from the previous stage may fail in detecting similarity if appearing in an inflected form. This requires these tokens to be converted to their root form or the lemma.

Lemmatization uses suffix stripping rules and the Malayalam wordnet. POS tagging is also combined in this stage.

B. Similarity computation

The Vector Space Model (VSM) is used for identifying plagiarized documents. The source and suspicious documents are converted to vectors, and comparison is done at sentence level.

For calculation of similarity the Jaccard similarity co-efficient, cosine similarity co-efficient and the Dice co-efficient is used. The non-matching n-grams are replaced with their synonyms to check for similarity.

Table 5.8 gives examples of two pairs of sentences (1a and 1b) and (2a and 2b) and the similarity score obtained using the three similarity metrics.

5.4.3 Classification using PNN model

For combining the similarity scores and predicting whether the given text is plagiarized or not, the PNN is used. PNN allows the classification of text pairs based on a combination of features generated by more than one similarity metric, which enables a more flexible approach and is much more beneficial than classifying a text pair based on only one similarity metric with a predetermined threshold.

- Architecture of PNN

The PNN was first introduced by Specht(1990) , and it is mainly based on Bayes Parzen classification. The PNN is one of the supervised learning networks. It is implemented using the probabilistic model, such as Bayesian classifiers.. (Hajmeer, M and Basheer, I. 2002)A probabilistic neural network (PNN) is primarily a classifier that maps any input pattern to a number of classifications. It is an implementation of a statistical algorithm called kernel discriminant analysis in which the operations are organized into a multilayered feed forward network with four layers:

- Input layer
- Pattern layer
- Summation layer
- Output layer

The architecture of PNN is shown in figure 5.6.

A simple probabilistic density function (pdf) for class k is as follows where

X = unknown (input),

X_k = "Kth" sample,

σ = smoothing parameter and

p = length of vectors

$$f_k(x) = \frac{1}{(2\pi)p^2.\sigma^p} e^{\frac{-\|x - xk\|^2}{2\sigma^2}} \qquad (5.9)$$

The accuracy of PNN classification depends mainly on probability density function. The probability density function for single population is described using the following equation where n = no of samples in the population.

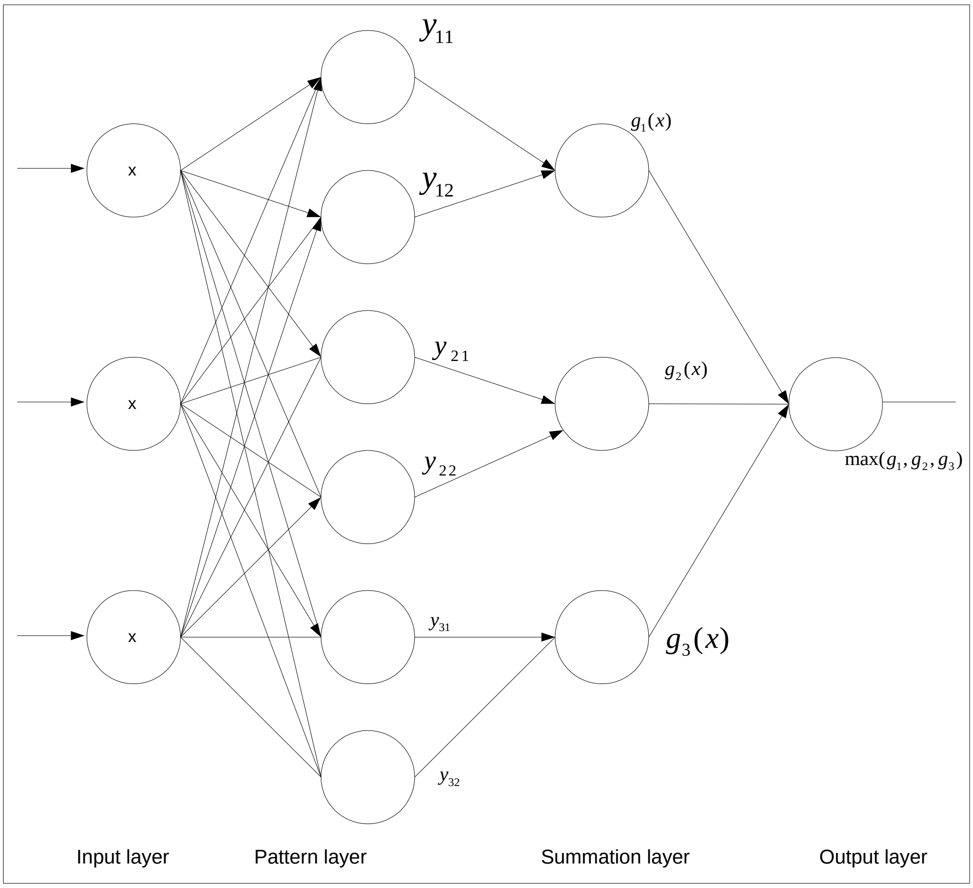


Figure 5.6: Architecture of PNN

$$g_i(x) = \frac{1}{(2\pi)^{\frac{p}{2}}.\sigma^p} \frac{1}{n_i} \sum_{k=1}^{n_i} e^{\frac{-\|x-x_k\|}{2\sigma^2}} \qquad (5.10)$$

If there are two classes i, j then classification criteria is decided using the following comparison:

$$g_i(X) > g_j(X) \text{ for all } j \neq i \qquad (5.11)$$

PNN is a simple probabilistic classification algorithm based on Bayes' theorem. This classifier uses the set of training examples to create a probabilistic model, which can be further used for the classification of new examples. The PNN is trained and tested using the three similarity measures as features. PNN is derived from Bayesian network and has several advantages like simple structure, fast and generates accurate predicted target probability scores.

It is appropriate for these experiments because it can operate on numeric features and the similarity measures generated are also numeric. Table 5.8 shows the similarity calculation for two sentences

Table 5.8: Example of Similarity calculation for two sentences

Sentence 1a	പരീക്കര് ചൈനീസ്നേതാക്കളുമായി ചര്ച്ച നടത്തും. (Pareekar will hold talks with Chinese leaders)	
Sentence 1b	പ്രതിരോധമന്ത്രി മനോഹര്പരീക്കര് ചൈനീസ്രാഷ്ട്രീയ സൈനിക നേതാക്കളുമായി ചര്ച്ച നടത്തും. (Defense minister Pareekar will hold talks with Chinese political and military leaders).	
Sentence2a	സുരേഷ്ഗോപിക്ക്പുറമെ മേരികോമും രാജ്യസഭയിലേക്ക്. (Mary Kom also along with Suresh Gopi to the Rajya Sabha.)	
Sentence2b	നടന് സുരേഷ്ഗോപിക്ക്പുറമെ ഒളിമ്പ്യന് ബോക്സര് മേരികോമും രാജ്യസഭയിലേക്ക്. (Olimpian Boxer Mary Kom also along with actor Suresh Gopi to the Rajya Sabha.)	
	Similarity between sentence 1a and 1b	Similarity between sentence 2a and 2b
Jaccard similarity	0.50	0.66
Dice similarity	0.66	0.80
Cosine similarity	0.70	0.80

5.4.4 Experimental Results

Corpus: The corpus comprises 2500 text documents in set-1 and 3500 text documents in set-2 of the training set and 900 text documents in set-1 and 1400 text documents in set-2 in the test set as shown in table .

Table 5.9 Data set for PNN training and Test

Sets	Number of Documents			
	Training set		Test set	
	Plagiarized	Source	Plagiarized	Source
Set-1	1000	1500	360	540
Set-2	2000	1500	700	700

To the best of our knowledge an annotated corpus in Malayalam language does not exist for plagiarism detection. This corpus is based on a paraphrase detection corpus released as part of DPIL shared task at FIRE 2016. The above said corpus is modified using text from Malayalam newspapers. We have constructed only short documents with different levels of plagiarism.

Similarity scores obtained using Jaccard, Dice and Cosine similarity metrics are given to the PNN for classifying the document as plagiarized or not.

The PDS evaluated on the four standard measures-Precision, Recall, accuracy and F-measure.

The performance of the three metrics with and without the use of NLP techniques are shown tables 5.10 and 5.11 respectively.

Table 5.10: Performance of PNN model with NLP

similarity measure	Precision	Recall	F-measure	Accuracy
Jaccard	0.79	0.82	0.81	0.83
Dice	0.81	0.86	0.91	0.91
Cosine	0.81	0.85	0.90	0.90

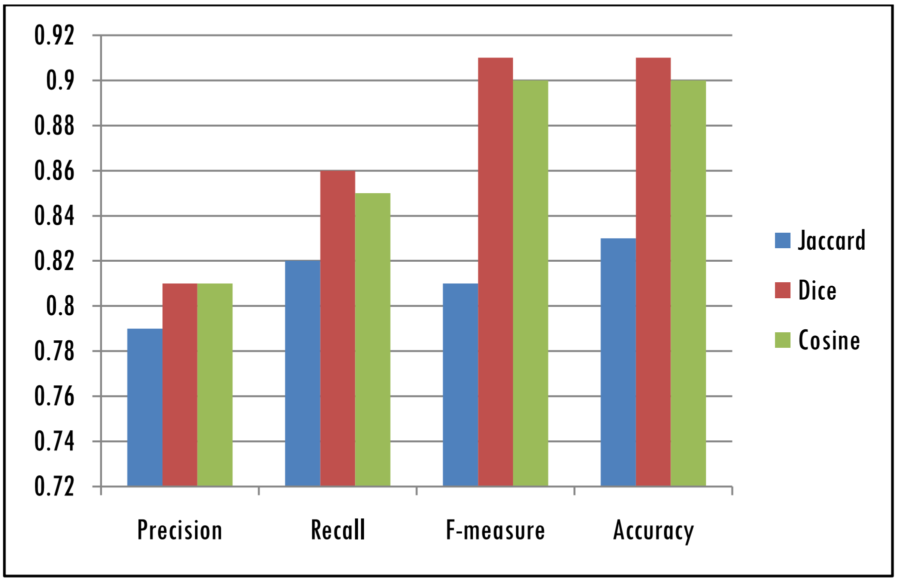


Figure 5.7: Performance of PNN with NLP

Table 5.11: Performance of PNN model without NLP

similarity measure	Precision	Recall	F-measure	Accuracy
Jaccard	0.58	0.71	0.639	0.64
Dice	0.61	0.73	0.67	0.68
Cosine	0.65	0.69	0.67	0.68

