MS (CS) THESIS

Use of LDA and POS Tags for Efficient Searching of Plagiarized Passages

By:
Jamal Ahmad Khan
Reg#: 613-FBAS/MSCS/F10

Supervisor:
Dr. Ali Daud
Assistant Professor

Department of Computer Science & Software Engineering,
Faculty of Basic & Applied Sciences, International Islamic University,
Sector H-10, Islamabad

- Computer Software
- LDA model
- Performance evaluation

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

A thesis submitted to the

Department of Computer Science

and Software Engineering,

International Islamic University, Islamabad

As a partial fulfillment of the requirements

for the award of the degree of

MS Computer Science

International Islamic University
Department of Computer Science & Software Engineering

Date: 21/05/2015

Final Approval

This is to certify that we have read and evaluated the thesis titled **"Use of LDA and POS Tags for Efficient Search of Plagiarized Passages"** submitted by **Jamal Ahmad Khan** under **Reg. No. 613-FBAS/MSCS/F10**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of MS in Computer Science.

Committee:

External Examiner
Dr. Sohail Asghar
Chief Technologist
Department of Computer Science
COMSATS Institute of Information Technology
Park Road, Chak Shahzad, Islamabad
Phone #: +92 51 9049 5295

Internal Examiner
Dr. Jamal Nasir
Assistant Professor
Department of Computer Science & Software Engineering
Faculty of Basic & Applied Sciences
International Islamic University
Sector H-10, Islamabad
Cell #: 0323-7233133

Supervisor
Dr. Ali Daud
Assistant Professor
Department of Computer Science & Software Engineering
Faculty of Basic & Applied Sciences
International Islamic University
Sector H-10, Islamabad
Cell #: 0332-5318403

Ali Daud

DECLARATION

I, hereby declare that **"Use of LDA and POS Tags for Efficient Searching of Plagiarized Passages"** neither as a whole nor as a part thereof has been copied out from any source. I have developed this project and the accompanied report entirely on the basis of my personal efforts made under the sincere guidance of my supervisor. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institution of learning.

Jamal Ahmad Khan

613-FBAS/MSCS/F-10

ACKNOWLEDGEMENTS

All praises and much gratitude to Almighty Allah, the most merciful and glorious, who granted me the potential to work hard and perseverance to accomplish this research work.

I would like to sprinkle special thanks on my supervisor **Dr. Ali Daud, Assistant Professor**, who always provided me greatest support and help whenever I needed throughout my research work. He was readily available for consultation, shared his knowledge with me as well as motivated me for successful completion of this research work.

I can't forget successful support of my affectionate parents and my wife, who always have shown desire and prayed for my success as well as provided me financial and moral support throughout my life.

I would like to thank all my respectable teachers, sincere friends and all those people in the faculty who helped me during this research project.

Jamal Ahmad Khan

613-FBAS/MSCS/F-10

PROJECT IN BRIEF

PROJECT TITLE	:	Use of LDA and POS Tags for Efficient Searching Plagiarized Passages
UNIVERSITY	:	Department of Computer Science & Software Engineering International Islamic University, Islamabad.
UNDERTAKEN BY	:	Jamal Ahmad Khan 613-FBAS/MSCS/F10
SUPERVISED BY	:	Dr. Ali Daud Assistant Professor Department of Computer Scienc & Software Engineering International Islamic University, Islamabad.
TOOLS USED	:	Microsoft C#.Net 2008(for development purpose) Hunspellx86.dll (for word stemming and synonyms generation) OpenNLP.dll (for sentence Tokenization) XML Linq (for querying xml documents) MS Office 2007 for documentation & presentation
OPERATING SYSTEM	:	Windows 7 (32-bit.) HP CANON Intel (R) Core (TM) 2 Duo CPU T2390 @ 1.87GHz RAM 1 GB
START DATE	:	January, 2013
COMPLETION DATE	:	May, 2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this research thesis a new method is presented to address the detection of external textual plagiarism between the suspicious and a number of source documents. In recent years a number of plagiarism detection models have been deployed in order to cater for the wide spread plagiarism of text documents; but most of the models use the n-gram based comparison approach in order to confirm detected plagiarism cases. Whereas the use of Syntactic features and Part of speech analysis at sentence level are so far mostly used for intrinsic plagiarism detection. Our method is based on the syntactic analysis of text using "Part of Speech (POS)" tags and Latent Dirichlet Allocation model (LDA) for topic extraction of text windows i.e. sentences, for external plagiarism detection. Our method is based on two steps, naming, preprocessing where we detect the topics from the sentences in documents using the LDA and convert each sentence in POS tags array then a post processing step where the suspicious cases are verified purely on the basis of semantic rules. Our main aim is to build an efficient model for the accurate and precise search of plagiarized areas of text within the suspicious documents (without the use of n-gram based technique at any stage) and compare its efficiency from some of the pre-existing n-gram based detection models. Also, by efficiency we mean only the accuracy plagiarism detection and we did not took into account the time required for our model to produce plagiarism detection results.

