Jestr

Journal of Engineering Science and Technology Review 9 (4) (2016) 150 - 164

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Study on Extrinsic Text Plagiarism Detection Techniques and Tools

Vani K[1] and Deepa Gupta[2]

Department of Computer Science & Engineering, Amrita School of Engineering, Amrita University, Amrita Vishwa Vidyapeetham, Bangalore, India
Department of Mathematics, Amrita School of Engineering, Amrita University, Amrita Vishwa Vidyapeetham, Bangalore, India

Received 13 November 2015; Accepted 10 September 2016

Abstract

The swift evolution of technology has facilitated the access of information through different means which has opened the doors to plagiarism. In today's world of technological outburst, plagiarism is aggravating and has become a serious concern in academia, research and many other fields. To curb this intellectual theft and to ensure academic integrity, efficient software systems to detect them are in urgent need. In this paper, a study on plagiarism is done with the focus on extrinsic text plagiarism detection, which is a fast emerging research area in this domain. The different extrinsic detection techniques and the methodologies involved are reviewed based on the current state of art. Further an overview of some of the available detection software tools, their features and detection efficiency is discussed with some of the output demos. The paper also throws light on the popular PAN competition, which is conducted yearly since 2009 in plagiarism domain and the major tasks involved in it. Further it attempts to identify the problems existing in available tools and the research gaps where immense explorations can be done.

Keywords: Text Plagiarism; Extrinsic Plagiarism Detection; Obfuscations; Software Tools; PAN systems

1. Introduction

With the onset of World Wide Web (WWW), ingress to information has become much easier. Further the hasty developments in technology lead to the swift access of information through various search engines, digital libraries and other databases. This profusion of knowledge and information has lead to the breach of information content, which is generally termed as 'plagiarism'. In the early 17th century, the English word "Plagiarism" came as an evolution from the Greek word "Plagion", then to the Latin words "Plagium" and "Plagiarius" which means kidnapping and kidnapper respectively. The synonym list found for plagiarism is the following:-"copying, infringement of copyright, piracy, theft, staling, poaching, appropriation and informal cribbing". It is a "serious intellectual and academic transgression" dictionary.com [1]. According to Merriam-Webster online dictionary Webster [2] "Plagiarize" means:

- to steal and pass off (the ideas or words of another) as one's own
- to use (another's production) without crediting the source
- to commit literary theft
- to present as new and original an idea or product derived from an existing source.

Plagiarism is not only a serious issue in academia, but also in many other domains,viz., art, literature, journalism and so on. Plagiarism is usually defined as the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work" (Random House Compact Unabridged Dictionary, 1995; Oxford English Dictionary, 1999). WPA (http://wpacouncil.org/positions/WPAplagiarism.pdf) defines plagiarism as a multifaceted and ethically complex problem. It claims that current discussions fail to distinguish between intentional plagiarism and unintentional/ careless writings that lead to plagiarism. But a good writer always tries to keep up with rules and take all his efforts to follow the ethics and avoid plagiarism. A survey was done by Guo[3] focusing on student plagiarism mainly in accounting education. It concludes that educators must motivate the students to follow ethical ways of writing. A quantitative study was conducted by Newton [4] to study the academic dishonesty performed among students in higher education. Another survey conducted by Kauffman & Young [5] indicated that overall 79.5% of the writers are involved in digital plagiarism.

The restriction of access to knowledge and information is impossible. Thus to ensure the academic integrity and quality of research work, efficient detection systems is in its urgency. Plagiarism is categorized into text plagiarism and source code plagiarism based on the domain of application Bin-Habtoor and Zaher [6]. In source code plagiarism or generally termed as software plagiarism, the code segments are copied. The detection methods for these two plagiarisms are entirely different, since software plagiarism is more restricted. In other words, here the focus shifts to the language used, set of key words, coding structure etc. Text plagiarism on the other hand extends to various possibilities and obfuscation complexities and even inter-language plagiarism can happen here, i.e., cross-language plagiarism. The current work focuses on the study and analysis of text plagiarism.

In doing text plagiarism, a plagiarist tries to obfuscate or manipulate the text and present the content in different ways possible. In the simplest scenario, the content is copied as such and presented. Mainly students when submitting assignments and projects practice this. The type of plagiarism is termed as literal plagiarism /verbatim plagiarism. When it comes to more complex cases, the plagiarist manipulates the content in different ways to present it as his own original work and thus making the plagiarism detection even harder. These obfuscations fall under the category of intelligent/paraphrase plagiarism. Here the source contents are modified and obfuscated in different complex ways, viz., synonym substitutions, idea adoptions, translations, summarizations etc. This can be done either algorithmically or manually or as a combination of both [7]. The general classification of plagiarism types, detection systems and techniques is shown in Fig.1.

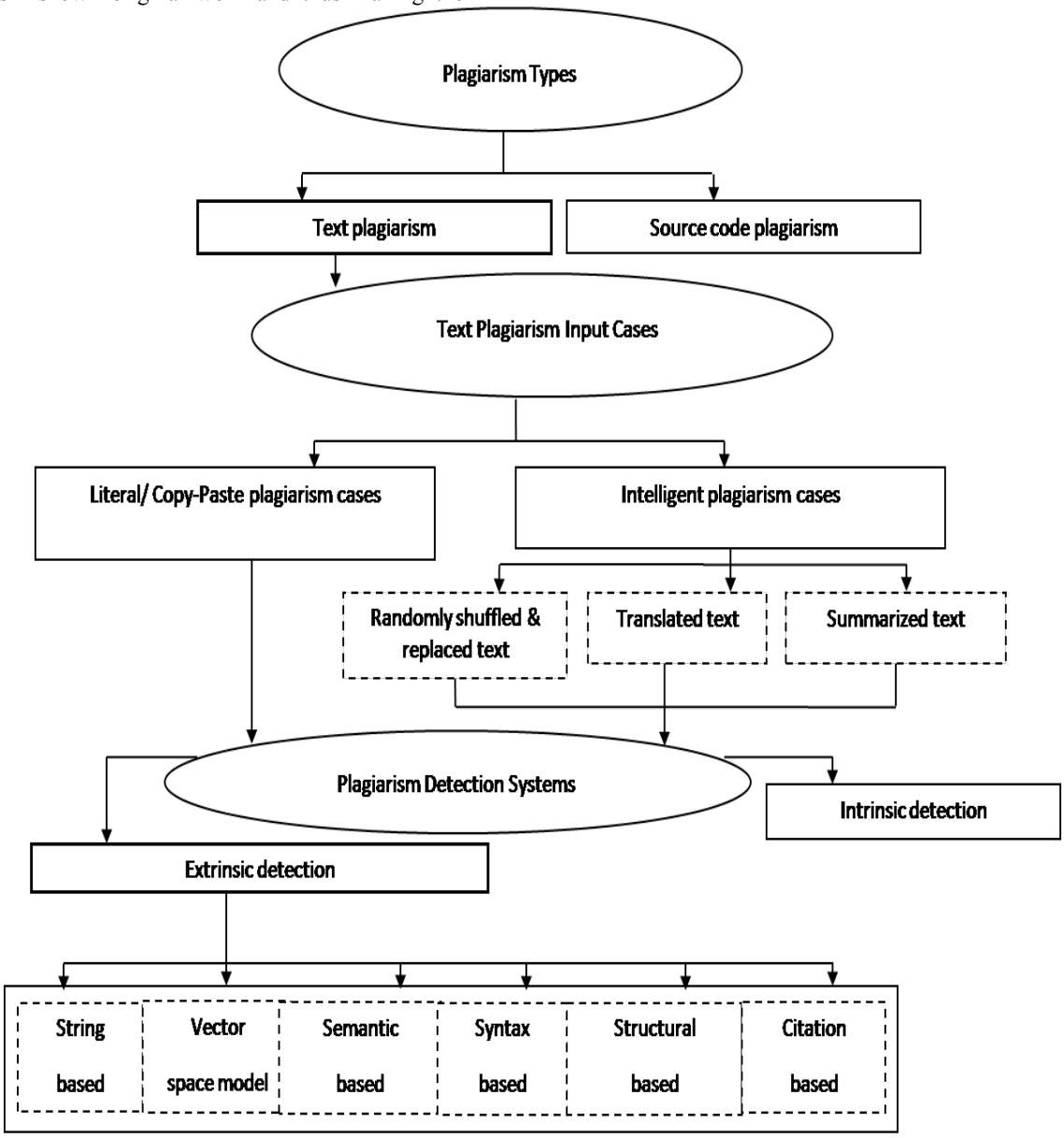

Fig.1. General Classification of Plagiarism Types, Detection Systems & Techniques

The main modules focused in our study are text plagiarism and extrinsic plagiarism detection systems (PDS) and the techniques involved. The degree of obfuscations is categorized in different ways. Broadly plagiarism types are divided as literal/ verbatim/ copy-paste plagiarism and intelligent / paraphrased plagiarism. The different types of input plagiarism cases that can be possibly fed into a plagiarism detection system or software are shown in Fig.1. Intelligent manipulations can be done in different ways such as shuffling of words, synonym replacements, translations, summarizations and various other means of idea adoptions and paraphrasing. The neologism for random manipulations by synonym substitutions is rogeting. Maurer, Kappe and Zaka [8] differentiate the types of plagiarism as copy-paste, paraphrasing, idea adoption, artistic plagiarism, translated and code plagiarism. The author also points out that always plagiarism is not an intentional act; it can be accidental where the person is unaware of proper means of citing and referencing or unintentional where he misses some information. Further it can be even a self-plagiarism where one's own work is published in some other form.

In text plagiarism detection, mainly two formal tasks are defined which are extrinsic/external detection and intrinsic/internal detection, which in turn defines the two types of PDS [9]. In the former, the suspected documents are compared against a reference source corpus. Unlike extrinsic

PDS, a reference corpus is unavailable for intrinsic PDS. Here the suspicious document is analyzed single-handedly without being compared with any sources. The writing styles of the author, structural distributions, vocabulary richness etc. are analyzed here [7]. Thus different stylometric features are extracted for identifying these plagiarism cases. In extrinsic PDS, various detection techniques can be employed, viz., string based, vector space model (VSM) based, syntax based, semantic based, structural based and citation based techniques or a combination of these techniques. The current study focuses on the extrinsic text plagiarism detection techniques, methodologies and its state of art. Further it analyzes the limitations of the current plagiarism checkers.