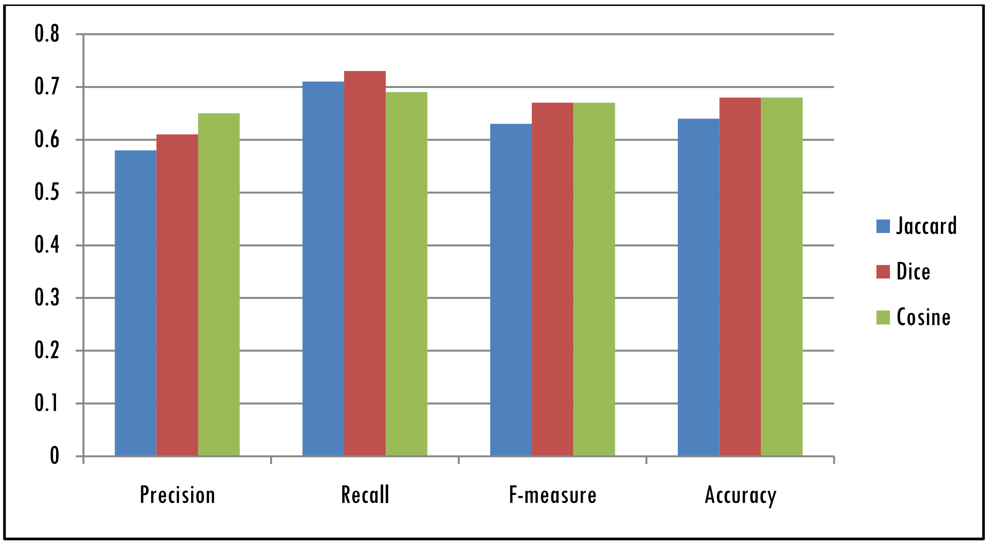


Figure 5.8: Performance of PNN without NLP

The performance of combined similarity with and without the use of NLP techniques are shown table 5.

Table 5.12: Performance of PNN model with and without NLP

similarity measure	Precision	Recall	F-measure	Accuracy
Without NLP	0.6	0.71	0.65	0.64
With NLP	0.93	0.95	0.94	0.95

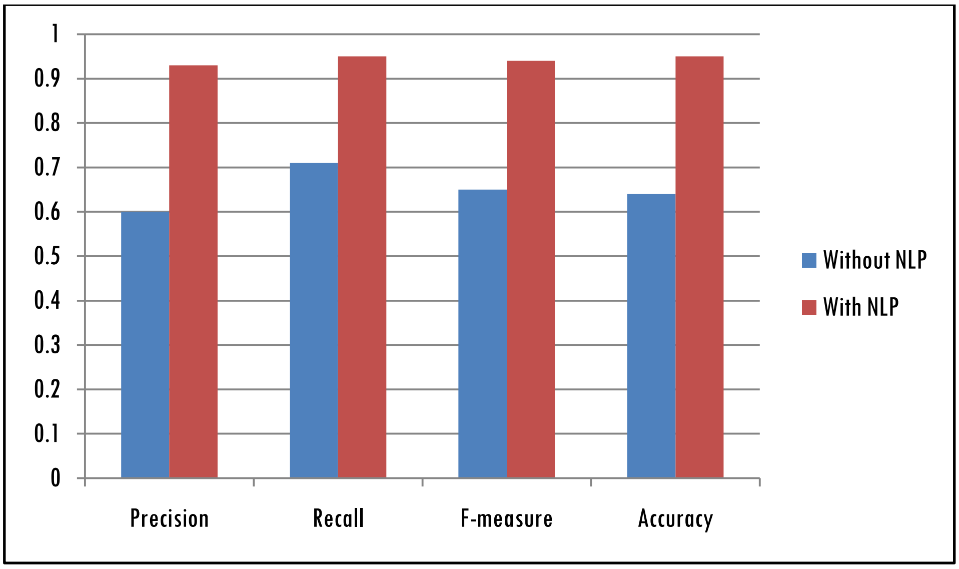


Figure 5.9: Performance of PNN model with and without NLP

It is clear from the results that plagiarism detection can be done efficiently by using NLP techniques like lemmatization and synonym replacement. Moreover the combined similarity classification gives better results than the individual similarity measures.

5.5 Chapter Summary

Four models for plagiarism detection in Malayalam was presented namely

- Plagiarism Detection System using modified n-grams model
- Plagiarism Detection System using fingerprinting method
- Plagiarism Detection System using Semantic Role Labelling model and
- Plagiarism Detection System using PNN model

A good corpus is essential for best results. The use of NLP techniques like lemmatization and synonym replacement has improved the detection rate. The possibility of combining three individual similarity measures using a PNN algorithm for detecting plagiarized documents in Malayalam Language is presented . Experimental results show that the PNN is fast and gives optimal classification. The results obtained from combining the similarity metrics and the use of NLP techniques is better than results obtained with single metrics . In future the effect of other semantic similarity measures for improving plagiarism detection can also be experimented with .

…….ꙮ…….

Chapter

6 CONCLUSIONS

Contents

This chapter summarises this study and provides an outline of further research directions.

The internet is a vast repository of digital information. It provides people free and easy access to information. But it is found that information thus accessed is often misused. Plagiarism is the process of using existing text without proper reference to the original source of the text. In recent years plagiarism has become a serious problem in the area of higher education and the research community in particular. Manual detection of plagiarism is not feasible because of the large number of sources available in digital form. Hence it is necessary to develop automated plagiarism detection systems. A large number of PDS have been developed for English and other languages like Arabic and Persian but none is available for the Malayalam language.

So the main research question for this research was "“How can we effectively detect plagiarism in Malayalam text taking into account the linguistic features of the language?” In order to answer this research question, six sub-questions were formulated in chapter one.

For answering the first, third and fifth sub-questions, we had studied the grammatical features of Malayalam in detail and analysed the literature to understand the available Malayalam language resources. Chapter 4 describes the details of this study. For dealing with the second sub question, we had studied the important concepts and different types of plagiarism in the research context. Chapter 2 explains the details of it. For answering the fourth sub-question, we had done a literature survey of the different plagiarism detection systems available for different languages. This is explained in Chapter 3. The sixth sub-question is answered in chapter 5. Conclusions and suggestions for future work are given in chapter 6.

6.1 Research Findings

Plagiarism can be committed at different levels: direct copy from existing text and intelligently masking the copied text using different methods. Plagiarism due to direct copy of text is easy to identify while it becomes difficult when the masking is very cleverly done.

Simple techniques cannot identify intelligent plagiarism.

A literature survey was conducted to identify the techniques available in the literature for different levels of plagiarism detection. From the survey it was found that a plagiarism detection system for Malayalam text is not yet available. Also it was found that the available plagiarism detection systems are not suitable for Malayalam because of the difference in the language characteristics. Hence a framework for plagiarism detection in Malayalam text was proposed.

For this a detailed study was conducted on the grammatical features of the Malayalam language and also found that Malayalam being a less resource language, tools like morphological analyser, tagger , lemmatiser, semantic role labeller etc were to be developed. So those tools were developed so that they can be used in the PDS.

Four models were developed for Malayalam PDS. They are :

(i) N-gram based model

(ii) Fingerprinting based model

(iii) Semantic role labelling based model and

(iv) PNN based model

On evaluation it is found that N-gram based model is the simplest to implement are gives best results for direct copy plagiarism. Fingerprinting based model can be used while comparing large files because the fingerprinting algorithm is a procedure that maps large to a much shorter fingerprint, that represents the original data. Semantic role labelling based model identifies plagiarism based on the semantic roles and so matching of irrelevant sentences can be reduced. The PNN based model combines different individual similarity measures to classify a text as plagiarised or not.

The n-gram based model and the fingerprinting based model do not consider the semantics of the language. Hence these models can be used for plagiarism detection in any language.

A plagiarism detection system should minimise the amount of cases mistakenly marked as plagiarised and it should only suggest what has been plagiarised, while the final verdict should be given only after manual investigation.

6.2 Suggestions for Further Work

Semantic similarity measures can improve the accuracy of PDS. But it requires standard language tools and resources. Malayalam being a less resource language, the better performance of Malayalam PDS demands the development of standard language tools like those available in English.

6.3 Conclusions

Four models have been developed for Malayalam text plagiarism detection and their performance were evaluated. The linguistic feature of Malayalam and the unavailability of standard language resources is a limitation for experimenting with more semantically oriented techniques.

REFERENCES

[1] Adeel Nawab, Rao Muhammad, Mark Stevenson, and Paul Clough. "Detecting text reuse with modified and weighted n-grams." *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2012.

[2] Alzahrani, Salha Mohammed, and Naomie Salim. "Plagiarism detection in Arabic scripts using fuzzy information retrieval." *Student Conf. Res. Develop., Johor Bahru, Malaysia*. 2008.

[3] Alzahrani, Salha Mohammed, and Naomie Salim. "On the use of fuzzy information retrieval for gauging similarity of arabic documents." *Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the*. IEEE, 2009.

[4] Alzahrani, Salha Mohammed. *Plagiarism Auto-detection in Arabic Scripts Using Statement-based Fingerprints Matching and Fuzzy-set Information Retrieval*. Diss. Universiti Teknologi Malaysia, 2009.

[5] Alzahrani, Salha, and Naomie Salim. "Statement-based fuzzy-set IR versus fingerprints matching for plagiarism detection in arabic documents." *Proc. 5th Postgraduate Annu. Res. Seminar*. 2009.

[6] Alzahrani, Salha, and Naomie Salim. "Fuzzy semantic-based string similarity for extrinsic plagiarism detection." *Braschler and Harman* (2010): 1-8.

[7] Alzahrani, Salha, Naomie Salim, Ajith Abraham, and Vasile Palade. "iPlag: intelligent plagiarism reasoner in scientific publications." In *Information and Communication Technologies (WICT), 2011 World Congress on*, pp. 1-6. IEEE, 2011.

[8] Alzahrani, Salha M., Naomie Salim, and Ajith Abraham. "Understanding plagiarism linguistic patterns, textual features and detection methods." *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 99 (2011): 1.

[9] Alzahrani, Salha, Vasile Palade, Naomie Salim, and Ajith Abraham. "Using structural information and citation evidence to detect significant plagiarism cases in scientific publications." *Journal of the Association for Information Science and Technology* 63, no. 2 (2012): 286-312.