Keywords: *Plagiarism detection, Extrinsic, LDA, Part of Speech, Syntactic analysis, n-gram*

CHAPTER 1

INTRODUCTION

1. Plagiarism

With the Advent of and popularity of search engines like Google, Yahoo, AltaVista etc. and online information sources like Wikipedia, the availability of documents on different topics has increased. People from different departments of education and research download such available documents as a supporting material in their related work. However, there is a drawback of such easy availability and access of these documents as more and more people either try to copy the parts of whole document or the whole document itself to show others that the copied work is related to them without giving a reference to the original work. So, basically this copying of other's work, statements or ideas in one's own document is called plagiarism. In Latin the word "plagiarius" means kidnapper from which word "plagiarism" is derived. It is defined as "the passing off of another person's work as if it were one's own, by claiming credit for something that was actually done by someone else" [1].

The Impact of plagiarism is huge at educational level as every year more and more graduating and under graduate students try to get involved in online plagiarism for the submission of their assignments as this is an easier and shorter way for them; but as a result of this new thoughts and ideas never get revealed as students copy and paste answers to questions or the solutions to the problems as is in their own assignments. Same behavior is seen at higher research level where some researchers try to publish someone else's work as their own with very minor changes. Moreover, the credit to original author is denied. As the cases of plagiarism increased with time; universities and other research institutes bothered to find a solution to this problem. Over the years different algorithms and solutions have been proposed to find plagiarism within the documents written in different languages, but still there are different complications for this task of plagiarism detection because the task itself gets complicated when people have to reference some previous work in their scholarly articles like equations, quotes, definitions or even algorithms etc. so the false positive results may increase.

Hence there is always a need of some standard technique to detect plagiarized passages within a document. The task of plagiarism detection can be divided into two main categories.

a) External Plagiarism Detection

This approach deals with the detection methods applied when we have some external reference to the suspicious document available from which text may have been copied. Also this type of reference oriented search is highly dependent on topic of document or the set of keywords used in suspicious document. This method can be divided into two additional classes; one which works at the local level of user's PC and analyzes the local archives of source documents or carry out internet exploring, the second approach is the one which facilitates the client to upload the suspicious documents at remote server and the plagiarism detection processes takes place remotely at other machine [2]. Figure 1a show how the suspicious documents are queried and compared in the form of segments from both local and online available resources that are referenced in the suspicious document. Also the two very basic techniques for plagiarism detection are shown. However, there are also a number of other plagiarism techniques that can be applied in this specific case.

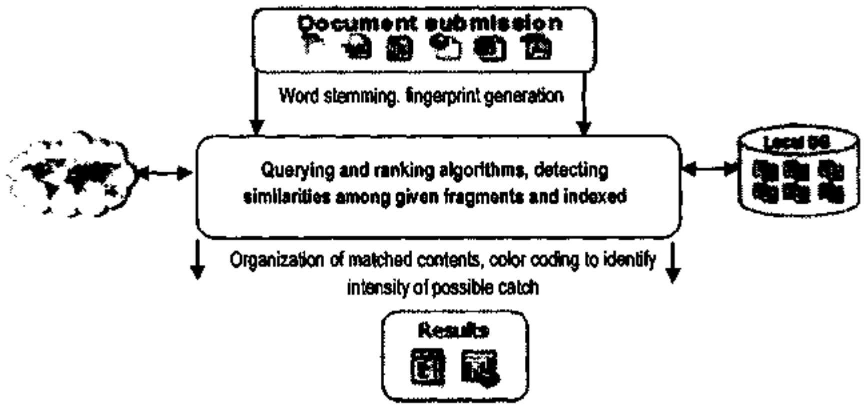

Document submission
Word stemming, fingerprint generation
Querying and ranking algorithms, detecting similarities among given fragments and indexed
Local DB
Organization of matched contents, color coding to identify intensity of possible catch
Results

Figure 1a: Shows the way a suspicious document is compared to external resources [2]

In case of online plagiarism detection one may identify several suspicious sentences from the write-up and feed them one by one as a query to a search engine to obtain a set of documents. Then human reviewers can manually examine whether these documents are truly the sources of the suspicious sentences.

b) Intrinsic Plagiarism Detection

Intrinsic plagiarism detection is a very recent technique used now days in order to detect the plagiarism within a document when there is no reference corpus is available. Different Stylometric features are used to detect plagiarized text pieces or sentences in a suspicious document. Some of the other intrinsic types where the Stylometric methods can be applied are Authorship attribution and Self -Plagiarism; another field where Stylometric features can be used is forgery in legal documents.