The study initially describes the stages employed in extrinsic plagiarism detection and then discusses the state of art in this domain based on the available detection techniques and systems. This is followed by the discussion of PAN (http://pan.webis.de/) plagiarism competition for providing an understanding about the different obfuscations or manipulations that can be imposed by the plagiarists. In the next section analysis of some of the online plagiarism tools is done using the text manipulated by the obfuscations described in PAN. Further the common problems and research gaps are pin-pointed and the discussion is concluded with insight to the future aspects.

2. Review of Extrinsic Plagiarism Detection Architecture

In an extrinsic PDS, the given suspicious document is compared against an available reference source document corpus or collection. This reference collection can be either online or offline, i.e., either the online sources in WWW or an offline database where the source documents are stored. Any detection system aims at finding the plagiarized suspicious passages and their corresponding counterparts in the available source document. Each input suspicious document is compared against the available sources to detect whether they are copied or manipulated from any of these reference documents. The source corpus or database can be the entire web, some specific libraries or databases particular to some domains and so on. With the availability of a database for comparison, it works more like a document comparison mechanism using some similarity schemes. Most of the online plagiarism checkers also work in a similar way and compares the suspected input to documents available in WWW or some data bases or a combination of both.

The general architecture of an extrinsic PDS is shown in Fig. 2. As shown in Fig. 2, the input suspicious document is compared against the reference sources, which can be either online / offline. Initially the documents are subjected to some pre-processing. In the offline case, when there are limited sources the reference documents may be also subjected to certain pre-processing. But as the size of reference corpus increases, mainly in case of online sources, say when the entire web needs to be crawled, an initial pre-processing of entire reference documents sounds tedious. Thus a heuristic retrieval procedure is employed that can identify the near duplicates which are referred as the candidate documents for the particular suspicious document at hand [9]. But in the general representation, pre-processing is followed by candidate retrieval. When online sources have to be searched, some query processing technique is used and this works similar to a search engine that outputs results related to the given query. Candidate retrieval reduces the search space and further the suspicious document needs to be compared only with their respective candidate set to detect the actual fragments or passages plagiarized. The detailed description of each stage and the techniques employed by available systems are given in following subsections.

2.1. Pre-Processing

The documents at hand are initially pre-processed, where the irrelevant information is removed which makes the document handling easier. These include techniques such as sentence segmentation, tokenization, stop word removal, punctuation removal, lowercasing etc. Natural language processing (NLP) techniques mainly stemming and lemmatization are also employed in this stage.

Based on the models or technique employed, pre-processing of the documents is done. If the technique used, performs sentence-based comparisons of documents, then sentence segmentation is performed. Here the document is divided into sentence units based on some rough sentence boundaries or applying some heuristics [10, 11, 12,13]. Tokenization considers a document at word-level by dividing it into tokens. Stop-word removal is carried out in most of the detection systems, which focus on intelligent plagiarism detection. Here content words or the words that convey some meaning is retained while the stop-words such as pre-positions, conjunctions, articles etc. are removed. But work that focus on stop-words are also reported in literature [14]. In his work, the content words are removed while stop words are retained to create stop-word N-gram profiles.

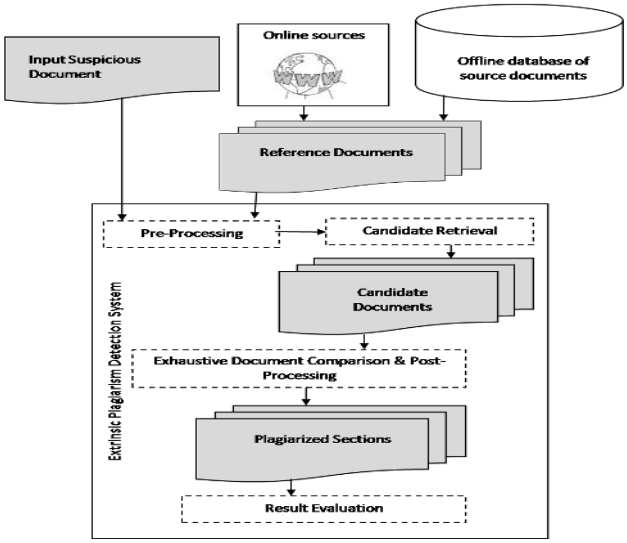

Fig.2. General Architecture of Extrinsic PDS

Further various NLP techniques are also employed for the effective document representation and handling [15, 12, 16, 17]. In pre-processing, the shallow procedures, viz., stemming or lemmatization are usually employed. Stemming is a heuristic process of removing the affixes from the words. Lemmatization produces the dictionary base forms of a word using vocabulary and morphology information. It is closely related to stemming but stemming operates only on a single word at a time while lemmatization operates on the full text. It can thus discriminate between words that have different meanings depending on part of speech [18].

2.2. Candidate Retrieval

After pre-processing the next important stage is the document-level plagiarism detection by retrieving the near duplicate sources. Usually in any practical scenario, for a detection system the suspected document has to be compared with large repositories or databases which can be some offline databases specific to an application or may be the entire web. In any case, the exhaustive comparison of a suspected document with all the documents in these databases will be quite time consuming. Thus to reduce this search space a document level comparison is done which retrieves the candidate sources for the given suspicious document at hand. In candidate retrieval task, the globally similar source documents with respect to a particular suspected document are retrieved. Thus each suspicious document is associated with a source set termed as candidate set. This process works similar to the information retrieval task in search engines, where the documents related to a particular query are retrieved.

Here two scenarios can be encountered where in the first case the reference source is an offline database and in the latter case wherein the entire web or some online databases are the references. In the former case we have a hermetic system where each suspicious document is compared at a document level with the each of the sources in offline database to retrieve the source set associated with it. This document level comparison is done using the different methods of document retrieval and similarity analysis. The second scenario is when the entire web or some online sources needed to be searched which is thus a web based system [19]. In this case, the method employed is similar to that of retrieval in search engines. Initially the suspected document is subjected for query formulation procedure. Different techniques for key phrase or key word extractions and query formulation are used here. Further the query processing is done through some search control mechanisms and then the related sources are retrieved [20, 21]. Even though in most of the practical scenarios, search of online sources are done, the other mechanism is also equally important. This is because many plagiarism checkers are employed specifically for certain applications, viz., plagiarism checking in student project reports with earlier reports, institutional reports, student thesis etc. where offline data bases can be employed. Candidate retrieval stage plays an important role in deciding the overall PDS efficiency. If the candidate retrieval is not done, then each suspicious document has to be compared exhaustively with all the available sources which will be quite time consuming. Further there will be many sources which are completely unrelated to the suspicious document at hand. Thus a document level comparison is always appreciated before the actual in depth comparisons.

Works for candidate document retrieval have been reported in both offline and online tasks. N-gram based models and Vector Space Models (VSM) are mainly used for this task. In N-gram models the documents are represented as word or character N-grams. To identify the candidate documents, similarity metrics such as jaccard, dice and overlap coefficient metrics are mainly employed [22, 23, 24]. Stop-word N-grams are used by Stamatatos [14] for candidate retrieval stage. Palkovskii and Belov [25] used sorted word 5-grams for the candidate retrieval task. In vector space model (VSM) based approaches initially document texts are represented in vector space. Further cosine similarity metric is used to find the candidate documents based on some defined thresholds [26, 27]. IR techniques, viz., clustering, IR ranking approaches and classification methods are also used in source document retrieval. A clustering based technique using K-means is proposed by Vani and Gupta [28] while a fuzzy clustering approach is experimented by Ravi, Vani and Gupta [29]. Machine learning (ML) based classifications are also employed in classifying plagiarized and non-plagiarized documents by viewing the task as a binary classification problem [15, 30]. Natural language processing techniques (NLP) is used for extracting dependency relations and various similarity scores are used for classification task by Chong [15]. Sánchez-Vega et al. [30] used various rewriting features, viz., overlapping degree, length of reused content and thematic of the rewritten text for identifying and classifying the reused text. These features are also extended for detecting the type of plagiarism imposed in the text thus modelling the task as a multi-class classification problem.

Similarly many detection systems are built to search over online resources using various techniques. As discussed, with this respect the main focus is on query formulation, where suspected document query is submitted to the given search engine API and further source retrieval is done. Different levels of document chunking, viz., line chunks, word chunks, sentence chunks or some combination of them are employed for retrieving near duplicate sources. A heterogeneous query formulation technique that combined key-word based, paragraph based and header-based queries for phrasal search is proposed by Suchomel and Brandejs [31]. Sentence and word chunking is mainly used here and further query ranking is done. Three different key phrase extractions is used by Elizalde [32], such as one query per 50- lines chunk containing the top 10 words scored by tf-idf values, first 8-gram with three words from 1 per chunk and 15 phrases based on head noun clusters. Noun phrases are extracted based on tf-idf values. In the download filtering process, the first 10 results are selected and those snippets with more than 90% of 4-grams as in suspected document are considered for retrieval. A query extraction method based on term frequency and word co-occurrence from a non-overlapping topically related sentence chunks is presented by Prakash and Saha[33]. A method that used paragraph chunks and tf-idf schemes with POS tagging for key word extraction is proposed by Ravi and Gupta [34].

Candidate retrieval task reduces the overall complexity of detection task, but at the same time implementing a well defined retrieval method is necessary. This is because any source document missed in this stage will not be accounted in the further stages also. Thus retrieving all the related source candidates is essential while maintaining the accuracy. When it comes to online sources, it is also important to reduce the overall costs of search engine usage. This means queries formulated must be limited but at the same time recall has to be maintained. Thus the candidate stage is an important building block of any PDS. Once the candidate documents are available, next stage is the exhaustive document comparison which can be considered as the heart of PDS.

