SRL based Plagiarism Detection System for Malayalam Documents

Sindhu L[1], Sumam Mary Idicula[2]

**[1] Computer Science Department, College of Engineering, Poonjar
India**

**[2] Department of Computer Science, Cochin University of Science and Technology
Cochin, India**

Abstract

Automatic techniques of measuring plagiarism between documents have gained importance in the recent years because of the availability of enormous volume of information over the internet. . The most general form of detecting plagiarism is by computing similarity between a source document and a possibly plagiarised document. Existing plagiarism detection systems are mainly designed for detection in English.. Moreover, plagiarism detection systems using natural language processing techniques are still very limited. Automated plagiarism detection systems so far have involved minimal syntactic and semantic linguistic techniques. Even though, in some systems shallow techniques have been included as part of the pre-processing stage, studies involving deep techniques are less. Very negligible research has been done for plagiarism detection in Malayalam text documents . This paper presents a method for plagiarism detection in Malayalam documents based on extracting the semantic roles and computing their similarity to detect plagiarism. The technique can detect documents created by direct copy methods, replacement of words with similar ones , changing the order of words or restructuring the sentences and also converting the sentence from active/ passive to passive/active.

Keywords: *Plagiarism detection, semantic role labelling, Malayalam, Karaka relations.*

1. Introduction

Plagiarism has become a common issue because of the availability of many documents on the internet which can easily be accessed. So anybody can use the information from these documents to create new documents. Documents created by copying and pasting from the existing documents without acknowledging the original are identified as plagiarised. Automated plagiarism detection is a wide research area spanning many fields like journalism, student assignments, scientific, engineering and other related documents.

Plagiarism can be defined as taking the ideas or words from a source without giving proper credit is an act of plagiarism. Plagiarism may be committed at different levels. At the lowest level, a plagiarist may add or delete words to the original sentence to create a text. . He may also cleverly replace some words with similar words or he may change the words or syntax of the original. Therefore, plagiarism is a very complex problem and has to be effectively recognised.

Plagiarism detection is the process of identifying a suspicious document by analyzing some of its features which are syntactic ,structural, semantic, or lexical in nature. Plagiarism detection can be categorized as intrinsic and extrinsic plagiarism detection . Whereas the intrinsic plagiarism detection focuses on determining plagiarism by analyzing the changes in writing style within the same document itself, the extrinsic plagiarism detection focuses on classifying plagiarized and original documents by using a reference document collection. The majority of the detection systems implemented rely only on exact-word or phrase matching to identify plagiarism. Their performance is not adequate to detect in instances of clever plagiarism by paraphrasing or words reordering.

The rest of the paper is structured as follows. Section 2 presents the related works in plagiarism detection. Existing approaches with and without natural language processing steps are briefed in this section. Section 3 details on the rules underlying semantic role labelling in Malayalam.. It discusses on how a noun is labelled based on it's relation to a verb. Section 4 describes the various stages of the architecture of our proposed system. In Section 5 we give a detailed explanation on the data set used and the results analysis. In the last section 6 we state our conclusions.

.

2. Related works

A lot of text plagiarism detection methodologies have been been implemented over the years and some commercial tools are also available. However most of the detection approaches are not based on natural language processing techniques. Subsequent to the suggestion by Clough(2003), that paraphrased texts can be more easily and accurately detected using techniques incorporating NLP, more study has been made in this area.

When a large number of documents are involved in the detection process, two tasks are important . Preprocessing is

done for the generalisation of texts and candidate filtering optimises the performance by reducing the search span . In the preprocessing stage, shallow NLP tasks like tokenisation to determine token boundaries, lowercasing of letters, stopword removal to remove articles,prepositions etc, stemming to convert words to their stems are done.. A plagiarism detection system for Slovak texts was proposed by Chuda and Navrat (2010) . The preprocessing done in their application were breaking sentences to tokens, eliminating common words known as stopwords and converting words to their stems.

In the method proposed by Ceska and Fox (2009), they have incorporated latent semantic analysis together with the text preprocessing tasks. N-gram matching with singular value decomposition is used for finding the similarity. It consists of easy tasks like removing numbers and punctuations , applying Natural Language Processesing tasks such as eliminating stopwords, lemmatisation, and including a thesaurus. The use of NLP in this technique did not show much improvement with respect to the word n-gram overlap approach because of the limitations of the NLP procedures used. Moreover, their corpora was small and the disambiguation methods were not efficient for generalising words.