1.1 Motivation

Plagiarism detection task can be divided into two broader categories one is called as External Plagiarism Detection and the other is called as Intrinsic Plagiarism detection. Both methods differ from each other by definition and methodology and both have different complications.

Most external plagiarism detection techniques that involve measuring the text based similarity among the documents, are based on two important steps of plagiarism detection. The first step is to represent the document in a way that it can provide a platform for second step which is comparison. These representations of documents include the "Bag-of-Word" model, document Fingerprints, N-grams, and probabilistic models. The second step is the similarity measure that is used to calculate the similarity among source and suspicious sentences. In this thesis we presented a new model based on the technique used mostly in intrinsic plagiarism detection i.e. use of parts of speech (POS) tags and limited topics to represent a document and all of its sentences as first step and then we calculate the similarity among source and suspicious sentences based on the matching topics and the percentage of matching sequences among different classes of POS tags e.g. nouns, verbs, pronouns etc.

1.2 Research Contribution

Syntactic-based methods do not consider the meaning of words, phrases, or sentence, thus the two words "Jamal" and "Japan" are considered same for this approach as both are nouns and hence may have same POS tag. This has been a major limitation of syntactic methods in detecting some external plagiarism. Hence, we used sentence topics in order to relate two or more sentences in suspicious and source documents with each other, it will be easier to find related text passages that may or may not be obfuscated when used in suspicious documents, because even if a sentence is fully rewritten or rearranged the topic of the phrase will always be the same. So, matching only the limited set of topics will not only simplify the search of related passages but it will also make it quicker.

An advantage of syntactic approach speeds up gain comparison of source and suspicious passages in post-processing stage especially for large data sets because the comparison does not involve deeper analysis of the structure and meaning of terms. So, comparing the POS tags in post processing stage speeds up the search of plagiarized passages.

As discussed in section 1.1, the use of POS tags for document representation will also be helpful in future to understand which of these POS classes can help most effectively to uncover plagiarism in text.

This model will be a new direction and a small step forward in the field of external plagiarism detection.

1.3 Research Objectives

The main objectives of this thesis are stated as follows:

1) Finding external plagiarism by using part of speech tags and querying the suspicious documents passages using generated topics through Latent Dirichlet Allocation model.
2) Improving plagiarism detection results through multiple experiments.
3) Comparing our results with other detection models over the self-built and standard datasets.
4) Show both qualitative and quantitative results in form of graphs and tables.
5) Finding limitations of our model through obtained results.

1.4 Research Delimitations

This research only focuses on finding plagiarized passages from pre-nominated source and suspicious documents present in local directories i.e. the task of text alignment. We have not considered searching for related source documents from large text documents archives first and then finding plagiarism in suspicious document i.e. the task of source document retrieval. Also we did not took into account the overall running time and the storage space requirements of our approach as the main focus was only the efficiency to find plagiarized passages.

1.5 Problem Background

Now days new plagiarism detection methods are discovered for academic purposes and many software models are used that are based on standard n-gram approach such as Turnitin [3], DocCop [4], CoReMo [5] etc. but the use of word n-gram based approach for external plagiarism detection can become unfruitful as the n-grams do not take into account semantic similarity and may fail in case someone change the plagiarized text with synonyms and obfuscate it significantly. Also the detection models such as Stop-words n-grams, can also prove worthless in case someone remove, alter or rearrange stop-words from plagiarized passages.

Moreover the use of word sense disambiguation (WSD) which makes it difficult to determine exact sense of word in the phrases and hence even more difficult to decide plagiarism cases for syntactic rules based detection models if these do not rely on n-gram based approach. Stylometric techniques have always been used for intrinsic plagiarism detection and these have not been fully applied in the field of external plagiarism.

1.6 Problem Statement

To cater the problems introduced in section 1.5, this thesis is carried out to answer the following questions:

1) What other approach to detect plagiarism should be adopted which do not have semantic sensitivity like the n-gram based approach?

2) How the Stylometric plagiarism detection techniques like Part of Speech analysis can be helpful to uncover plagiarism in external plagiarism detection?

3) What can be done in cases where someone replace or obfuscate the plagiarized passages with respective synonyms?

4) Can the intrinsic plagiarism detection technique perform as good as the classical n-gram and document fingerprint approaches?

5) How the use of topic base search can make the selection of related passages more efficient and fast?

6) How the use of WSD with POS tags can also be efficient in finding plagiarism without use of n-gram or VSM model approaches?