2.3. Exhaustive Document Comparison & Post-Processing

Once the candidate documents are retrieved, each suspicious document is compared against its candidate set exhaustively. This is where the suspected plagiarized segments and their corresponding source components are identified. In detailed document comparison stage, each suspected document is compared against its source candidates using various methodologies and detection techniques. The comparisons

can be on different levels including sentence level, N-gram level, word level and phrase levels. In this phase, deep NLP techniques such as Part of Speech (POS tagging), Chunking, Semantic Role labelling (SRL), named Entity recognition (NER) and various other NLP and artificial intelligence (AI) techniques has to employed for improving the detection efficiency. The source and suspicious components are compared using some similarity measures and plagiarized fragments are selected. Once the fragments are obtained, post processing is done which mainly includes passage boundary detection phase. Here the deductions of source and suspicious passages are done based on certain boundary thresholds and some split-merge conditions. It is important that plagiarized passages must be retrieved as a whole and not as pieces. Finally the PDS is evaluated on some standard data sets and performance is measured using standard metrics. The PAN data sets and measures are popular and used widely for evaluating PDS efficiency [9].

The reported work for exhaustive comparison stage in extrinsic plagiarism detection based on the different technique categorizations as shown in Fig.1 is described in detail in next sections.

3. Exhaustive Document Comparison Stage- State of Art

The state of art in exhaustive detection stage of a PDS is analyzed and studied based on different techniques and methodologies presented by renowned authors. The discussion is categorized based on the features extracted for comparisons or the level at which the comparison is made. Different techniques available and utilized by detection tools are discussed in given subsections.

3.1. String based detection technique

This includes the simplest level of comparison where character level/ word level comparisons are made. Mainly N-gram based comparisons either character N-grams or word N-grams fall into this category. N-grams are the group of N consecutive words / characters formed from the document text.

Torrejón and Ramos [35] extracted the contextual and surrounding N-grams which are extended N-gram models. They used sorted word 3-grams and sorted word 1-skip-3-grams. The accuracy dropped as the paraphrasing complexity increased. Non-overlapping 250 character chunks are extracted by Kuppers and Conrad [36]. Then the word-based similarity is computed using the dice coefficient and a threshold is used to detect plagiarized fragments. The overall system performance was poor due to the extremely low recall. Shrestha and Solorio [23] presented a detection system that utilized variety of N-grams such as stop word N-grams, N-grams with at least one named entity, and all words N-grams. As the manipulations increased, the performance degraded especially in terms of recall. Palkovskii and Belov [25] used regular N-grams, variable length stop word N-grams, named entity N-grams and most frequently used N-grams. A graphical clustering algorithm was used to define clusters of shared fingerprints or N-grams. Alvi, Stevenson and Clough [37] used a character based N-gram model with Rabin-Karp string matching algorithm. Stamatatos [14] used stop word N-gram profiles. All these detection systems were effective for detecting plagiarism cases with simple copy-paste and intelligent plagiarism cases with small random shuffling while the efficiency of detection dropped as plagiarism complexity increased. In general, N-gram based models were found to be less effective when it comes to complex obfuscation types. But the exhibition of good precision shows its potential to be combined and used in hybrid approaches.

3.2. Vector Space Models (VSM)

This is one of the popular techniques which utilizes the lexical and syntactic features and represent the document in a vector space. Then different weighting schemes are adopted for document representations and comparisons. Mainly the two weighting schemes used are term frequency-inverse document frequency (tf-idf) and term frequency-inverse sentence frequency (tf-isf), where the former operates at document level and latter at sentence level. The former is used in both candidate retrieval and exhaustive analysis stage while tf-isf is mainly used in exhaustive analysis.

A VSM model with tf-idf weighting for both candidate retrieval and exhaustive analysis is reported by Zechner et al. [26]. Here cosine similarity is used for document comparisons. Sanchez-Perez, Sidorov and Gelbukh [10] presented a tf-isf weighting scheme for the exhaustive analysis stage with cosine and dice similarity metrics. Vani and Gupta [38] presented an approach that uses tf-isf weights and POS tagging to retrieve the plagiarized fragments at sentence level. Authors also discuss the influence of various similarity metrics in deciding the detection efficiency. Kong et al. [39] used a tf-idf method for the candidate retrieval task and then scoring methods were used for ranking. Suchomel et al. [40] proposed a query formulation method for plagiarized source retrieval using tf-idf scheme. The top five keywords were used for formulation of initial query sets. The ranking was done based on tf-idf value of each word in the suspicious document. Kong et al. [41] presented a method that combined tf-idf, PatTree and weighted tf-idf to extract the keywords of suspicious documents as queries to retrieve the plagiarized source document. VSM approaches are also limited to detection of copy-paste and plagiarism by rogeting.

3.3. Syntax and Semantic based detection technique

In syntax based techniques, the document units at syntax level are extracted which can be sentences, phrases/chunks or it can be based on part of speech tagging (POS). Chunking and POS tagging provides the syntactic information within a document and facilitates in finding deeper manipulations. In chunking, parse trees of document are constructed and relevant phrases are extracted. In POS tagging, each token is labelled with their word classes which facilitates in more meaningful comparisons. In semantic based techniques the meaning representation of a document is focused and is found to be efficient for paraphrased detections. Semantic role labelling (SRL), machine learning techniques, soft computing techniques etc. fall into this category.

A PDS with tf-isf weighting and POS tagging is proposed by Vani and Gupta (2015). It was found that the PDS with POS tagging outperformed the one without mainly in terms of precision. This is because the system compared only the words with same tag and hence utilizing the syntax information to prune out false detections. A fuzzy based similarity approach was used in exhaustive analysis stage by Alzahrani and Salim [24] where fuzzy based semantic similarity metric computations are employed. Alzahrani, Salim and Palade [42] extended this similarity metric by

incorporating POS tag information and fuzzy-inference rules giving main focus on highly manipulated plagiarism cases. The statistical analysis using paired t-tests shows that this approach is statistically significant in comparison with the baselines and it also exhibits the potency of semantic-based models to detect plagiarism cases beyond the literal plagiarism. Gupta, Vani and Singh [43] used an improved fuzzy-semantic similarity metric using POS tagging. Semantic based detection systems and systems using semantic similarities mainly used WordNet (http://wordnet.princeton.edu/) thesaurus, semantic webs and other ontology's [44, 45, 46]. SRL based method is proposed by Osman, Salim and Binwahlanc [47] which did deep semantic analysis of document using role labelling. Kalleberg and Rune Borge[48] used ML classifiers with various similarity scores as the features to find plagiarized fragments. k-Nearest Neighbour Algorithm (k-NN) is used for detecting text plagiarism by clustering strings and detecting matches with neighbouring words by Sahu [49]. A detection system utilizing singular value decomposition (SVD) was presented by Ceska [50].

Even though these syntactic and semantic techniques are computationally expensive, it provides good improvement in detection efficiency mainly with respect to complex obfuscations.

3.4. Structural based detection technique

In this technique, tree structures and graphs are used to extract document structure information. Osman et al. [51] represented the text document as a graph which captures the semantic relationships. Each sentence is represented as a node and the sentence relationships with edges. Graph structures provide more detailed representations of document and facilitate in-depth analysis. The method has high potential when compared to other flat document representations. But usually a combined approach with text and structural information has to be used which is found to be effective. Methods that used structural information of documents based on generic classes and logical structure extraction (LSE) is used for detecting plagiarism in scientific publications [52, 53]. The exploration of structural information and techniques and tools that incorporate these techniques for detection are found less in literature. But when it comes to the detection of scientific publications and other scholarly articles the incorporation of this information can help to improve detection efficiency considerably.

3.5. Citation based detection technique

This technique is gaining popularity with its in depth analysis of document based on the citations used. The technique is mainly meant for scientific publications, where citations are used. Here the citation patterns are analyzed to identify plagiarism and are considered as an extension of text plagiarism or it is incorporated along with text based detections. This includes approaches that analyze citation using citation order analysis (COA), where order of citations in document with bibliographic coupling is exploited for plagiarism detection [54, 55]. A citation based PDS prototype called CitePlag that uses detection algorithms which analyze the citation sequences of academic documents for similar patterns that may indicate unethical text reuse is proposed by Meuschke, Gipp and Breitinger [56]. Alzahrani et al., [53] utilized citation evidences along with structural detection for detecting plagiarism cases. Four types of plagiarism, viz., self-reuse, self-plagiarism, reuse and plagiarism is detected using text based detection with citation analysis to detect copy-paste plagiarism in scientific articles of NLP domain by Mariani et al.[57]. They used papers from different websites such as ACL Anthology, ISCA archive and IEEE in NLP and speech processing.

Incorporation of citation and structural analysis has high scope to be explored as most of the unethical acts of plagiarism are found in educational domains. It is very important that the research work submitted by different individuals must be unique and original. Further most of the available plagiarism tools do not consider the references and citations which is an important part of any research publication. Plagiarism arises when the author copies the work without giving proper citation or acknowledgment to the original work. Thus presence and absence of citations plays an important role in plagiarism decision making.

Since most of the existing PDS is evaluated using the standard plagiarism corpus provided by the PAN competition and system performance is evaluated using the PAN standard measures, the paper briefly describes the PAN tasks and data set used in these tasks.

4. Pan Task- an Overview

As discussed, most of the available works is evaluated on PAN data sets and efficiency is measured using PAN measures. PAN is an international competition held yearly since 2009 in plagiarism detection domain. It evaluates the plagiarism detection systems submitted and ranks them based on defined measures. The plagiarism detection task is categorized under two subtasks, viz., text alignment and source retrieval. The systems submitted under these tasks are evaluated separately and ranked. Basically the text alignment focuses on the exhaustive comparison stage while the source retrieval task focuses on the candidate retrieval stage with online resources.