Deep Natural Language Processesing procedures can be used to analyze the syntax of texts. Using parse trees to study the structural relations between documents was suggested by Leung and Chan (2007) and Mozgovoy et al(2007). Uzuner et al. (2005) proposed shallow semantic and syntactic rules to detect paraphrasing in text. A part-of-speech tagger is used to determine the semantic class of each verb, that is a group of verbs which are similar in meaning, and also the syntactic structures are identified for each sentence. The similarity matching is then based on the verb classes, and matching is done on synonyms that remain in the same word order. The results obtained from the experiments showed that syntactic features better than tf-idf, and also that linguistic techniques identified paraphrases better than statistical methods. Translated texts from 49 books with different levels of paraphrasing were used in the experiment.

A method described by Mozgovoy et aI. (2006) applies preprocessing and NLP techniques for plagiarism detection in the Russian language which includes tokenising, converting words to their hierarchical class names and then extracting the functional words and argumentative words for matching. Mozgovoy (2007) suggests that string matching algorithms can be improved by including tokenisation and syntactic parsing into document copy detection . In Mozgovoy et aI. (2007), they proposed the use of natural language parsers. The process of detection works in two stages. In the first stage, the Stanford Parser parses all the documents and it generates the grammatical relations . In the second stage, the similarity is computed between the documents based on the results of the first stage. The final results of the experiments showed that even though parsing may discover sentence re-ordering, it is not capable of detecting the paraphrasing. Moreover, after parsing ,the original ordering of words in every sentence is changed .Hence their detection system cannot highlight similar segments of text. They used a corpus based on journalism text reuse in their experiment .

In the method by Leung and ChanI (2007) , sentences are compared at semantic level . They suggested the application of both shallow NLP and deep NLP which involves synonym generalisation and extraction of syntactic structure. Semantic processing involves converting parse tree of a sentence into case grammar structure in order to identify the deep structure. However, the non availability of semantic analysis tools and a suitable corpus have been a restriction on evaluating the actual performance of the method.

Ceska (2009) performed experiments using a Czech thesaurus. Alzahrani and Salim (2010) used WordNet which is semantically structured and gives information on relationships between words which can be used for the matching of synonyms and hyponyms. For most words, WordNet has a set of synsets which is a group of synonyms.

In the experiments conducted by Chen et al. (20I0), they used WordNet for substituting words with their synonym, hyponym and hypernym and included these into a metric called ROUGE (Lin, 2004).

In Chong et al. (2010) both shallow and deep NLP techniques were used in an experiment using small texts. Procedures such as chunking and parsing, are compared against an overlapping word 3-gram baseline. In addition, language models are applied to generate probabilities for word n-grams, perplexities and out of lexicon rates. A similarity metric such as the Jaccard coefficient, is applied to the extracted features to generate similarity scores for use in the machine learning algorithm. The results showed that the best performing features included a combination of word n-grams(3grams), lemmatisation, language model perplexities and parsing.

Chong and Specia (2011) explored lexical generalisation for word-level matching in identifying plagiarism. Here ,lexical generalisation replaces each content word with the set of it's synonyms. The purpose of this is to deal with paraphrased plagiarism. This differs from other similar works in that , the technique does not incorporate any Word Sense Disambiguation. Similarity check is carried out at the word level, which disregards the ordering of words, and the results were compared against an overlapping word n-gram(5-gram) metric. The results of the experiment proved that lexical generalisation reduces the false negatives and improves recall .

In Osman et al. (2012) semantic role labelling has been used for identifying plagiarism . The arguments of sentences are extracted using SRL and compared . Arguments weighting was done and only the important arguments were used in the similarity calculation process. Experiments conducted on PAN-PC-09 data sets showed that their method performs better than modern semantic methods for plagiarism detection in when evaluated for Recall, Precision and F-measure.

3. SRL for Malayalam

3.1 Malayalam language characteristics

Malayalam , a language spoken in south India, is both an agglutinative as well as a n inflectional language. Based on the tense, number, gender etc, the root word is inflected , to produce new words. Furthermore two or more words can combine together to form a single compound word. 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 kaaraka 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.