[10] Alzahrani, Salha M., Naomie Salim, and Vasile Palade. "Uncovering highly obfuscated plagiarism cases using fuzzy semantic-based similarity model." *Journal of King Saud University-Computer and Information Sciences* 27.3 (2015): 248-268.

[11] Ampazis, Nikolaos, and Stavros J. Perantonis. "LSISOM—A Latent Semantic Indexing Approach to Self-Organizing Maps of Document Collections." *Neural Processing Letters* 19.2 (2004): 157-173.

[12] Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. "Modern Information Retrieval: The Concepts and Technology behind Search (ACM Press Books)." (2011).

[13] Balaguer, Enrique Vallés. "Putting ourselves in SME’s shoes: Automatic detection of plagiarism by the WCopyFind tool." *Proc. SEPLN*. 2009.

[14] Barrón-Cedeno, Alberto, Paolo Rosso, David Pinto, and Alfons Juan. "On Cross-lingual Plagiarism Analysis using a Statistical Model." In PAN, pp. 1-10. 2008.

[15] Barrón-Cedeño, Alberto, and Paolo Rosso. "On automatic plagiarism detection based on n-grams comparison." *Advances in Information Retrieval* (2009): 696-700.

[16] Barrón-Cedeño, Alberto, Martin Potthast, Paolo Rosso, Benno Stein, and Andreas Eiselt. "Corpus and Evaluation Measures for Automatic Plagiarism Detection." In *LREC*. 2010.

[17] Basile, Chiara, et al. "A plagiarism detection procedure in three steps: Selection, matches and “squares”." *Proc. SEPLN*. 2009.

[18] Bell, Roger T. *Translation and translating: Theory and practice*. Taylor & Francis, 1991.

[19] Bensalem, Imene, Paolo Rosso, and Salim Chikhi. "Intrinsic Plagiarism Detection using N-gram Classes." *EMNLP*. 2014.

[20] Brin, Sergey, James Davis, and Hector Garcia-Molina. "Copy detection mechanisms for digital documents." *ACM SIGMOD Record*. Vol. 24. No. 2. ACM, 1995.

[21] Broder, Andrei Z. "On the resemblance and containment of documents." *Compression and Complexity of Sequences 1997. Proceedings*. IEEE, 1997.

[22] Broder, Andrei Z., et al. "Syntactic clustering of the web." *Computer Networks and ISDN Systems* 29.8-13 (1997): 1157-1166.

[23] Cavnar, William B., and John M. Trenkle. "N-gram-based text categorization." *Ann Arbor MI* 48113.2 (1994): 161-175.

[24] Ceska, Zdenek. "The future of copy detection techniques." *Proc. YRCAS* (2007): 5-10.

[25] Ceska, Zdenek. *Automatic plagiarism detection based on latent semantic analysis*. Diss. Ph. D. dissertation, Faculty Appl Sci., Univ. West Bohemia, Pilsen, Czech Republic, 2009.

[26] Ceska, Zdenek, and Chris Fox. "The influence of text pre-processing on plagiarism detection." Association for Computational Linguistics, 2011.

[27] Chen, Hsinchun, et al. "Generating, integrating, and activating thesauri for concept-based document retrieval." *IEEE Expert*8.2 (1993): 25-34.

[28] Chong, Miranda, Lucia Specia, and Ruslan Mitkov. "Using natural language processing for automatic detection of plagiarism." *Proceedings of the 4th International Plagiarism Conference (IPC-2010)*. 2010.

[29] Chong, Miranda, and Lucia Specia. "Lexical Generalisation for Word-level Matching in Plagiarism Detection." *RANLP*. 2011.

[30] Chow, Tommy WS, and M. K. M. Rahman. "Multilayer SOM with tree-structured data for efficient document retrieval and plagiarism detection." *IEEE Transactions on Neural Networks*20.9 (2009): 1385-1402.

[31] Chuda, Daniela, and Pavol Navrat. "Support for checking plagiarism in e-learning." *Procedia-Social and Behavioral Sciences* 2.2 (2010): 3140-3144.

[32] Clough, Paul, et al. "Meter: Measuring text reuse." *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002.

[33] Clough, Paul. "Old and new challenges in automatic plagiarism detection." *National Plagiarism Advisory Service, 2003; http://ir. shef. ac. uk/cloughie/index. html*. 2003.

[34] Clough, Paul, and Mark Stevenson. "Developing a corpus of plagiarised short answers." *Language Resources and Evaluation* 45.1 (2011): 5-24.

[35] Cohen, William, Pradeep Ravikumar, and Stephen Fienberg. "A comparison of string metrics for matching names and records." *Kdd workshop on data cleaning and object consolidation*. Vol. 3. 2003.

[36] Cross, Valerie. "Fuzzy information retrieval." *Journal of Intelligent Information Systems* 3.1 (1994): 29-56.

[37] Ding, Chris HQ. "A similarity-based probability model for latent semantic indexing." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.

[38] Ehsan, Nava, Frank Wm Tompa, and Azadeh Shakery. "Using a dictionary and n-gram alignment to improve fine-grained cross-language plagiarism detection." *Proceedings of the 2016 ACM Symposium on Document Engineering*. ACM, 2016.

[39] Elhadi, Mohamed, and Amjad Al-Tobi. "Use of text syntactical structures in detection of document duplicates." *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*. IEEE, 2008.

[40] Elhadi, Mohamed, and Amjad Al-Tobi. "Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures." *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*. 2009.

[41] Esteki, Fezeh, and Faramarz Safi Esfahani. "A Plagiarism Detection Approach Based on SVM for Persian Texts." *FIRE (Working Notes)*. 2016.

[42] Finkel, Raphael A., et al. "Signature extraction for overlap detection in documents." *Australian Computer Science Communications*. Vol. 24. No. 1. Australian Computer Society, Inc., 2002.

[43] Fox, Edward A., and Joseph A. Shaw. "Combination of multiple searches." *NIST SPECIAL PUBLICATION SP* 243 (1994).

[44] Gaizauskas, Robert, et al. "The METER corpus: a corpus for analysing journalistic text reuse." *Proceedings of the corpus linguistics 2001 conference*. 2001.

[45] Gharavi, Erfaneh, et al. "A Deep Learning Approach to Persian Plagiarism Detection." *FIRE (Working Notes)*. 2016.

[46] Gillam, Lee, and Anna Vartapetiance. "From English to Persian: Conversion of Text Alignment for Plagiarism Detection." *PAN@ FIRE2016 Shared Task on Persian Plagiarism Detection and Text Alignment Corpus Construction. Notebook Papers of FIRE 2016* (2016).

[47] Gipp, Bela, and Jöran Beel. "Citation based plagiarism detection: a new approach to identify plagiarized work language independently." *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. ACM, 2010.

[48] Gipp, Bela, and Norman Meuschke. "Citation pattern matching algorithms for citation-based plagiarism detection: greedy citation tiling, citation chunking and longest common citation sequence." *Proceedings of the 11th ACM symposium on Document engineering*. ACM, 2011.

[49] Gipp, Bela, Norman Meuschke, and Corinna Breitinger. "Citation-based plagiarism detection: Practicability on a large-scale scientific corpus." *Journal of the Association for Information Science and Technology* 65.8 (2014): 1527-1540.

[50] Grman, Jan, and Rudolf Ravas. "Improved implementation for finding text similarities in large collections of data." *Proceedings of PAN* (2011).

[51] Grozea, Cristian, Christian Gehl, and Marius Popescu. "ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection." *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*. 2009.

[52] Grozea, Cristian, and Marius Popescu. "The encoplot similarity measure for automatic detection of plagiarism." *Notebook for PAN at CLEF* 2011 (2011).

[53] Gupta, Parth, and Paolo Rosso. "Text reuse with ACL:(upward) trends." *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*. Association for Computational Linguistics, 2012.

[54] HaCohen-Kerner, Yaakov, Aharon Tayeb, and Natan Ben-Dror. "Detection of simple plagiarism in computer science papers." *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010.

[55] Hannabuss, Stuart. "Contested texts: issues of plagiarism." *Library management* 22.6/7 (2001): 311-318.

[56] Heintze, Nevin. "Scalable document fingerprinting." *1996 USENIX workshop on electronic commerce*. Vol. 3. No. 1. 1996.

[57] Hoad, Timothy C., and Justin Zobel. "Methods for identifying versioned and plagiarized documents." *Journal of the Association for Information Science and Technology* 54.3 (2003): 203-215.

[58] Joy, Mike, and Michael Luck. "Plagiarism in programming assignments." *IEEE Transactions on education* 42.2 (1999): 129-133.

[59] Kaski, Samuel, et al. "WEBSOM–self-organizing maps of document collections." *Neurocomputing* 21.1 (1998): 101-117.

[60] Kasprzak, Jan, Michal Brandejs, and Miroslav Kripac. "Finding plagiarism by evaluating document similarities." *Proc. SEPLN*. Vol. 9. No. 4. 2009.

[61] Kasprzak, Jan, and Michal Brandejs. "Improving the reliability of the plagiarism detection system." *Lab Report for PAN at CLEF* (2010): 359-366.

[62] Khoshnavataher, Khadijeh, et al. "Developing monolingual Persian corpus for extrinsic plagiarism detection using artificial obfuscation." *Notebook for PAN at CLEF* (2015).

[63] Koberstein, Jonathan, and Yiu-Kai Ng. "Using word clusters to detect similar web documents." *KSEM* 4092 (2006): 215-228.

[64] Kohonen, Teuvo. *Self-organization and associative memory*. Vol. 8. Springer Science & Business Media, 2012.

[65] Kong, Leilei, et al. "Approaches for Source Retrieval and Text Alignment of Plagiarism Detection Notebook for PAN at CLEF 2013." *CLEF (Working Notes)*. 2013.

[66] Koroutchev, Kostadin, and Manuel Cebrián. "Detecting translations of the same text and data with common source." *Journal of Statistical Mechanics: Theory and Experiment*2006.10 (2006): P10009.

[67] Lancaster, Thomas, and Fintan Culwin. "Classifications of plagiarism detection engines." *Innovation in Teaching and Learning in Information and Computer Sciences* 4.2 (2005): 1-16.

[68] Leilei, Kong, et al. "Approaches for candidate document retrieval and detailed comparison of plagiarism detection." *Notebook for PAN at CLEF 2012* (2012).