In the text alignment task, the extrinsic plagiarism detection is carried out as an offline process. The suspicious and source document corpus is provided as downloadable databases and it aims at finding the exact plagiarized suspicious passages and their corresponding counterparts in the source document. The plagiarized data available here is categorized based on the level of their complexity as: a) No obfuscation b) Random obfuscation c) Translation obfuscation and d) Summary obfuscation [9]. No obfuscation refers to simple copy-paste which is a literal/ verbatim plagiarism type. Most of the detection systems can find out this with simple algorithms. In random obfuscation, the text is manipulated using synonym replacements, word shuffling, active to passive transformations etc. while in translation obfuscation the source text is passed through some translators and then back translated to the original language. Further some manual modifications may be also done. Summary obfuscation is a complex case where the source idea is adopted and summarized. With the availability of online translators and automatic summarizers these tasks have become much easier for the plagiarist. As the plagiarism complexity increases, it is obvious that the detection becomes more challenging. Thus this PAN task is mainly aligned with exploring the detection techniques for exhaustive comparisons and to deal with manipulations of high complexity. The PAN measures used for this task are recall, precision, granularity and plagdet_score [9].

Source retrieval is an online task that aims to retrieve the plagiarized source with respect to a suspicious document query. It refers to the candidate retrieval task using online

resources as reference corpus. Here exhaustive comparison is not the focus, whereas query formulation is the main procedure. Efficient queries facilitate in accelerating the retrieval process while maintaining the system accuracy. This is closely related to the general information retrieval (IR) process used in search engines. PAN provides its own API and search engine for this task. The search engines, viz., Indri and ChatNoir [58] were built upon ClueWeb corpus 2009(ClueWeb09) (http://lemurproject.org/clueweb09) for this evaluation. Evaluation is done based on recall, precision, F-measure which is in turn based on the number of downloaded sources, the total workload based on number of downloads and number of queries formulated for the search [59, 60].

Both these tasks together contribute to the development of effective plagiarism detection software. The source retrieval task facilitates in retrieving the sources with respect to a plagiarized document query which constitutes the candidate stage for any online extrinsic PDS or plagiarism checkers. The actual segments of plagiarism within a source and suspicious document are identified in text alignment task which corresponds to the exhaustive comparison stage. Thus both tasks facilitate in availing a PDS giving attention to the major building blocks of a PDS, viz., candidate retrieval and exhaustive comparison stages.

From the discussion of different plagiarism detection techniques it can be noted that mostly N-gram models and VSM is employed in detection process. In some systems, more semantic based and linguistic approaches are utilized while many others focuses on potential of utilizing different NLP techniques. In the next section, an overview of some of the plagiarism software's is given and further in-depth analysis of some of these tools are done to identify the existing limitations and emphasizing the need of intelligent techniques.

5. Software Tools

Many plagiarism detection tools are available for text plagiarism detection which are either online or offline and commercial or plagiarism checking services. Studies report that the most of these available detection tools could not detect plagiarism imposed by structural variations and paraphrasing [61, 62, 63, 64].

Tab.1 shows some of the available plagiarism software for text plagiarism detection and their relevant features. The features and specifications of these tools found as a part of the study from their respective websites are reported here. Plagiarism checkers, viz., Small Seo, PlagScan, and Plagiarisma are freely available services but impose some text limits. Others such as Turnitin, iThenticate, Copycatch, EVE2 and CheckForPlagiarism are commercial.

Table 1. Plagiarism Tools and their Features

Tools	Features
Small Seo http://Small Seotools.com/plagiarism-checker/	• Freely available online plagiarism checker • Text limit of 1000-1500 words • Outputs the text as Existing/ Good or Plagiarized/ Unique • Supported documents- Only TXT
Plagiarisma http://plagiarisma.net/	• Free online checker • Uses simple string matching algorithms • Supported documents - TXT, HTML, RTF, DOC, DOCX, PDF, ODT. • Outputs the text as Unique if not plagiarized
Plagscan http://www.plagscan.com/	• Only about 2000 words can be checked as a part of free trial • Supported documents - MS Word, PDF and many more
Copycatch http://www.cflsoftware.com/GoldFull.html	• Mainly focus on student based plagiarism detection, viz, essays, projects etc. • Do not compare with web, only with other students work. • Different levels of similarity are represented by colors. Red is used for the sentences from the most matched statement. Blue for the next best match and pink for the third best match. Brown for any other matches if there are at least three sentences.
Turnitin http://turnitin.com/	• Used for document analysis • Document is compared against different sources from web and its own data base and with different algorithms plagiarism is checked. • The final report underlines or colors the similar sentences and with links to the suspected sources.
EVE2- Essay Verification Engine http://www.canexus.com/	• Compares the submitted text with internet sources and underlines the suspected sentences. • Supported documents-TXT and DOC
CheckForPlagiarism http://www.checkforplagiarism.net/	• Uses sentence structure assessment and synonym identifications • Database of books, articles, magazines and live internet sources • Supports multiple languages and document formats
iThenticate http://www.ithenticate.com/	• A paid plagiarism checker • 35+ millions documents checked • Used by most of the publishers like Elsevier, Springer, Wiley, IEEE etc.

Further the performance of three of these tools is checked using a small text fragment extracted from the abstract section of *Alzahrani & Salim, 2010* [24]. The text is then modified based on four main degrees of obfuscations as

defined in PAN text alignment task. The input used is given in Fig. 3.The text manipulated with different obfuscations is given in Fig. 4(a), (b), (c) and (d).

Fig. 4(a) and (b) shows the plagiarism by mere copy-paste or verbatim plagiarism and those by random obfuscations respectively. Fig. 4(c) shows the passage obfuscated using translation plagiarism and 4(d) represents summary obfuscations. For translation obfuscation, as given in given in Fig.5, the input English passage is initially translated to Hindi and then this is back translated to English using Google translate (https://translate.google.co.in/). Doing this back translation, it was found that the complete word order changed and many meaningless sentences were produced as seen in second text of Fig. 5. Considering the real plagiarism cases, the plagiarist may do some manual reordering to make the sentences meaningful. With this view, after the translation and back translation, the obtained text is slightly modified manually to make them meaningful and used for current experiment (Fig. 4 (c)). In Fig. 4 (d), the summary obfuscated passage is shown which is obtained by summarizing the input content manually. Automatic summarizers can be also used for this. But when checked with some of the online summarizers, it was found that the summary obtained for this input was not conveying the complete idea. Instead it was just a group of some of the randomly selected sentences from the actual input. This may be because the input size is too small. The details are not analyzed as our research focus is not on these summarization tools.

Abstract. This report explains our plagiarism detection method using fuzzy semantic-based string similarity approach. The algorithm was developed through four main stages. First is pre-processing which includes tokenisation, stemming and stop words removing. Second is retrieving a list of candidate documents for each suspicious document using shingling and Jaccard coefficient. Suspicious documents are then compared sentence-wise with the associated candidate documents. This stage entails the computation of fuzzy degree of similarity that ranges between two edges: 0 for completely different sentences and 1 for exactly identical sentences. Two sentences are marked as similar (i.e. plagiarised) if they gain a fuzzy similarity score above a certain threshold. The last step is post-processing whereby consecutive sentences are joined to form single paragraphs/sections.

Fig.3. Actual Source Fragment [Taken from Abstract Section of Alzahrani & Salim, 2010]

Table 2 shows the approximate statistics of number of single word and n consecutive word matching (n > 3) between the texts in Fig. 4 (b), 4(c) and 4(d) and the original input in Fig.3. It compares the number of matching words based on the bag of words (BOW) concept. Fig. 4 (a), i.e., no obfuscation text is not considered as it is an exact copy-paste of original text.

The texts with different manipulations are fed to the two online free text plagiarism tools, viz., Small Seo and Plagiarisma and a paid commercial tool, viz., Turnitin. The output results obtained with each complexity levels are analyzed, studied and compared. The results obtained by Small Seo, Plagiarisma and Turnitin are shown in Fig. 6, 7 and 8 respectively. The output of each of these tools is analyzed and discussed in subsequent sections.

5.1. Small Seo Output Analysis

Initially the results of Small Seo plagiarism tool with each obfuscated text are verified. As observed from Fig.6, the technique used for comparison is not given and not so clear from the output, i.e., whether it is sentence based/ N-gram based or other methods. In the tool, the suspicious text has to be pasted in the GUI provided for comparison. The tool shows whether the compared fragment is Plagiarized/ Unique. In the earlier version it was shown as Existing /Good. The plagiarized fragments are marked in dark red colour and unique fragments in green colour. From the plagiarized one's, the links to suspected sources can also be accessed.

4(a) No obfuscation (Copy-Paste): Our plagiarism detection method using fuzzy semantic-based string similarity approach. The algorithm was developed through four main stages. First is pre-processing which includes tokenisation, stemming and stop words removing. Second is retrieving a list of candidate documents for each suspicious document using shingling and Jaccard coefficient. Suspicious documents are then compared sentence-wise with the associated candidate documents. This stage entails the computation of fuzzy degree of similarity that ranges between two edges: 0 for completely different sentences and 1 for exactly identical sentences. Two sentences are marked as similar (i.e. plagiarised) if they gain a fuzzy similarity score above a certain threshold. The last step is post-processing whereby consecutive sentences are joined to form single paragraphs/Sections

4(b) Random obfuscation: A fuzzy semantic-based string similarity based method is used here. The algorithm constitutes four steps. Initially, pre-processing is done with tokenisation, stemming and stop words removal. Then candidate documents for each suspicious document using shingling and Jaccard coefficient is computed. Next, sentence based comparison of each suspected document is done. Here fuzzy similarity is computed that ranges between 0 for different sentences and 1 for exactly same sentences. Two sentences are considered as plagiarised if they have a fuzzy similarity score above a certain threshold. The final step is post-processing in which consecutive sentences are combined to form single paragraphs

4(c) Translation obfuscation: A meaning-based approach using fuzzy string similarity is used for our plagiarism detection method. The algorithm was developed through four main steps. First stemming, stop word removal and tokenisation which is the pre-processing stage. The second is retrieving candidate list of documents for each suspect document using shingling and using Jaccard coefficient. Suspicious documents are compared with candidate documents associated in terms of the sentences. Next stage entails calculation of the degree of fuzzy similarity ranging between sides those absolutely completely different sentences 0 and for the same phrase 1. If two sentences achieved a fuzzy similarity score above a certain threshold then are marked similar (i.e., plagiarized). The final step included the subsequent processing of consecutive sentences as single paragraph / Section.