3.2 Vibhakthi to Kaaraka mapping

Case endings differentiate the vibhakthis. 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 Table1 .

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

Table 1. Vibhakthi – Kaaraka relation

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

Karthaav:- Subject of the sentence has nirdesika (nominative) as Vibhakthi in active voice.
Eg. Ramu vannu. (Ramu came.)
Karmam:- Object of the sentence has Prathigrahika (accusative) as Vibhakthi in active voice.
Eg. Avan Ramuvine adichu. (He beat Ramu)
Saakshi Kaarakam:- 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)
Swaami Kaarakam:- 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)

4. Plagiarism Detection Using Srl - Proposed Method

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 labeling 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 1 shows the architecture of the proposed system.

4.1 Preprocessing

Pre-processing is an essential step in Natural Language Processing tasks.

4.1.1 Text segmentation

The first step in pre-processing is dividing the text into segments. The text is split into sentences and then into words. Our system compares a suspected text with original text based at the sentence level.

4.1.2 Stop words removal

Stopwords are those words that occur frequently in documentt. They do not contain any substantial meaning and so can be deleted without compromising the significance of the text. Since no stop words list is available for Malayalam language, one was created based on which the stop words were deleted from the documents. As a result the speed of processing and accuracy of the system is increased and it also saves memory space.

4.1.3 Lemmatization and POS tagging

Lemmatization ia a very important step in the processing of Malayalam words. This is because Malayalam is a highly agglutinative language and highly complex words are formed by the continuous addition of suffixes to the root(base) word. These various word forms of the same root can affect the accuracy while matching. Lemmatization on the words to obtain the root words and the root words are used for further processing. The root words are tagged as either noun, verb, adjective or adverb using a rule-based tagger.

4.2 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).

4.3 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.

4.4 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.