1.7 Structure of Thesis

This Thesis is organized as follows:

Chapter 1 formulates the problem and outlines the framework and main objectives of the project.

Chapter 2 consists of detailed literature review of different research papers; we have not only analyzed and compared different plagiarism detection methodologies but also have identified limitations in the chapter.

Chapter 3 illustrates the methodology that will be used to fulfill the objectives of this research thesis.

Chapter 4 presents the experimental results of this thesis, and finally chapter 5 concludes this research.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

This chapter focuses on some of prevalent techniques (both old and new) and methods used in the field of extrinsic plagiarism detection; here we have reviewed a number of research papers which have been divided into different plagiarism detection categories. These categories are explained separately as follows.

2.1.1 Document Fingerprint Analysis

A fingerprinting algorithm produces shorter representative bit strings (chunks) for large textual documents and these bit strings are called its fingerprint. Following figure shows how a string can be converted into a representative compact string. This approach has been used to fulfill the task of plagiarism detection because fingerprinting technique offers fast comparison of bit strings called chunks through hashing.

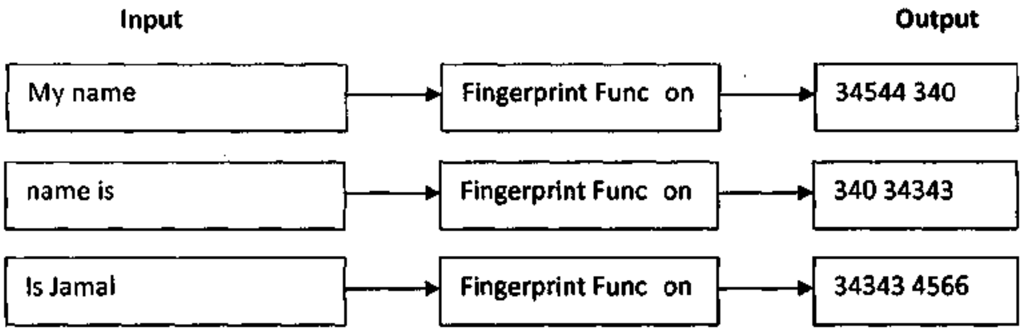


Figure 2a: Shows the way a of chunking and hashing

Effectiveness of this approach depends on the size of chunks chosen because smaller chunk size will increase false positive and larger size will increase false negative detection results.

In past many techniques with different sizes of chunks were introduced. The "shingling approach" [6] considers fix sized chunks of 10 consecutive words and compared the hash values to detect similarity. Authors [7] compare different chunking strategies, such as removal of stop-words and non-overlapping chunks i.e. sentences, they also introduced hashed-breakpoint chunking by calculating a hash value on each word and whenever this hash value modulo *k* is 0, it is taken as a chunk boundary where *k* could be 5 to 10.

A model [8] was introduced which compared above mentioned fingerprinting techniques and the sub-string based comparison based on suffix trees was used to detect plagiarism.

A new class of fuzzy-fingerprints [9] was introduced that was based on the classical vector space model. These fingerprints allowed for chunks of larger sizes than the typical ones used in previous fingerprinting models which boosted the speed for search of plagiarism candidates. The main limitation of this model was that it did not improve the detection results.

A model [10] for detection of plagiarism in Arabic text was proposed that was based on inverted index model as used in search engines. In this approach an index term represented a single fingerprint i.e. each index term represented n normalized words and each inverted list consisted of the set of sentences that contained the fingerprint. This way querying for the search of related chunks became easier.

A recent approach [11] with fingerprints of word 3-grams was used to detect plagiarized passages. The documents were first broken into passages and then extracted the 3-gram fingerprints of texts. After full fingerprinting of the document the method selects k-th fingerprint from each sentence where the value of $k = 5$. The importance of passages was determined by hash values assigned to all k-grams and then the similarity index was calculated.

A limitation of this approach is that there is no surety that matches between documents are detected specially in case of word and sentence n-gram chunks are chosen for fingerprinting because a fingerprint also contains positional information, which describes the document and the location within that document that the fingerprint came from, so, in case of higher obfuscation where a plagiarist reorders or change a phrase with translated words or synonyms, the probability of fingerprint matching will lessen.

2.1.2 Syntactic Features Analysis

This approach was used first time by authors [14] to explore sets of syntactic arrangement of text that contain information about different ways of writing and they showed how this information could help to find similarities between two texts. They observed how different authors try to express same content in a translated and paraphrased passages using different syntactic arrangements of words. They chose one book from each title and used this book trained a model that learned the syntactic elements of expression used in this title and afterwards remaining books paraphrasing the title were used as the test set.