[69] Leung, Chi-Hong, and Yuen-Yan Chan. "A natural language processing approach to automatic plagiarism detection." *Proceedings of the 8th ACM SIGITE conference on Information technology education*. ACM, 2007.

[70] Li, Yuhua, David McLean, Zuhair A. Bandar, James D. O'shea, and Keeley Crockett. "Sentence similarity based on semantic nets and corpus statistics." *IEEE transactions on knowledge and data engineering* 18, no. 8 (2006): 1138-1150.

[71] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." *Text summarization branches out: Proceedings of the ACL-04 workshop*. Vol. 8. 2004.

[72] Lyon, Caroline, James Malcolm, and Bob Dickerson. "Detecting short passages of similar text in large document collections." *Proceedings of the*. 2001.

[73] Magooda, Ahmed, Ashraf Y. Mahgoub, Mohsen Rashwan, Magda B. Fayek, and Hazem M. Raafat. "RDI System for Extrinsic Plagiarism Detection (RDI_RED), Working Notes for PANAraPlagDet at FIRE 2015." In *FIRE Workshops*, pp. 126-128. 2015.

[74] Manber, Udi. "Finding similar files in a large file system." *Usenix Winter*. Vol. 94. 1994.

[75] Manning, C. D., P. Raghavan, and H. Schütze. "Web search basics: Near-duplicates and shingling." *Introduction to Information Retrieval* (2008): 437-442.

[76] Manning, C. D., P. Raghavan, and H. Schütze. "An Introduction to Information Retrieval-Chapitre 18: Matrix decompositions and latent semantic indexing." (2009): 403-419.

[77] Mashhadirajab, Fatemeh, and Mehrnoush Shamsfard. "A Text Alignment Algorithm Based on Prediction of Obfuscation Types Using SVM Neural Network." *FIRE (Working Notes)*. 2016.

[78] Maurer, Hermann A., Frank Kappe, and Bilal Zaka. "Plagiarism-a survey." *J. UCS* 12.8 (2006): 1050-1084.

[79] Menai, Mohamed El Bachir, and Manar Bagais. "APlag: A plagiarism checker for Arabic texts." *Computer Science & Education (ICCSE), 2011 6th International Conference on*. IEEE, 2011.

[80] Meuschke, Norman, et al. "CitePlag: A citation-based plagiarism detection system prototype." *Proceedings of the 5th International Plagiarism Conference*. 2012.

[81] Meyer zu Eissen, Sven, Benno Stein, and Marion Kulig. "Plagiarism detection without reference collections." *Advances in data analysis* (2007): 359-366.

[82] Miller, George A. "WordNet: a lexical database for English." *Communications of the ACM* 38.11 (1995): 39-41.

[83] Momtaz, Mozhgan, Kayvan Bijari, Mostafa Salehi, and Hadi Veisi. "Graph-based Approach to Text Alignment for Plagiarism Detection in Persian Documents." In *FIRE (Working Notes)*, pp. 176-179. 2016.

[84] Mozgovoy, Maxim. "Desktop tools for offline plagiarism detection in computer programs." *Informatics in education* 5.1 (2006): 97.

[85] Mozgovoy, Maxim. *Enhancing computer-aided plagiarism detection*. University Of Joensuu, 2007.

[86] Nawab, R., Mark Stevenson, and Paul Clough. "University of sheffield: Lab report for PAN at CLEF 2010." *CLEF 2010 LABs and Workshops, Notebook Papers*. CLEF, 2010.

[87] Ogawa, Yasushi, Tetsuya Morita, and Kiyohiko Kobayashi. "A fuzzy document retrieval system using the keyword connection matrix and a learning method." *Fuzzy sets and systems* 39.2 (1991): 163-179.

[88] Pereira, Álvaro R., and Nivio Ziviani. "Retrieving similar documents from the web." *Journal of Web Engineering* 2.4 (2003): 247-261.

[89] Pertile, Solange de L., Paolo Rosso, and Viviane P. Moreira. "Counting Co-occurrences in Citations to Identify Plagiarised Text Fragments." *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, Berlin, Heidelberg, 2013.

[90] Potthast, Martin, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. "An evaluation framework for plagiarism detection." In *Proceedings of the 23rd international conference on computational linguistics: Posters*, pp. 997-1005. Association for Computational Linguistics, 2010.

[91] Potthast, Martin, Benno Stein, and Teresa Holfeld. "Overview of the 1st International Competition on Wikipedia Vandalism Detection." *CLEF (Notebook Papers/LABs/Workshops)*. 2010.

[92] Potthast, Martin, Alberto Barrón-Cedeño, Benno Stein, and Paolo Rosso. "Cross-language plagiarism detection." *Language Resources and Evaluation* 45, no. 1 (2011): 45-62.

[93] Potthast, Martin, Matthias Hagen, Tim Gollub, Martin Tippmann, Johannes Kiesel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. "Overview of the 4th International Competition on Plagiarism Detection." In *CLEF (Online Working Notes/Labs/ Workshop)*. 2012.

[94] Rahman, M. K. M., et al. "A flexible multi-layer self-organizing map for generic processing of tree-structured data." *Pattern Recognition* 40.5 (2007): 1406-1424.

[95] Rahman, M. K. M., and Tommy WS Chow. "Content-based hierarchical document organization using multi-layer hybrid network and tree-structured features." *Expert Systems with Applications* 37.4 (2010): 2874-2881.

[96] Rodríguez-Torrejón, D. A., and J. M. Martín-Ramos. "CoReMo system (contextual reference monotony) a fast, low cost and high performance plagiarism analyzer system: Lab report for PAN at CLEF 2010." *Notebook Papers of CLEF* (2010).

[97] Sanchez-Perez, Miguel A., Grigori Sidorov, and Alexander F. Gelbukh. "A Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014." *CLEF (Working Notes)*. 2014.

[98] Scherbinin, Vladislav, and Sergey Butakov. "Using Microsoft SQL server platform for plagiarism detection." *Proc. SEPLN*. 2009.

[99] Schleimer, Saul, Daniel S. Wilkerson, and Alex Aiken. "Winnowing: local algorithms for document fingerprinting." *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003.

[100] Shivakumar, Narayanan, and Hector Garcia-Molina. "SCAM: A copy detection mechanism for digital documents." (1995).

[101] Shivakumar, Narayanan, and Hector Garcia-Molina. *The SCAM approach to copy detection in digital libraries*. Stanford InfoLab, 1995.

[102] Shivakumar, Narayanan, and Hector Garcia-Molina. "Building a scalable and accurate copy detection mechanism." *Proceedings of the first ACM international conference on Digital libraries*. ACM, 1996.

[103] Siddiqui, Muazzam Ahmed, Imtiaz Hussain Khan, K. Mansoor Jambi, S. Omar Elhaj, and Abobakr Bagais. "Developing an Arabic Plagiarism Detection Corpus." *Computer Science & Information Technology (CS & IT)* 4, no. 2014 (2014): 261-269.

[104] Sorokina, Daria, Johannes Gehrke, Simeon Warner, and Paul Ginsparg. "Plagiarism detection in arXiv." In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pp. 1070-1075. IEEE, 2006.

[105] Specht, Donald F. "Probabilistic neural networks." *Neural networks* 3.1 (1990): 109-118.

[106] Stamatatos, Efstathios. "Plagiarism detection using stopword n-grams." *Journal of the Association for Information Science and Technology* 62.12 (2011): 2512-2527.

[107] Stein, Benno, and Sven Eissen. "Near similarity search and plagiarism analysis." *From data and information analysis to knowledge engineering* (2006): 430-437.

[108] Stein, Benno, Sven Meyer zu Eissen, and Martin Potthast. "Strategies for retrieving plagiarized documents." *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

[109] Stein, Benno. "Principles of hash-based text retrieval." *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2007.

[110] Stein, Benno, et al. "3rd PAN workshop on uncovering plagiarism, authorship and social software misuse." *25th Annual Conference of the Spanish Society for Natural Language Processing (SEPLN)*. 2009.

[111] Su, Zhan, et al. "Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm." *Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on*. IEEE, 2008.

[112] Talebpour, Alireza, Mohammad Shirzadi Laskoukelayeh, and Zahra Aminolroaya. "Plagiarism Detection Based on a Novel Trie-based Approach." *FIRE (Working Notes)*. 2016.

[113] Torrejón, Diego A. Rodríguez, and José Manuel Martín Ramos. "Text Alignment Module in CoReMo 2.1 Plagiarism Detector." *Notebook for PAN at CLEF* (2013).

[114] Uzuner, Özlem, Boris Katz, and Thade Nahnsen. "Using syntactic information to identify plagiarism." *Proceedings of the second workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, 2005.

[115] Weber-Wulff, Debora. "On the utility of plagiarism detection software." *Third International Plagiarism Conference, Gateshead. Retrieved from http://www. plagiarismconference. co. uk/images/conferenceimages/028_P21% 20W eber-Wulff. pdf*. 2008.

[116] Yerra, Rajiv, and Yiu-Kai Ng. "A sentence-based copy detection approach for web documents." *Fuzzy systems and knowledge discovery* (2005): 481-482.

[117] Zadrożny, Sławomir, and Katarzyna Nowacka. "Fuzzy information retrieval model revisited." *Fuzzy Sets and Systems* 160, no. 15 (2009): 2173-2191.

[118] Zechner, Mario, et al. "External and intrinsic plagiarism detection using vector space models." *Proc. SEPLN*. Vol. 32. 2009.

[119] Zhang, Haijun, and Tommy WS Chow. "A coarse-to-fine framework to efficiently thwart plagiarism." *Pattern Recognition* 44.2 (2011): 471-487.

[120] Zu Eissen, Sven Meyer, and Benno Stein. "Intrinsic Plagiarism Detection." *ECIR*. Vol. 3936. 2006.

…….ജൻ…….

LIST OF PUBLICATIONS

Papers in International Journals

[1] Sindhu, L., and Sumam Mary Idicula. "SRL based Plagiarism Detection System for Malayalam Documents." *International Journal of Computer Science Issues (IJCSI)* 12.6 (2015): 91.

[2] Sindhu, L., Bindu Baby Thomas, and Sumam Mary Idicula. "Automated plagiarism detection system for malayalam text documents." *International Journal of Computer Applications* 106.15 (2014).