4(d) Summary obfuscation: A fuzzy semantic-based string similarity based plagiarism detection method with four stages is used here. Pre-processing done with tokenisation, stemming and stop words removal, which is followed by candidate document retrieval with shingling and jaccard coefficient. Then sentence based similarity computation is done to find the plagiarized sentences and finally consecutive sentences are merged

Fig. 4. (a). No Obfuscation Text. (b) Random Obfuscation Text. (c). Translation Obfuscation text (d). Summary Obfuscation Text.

In this tool, with exact copy-paste, i.e., no obfuscation case it is observed that an accurate detection is done. The tool shows 0% uniqueness or 100% detection in this case, which means the submitted text is a completely plagiarized version, and all the fragments compared are shown as

'Plagiarized' as observed from Fig.6. Hence the detection is effective when the text is a simple copy-paste or literal plagiarism. With random obfuscation, the detection efficiency is found to drop. It shows 91% unique content, while the entire text is actually randomly plagiarized as given in Fig.4 (b).

This means only 9% similarity detection was possible when the complexity of obfuscation increased or the text got slightly manipulated. In this case, most of the fragments are marked as '*Unique'*, which is not the actual case. As the plagiarism complexity further increases, viz., summarized texts where the idea is presented by summarizing the actual content, only 33% detection (67% unique) is presented. But the link to the portion shown as '*Plagiarized*' was showing as *'did not match any documents'*. This means the matching is still inappropriate. With translated texts, the tool shows a 100% uniqueness marking all fragments as '*Unique*'. This means with intelligently manipulated texts the detection efficiency is less or even nullified. This can be because the algorithm employed is inefficient for higher obfuscation detections.

Translation (English- Hindi): फजी अर्थ-आधारित स्ट्रिंग समानता दृष्टिकोण का उपयोग हमारे साहित्यिक चोरी का पता लगाने के विधि। एल्गोरिथ्म चार मुख्य चरण के माध्यम से विकसित किया गया था। सबसे पहले stemming और हटाने शब्द रोक, tokenisation भी शामिल है जो पूर्व प्रसंस्करण है। दूसरा shingling और Jaccard गुणांक का उपयोग करते हुए प्रत्येक संदिग्ध दस्तावेज के लिए उम्मीदवार दस्तावेजों की एक सूची को पुन: प्राप्त कर रहा है। संदेहास्पद दस्तावेजों तो वाक्य के लिहाज से जुड़े उम्मीदवार दस्तावेजों के साथ तुलना कर रहे हैं। बिल्कुल समान वाक्यांश के लिए पूरी तरह से अलग वाक्यों के लिए 0 और 1:इस चरण में दो किनारों के बीच पर्वतमाला कि समानता के फजी डिग्री की गणना पर जोर देता। वे एक निश्चित सीमा से ऊपर एक फजी समानता स्कोर हासिल अगर दो वाक्य भी इसी तरह के रूप में चिह्नित कर रहे हैं (यानी plagiarized)। अंतिम चरण के लिए लगातार वाक्य एकल पैराग्राफ / अनुभाग के रूप में शामिल कर रहे हैं, जिससे बाद के प्रसंस्करण है।

Back Translation (Hindi- English): Meaning-based approach using fuzzy string similarity of our plagiarism detection method. The algorithm was developed through four main steps. First stemming and stop word removal, tokenisation which includes pre-processing. The second shingling and using Jaccard coefficient candidate a list of documents for each suspect document retrieving is. Suspicious documents in terms of the sentence candidate associated with the documents are compared. Absolutely completely different sentences for the same phrase 0 and 1: In this stage ranges between sides that entails calculating the degree of similarity fuzzy. They have a fuzzy similarity score above a certain threshold achieved if two sentences are similar marked (ie plagiarized). The final step consecutive sentences single paragraph / Section are included, the subsequent processing.

Fig.5. Method used for Imposing Translation Obfuscation

5.2. Plagiarisma Output Analysis

The next tool surveyed is Plagiarisma, which again is freely available. Here basically a rough sentence-based approach is utilized as seen from the outputs in Fig. 7. Each sentence is given as a query and the results are retrieved if some similarity is detected. The number of results retrieved and the domain links are also given with the output. If the compared fragment is marked as non-plagiarized by tool, it shows it as Unique with highlighting. In addition, highlighting the matched fragments also shows the entire input text. In no obfuscation text, 9% uniqueness is shown. This means that even with mere copy-paste the similarity detection of the tool is not 100% accurate. With random obfuscation 78% uniqueness is detected which means only 22% detection efficiency is shown. As observed from the output demo, the detected text portions are almost the exact matching cases only. While with the other two complex manipulations, detection is 0%, presenting 100% uniqueness. It can be noted that with these online plagiarism checkers it is quite difficult to detect intelligent plagiarism cases. This is because most of these tools utilize string-matching algorithm for detections, which cannot capture structural and semantic concepts. As observed, it is somehow matching the longest phrase or some subsequence, which is exactly or almost similar to the input given. From Table 2, it is noted that based on single word matching, translated text is having about 71 similar words. But the number of contiguous matching is less for both translation and summary. Thus simple plagiarism cases are detected but with even small manipulations the detection efficiency decreases considerably.

5.3. Turnitin Output Analysis

Next we verified the output of a paid plagiarism checker widely used in various popular journals, conferences etc., Turnitin whose output demo is given in Fig. 8. The output is obtained as an entire submitted text with plagiarized segments highlighted. The similarity index with the links to the detected sources of plagiarism is also given. Here, in the literal plagiarism case, 100% similarity index is reported and the right source of plagiarism is retrieved. Hence complete detection is done in case of copy-paste plagiarism or a no obfuscation case. With random obfuscations, 46% detection is shown surpassing the other two tools. But still, the efficiency of detection dropped with intelligent manipulations. It is also found that the detected source is not the same as that of simple copy-paste. With increased complexity level of plagiarized data, i.e., with summary obfuscation, 34% similarity is presented. Even though the Small Seo tool gave 33% detection, in this case, no source was found to be retrieved with respect to the plagiarized segment reported, which is again questionable. With translated text, the detection was not possible here also and the tool reported a 0% similarity as it cannot identify any similar sources corresponding to this text