Table 2a: Sample syntactic formulae [14]

Syntactic Formula	Example
NP + Vh + NP + from + Partcp	The belt kept him from dying
NP + Vh + that + IS	He admitted that he was guilty
NP + Vh + that + Subjunct	I request that she go alone

Although the detection results for the detection model discussed above were excellent but the main limitation of the model was its limited dataset, the authors trained a model to make rules first over the pattern of syntactic elements but in real world where the datasets are huge, such model will fail to get high results.

The model may not have worked for well for author identification tasks but not for external plagiarism because for external plagiarism one has to analyze one suspicious document against a number of source documents.

2.1.3 Bag of Words Analysis

Bag of words analysis is the use of vector space (VSM) retrieval in the domain of external plagiarism detection where the documents and phrases are represented as one or multiple vectors, e.g. for different document parts, which are used for pair wise similarity analysis. Similarity computation may be computed through cosine similarity measure, or on more sophisticated similarity measures like Jaccard's similarity [12]. Also it is important to mention the term **n-grams** which are the building blocks for this approach. N-gram is a continuous sequence of *n* items which typically are collected from a text corpus [13]. The value of n can be 1 (unigram), 2 (bigram), 3 (trigram) and so on depending on the type of model to detect plagiarism. Following are some detection models that use the n-grams for plagiarism detection.

2.1.3.1 Use of Syntactic Features with N-Grams

An online plagiarism detection framework [15] was introduced which incorporated a number text analysis techniques like the classical n-gram analysis with $n = 2$ and syntactic features analysis like Part of Speech (POS) and phrasal tags for words in sentences of documents. They used POS tagging in order to decrease the false positive results that could have emerged after alignment and reordering of two word sequences. The similarity measure for POS tags matching is shown as follows.

$$sim = \frac{num\ (matc\ hed\ words\ \ wit\ h\ similar\ \ tags)}{num\ (matc\ hed\ words\ )} \qquad (1)$$

The alignment and reordering was done to cater the cases where the plagiarist may add or delete the words or just rewrites the phrases using the same words. They also used Latent Dirichlet Allocation (LDA) model in order to obtain the topic distribution of a query and a candidate snippet, and compare the cosine similarity of them but however this feature was not used in final evaluation of model. Final results over the test dataset showed that only the n-gram based analysis results outperformed all other syntactic based approaches when each was tested separately.

In another approach [16], authors make use of combination of lexical analysis tools and n-grams techniques for detection of both verbatim i.e. copy paste and slightly modified plagiarism passages. They divided their work into two steps i.e. preprocessing step where they used an NLP tool to performed sentence splitting, word tokenization and lemmatization. Later the lemmatized tokens were used to tag with its respective POS. In the post-processing step they used the conventional tri-gram based approach to find related text passages. Each pair of sentences from source and suspicious document that have at least three overlapping tri-grams and a similarity degree over the threshold of 0.25 was considered as a plagiarism case. They used cosine similarity index for measuring similarity.

$$similarity\ (d1, d2) = \frac{d1 * d2}{|d1| * |d2|} \qquad (2)$$

Another approach [17] was used to find related passages of source and suspicious documents where the bigrams matched with dice similarity measure as shown in equation (3), where C is the number of similar words.

$$Q = \frac{2C}{(A+B)} \tag{3}$$

All passages retrieved through the similarity measure were chosen for post-processing where phrasal POS tagging was deployed in order to analyze the syntactic relatedness of source and suspicious sentences. The cases where the words in the suspicious sentences did not match with those in source sentences were examined with synonyms produced with WordNet, taking into account the POS category of the word. The approach showed high precision with low recall which meant that the rate of false negative cases was higher.

The detection models [16, 17] discussed above seems to rely on n-gram based approach as post and pre processing steps which limited the performance in cases where the semantic and syntactic obfuscation with synonyms replacement was used.

2.1.3.2 Stop-Words N-grams (SWNG) Analysis

A novel approach was presented [18] that showed how the stop-words n-grams can play a vital role for plagiarism detection since these stop-words expose syntactic resemblance among suspicious and source documents. The author used 50 stop words is mentioned that are most frequently used in English text.

As the preprocessing step each passage of a document was condensed to the appearances of the stop-words in that document by removing all other words as shown in table 2b. After that these condensed SWNGs were sliced into overlapping chunks of length n1 and to find similar passages (for post-processing) between the source and suspicious documents, these SWNG chunks were compared.