Papers in International Conferences

[1] Sindhu, L., and Sumam Mary Idicula. "Plagiarism Detection in Malayalam Language Text using a Composition of Similarity measures." *Proceedings of the 9th International Conference on Machine Learning and Computing*. ACM, 2017.

[2] Sindhu, L., and Sumam Mary Idicula. "CUSAT_NLP@ DPIL-FIRE2016: Malayalam Paraphrase Detection." *FIRE (Working Notes)*. 2016.

[3] Sindhu, L., and Sumam Mary Idicula. "A Plagiarism Detection System for Malayalam Text Based Documents with Full and Partial Copy." *Procedia Technology* 25 (2016): 372-377.

[4] Sindhu, L., and Sumam Mary Idicula. "Fingerprinting based detection system for identifying plagiarism in Malayalam text documents." *Computing and Network Communications (CoCoNet), 2015 International Conference on*. IEEE, 2015.

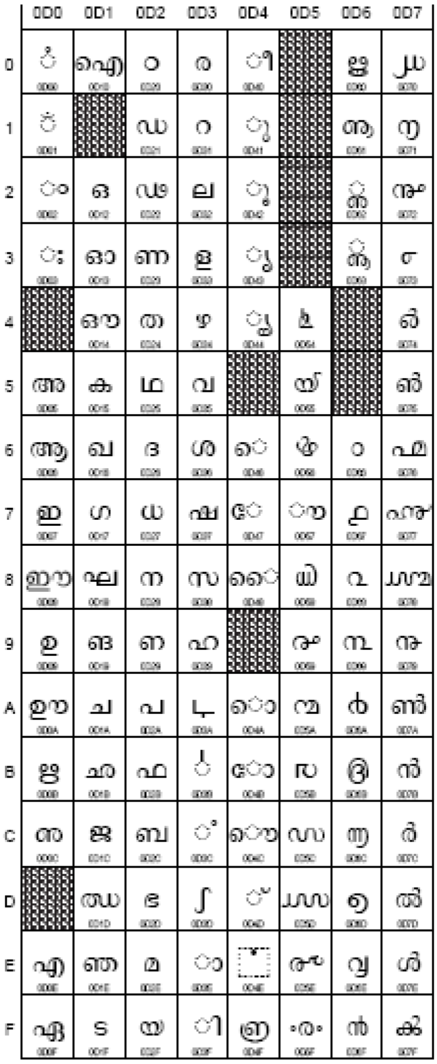
National Conferences

[1] Sindhu, L., Bindu Baby Thomas, and Sumam Mary Idicula . "A Copy detection Method for Malayalam Text Documents using N-grams Model." *National conference on Indian Language Computing (NCILC)*, 2013.

Appendix A

MALAYALAM UNICODE

0D00 Malayalam 0D7F



	0D0	0D1	0D2	0D3	0D4	0D5	0D6	0D7
0	◌ഀ 0D00	ഐ 0D10	ഠ 0D20	ര 0D30	◌ീ 0D40		ൠ 0D60	൰ 0D70
1	◌ഁ 0D01		ഡ 0D21	റ 0D31	◌ു 0D41		ൡ 0D61	൱ 0D71
2	◌ം 0D02	ഒ 0D12	ഢ 0D22	ല 0D32	◌ൂ 0D42		◌ൢ 0D62	൲ 0D72
3	◌ഃ 0D03	ഓ 0D13	ണ 0D23	ള 0D33	◌ൃ 0D43		◌ൣ 0D63	൳ 0D73
4		ഔ 0D14	ത 0D24	ഴ 0D34	◌ൄ 0D44	ൔ 0D54		൴ 0D74
5	അ 0D05	ക 0D15	ഥ 0D25	വ 0D35		ൕ 0D55		൵ 0D75
6	ആ 0D06	ഖ 0D16	ദ 0D26	ശ 0D36	◌െ 0D46	ൖ 0D56	൦ 0D66	൶ 0D76
7	ഇ 0D07	ഗ 0D17	ധ 0D27	ഷ 0D37	◌േ 0D47	◌ൗ 0D57	൧ 0D67	൷ 0D77
8	ഈ 0D08	ഘ 0D18	ന 0D28	സ 0D38	◌ൈ 0D48	൘ 0D58	൨ 0D68	൸ 0D78
9	ഉ 0D09	ങ 0D19	ഩ 0D29	ഹ 0D39		൙ 0D59	൩ 0D69	൹ 0D79
A	ഊ 0D0A	ച 0D1A	പ 0D2A	ഺ 0D3A	◌ൊ 0D4A	൚ 0D5A	൪ 0D6A	ൺ 0D7A
B	ഋ 0D0B	ഛ 0D1B	ഫ 0D2B	◌഻ 0D3B	◌ോ 0D4B	൛ 0D5B	൫ 0D6B	ൻ 0D7B
C	ഌ 0D0C	ജ 0D1C	ബ 0D2C	◌഼ 0D3C	◌ൌ 0D4C	൜ 0D5C	൬ 0D6C	ർ 0D7C
D		ഝ 0D1D	ഭ 0D2D	ഽ 0D3D	◌് 0D4D	൝ 0D5D	൭ 0D6D	ൽ 0D7D
E	എ 0D0E	ഞ 0D1E	മ 0D2E	◌ാ 0D3E	ൎ 0D4E	൞ 0D5E	൮ 0D6E	ൾ 0D7E
F	ഏ 0D0F	ട 0D1F	യ 0D2F	◌ി 0D3F	൏ 0D4F	ൟ 0D5F	൯ 0D6F	ൿ 0D7F

Appendix

B SAMPLE DOCUMENT

പുഴകളുടെ മരണം ലോകത്തിന്റെ തന്നെ സ്വസ്ഥത കെടുത്തുന്ന വിഷയമായി മാറിക്കഴിഞ്ഞു. ലോകത്തിലെ അഞ്ഞൂറിലേറെ വൻനദികളിൽ പകുതിയിലേറെയും വരളുകയാണെന്ന് ഐക്യരാഷ്ട്ര സംഘടന മുന്നറിയിപ്പു തരുന്നുണ്ട്. മറയുകയാണു ജലസമൃദ്ധി.അല്ലെങ്കിൽ തന്നെ മെലിഞ്ഞുപോയ പുഴയൊഴുക്കിലേക്കു മാലിന്യങ്ങൾ തള്ളാൻ നമുക്കൊരു മടിയുമില്ലാതെയായിക്കഴിഞ്ഞു. നോവലിസ്റ്റ് എം. മുകുന്ദൻ ദീർഘകാലത്തെ ഡൽഹി ജീവിതത്തിനു വിരാമമിട്ടു നാട്ടിൽ തിരിച്ചെത്തിയപ്പോൾ ഒരു കാര്യത്തിലാണ് അത്യധികം ആഹ്ലാദിച്ചത്: ബാല്യവും ഓർമയുമൊഴുകുന്ന മയ്യഴിപ്പുഴയുടെ സുന്ദരതീരത്ത് ഇനിയുള്ള കാലം സന്തോഷമായി ജീവിക്കാം. പക്ഷേ, നാട്ടിലെത്തിയതിന്റെ തൊട്ടടുത്ത സൂര്യോദയത്തിൽ തന്നെ മയ്യഴിപ്പുഴ കാണാൻ പോയ മുകുന്ദനെ വരവേറ്റത് പുഴയിൽ നിക്ഷേപിച്ച മാലിന്യങ്ങളുടെയും മൃഗാവശിഷ്ടങ്ങളുടെയും ദാരുണ കാഴ്ചയായിരുന്നു. മയ്യഴിപ്പുഴയുടെ മാത്രം ദുരവസ്ഥയല്ല ഇത്. കേരളത്തിലെ എല്ലാ പുഴകളിലും ഇന്നു പൊങ്ങിക്കിടക്കുന്നത് അറവുശാലകളിലെയും വീടുകളിലെയും മാലിന്യച്ചാക്കുകളാണ്. പുഴകളോട് ഇത്രയധികം ക്രൂരമായി പെരുമാറുന്ന മറ്റൊരു സമൂഹം ലോകത്തുണ്ടാകുമോ?ആമസോണും നൈലും ഗംഗയും ഡാന്യൂബുമടക്കം ലോകത്തിലെ മഹാനദികളിൽ പലതും ശോഷിച്ചുകൊണ്ടിരിക്കുകയാണെന്നു കണ്ടെത്തിക്കഴിഞ്ഞു. മറക്കാൻ പാടില്ലാത്ത ജലപാഠങ്ങൾ നാം മറന്നുപോയതിന്റെ ദുരന്തസാക്ഷ്യങ്ങളായി കേരളത്തിലെയും പല പുഴകളും മരണത്തിലേക്കാണിപ്പോൾ ഒഴുകുന്നത്. ഈ മണ്ണിന്റെ ജലഞരമ്പുകളായി ഒരിക്കൽ സന്തോഷത്തോടെ ഒഴുകിയിരുന്ന നദികളുടെ ഇപ്പോഴത്തെ ദുർവിധിയറിയാൻ ഈ വേനലിൽ ഒരൊറ്റ പുഴക്കാഴ്ച മതിയാകും.കേരളത്തിലെ മറ്റു പുഴകളെ അപേക്ഷിച്ചു പരിശുദ്ധി നഷ്ടപ്പെടാത്തവയായിരുന്നു അടുത്തകാലം വരെ ഉത്തരകേരളത്തിലെ പുഴകൾ.