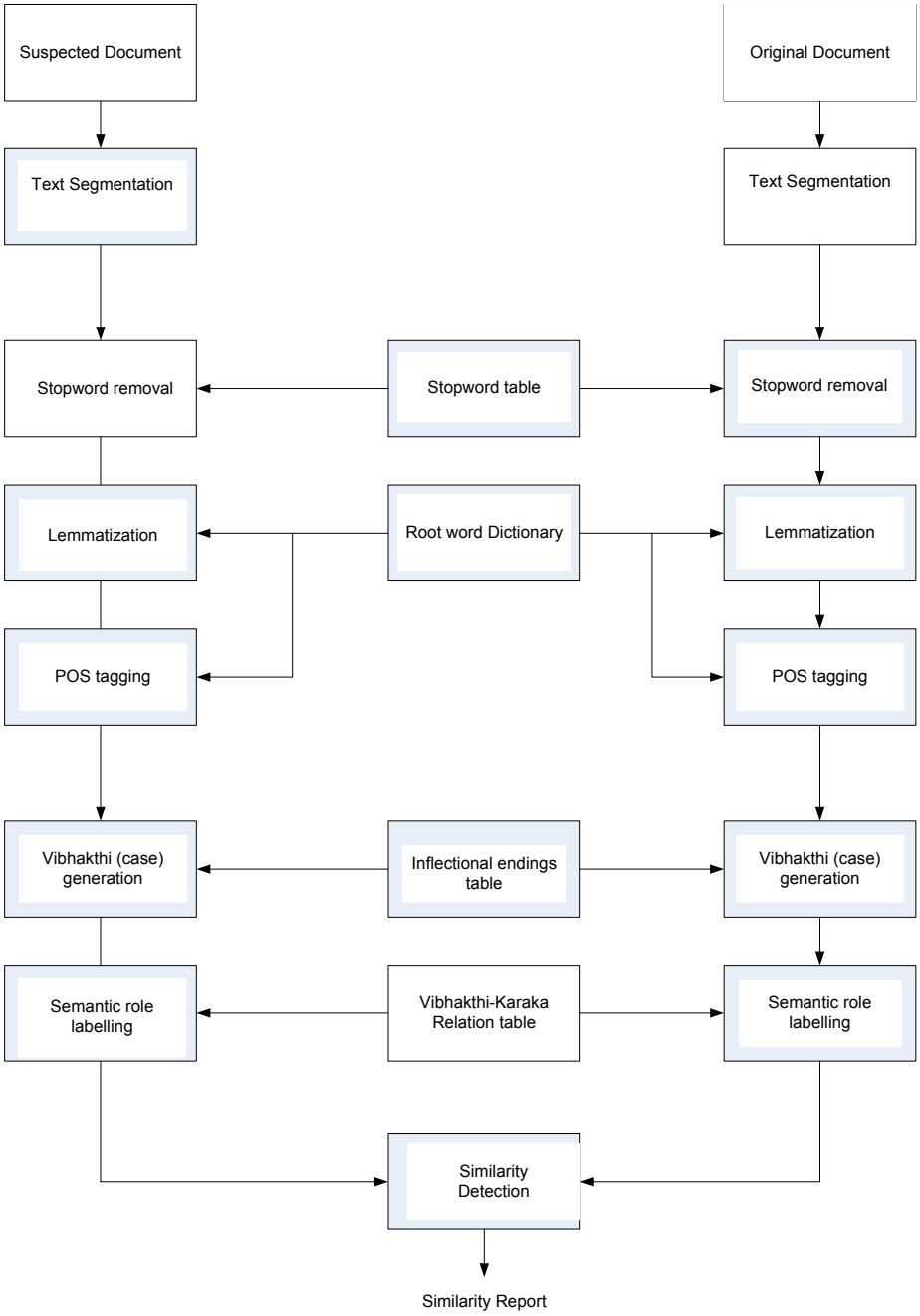

Fig1. Architecture of the proposed method

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

Let Arg(S_a) be the set of arguments in the sentence S_a in the suspected document and

Let Arg(S_b) be the set of arguments in the sentence S_b in the original document

$$\text{Jaccard}(S_a,S_b) = \frac{|Arg(Sa) \cap Arg(Sb)|}{|Arg(Sa) \cup Arg(Sb)|} \quad (1)$$

5. 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, The correctly classified plagiarised texts (True Positives: TP), correctly classified original texts (True Negatives: TN), original texts incorrectly classified as plagiarised (False Positives: FP), plagiarised texts incorrectly classified as original (False Negatives: FN) are used in the standard calculation of precision., recall., F-score., and accuracy as follows:

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

Precision calculates the the number of correctly identified sentences as plagiarised, normalised by the total number of texts both correctly and incorrectly identified as plagiarised.

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

Recall calculates the number of correctly identified sentences as plagiarised , normalised by the total number of sentences that have been correctly identified and those that have not been identified as plagiarised, but are actually plagiarised.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

F-measure is the harmonic mean of precision and recall.

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN}$$

Accuracy gives the proportion of the total number of correctly identified sentences over all the sets.

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

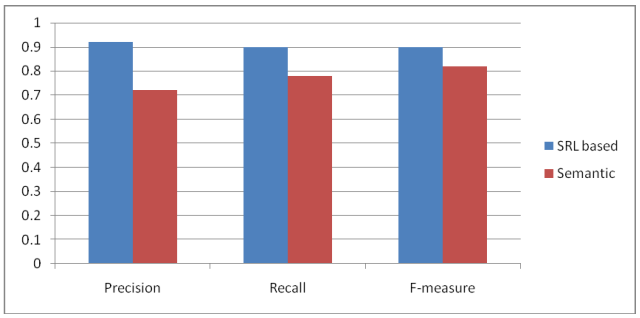

Fig2. Comparison of SRL based and semantic similarity

6. Conclusions and Future Work

A plagiarism detection method for Malayalam text documents based on semantic role labelling was presented. Though SRL based plagiarism detection has been experimented for English, this is the first of its kind for Malayalam. The arguments of the sentences were extracted and corresponding arguments were compared at the sentence level. Tests were conducted on texts extracted from online Malayalam newspapers since a standard dataset is not available for the language. The proposed system is capable of detecting direct text copy, copy with words replaced by synonyms , words reordering and also copying by changing the voice of the sentence.

The efficiency of the system can further be enhanced by incorporating a more elaborate tool to assist SRL and also the system can be tested on a standard corpus.

References

[1] A.H. Osman, N. Salim M.S An improved plagiarism detection scheme based on semantic role labelling, in Journal of Applied Soft Computing Elsevier vol:12, p. 1493-1502, 2012.