Table 2b: Conversion Sentences into SWNG chunks

Sentence	SWNG Chunks
My country is Pakistan and I live there	My is and I there
The boys were playing in the street	The were in the

In order to escape from any coincidental matches of different passages comprising of same SWNGs, following equation was used.

$$\exists g \in \{P(n_1,dx) \cap P(n_1,ds)\}: member(g,C)<n_1\text{-}1 \ \wedge \max seq(g,C)<n_1\text{-}2 \qquad \textbf{(4)}$$

Where g is the chunk of size n1 that exists commonly in documents *dx* and *ds* and *C* is the sequential list of 6 most frequent stop-words. Once the suspicious passages are retrieved the similarity measure was calculated depending on the tri-gram matching approach.

This model showed a new approach to detect similarity among the passages specially where there are least changes in a passage, but the main limitation of this approach is the length of smaller plagiarized passages where the numbers of SWNGs will not match the criteria of equation 4. Also in cases where the order of sentences in a plagiarized paragraph is changed this approach will fail to show significant results.

2.1.3.3 Word N-Gram Based Analysis

In this approach the authors [19] used the vector space model for the detection of external plagiarism detection. The aim was to speed up the search in high dimensional vector spaces at the cost of precision. First, each sentence of each document in the reference corpus was vectorized and second, each sentence of a suspicious document was vectorized and queried to find each passage's nearest neighbor(s) in the reference corpus vector space by cosine similarity index.

Same approach [20] was introduced in PAN-10 competition with the difference that value of word 4-grams were used as vectors and the values of these vectors were determined through the term frequency and inverse document frequency (tf-idf) method and finally cosine similarity measure was used to determine similarity.

Discourse Markers based Approach [21] proposed a plagiarism detection system which can detect external plagiarism using the VSM and word 7-grams as discourse markers. The authors used the VSM model to retrieve the related source documents from corpus. After the retrieval they took a 7-gram from the suspicious document and searched for it in source documents, if it matched then from that matching point they took 25 words window in both suspicious and source documents and compute the similarity index.

Name-Entity (NE) relationship based approach [22] used along with the conventional n-gram approach. The suspicious documents were tagged with Lingpipe NE Tagger and queried to source corpus for retrieval of source documents and overlapping n-grams (n=7) of suspicious documents were compared with those of source documents to find plagiarized passages.

In another approach [23] all source documents were transformed into overlapping blocks of 40 words and these blocks along with offset, length and document id were indexed by using an indexing and query software called "Lucene". After that, suspicious documents were tokenized and overlapping blocks of tokens were transformed to Boolean queries and terms of the Boolean query were sorted by their corpus frequency in increasing order.

In order to detect plagiarism from online submission of assignments, a new model [24] based on k-means clustering with n-grams was introduced. This model clustered documents into related cluster and improved time latency. The approach to find similarity among source and suspicious documents was conventional with $n = 3$.

Most of the approaches discussed above showed a low recall score which clearly means that a high number of suspicious passages were not detected i.e. high false negatives. This means that the VSM based approach traded fast detection with a low recall. The main limitation of all the approaches seems to be the value of n-gram which is greater than the one recommended i.e. $n = 3$ [25]. Also one main limitation of n-gram based approach is that this approach is not syntactically sensitive i.e. the words "U.S.A" and "USA" are different for this approach also the words 'effective" and "useful" would be different, this is because "n-gram models are not designed to model linguistic knowledge" [25]. Also most of these approaches have to remove stop-words as a noise reduction step, whereas the stop-words patterns can depict important information about text passages.

2.1.4 Using Word-Net based semantic similarity

Word-Net is an important tool to counter for the paraphrased cases in external plagiarism detection. Using this tool the authors [26] performed the semantic text comparison for plagiarism detection. Given two sentences X and Y, they denoted m and n as the lengths of X and Y, they constructed a matrix R[m, n] which showed semantic similarity among each word pairs. This approach however failed to show any significant results.

A recent approach proposed by authors [27] in terms that they represented the chunks of texts in documents by bitmaps where each occurrence of term in the document was represented by "1" and absence by "0". While converting the document sentences to respective bitmaps each term in the sentence was consulted with Word-Net and in case the Word-Net offered synonyms to that word, additional 1's are added to the bitmap of that sentence. Also the terms that occur more were added to in the start of bitmaps; this allowed the authors to skip first n-terms while comparing source and suspicious bitmaps. The similarity measure among two bitmaps was calculated through Jaccard's coefficient shown in equation 5.

$$J(A,B) = |\frac{A \cap B}{A \cup B}| \qquad (5)$$

The authors did not used any standard dataset to prove the efficacy of this model.