കാസർകോടു ജില്ലയിലെ തേജസ്വിനി മുതൽ മാഹിയിലെ മയ്യഴിപ്പുഴ വരെ തെളിനീരുവറ്റാതെ അറബിക്കടലിൽ എത്തിച്ചേർന്നു. കണ്ണാടിയിൽ കാണുന്നതുപോലെ ഈ പുഴകൾ തീരങ്ങളുടെ സംസ്കാരത്തെയും ജീവിതത്തെയും പ്രതിഫലിപ്പിച്ചു. പക്ഷേ, അതെല്ലാം ഇനി ഭൂതകാലത്തിൽ മാത്രം പറയേണ്ട പുഴയോർമകൾ. ശുദ്ധജലത്തിനുവേണ്ടി മനുഷ്യർ പരക്കം പായുന്ന കാലം. ജല യുദ്ധങ്ങൾ സാധാരണമാവുന്ന കാലം... ഇങ്ങനെയൊരു ദുരന്തകാലത്തിലേക്ക് അധികം അകലമൊന്നുമില്ല എന്ന ഓർമപ്പെടുത്തലുമായി ഒരു ലോക ജലദിനം കൂടി കടന്നുവരുന്നു. നദികളും ധാരാളം കിണറുകളും കുളങ്ങളും ജലാശയങ്ങളുമൊക്കെയുള്ള കേരളത്തിന്റെ കാര്യംതന്നെ നോക്കൂ. വേനലാവുമ്പോൾ പാത്രങ്ങളുമായി വെള്ളം കൊണ്ടുവരുന്ന ടാങ്കർ ലോറിക്കു മുന്നിൽ കാത്തുനിൽക്കേണ്ട ഗതികേടിലാണു വലിയൊരു വിഭാഗം ജനങ്ങൾ. ഭാരതപ്പുഴയാണെങ്കിൽ വേനലിനു മുൻപേ തന്നെ നീർച്ചാലായി മാറുന്നു. എന്നിട്ടും നമ്മൾ അവശേഷിക്കുന്ന ജലസ്രോതസ്സുകളെ വികസനത്തിന്റെ പേരിൽ ഇല്ലാതാക്കുന്നു. കുന്നിടിച്ചു നിരത്തുന്നു. ബാക്കിയുള്ള പച്ചപ്പിന്റെ കടയ്ക്കലും മഴു വയ്ക്കുന്നു.ബ്രസീലിലെ റിയോവിൽ ചേർന്ന യുഎൻ കോൺഫറൻസ് ഓൺ എൻവയൺമെന്റ് ആൻഡ് ഡവലപ്മെന്റിലാണ് ലോക ജലദിനം ആചരിക്കുന്നതിനെക്കുറിച്ചുള്ള നിർദേശങ്ങൾ ഉയർന്നത്. നഗരവൽക്കരണവും ജലവുമാണ് ഈ വർഷത്തെ ലോക ജലദിനാചരണ വിഷയം. ദക്ഷിണാഫ്രിക്കയിലെ കേപ്ടൗൺ ആണ് ലോകജലദിനത്തിന്റെ ഔദ്യോഗിക പരിപാടികൾക്ക് ഇത്തവണ ആതിഥ്യമരുളുന്നത്. പെരുകുന്ന ജനസംഖ്യ, കാലാവസ്ഥാ വ്യതിയാനങ്ങൾ, പ്രകൃതി ദുരന്തങ്ങൾ, മറ്റു പ്രശ്നങ്ങൾ എന്നിവയൊക്കെ നഗര ജലസ്രോതസ്സുകളിലുണ്ടാക്കുന്ന പ്രശ്നങ്ങളിലേക്കു രാജ്യാന്തര ശ്രദ്ധ ക്ഷണിക്കാനാണ് ഇത്തവണത്തെ ദിനാചരണം ലക്ഷ്യമിടുന്നത്.വരാനിരിക്കുന്ന വേനലുകൾക്കും പിറക്കാനിരിക്കുന്ന തലമുറകൾക്കും വേണ്ടി ഇന്ന്, ലോക ജലദിനത്തിൽ, നമ്മുടെ പുഴകളുടെ ആസന്നമരണ വിലാപങ്ങൾക്കായി നമുക്കു കാതോർക്കാം.പറശ്ശിനി പുഴയിൽ നിന്നു മണൽ ഊറ്റിയൂറ്റി മടപ്പുര മുത്തപ്പൻ ക്ഷേത്രത്തിന്റെ അടിവാരം

വരെയെത്തി. പുഴയോരത്തെ കണ്ടലുകളെല്ലാം വെട്ടി മണ്ണിട്ടുനികത്തി. ഉപയോഗശൂന്യമായ കീടനാശിനി തള്ളിയാണു കണ്ണൂർ ജില്ലയിലെ ആലക്കോടു പുഴയെ കഴിഞ്ഞ വർഷം കാളിന്ദിയാക്കിയത്. തലശേരിയിലെ കുയ്യാലിപ്പുഴയിലും എരഞ്ഞോളിപ്പുഴയിലും ആശുപത്രി മാലിന്യങ്ങൾ തള്ളിനിറച്ചു. പുഴയിലെ മീനുകൾക്കും പുഴയോര സസ്യങ്ങൾക്കുമൊക്കെ ആന്റിബയോട്ടിക്കിന്റെ മണമായി. കൈക്കുമ്പിളിൽ കോരിയെടുത്താൽ കിട്ടുന്നതു വെറും അഴുക്കു മാത്രം. കുട്ടിക്കാലത്തു പുഴയുടെ തെളിമയിൽ ചാടിമറിഞ്ഞു നീന്തിത്തുടിച്ചവർ പെട്ടി ഓട്ടോറിക്ഷകളിൽ മാലിന്യം കൊണ്ടുവന്നു പുഴയിൽ തള്ളാൻ കരാറെടുത്തു. വളപട്ടണം പുഴയുടെ കൈവഴിയായി ഒഴുകിയിരുന്ന കക്കാട് പുഴ, മധ്യതിരുവിതാംകൂറിലെ വരട്ടാറിന്റെ വിധി പിന്തുടർന്നു മണ്ണിൽ നിന്നേ ഏതാണ്ടു മാഞ്ഞുകഴിഞ്ഞു. കുറെ പരിസ്ഥിതി സ്നേഹികൾ കക്കാട് പുഴയെ വീണ്ടെടുക്കാനുള്ള ശ്രമത്തിലാണെന്നത് ആശ്വാസകരം.ഉത്തര കേരളത്തിലെ നദികളുടെ പരിശുദ്ധിയും പ്രതാപവും തിരികെ കൊണ്ടുവരാനുള്ള ആഗ്രഹവുമായി പൊതുജനപങ്കാളിത്തത്തോടെ മലയാള മനോരമ മുന്നിട്ടിറങ്ങുകയാണ്. ‘പുഴയോടൊപ്പം എന്ന ഈ പദ്ധതി യുണിസെഫിന്റെയും സംസ്ഥാന ജലവിഭവ വകുപ്പിന്റെയും സഹകരണത്തോടെയാണു നടപ്പാക്കുന്നത്. കണ്ണൂർ, കാസർകോട് ജില്ലകളിലെ പരിസ്ഥിതി പ്രവർത്തകരും സാംസ്കാരിക നായകരും സന്നദ്ധസംഘടനകളുമെല്ലാം ഈ നാടുണർത്തലിൽ കൈകോർക്കുന്നു.ജനപങ്കാളിത്തത്തോടെ ഈ പദ്ധതി നിറഞ്ഞുകവിഞ്ഞൊഴുകട്ടെ എന്നാണു ഞങ്ങളുടെ വിനീതമായ പ്രാർഥന. പുഴയ്ക്കും മിടിക്കുന്ന ഒരു ഹൃദയമുണ്ടെന്നു തിരിച്ചറിയുമ്പോഴല്ലേ നമ്മുടെ മനസ്സിലും നനവു പടരുക?

Appendix

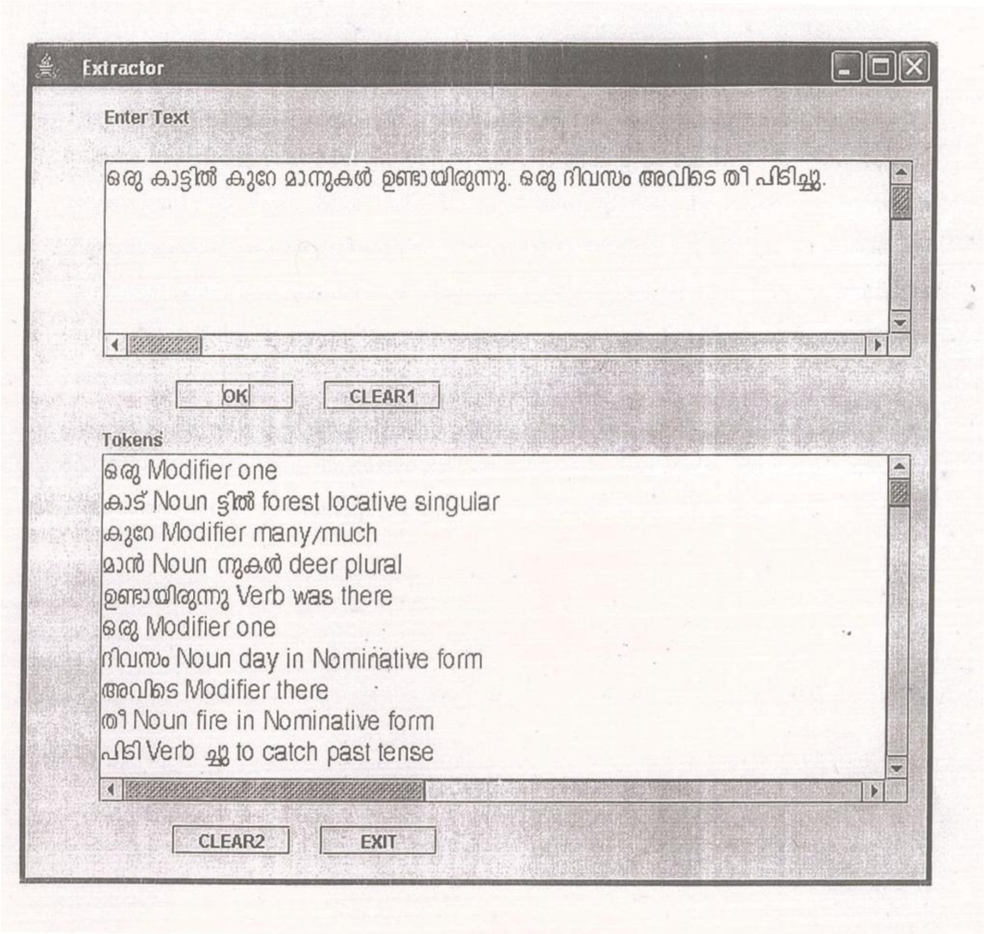
C

STOP WORD LIST

അവിടെ	പിന്നീട്	ഒട്ടേറെ	ആണ്
ഇവിടെ	എന്നിവ	മാത്രം	അല്ല
എവിടെ	തീരെ മാറ്റ്	അതേ	എന്തെന്നാൽ
ഇപ്പോൾ	ഇതിനിടെ	ആയ	മുൻപേ
എപ്പോൾ	വളരെ	ഒരിക്കലും	താഴെ
അപ്പോൾ	അതിന്റെ	പലപ്പോഴും	ഇടയ്ക്ക്
വരെ ഈ	എത്രയോ	അഥവാ	പക്ഷെ
ഇളം	ധാരാളം	കൂടി	പറ്റും
ചെറിയ	ഇതോടെ	കൂടെ	പറ്റില്ല
അന്ന്	ഒരു	എന്നും	വേളയിൽ
അതിനിടെ	ഇനി	ഏതു	ഓരോ
തന്നെ	പല	പറ്റി	കുറച്ച്
പെട്ടന്ന്	ക്രമേണ	മുകളിൽ	വേണ്ടി
ഇത്തരം	തുടർന്ന്	ശേഷം	നിന്നും
അതുപോലെ	എന്നാൽ	വീണ്ടും	കൂടുതൽ
ഇതിന്റെ	ഏതാനും	എല്ലാം	അവൻ
അവന്റെ	കൂടുതൽ	ഏതെങ്കിലും	അവൾ
അവളുടെ	ഏറ്റവും	അതുകൊണ്ടു	എവിടെ
എങ്ങനെ	ഇല്ല	അത്	എന്ത്കൊണ്ട്
ഞാൻ	ഒരിക്കൽ	അവർ	അത്
എന്നിക്കു	മാത്രം	അവരുടെ	നമ്മുടെ
എൻന്റെ	അല്ലങ്കിൽ	അതെല്ലാം	നമ്മൾ
എങ്കിൽ	മറ്റേ	അതിലൂടെ	എന്ത്