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

[3] Naomie Salim, Ahmed Hamza Osman, Plagiarism Detection Scheme Based on Semantic Role Labeling, International conference march 2012.

[4]. C. J. Fillmore, The case for case. In Emmon Bach and Robert T, *Universals in Linguistic Theory. Holt, Rinehart, and Winston, NewYork,* p.1–210, 1968.
[5]. C. F. Baker*, et al.*, The Berkeley FrameNet Project, presented at the Proceedings of the 17th international conference on Computational linguistics - Volume 1, Montreal, Quebec, Canada, 1998.
[6]. M. Palmer*, et al.*, The Proposition Bank: An Annotated Corpus of Semantic Roles, *Comput. Linguist.,* vol. 31, p. 71-106, 2005.
[7]. A. Si, H. V Leong, and R. W. Lau. CHECK: A Document Plagiarism Detection System. In Proceedings of ACM Symposium for Applied Computing, pages 70-77, February 1997.
[8]. Chong, B. M., Specia, L., & Mitkov, R. (2010).A Study on Plagiarism Detection and Plagiarism Direction Identification Using Natural Language Processing Techniques. In Proceedings of the 4th international plagiarism conference. Newcastleupon- Tyne, UK.
[9]. Maxim Mozgovoy, Tuomo Kakkonen, and Erkki Sutinen. Using natural language parsers in plagiarism detection. In Proceedings of the Workshop on Spoken.
[10]. Maxim Mozgovoy, Vitaly Tusov, and Vitaly Klyuev. The Use of Machine Semantic Analysis in Plagiarism Detection. In Proceedings of the 9th International Conference on Humans and Computers, pages 72-77, Aizu-Wakamatsu, Japan, 2006.
[11]. Chien-Ying, C., Jen-Yuan, Y., & Hao-Ren, K. (2010). Plagiarism detection using ROUGE and WordNet. Journal of Computing, 2(3), 34-44.
[12]. Ceska, Z., & Fox, C. (2009). The Influence of Text Preprocessing on Plagiarism Detection. In Recent Advance in Natural Language Processing, RANLP '09.
[13].
[14]. Asim M. El Tahir Ali, Hussam M. Dahwa Abdulla, Vaclav Snasel Survey of Plagiarism Detection Methods IEEE 2011 39-42.
[15]. Clough, P. (2003). Old and new challenges in automatic plagiarism detection. National Plagiarism Advisory Service, 391-407.
[16]. Fernando Sanchez-Veg, Esau Villatoro-Tello, Manuel Montesy-Gomez, Luis Villasenor-Pineda, Paolo Rosso. Determining and characterizing the reused text for plagiarism detection, Expert Systems with Applications 40 (2013) 1804-1813 Elsevier.
[17]. Chong, B. M., Specia, L., & Mitkov, R. (2010).Using natural language processing for automatic detection of plagiarism. In Proceedings of the 4th international plagiarism conference. Newcastle-upon-Tyne, UK.
[18].S.M. Alzahrani, N. Salim, A. Abraham, Understanding Plagiarism linguistic patterns, textual features, and detection methods, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews (99) (2011)
[19]. Sindhu.L, Bindu Baby Thomas and Sumam Mary Idicula. Article: Automated Plagiarism Detection System for Malayalam Text Documents. *International Journal of Computer Applications* 106(15):13-16, November 2014.
[20] Sindhu.L, Bindu Baby Thomas and Sumam Mary Idicula. "A Copy detection Method for Malayalam Text Documents using N-grams Model." (2013).

First Author Sindhu.L is working as Assistant Professor at College of Engineering, Poonjar. She took B.E degree in Computer Engineering from Madurai Kamaraj University and M.Tech in Software Engineering from Cochin University of Science and Technology.

Second Author Prof . Dr. Sumam Mary Idicula is currently the Head of the Department of Computer Science at Cochin University of Science and Technology. Dr. Sumam Mary Idicula took B.Sc(Engg) degree in Electrical Engineering from College of Engineering Trivandrum and took M.Tech degree in Computer and Information Science from Cochin University of Science & Technology. She took PhD degree in Computer Science from the same department. She is an active researcher in the field of Natural Language Processing, Datamining and Human Computer Interaction