The following table presents year wise comparison of above mentioned plagiarism detection models while the advantages and limitations are represented by following characters.

a) **Advantages:** *A* = Fast search of plagiarized passages, *B* = Good detection results, *C* = No Semantic sensitivity

b) **Limitations:** *a* = Bad detection results, *b* = Limited Dataset, *c* = Time Consumption, *d* = Semantic Sensitivity

Table 2c: Summarized Comparison of different Plagiarism detection models discussed

Year	Model	Methods Used	Advantages	Limitations
1996	Building a Scalable and Accurate Copy Detection Mechanism	Hashed-breakpoint	*A*	*a, b, d*
2005	Using Syntactic Information to Identify Plagiarism	Phrasal tagging, Rule based Learning	*B, C*	*b, c*
2006	Near Similarity Search and Plagiarism Analysis	Fuzzy fingerprints, VSM	*A, B*	*b, d*
2009	External and Intrinsic Plagiarism Detection Using Vector Space Models	VSM, N-Gram, Nearest Neighbor search	*A, B*	*d*
2010	External Plagiarism Detection	VSM, N-Gram	*A, B*	*d*
	External Plagiarism Detection: N-Gram Approach using Named Entity Recognizer	N-Gram, NE relationship	*A, B*	*d*

	Automatic External Plagiarism Detection Using Passage Similarities	N-Gram	A	a, d
2011	Plagiarism Detection Using Stopword n-grams	SWNG, N-Gram	*A, B*	*d*
	External & Intrinsic Plagiarism Detection: VSM & Discourse Markers based Approach	VSM, N-Gram	*A, B*	*d*
	WordNet-based semantic similarity measurement in External Plagiarism Detection	N-Grams, word-net	*A, B*	*c*
2012	A Fingerprinting-Based Plagiarism Detection System for Arabic Text-Based Documents	Hashed reverse indexes	*A, B*	*b, d*
	Online Plagiarism Detection Through Exploiting Lexical, Syntactic, and Semantic Information	N-Gram, Phrasal tagging, LDA,	*A, B, C*	*c*
2013	AntiPlag: Plagiarism Detection on Electronic Submissions of Text Based Assignments	N-Gram, data clustering	*A, B*	*d*
2014	Plagiarism Detection	Text Fingerprints	*A, B*	*d*
	NLP Applications in External Plagiarism Detection	N-gram, Phrasal tagging	*A, B*	*c*
	Identification of Plagiarism using Syntactic and Semantic Filters	POS Tagging, Word-Net	*A, B, C*	*c*
	A Concept for Plagiarism Detection Based on Compressed Bitmaps	VSM, N-Grams, Word-Net	*A, B, C*	*c*

CHAPTER 3

RESEARCH METHODOLOGY

In this section we will present our model "*Use of LDA and POS Tags for Efficient Search of Plagiarized Passages*". Our method will syntactically analyze the word windows in order to address the above problems that may occur while using n-grams profile and in case of changing the order of words in plagiarized passages. Replacing words with their respective POS tags will allow us to not to be dependent on the software like Word-Net and hence attaining the target of rapid searching of plagiarized chunks within a matrix space. We will also use Latent Dirichlet Allocation model along with POS tags (of text windows) to get the actual text context. Let's discuss these two components of proposed framework in detail.

3.1 POS Tagging

Through POS tagging (which is also known as grammatical tagging) we mark a word or term in a text document as its related part of speech. This approach is based on the term classification and the context in which it's used i.e. association with neighboring words in the same phrase, sentence, or article. POS tagging is a difficult task than merely having a catalog of terms and their related parts of speech, because some terms can express more than one part of speech when written in different situations, and also because some are difficult to express. However the most common parts of speech are 9 which are noun, verb, article, adjective, preposition, adverb, pronoun, conjunction and interjection [28].

Here we have divided each type of POS in different groups and allocated a decimal number for each one in order to make the post processing and comparison more easier and faster. Let's discuss some of the important POS that we have used in our project.

3.1.1 Adjectives

An adjective is used to change, qualify, describe or quantify a noun or a pronoun. Tags assigned to different forms are JJ, JJR and JJS.

3.1.2 Adverbs

Like the adjective for nouns and pronouns; an adverb modifies a verb, an adjective, another adverb, a sentence, or a condition. An adverb specifymode, time, place, reason, or degree and answers questions such as "how," "when," "where," "how much".A class of adverbs can be recognized by their "ly" suffix. . Tags assigned to different forms are RB, RBR and RBS.

3.1.3 Stop Words

In case of plagiarism detection stopwords mean all those words that occur most frequently in a given text corpora except nouns and verbs, but in case of grammer each of these words are categorized as different class of speech. We however, assumed following POS cotegoris as stop-words while processing text in our algorithm.