Appendix

D SCREEN SHOT – MORPHOLOGICAL ANALYSIS

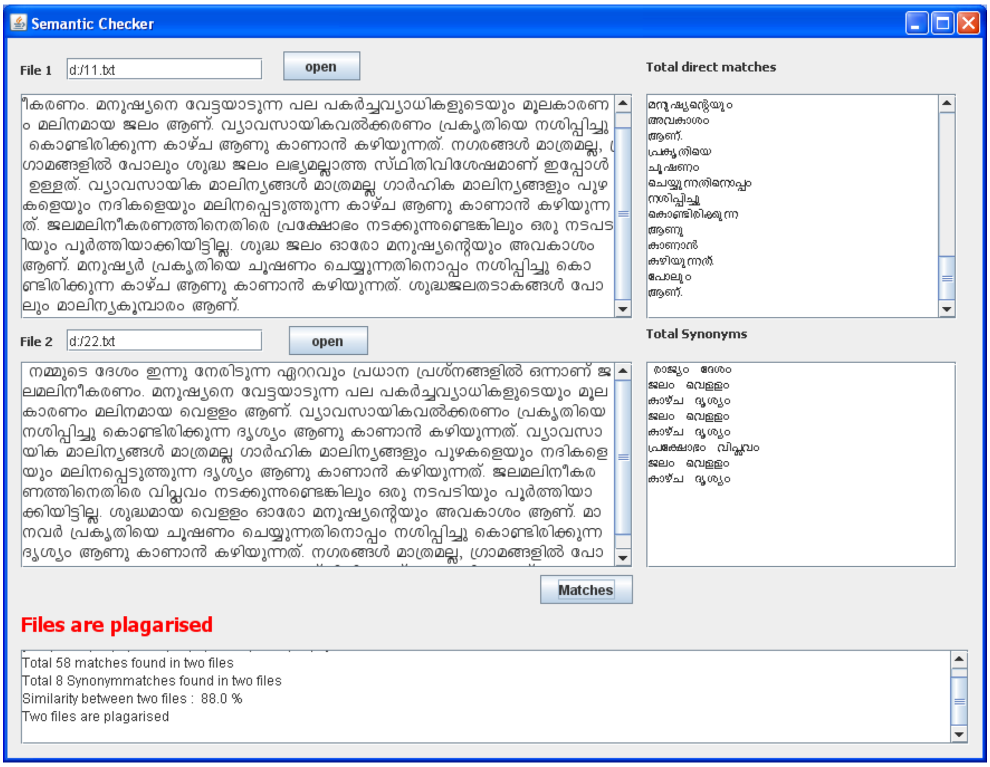


ഒരു കാട്ടിൽ കുറേ മാനുകൾ ഉണ്ടായിരുന്നു. ഒരു ദിവസം അവിടെ തീ പിടിച്ചു.

ഒരു Modifier one
കാട് Noun ട്ടിൽ forest locative singular
കുറേ Modifier many/much
മാൻ Noun നുകൾ deer plural
ഉണ്ടായിരുന്നു Verb was there
ഒരു Modifier one
ദിവസം Noun day in Nominative form
അവിടെ Modifier there
തീ Noun fire in Nominative form
പിടി Verb ച്ചു to catch past tense

Appendix

E SCREEN SHOT – SYNONYM REPLACEMENT



File 1 d:/11.txt

ീകരണം. മനുഷ്യനെ വേട്ടയാടുന്ന പല പകർച്ചവ്യാധികളുടെയും മൂലകാരണ
ം മലിനമായ ജലം ആണ്. വ്യാവസായികവൽക്കരണം പ്രകൃതിയെ നശിപ്പിച്ചു
കൊണ്ടിരിക്കുന്ന കാഴ്ച ആണു കാണാൻ കഴിയുന്നത്. നഗരങ്ങൾ മാത്രമല്ല, ഗ്ര
ഗാമങ്ങളിൽ പോലും ശുദ്ധ ജലം ലഭ്യമല്ലാത്ത സ്ഥിതിവിശേഷമാണ് ഇപ്പോൾ
ഉള്ളത്. വ്യാവസായിക മാലിന്യങ്ങൾ മാത്രമല്ല ഗാർഹിക മാലിന്യങ്ങളും പുഴ
കളെയും നദികളെയും മലിനപ്പെടുത്തുന്ന കാഴ്ച ആണു കാണാൻ കഴിയുന്ന
ത്. ജലമലിനീകരണത്തിനെതിരെ പ്രക്ഷോഭം നടക്കുന്നുണ്ടെങ്കിലും ഒരു നടപട
ിയും പൂർത്തിയാക്കിയിട്ടില്ല. ശുദ്ധ ജലം ഓരോ മനുഷ്യന്റെയും അവകാശം
ആണ്. മനുഷ്യർ പ്രകൃതിയെ ചൂഷണം ചെയ്യുന്നതിനൊപ്പം നശിപ്പിച്ചു കൊ
ണ്ടിരിക്കുന്ന കാഴ്ച ആണു കാണാൻ കഴിയുന്നത്. ശുദ്ധജലതടാകങ്ങൾ പോ
ലും മാലിന്യകൂമ്പാരം ആണ്.

File 2 d:/22.txt

നമ്മുടെ ദേശം ഇന്നു നേരിടുന്ന ഏററവും പ്രധാന പ്രശ്നങ്ങളിൽ ഒന്നാണ് ജ
ലമലിനീകരണം. മനുഷ്യനെ വേട്ടയാടുന്ന പല പകർച്ചവ്യാധികളുടെയും മൂല
കാരണം മലിനമായ വെള്ളം ആണ്. വ്യാവസായികവൽക്കരണം പ്രകൃതിയെ
നശിപ്പിച്ചു കൊണ്ടിരിക്കുന്ന ദൃശ്യം ആണു കാണാൻ കഴിയുന്നത്. വ്യാവസാ
യിക മാലിന്യങ്ങൾ മാത്രമല്ല ഗാർഹിക മാലിന്യങ്ങളും പുഴകളെയും നദികളെ
യും മലിനപ്പെടുത്തുന്ന ദൃശ്യം ആണു കാണാൻ കഴിയുന്നത്. ജലമലിനീകര
ണത്തിനെതിരെ വിപ്ലവം നടക്കുന്നുണ്ടെങ്കിലും ഒരു നടപടിയും പൂർത്തിയാ
ക്കിയിട്ടില്ല. ശുദ്ധമായ വെള്ളം ഓരോ മനുഷ്യന്റെയും അവകാശം ആണ്. മാ
നവർ പ്രകൃതിയെ ചൂഷണം ചെയ്യുന്നതിനൊപ്പം നശിപ്പിച്ചു കൊണ്ടിരിക്കുന്ന
ദൃശ്യം ആണു കാണാൻ കഴിയുന്നത്. നഗരങ്ങൾ മാത്രമല്ല, ഗ്രാമങ്ങളിൽ പോ

Total direct matches

മനുഷ്യന്റെയും
അവകാശം
ആണ്.
പ്രകൃതിയെ
ചൂഷണം
ചെയ്യുന്നതിനൊപ്പം
നശിപ്പിച്ചു
കൊണ്ടിരിക്കുന്ന
ആണു
കാണാൻ
കഴിയുന്നത്.
പോലും
ആണ്.

Total Synonyms

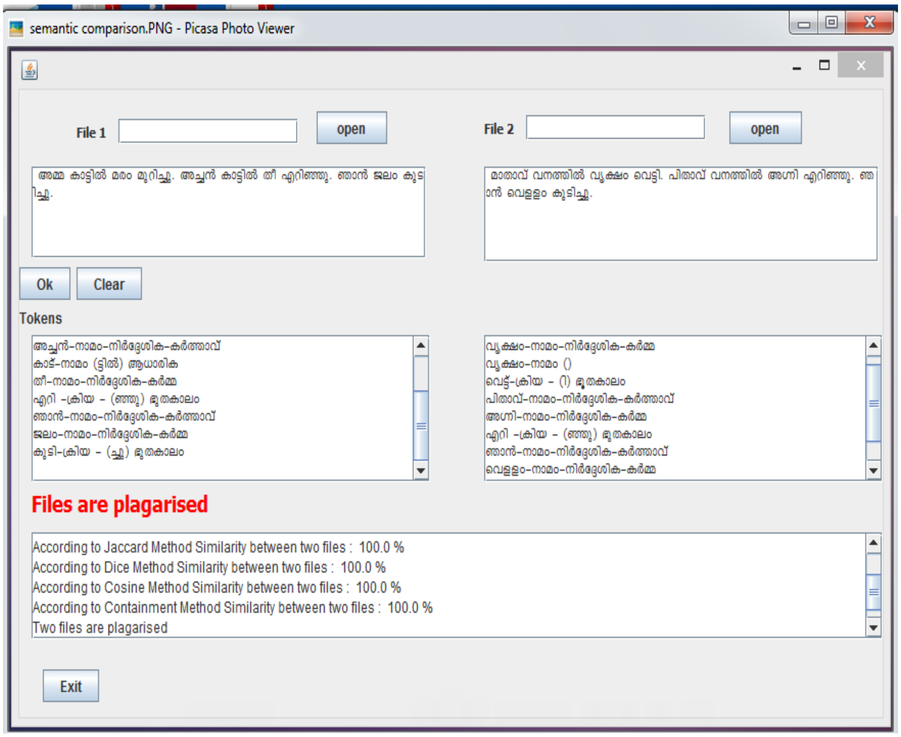
രാജ്യം ദേശം
ജലം വെള്ളം
കാഴ്ച ദൃശ്യം
ജലം വെള്ളം
കാഴ്ച ദൃശ്യം
പ്രക്ഷോഭം വിപ്ലവം
ജലം വെള്ളം
കാഴ്ച ദൃശ്യം

Files are plagarised

```
Total 58 matches found in two files
Total 8 Synonymmatches found in two files
Similarity between two files :  88.0 %
Two files are plagarised
```

Appendix F

SCREEN SHOT – SEMANTIC ROLE LABELLING



അമ്മ കാട്ടിൽ മരം മുറിച്ചു. അച്ചൻ കാട്ടിൽ തീ എറിഞ്ഞു. ഞാൻ ജലം കുടിച്ചു.