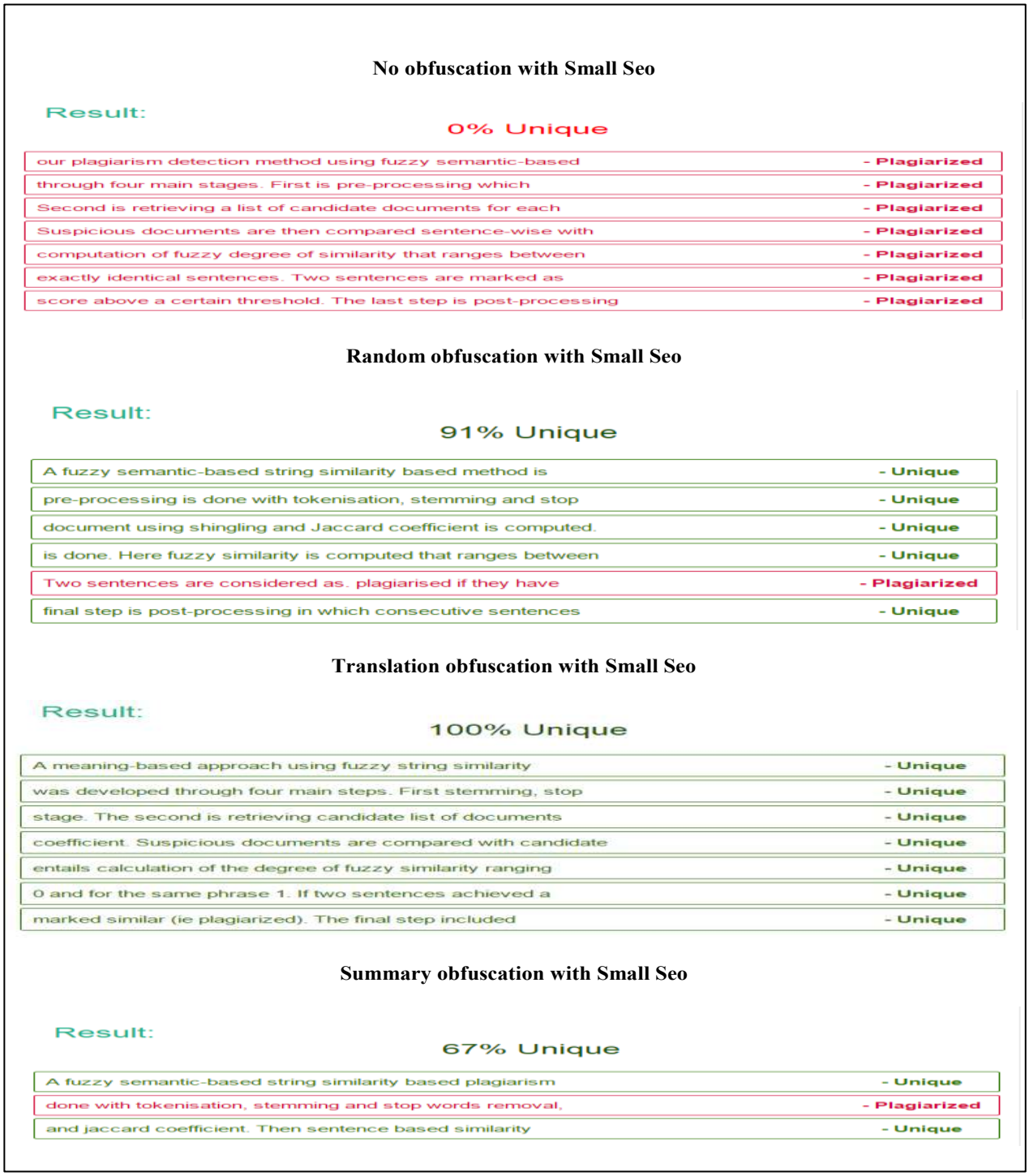

Fig.6. Outputs of Small Seo Tool Using Text with Different Obfuscations

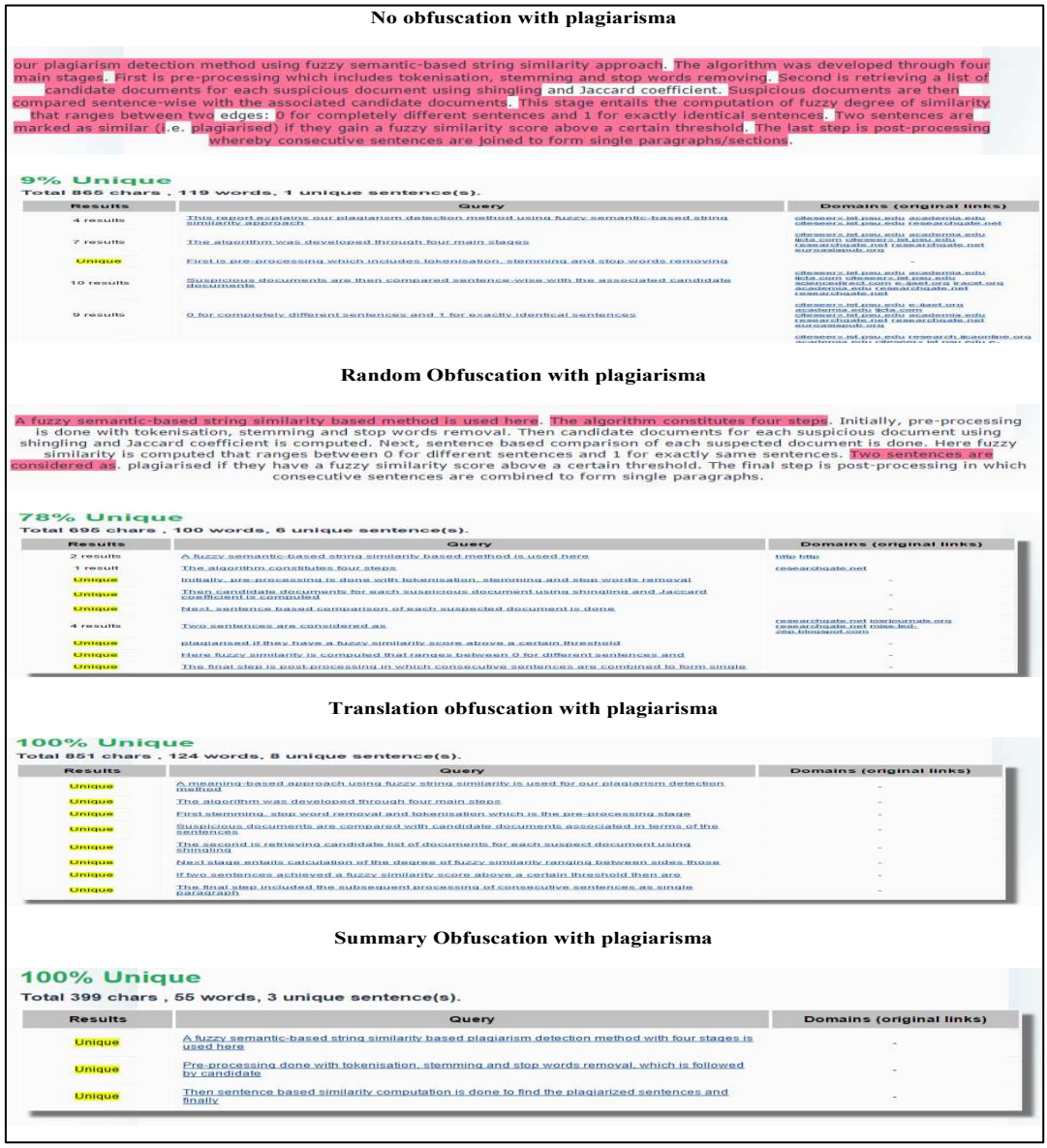

Fig.7. Outputs of Plagiarisma Tool Using Text with Different Obfuscations

Table 2. Statistics of # of Matched Fragments between Actual & Intelligently Manipulated Texts

Intelligent Manipulations	# of words in BOW	Approx. # of single words matches	Approx. # of n consecutive word matches (n >3)
Random	100	60	9
Translation	123	71	6
Summary	54	26	4

5.4. Analysis & Discussion

The discussion is based on the analysis of output demos of Turnitin tool. It is found that in copy-paste plagiarism, i.e., no obfuscation case, the detection is accurate and this is almost obvious as the text is exactly similar to input and any simple string matching algorithm should detect it. Coming to random obfuscations, it is found from Table 2, that about 60 single word matches and 9 contiguous matches are present compared to the original input. Even some of these contiguous fragments are not identified as plagiarized or similar by the detection software's.

For instance consider the third sentence in Fig. 4 (b). Here the fragment "tokenisation, stemming and stop words" is similar with the input text but match is not detected in random obfuscation case of turnitin. Now consider the second last sentence from the original and random texts, viz., *"Two sentences are marked as similar"* and *"Two sentences are considered as plagiarised"* respectively. The sentences are semantically the same but the detection is shown only in the exact matching part *"Two sentences are"* as noted from Fig.8. Thus it is obvious that only some of the phrases are identified which forms the exact match with input words while semantic concepts are not captured. Again consider another example, input text sentence *"The algorithm was developed through four main stages"* which is modified in random obfuscation as *"The algorithm constitutes four steps"*. Here the sentences are modified by replacement with synonyms but this paraphrasing could not be identified by the tool. Basically exact match identification is done and some heuristics must have been applied to match

a fragment based on the number of consecutively matching words or some notion based on length of matched items or surrounding words.

In summary obfuscation, as observed from Tab. 2, only 26 single word matches and 4 contiguous fragments are analyzed. This is mainly because it is a summarized version and the text content is small, as seen, only 54 BOW is present. Even though it is an idea plagiarism, 34% detection is presented by the Turnitin. But analyzing the detected fragments from Fig. 8, again it is found that only exact phrases are matched. As the sentence restructuring included some of the phrases and words of original input (Fig. 4(d)), the tool figured out it and highlighted it as plagiarized, viz., *"fuzzy semantic-based string similarity", "shingling and jaccard coefficient"* etc. The tool failed to detect other fragments which actually convey the same idea as the input. It is also observed that even the stop word *'is'* is highlighted as similar or duplicate which is not correct. Thus the manipulations created by restructuring and merging of sentences are skipped by the tool.

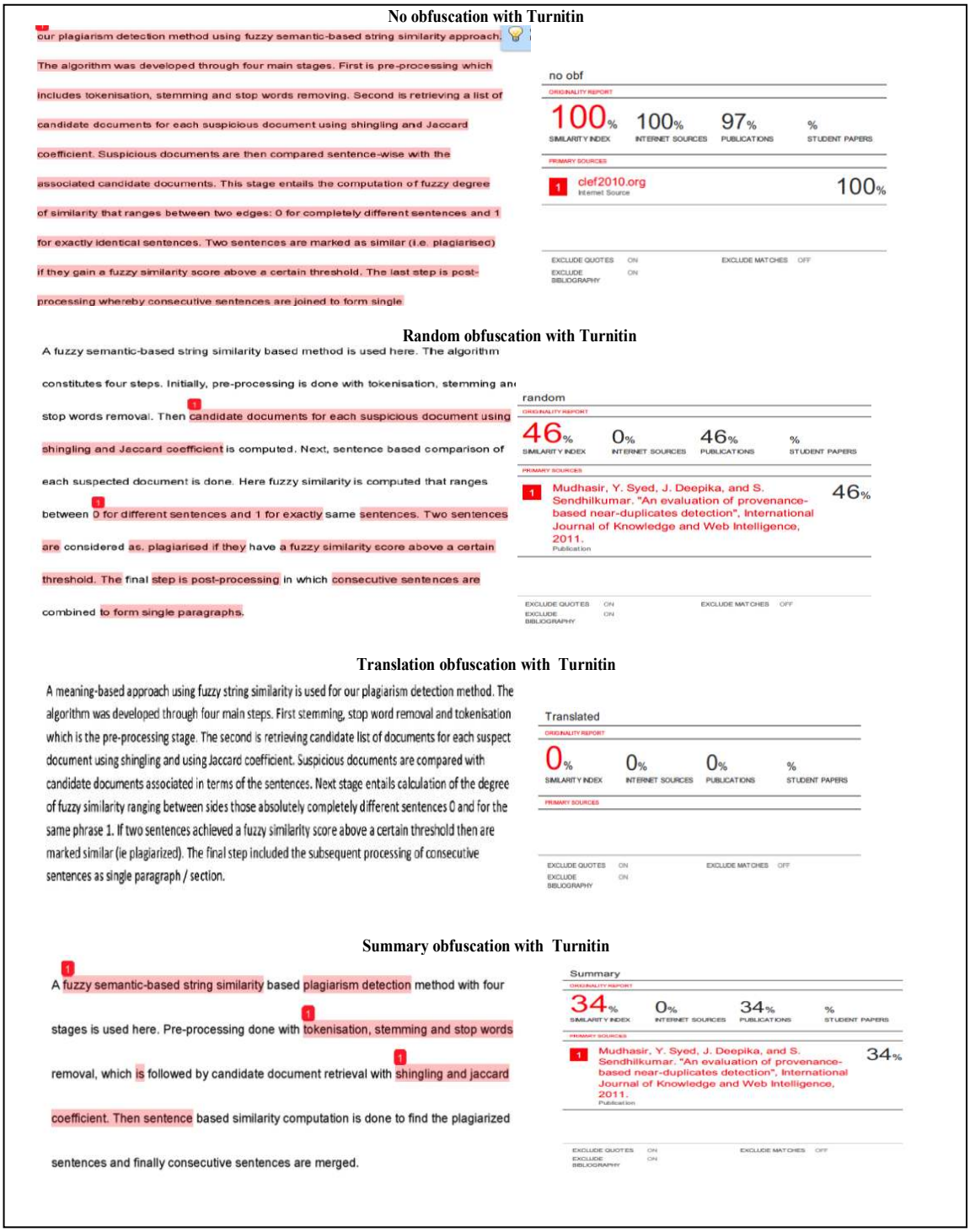

Fig.8. Outputs of Turnitin Tool Using Text with Different Obfuscations

With translation obfuscation, the detection dropped completely. In translated text, as observed from Fig. 4 (c), the structuring of sentences is changed and so many shuffling in word positions can be seen. Further variations of word phrases are also visible compared to the original input text. But it is noted that the idea conveyed is still semantically similar. Further from the statistics in Tab.2, it is found that very high single word matching is available in this translated text and 6 contiguous fragments are also noted. For example, the second sentence in Fig. 4 (c) *"The algorithm was developed through four main steps"* which is similar to that of actual text *"The algorithm was developed through four main stages"* except that the last word is changed from *"stages"* to *"steps"*. But even this fragment is not identified by the analyzed tools. Further the word sequence in input text *"tokenisation, stemming and stop words removing"* is modified as *"stemming, stop word removal and tokenisation"* which is basically a reordering of words. The reasons of these detection failures may be many. It can be because the algorithm implemented by the tool is not able to detect plagiarized fragments with high restructuring and paraphrasing. It is not able to capture the semantic and linguistic variations and thus detection efficiency drops.