മാതാവ് വനത്തിൽ വൃക്ഷം വെട്ടി. പിതാവ് വനത്തിൽ അഗ്നി എറിഞ്ഞു. ഞാൻ വെള്ളം കുടിച്ചു.

Tokens

```
അച്ചൻ-നാമം-നിർദ്ദേശിക-കർത്താവ്
കാട്-നാമം (ട്ടിൽ) ആധാരിക
തീ-നാമം-നിർദ്ദേശിക-കർമ്മ
എറി -ക്രിയ - (ഞ്ഞു) ഭൂതകാലം
ഞാൻ-നാമം-നിർദ്ദേശിക-കർത്താവ്
ജലം-നാമം-നിർദ്ദേശിക-കർമ്മ
കുടി-ക്രിയ - (ച്ചു) ഭൂതകാലം
```

```
വൃക്ഷം-നാമം-നിർദ്ദേശിക-കർമ്മ
വൃക്ഷം-നാമം ()
വെട്ട്-ക്രിയ - (ി) ഭൂതകാലം
പിതാവ്-നാമം-നിർദ്ദേശിക-കർത്താവ്
അഗ്നി-നാമം-നിർദ്ദേശിക-കർമ്മ
എറി -ക്രിയ - (ഞ്ഞു) ഭൂതകാലം
ഞാൻ-നാമം-നിർദ്ദേശിക-കർത്താവ്
വെള്ളം-നാമം-നിർദ്ദേശിക-കർമ്മ
```

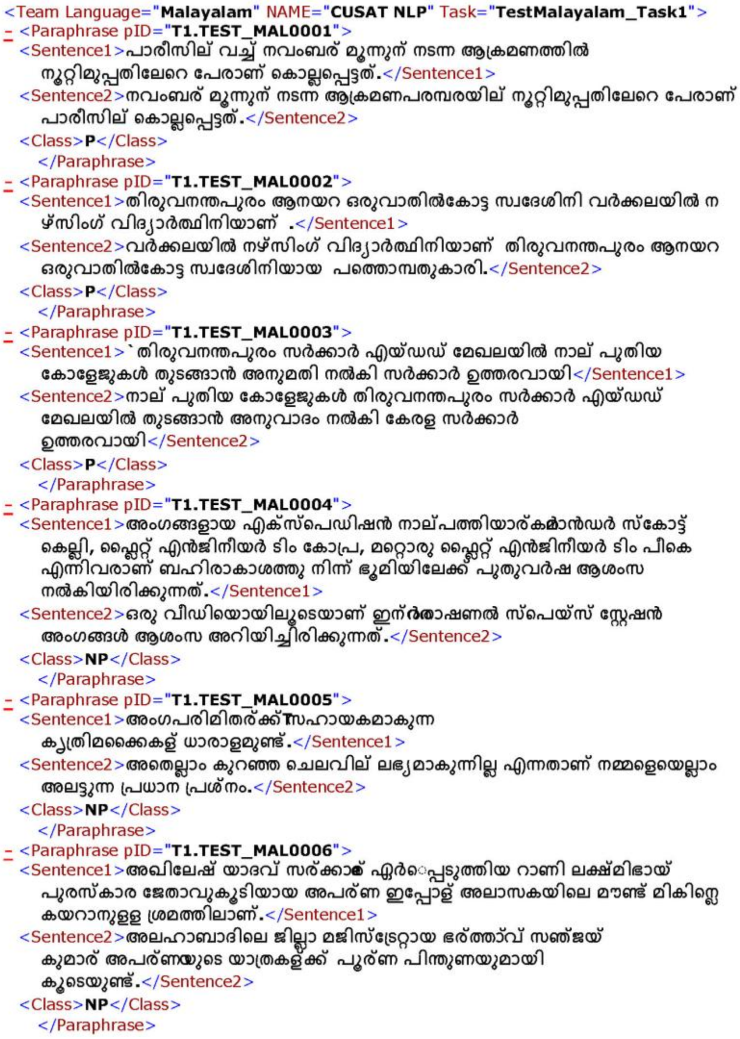
Files are plagarised

```
According to Jaccard Method Similarity between two files :  100.0 %
According to Dice Method Similarity between two files :  100.0 %
According to Cosine Method Similarity between two files :  100.0 %
According to Containment Method Similarity between two files :  100.0 %
Two files are plagarised
```

Appendix

G PARAPHRASE CORPUS

(DPIL shared task at FIRE 2016)



```
<Team Language="Malayalam" NAME="CUSAT NLP" Task="TestMalayalam_Task1">
- <Paraphrase pID="T1.TEST_MAL0001">
  <Sentence1>പാരീസില് വച്ച് നവംബര് മൂന്നുന് നടന്ന ആക്രമണത്തിൽ
    നൂറ്റിമുപ്പതിലേറെ പേരാണ് കൊല്ലപ്പെട്ടത്.</Sentence1>
  <Sentence2>നവംബര് മൂന്നുന് നടന്ന ആക്രമണപരമ്പരയില് നൂറ്റിമുപ്പതിലേറെ പേരാണ്
    പാരീസില് കൊല്ലപ്പെട്ടത്.</Sentence2>
  <Class>P</Class>
    </Paraphrase>
- <Paraphrase pID="T1.TEST_MAL0002">
  <Sentence1>തിരുവനന്തപുരം ആനയറ ഒരുവാതിൽകോട്ട സ്വദേശിനി വർക്കലയിൽ ന
    ഴ്സിംഗ് വിദ്യാർത്ഥിനിയാണ് .</Sentence1>
  <Sentence2>വർക്കലയിൽ നഴ്സിംഗ് വിദ്യാർത്ഥിനിയാണ്  തിരുവനന്തപുരം ആനയറ
    ഒരുവാതിൽകോട്ട സ്വദേശിനിയായ  പത്തൊമ്പതുകാരി.</Sentence2>
  <Class>P</Class>
    </Paraphrase>
- <Paraphrase pID="T1.TEST_MAL0003">
  <Sentence1>`തിരുവനന്തപുരം സർക്കാർ എയ്ഡഡ് മേഖലയിൽ നാല് പുതിയ
    കോളേജുകൾ തുടങ്ങാൻ അനുമതി നൽകി സർക്കാർ ഉത്തരവായി</Sentence1>
  <Sentence2>നാല് പുതിയ കോളേജുകൾ തിരുവനന്തപുരം സർക്കാർ എയ്ഡഡ്
    മേഖലയിൽ തുടങ്ങാൻ അനുവാദം നൽകി കേരള സർക്കാർ
    ഉത്തരവായി</Sentence2>
  <Class>P</Class>
    </Paraphrase>
- <Paraphrase pID="T1.TEST_MAL0004">
  <Sentence1>അംഗങ്ങളായ എക്സ്പെഡിഷൻ നാല്പത്തിയാര്കമാൻഡർ സ്കോട്ട്
    കെല്ലി, ഫ്ലൈറ്റ് എൻജിനീയർ ടിം കോപ്ര, മറ്റൊരു ഫ്ലൈറ്റ് എൻജിനീയർ ടിം പീകെ
    എന്നിവരാണ് ബഹിരാകാശത്തു നിന്ന് ഭൂമിയിലേക്ക് പുതുവർഷ ആശംസ
    നൽകിയിരിക്കുന്നത്.</Sentence1>
  <Sentence2>ഒരു വീഡിയൊയിലൂടെയാണ് ഇന്ർനാഷണൽ സ്പെയ്സ് സ്റ്റേഷൻ
    അംഗങ്ങൾ ആശംസ അറിയിച്ചിരിക്കുന്നത്.</Sentence2>
  <Class>NP</Class>
    </Paraphrase>
- <Paraphrase pID="T1.TEST_MAL0005">
  <Sentence1>അംഗപരിമിതര്ക്ക്സഹായകമാകുന്ന
    കൃത്രിമക്കൈകള് ധാരാളമുണ്ട്.</Sentence1>
  <Sentence2>അതെല്ലാം കുറഞ്ഞ ചെലവില് ലഭ്യമാകുന്നില്ല എന്നതാണ് നമ്മളെയെല്ലാം
    അലട്ടുന്ന പ്രധാന പ്രശ്നം.</Sentence2>
  <Class>NP</Class>
    </Paraphrase>
- <Paraphrase pID="T1.TEST_MAL0006">
  <Sentence1>അഖിലേഷ് യാദവ് സര്ക്കാര് ഏർപ്പെടുത്തിയ റാണി ലക്ഷ്മിഭായ്
    പുരസ്കാര ജേതാവുകൂടിയായ അപര്ണ ഇപ്പോള് അലാസകയിലെ മൗണ്ട് മികിന്ലെ
    കയറാനുള്ള ശ്രമത്തിലാണ്.</Sentence1>
  <Sentence2>അലഹാബാദിലെ ജില്ലാ മജിസ്ട്രേറ്റായ ഭര്ത്താവ് സഞ്ജയ്
    കുമാര് അപര്ണയുടെ യാത്രകള്ക്ക് പൂര്ണ പിന്തുണയുമായി
    കൂടെയുണ്ട്.</Sentence2>
  <Class>NP</Class>
    </Paraphrase>
```