Thus comparing the detections done by the tool in each obfuscation type, it is found that the detection of exact phrases or almost similar ones are done while the semantic and structural variations are not captured by them. Even a simple reordering or shuffling is not identified in many cases. Only contiguous fragment matches are identified. These detection failures point out the limitations of these tools which can be easily surpassed by plagiarists. Even with the paid tool turnitin used by many academic institutions for student plagiarism checking, limitations are figured out mainly when it comes to complex manipulations. Further with complex methods of rogeting, plagiarist substitutes words and modify them with synonyms even in the internal binary codes of saved electronic files which have aggravated the issue. These manipulations are claim to cheat even detection systems such as turnitin. Cheatturnitin (http://cheatturnitin.blogspot.in/) describes how turnitin can be cheated using its limitations. The limitations pointed out include:

- Inability to detect intelligent parahrasing & rogeting
- Cannot trace out and analyze citations and quotes, hence giving false detections
- It detects headers, footers,references, acknowledgements etc. as plagiarized, as it doesn't consider structural information

Other problems include cheating turnitin using word functionalities such as macro-enabling and disabling, getting papers from cheat sites or essay mills which can prevent turnitins crawling etc. Many of these serious limitations can be countered using effective AI techniques such as NLP, ML, Soft Computing and other intelligent techniques for plagiarism detection. NLP and ML techniques as the future of plagiarism detection [61,15]. Further along with text based detections structural and citation analysis has to be incorporated which is important in scholarly articles mainly to avoid false detections.

6. Common Problems & Research Gap

The common problem noted with most of these tools is their lack of ability to detect intelligent manipulations, even though they claim to be. Most of the tools, even paid, fail when it comes to translation and summary obfuscations. In today's world, with the ease of access to online translation and summarization tools a plagiarist can easily perform intelligent and complex manipulations in source text which can surpass the detection capacity of these tools. Further the condition can be still complex when thses obfuscations are manully combined. Patents with efficient text plagiarism detection tools are not found while some for source code plagiarism were there. During the survey it was surprising to see that some accepted publications found in web were exact copies of original piece of work and even the citations to those works were not given in them. These scenarios cannot be treated as unintentional because some of the basic ethics of writings must be followed at least when publishing papers. One reason for the growth of this kind of work may be that either some journals or conferences do not employ any sort of plagiarism checking or the tools used are inefficient. Thus there is still a lot to explore and improve in this domain to improve the efficiency of detection tools. From analysis done with some of the available tools, it is clear that a lot has to be improved to tackle high obfuscation plagiarism cases. A lot of research gaps can be analyzed, mainly in:

- Improving detection techniques mainly focusing on paraphrase and intelligent manipulation detection.
- Structural and semantic variations or manipulations are least captured by the available tools. Thus algorithm efficiency should be improved in these terms.
- Focusing on plagiarism using idea adoptions, viz, summary obfuscations which are hard to tackle. In these aspects computational intelligence, soft computing and advanced NLP techniques can be explored. From the literature, it is found most of the works done are with N-gram models, VSM etc. Only very few works with semantic and intelligent implementation were found.
- Citation based techniques are very less explored and has good scope in facilitating the improvement of detection efficiency, when coupled properly with text based techniques.
- Focus on candidate retrieval stage techniques, specifically when dealing with online resources. Techniques for query formulation and proper key phrase extractions have to be explored for regulating and improving the performance efficiency of a PDS.

These are some of the few research potentials that we came across during the studies and analysis. The main problem with intensive intelligent technique usage is the computational expensiveness. But with different parallelization technologies, cloud computing, big data analytics etc, this problem is solved and can be easily implemented in research labs. Plagiarism reduces the amount of effective original pieces of work. The tendency of people to copy things increases, if it is not detected properly and punished. Thus to ensure the protection of the original work of ethical researchers, a detection system with intelligent algorithm application is highly needed.

7. Conclusion

This paper presents a brief review about the tools and techniques in extrinsic text plagiarism. It attempts to provide some insight to the current state of art in this domain, the techniques used, the tools etc. Study and analysis of some of the tools are done, further pointing the main problems with these tools and the research gaps Intelligent techniques for detection of high obfuscations are still in its infancy and most of the available online, stand alone and web based tools fail to detect complex manipulations. The paper thus throws light on the immense research potential in this field for developing efficient intelligent detection systems so as to curb this unethical act.

Acknowledgements

This work was supported by Department of Science and Technology, Govt. of India (www.dst.gov.in), under Grant (number SERB/F/1511/2014-2016).

References

1. Random House Compact Unabridged Dictionary (1995): qtd. in Stepchyshyn, Vera; Nelson, Robert S.. Library plagiarism policies. Assoc. of College & Research Libraries, 65, ISBN 0838984169(2007).
2. Oxford English Dictionary :qtd. in Lands(1999).
3. Guo, X. Understanding Student Plagiarism: An Empirical Study in Accounting Education. Journal of Accounting Education, Taylor and Francis, 20(1), 17-37. 10.1080/09639284.2010.534577 (2011).
4. Newton, P. Academic integrity: a quantitative study of confidence and understanding in students at the start of their higher education. Journal of Assessment and Evaluation in Higher Education, Taylor and Francis, 1-16. doi:10.1080/02602938.2015.1024199 (2015)..
5. Kauffman, Y., and Young, M.F. Digital Plagiarism: An Experimental Study of the Effect of Instructional Goals and Copy-and-Paste Affordance. Journal of Computers and Education, 83, 44-56(2015).
6. Bin-Habtoor, A.S., and Zaher, M.A. A survey on Text Plagiarism Detection Systems. International Journal of Computer Theory and Engineering, 4 (2) (2012).
7. Alzahrani, S. M., Salim,N., and Abraham, A. Understanding plagiarism linguistic patterns, textual features, and detection methods. IEEE transactions on systems, man, and cybernetics part c: application and reviews, 42 (2) (2012).
8. Maurer, H., Kappe, F., and Zaka, B. Plagiarism - A Survey. Journal of Universal Computer Science, 12(8), 1050-1084(2006).
9. Potthast,M., Stein, B.,. Cedeno A.B, and Rosso, P. An evaluation framework for plagiarism detection. Proc. of 23rd Int. Conf. on Computational Linguistics, COLING , Beijing, China (2010).
10. Sanchez-Perez, Sidorov,G., and Gelbukh, A. A Winning Approach to Text Alignment for Text Reuse Detection - lab report for PAN at CLEF 2014. Proc. of 6th Int. Workshop PAN-14, Sheffield, UK (2014)..
11. Pera,M.S. and Ng,Y.K. Sentence-Based Plagiarism Detection Tool on Web Documents. Web Intelligence and Agent Systems: An International Journal, IOS Press ,1(1) (2009)..
12. White, D. R. and Joy, M.S. Sentence-based natural language plagiarism detection. ACM Journal on Educational Resources in Computing, 4(4), 1-20 (2004). .
13. Yokoi,T., Oikawa, G., Iwata, M., Sato, T. and Kobayakawa, M. Sentence based Plagiarism Detection focusing on Nouns and Part of Speech Structures. New Trends in Software Methodologies, Tools and Techniques, IOS Press (2014)..
14. Stamatatos, E. Plagiarism Detection Using Stopword N-grams. Journal of the American Society for Information Science and Technology, Wiley, 62(12), 2512-2527(2011).
15. Chong, M., Specia, L. and Mitkov, R. Using Natural Language Processing for Automatic Plagiarism Detection, Proc. of 4th International Plagiarism Conference, Northrumbia University Newcastle-upon-Tyne, UK (2010).
16. Mozgovoy , M., Kakkonen, T., and Sutinen, E. Using Natural Language Parsers in Plagiarism Detection. Proc. of SLaTE'07 Workshop, USA(2007).
17. Adam, AR. and Suharjito,M. Plagiarism Detection Algorithm Using Natural Language Processing Based on Grammar Analyzing. Journal of Theoretical and Applied Information Technology, 63 (1), 168 – 180(2014).
18. Manning,C.D., Prabhakar,R., and Schiitze, H. Introduction to Information Retrieval. Cambridge University Press(2008)..
19. Mozgovoy , M., Kakkonen, T. and Cosma, G. Automatic student plagiarism detection: future perspectives, Journal of Educational Computing Research, 43 (4), 511-531(2010).
20. Potthast, M., Gollub, T., Hagen, M., Graßegger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P. and Stein, B. Overview of the 4th International Competition on Plagiarism Detection.Working Notes Papers of the CLEF 2012 Evaluation Labs (2012).
21. Hagen, M., Potthast,M., and Stein, B. Source Retrieval for Plagiarism Detection from Large Web Corpora: Recent Approaches. Proc. of CLEF 2015 Labs and Workshops, Notebook Papers, 8-11 September, Toulouse, France (2015)..
22. Barrón-Cedeño, A., Basile,C., Esposti, M.D., and Rosso,P. Word Length n-Grams for Text Re-use Detection. CICLing 2010, LNCS 6008, 687–699(2010).
23. Shrestha,P and Soloria, T. Using a variety of N-grams for the detection of different kinds of plagiarism -lab report for PAN at CLEF 2013. Proc. of 5th International Workshop PAN-13, Valencia, Spain. (2013).
24. Alzahrani,S.M., and N. Salim. Fuzzy semantic-based string similarity for extrinsic plagiarism detection- lab report for PAN at CLEF 2010. Proc. of 2nd Int. Workshop PAN-10, Padua, Italy(2010)..
25. Palkovskii,Y., and Belov,A. Developing High-Resolution Universal Multi- Type N-Gram Plagiarism Detector- lab report for PAN at CLEF 2014. Proc. of 6th Int.Workshop PAN-14, Sheffield, UK(2014)..
26. Zechner,M., Muhr,M., Kern,R., and Granitzer,M. External and intrinsic plagiarism detection using vector space models. Proc. of SEPLN, Spain, 47–55. (2009).
27. Ekbal, A., Saha, S., and Choudhary, G. Plagiarism detection in text using vector space model. Proc. of 12th Int. Conf. on Hybrid Intelligent Systems (HIS), 366-371(2012).
28. Vani,K., Gupta, D. Using K-means Cluster based Techniques in External Plagiarism Detection. Proc. of Int. Conf. on Contemporary Computing and Informatics (IC3I), 27-29(2014)..
29. Ravi,N.R., Vani,K., and Gupta,D. Exploration of Fuzzy C Means Clustering Algorithm in External Plagiarism Detection System. Int. Symposium on Intelligent Systems Technologies and Applications (ISTA-2015), 384,127-138 (2015).
30. Sánchez-Vega, F., Villatoro-Tello, E., Montes-y-Gómez, M., Pineda,L.V. and Rosso, P. Determining and characterizing the reused text for plagiarism detection. Expert Systems with Applications, 40(5), 1804-1813(2013)..
31. Suchomel, S., and Brandejs, M. Heterogeneous Queries for Synoptic and Phrasal Search- Notebook for PAN at CLEF 2014.Proc. of 6th International Workshop PAN-14, Sheffield, UK(2014)..
32. Elizalde,V. Using noun phrases and tf-idf for plagiarized document retrieval- Notebook for PAN at CLEF 2014. Proc. of 6th International Workshop PAN-14, Sheffield, UK(2014)..
33. Prakash, A., and Saha,S.K. Experiments on Document Chunking and Query Formation for Plagiarism Source Retrieval- lab report for PAN at CLEF 2014.Proc. of 6th International Workshop PAN-14, Sheffield, UK(2014)..
34. Ravi N, R., and Gupta, D. Efficient Paragraph based Chunking and Download Filtering for Plagiarism Source Retrieval -Notebook for PAN at CLEF 2015. Proc. of 7th International Workshop PAN-15, Toulouse, France(2015)..
35. Torrejon, R. D. A., and Ramos, M. J. M. Text Alignment Module in CoReMo 2.1 Plagiarism Detector- lab report for PAN at CLEF 2013. Proc. of 5th Int.Workshop PAN-13, Valencia, Spain (2013).
36. Kuppers, R., and Conrad, S. A Set-Based Approach to Plagiarism Detection- lab report for PAN at CLEF 2012. Proc. of 3rd Int.Workshop PAN-12, Rome, Italy(2012)..

37. Alvi, F., Stevenson, M., and Clough, P. Hashing and Merging Heuristics for Text Reuse Detection -Notebook for PAN at CLEF-2014. Proc. of 6th Int. Workshop PAN-14, Sheffield, UK(2014)..
38. Vani,K., and Gupta,D. Investigating the Impact of Combined Similarity Metrics and POS tagging in Extrinsic Text Plagiarism Detection System. Proc. of Int. Conf. on Advances in Computing, Communication and Informatics, Kochi, India, 1578-1584(2015)..
39. Kong, L., Haoliang,Q., Shuai, W., Cuixia,D., Suhong, W., and Yong,H. Approaches for Source Retrieval and Text Alignment of Plagiarism Detection-lab report for PAN at CLEF 2013. Proc. of 5th Int. Workshop PAN-13, Valencia, Spain (2013)..
40. Suchomel, S., Kasprzak, J., and Brandejs, M. Diverse Queries and Feature Type Selection for Plagiarism Discovery- lab report for PAN at CLEF 2013. Proc. of 5th Int. Workshop PAN-13, Valencia, Spain.(2013).
41. Kong, L., Haoliang,Q., Shuai, W., Cuixia,D., Suhong, W., and Yong,H .Source Retrieval Based on Learning to Rank and Text Alignment Based on Plagiarism Type Recognition for Plagiarism Detection- lab report for PAN at CLEF 2014.Proc. of 6th Int. Workshop PAN-14, Sheffield, UK(2014).
42. Alzahrani, S.M., Salim, N., and Palade, V. Uncovering highly obfuscated plagiarism cases using fuzzy semantic-based similarity model. Journal of King Saud University – Computer and Information Sciences, 27(3), 248-268. doi: 10.1016/j.jksuci.2014.12.001.(2015).
43. Gupta, D., Vani, K., and Singh, C.K. Using Natural Language Processing techniques and fuzzy-semantic similarity for automatic external plagiarism detection. Proc. of Int. Conf. on Advances in Computing, Communication and Informatics, Noida, 2694-269(2014)..
44. Agarwal ,J., Goudar, R. H., Kumar,P., Sharma,N., Parshav, V., Sharma, R., Srivastava, A. and Rao, S. Intelligent plagiarism detection mechanism using semantic technology: A different approach. Proc. of Advances in Computing, Communications and Informatics (ICACCI), Mysore, 2013, 779-783(2013)..
45. Al-Shamery, E.S. and Gheni,H.Q. Plagiarism Detection using Semantic Analysis. Indian Journal of Science and Technology, 9(1) (2016).
46. Shenoy K. M. and Shet, K.C., Acharya, U.D. Semantic Plagiarism Detection System Using Ontology Mapping. Advanced Computing: An International Journal (ACIJ), 3(3) (2012).
47. Osman, A.H., Salim,N., Binwahlanc,M.S., Alteebd,R., and Abuobieda,A. An improved plagiarism detection scheme based on semantic role labelling. Journal of Applied Soft Computing, 12, 1493–1502(2012)..
48. Kalleberg and Rune Borge. Towards Detecting Textual Plagiarism Using Machine Learning Methods, Master thesis, University of Agder (2015).
49. Sahu, M. Plagiarism Detection Using Artificial Intelligence Technique In Multiple Files. International Journal 0f Scientific and Technology Research, 5(4) (2016).
50. Ceska, Z. Plagiarism detection based on singular value decomposition. Advances in Natural Language Processing, Springer Berlin Heidelberg Lecture Notes in Computer Science, 52(21),108–119(2008)..
51. Osman, A.H., Salim,N., and Binwahlanc,M.S. Plagiarism Detection Using Graph-Based Representation. Journal of Computing, 2(4) (2010)..
52. Alzahrani, S. M., Salim,N., Abraham, A. and Palade,V. iPlag: Intelligent Plagiarism Reasoner in Scientific Publications. Proc. of Information and Communication Technologies (WICT), 2011 World Congress, 1- 6(2011).
53. Alzahrani, S.M., Palade, V., Salim, N., and Abraham,A. Using structural Information and Citation Evidence to Detect Significant Plagiarism cases in Scientific Publications. Journal of the American Society for Information Science and Technology, 63(2), 217-430(2011).
54. Gipp, B. Citation-based Plagiarism Detection – Detecting Disguised and Cross-language Plagiarism using Citation Pattern Analysis. Springer Vieweg Research, 2014(2014)..
55. Gipp, B.,and Beel,J. Citation Based Plagiarism Detection – A New Approach to Identify Plagiarized Work Language Independently.Proc. of the 21st ACM Conference on Hypertext and Hypermedia (HT'10), New York, NY, USA(2010).
56. Meuschke, N., Gipp, B., and Breitinger, C. CitePlag: A citation-based plagiarism detection system prototype. Proc. of the 5th Int.Plagiarism Conf., Newcastle upon Tyne, UK2012.
57. Mariani,J., Francopoulo, G. and Paroubek,P. A Study of Reuse and Plagiarism in Speech and Natural Language Processing papers, BIRNDL 2016 Joint Workshop on Bibliometric-enhanced Information Retrieval and NLP for Digital Libraries(2016).
58. Potthast, M., Hagen, M., Stein, B., Grabegger, J., Michel, M., Tippmann, M., and Welsch, C. ChatNoir: A Search Engine for the ClueWeb09 Corpus. Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12), 1004(2012)..
59. Potthast,M., Hagen,M., Beyer,A., Buss,M., Tippmann, M., Rosso, P., and Stein, B. Overview of the 6th International Competition on Plagiarism Detection. CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers. 15-18 September, Sheffield, UK(2014).
60. Potthast,M., Hagen,M., Gollub,T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., and Stein, B. Overview of 5th International Competition on Plagiarism Detection. CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers. 23-26 September, Valencia, Spain(2013).
61. Clough, P. Plagiarism in natural and programming languages: An overview of current tools and technologies. (Research Memoranda: CS-00–05), Department of Computer Science, University of Sheffield,U.K(2000).
62. Marsh,B. Turnitin.com and the scriptural enterprise of plagiarism detection. Computers and Composition, 21(4), 427-438(2004)..
63. Weber-Wulff, D. On the utility of plagiarism detection software. Proc. of 3rd International Plagiarism Conference, Newcastle Tyne, 1-11(2008).
64. Williams,J. The plagiarism problem: are students entirely to blame? In Proc. of the 19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education (ASCILITE), 2, 721-730, Auckland, New Zealand(2002).