- **Conjuction:** You can use a conjunction to join words or sentences e.g. I ate the apple "and" the mango. Tags assigned to different forms are CC, IN and TO.
- **Determiner:** This category express terms "a(n), every, no and the, another, any, and, some, each, either, neither, that, these, this and those. Tags assigned to different forms are DT and EX.
- **Cardinals:** Numerals and digits are included in this category.Tag assigned to this category is CD.
- **Foreign:** Words which are non-english but are included from other languages e.g. noir, beta, gama; Tag assigned to this category is FW.
- **Posessive Ending:** When we have possessive ending at the end of nouns in 's which is usually split from noun or pronoun by tagging algorithm. Tag assigned to this category is POS

3.1.4 Pronouns

The following types of pronouns are taken into account.

- **Personal Pronoun:** Personal pronouns are represented without the taking into account for case difference e.g. I, me, you, he, him, etc. Tag assigned to this category is PRP.

- **Posessive Pronoun:** The adjectival possessive forms my, your, his, her, its, our and their, on the other hand, are tagged PRP$.

3.1.5 Nouns

Indicates the name of any thing, person, personality and relation e.g. parents, tables, jamal etc. Tags assigned to this category according to types are NN, NNP, NNS and NNPS.

3.1.6 Verbs

This tag subsumes any work performed by any thing or person; this includes Imperative. It has following forms e.g. Do it; present participle-VBG, I am working; past participle e.g. gone, done; 3rd person singular e.g. goes, cares. Tags assigned to this category according to types are VB, VBD, VBG, VBN, VBP and VBZ.

- **Modal verb:** This type takes into account all verbs that do not have an -s closing in the third person singular present e.g. can, could, (dare), may, might, must, ought, shall, should, will, would. Tags assigned to this category according to types are WP, WP$ and WDT.

3.2 LDA Model

Latent Dirichlet Allocation is a prevailing algorithm that can learn by itself by clustering group of words into "topics" and documents into fusion of "topics". This model is applied successfully in different scientific areas. We can describe a topic model as a Bayesian model that links probability allocation over the topics in each document. Topics are in fact distributions over words [29]. An example of how LDA works is presented below by a number of sentences.

1. I like to eat mangoes and apples.

2. I ate an apple and bread for my breakfast.

3. Tigers and puppies are pretty.

4. My brother bought a cat today.

5. Look at this pretty cat chewing on a piece of bread.

So, LDA is simply the model to find topics hidden in these sentences.

Sentences 1 and 2: 100% Topic A

Sentences 3 and 4: 100% Topic B

Sentence 5: 60% Topic A, 40% Topic B

Topic A: 30% apples, 15% mangoes, 10% breakfast, 10% chewing … (here we can guess that first is related to food)

Topic B: 30% cats, 20% tigers, 20% pretty, 15% puppies,(here we can guess that first is related to animals)

We have used Gibb's LDA [30] written in C++ in our software for generating topics ranging from 1 to 10 for each sentence depending on the type of configuration we choose. The parameters that are set for generation of topics by Gibb's LDA are following.

Table 3a: Gibb's LDA Parameters Settings

Parameter	Description	Values
-est	Estimate the LDA model from scratch	-est
-ntopics	The number of topics to produce for each sentence	Values from 1 to 10 per sentence
1	The number of most likely words for each topic	_
Data.txt	File name to write topics	_

3.3 Steps Performed for Plagiarism Detection

In this section we will discuss the methodology steps of proposed framework. The task is divided into two steps i.e. Pre-Processing of source and suspicious documents which produces representative XML files for each document processed. These XML files includes information about every sentence of processed document including sentence number, sentence topics, topics synonyms and array of POS tags.

Post-Processing is done after the XML files are generated in order to find out the related passages on the basis of similar topics among source and suspicious passages. After retrieval of source and suspicious passages we apply certain syntactic rules to relate the two passages and find if the suspicious passage is plagiarized or not.

3.3.1 Pre Processing

1. POS tags are assigned to every sentence present in both source and suspicious documents as shown in following table. This step of tokenization and tagging will be done using Open-NLP library for windows.

Table 3b: Showing Sentences with their respective Tags

No.	Sentence	POS Tags
1	Maria can lastly put some cash in the bank	NNP MD RB VB DT NN IN NN
2	Finally, Martha can lay some money in bank	RB NNP MD VB DT NN IN NN
3	Nora sent the book to Richard	NNP VBD DT NN TO NNP
4	Noor lead the volume to Rashid	NNP VBD DT NN TO NNP
5	That's her mistake	DT PRP NN
6	That's her fault	DT PRP NN

2. Each tag will be represented by a one unique number e.g. NN will be represented by '1' and VB will be represented by '2' and so on. Hence we will get an integer string or array of variable length for each sentence's POS tags window. This step will fasten up the post-processing.

3. To find out the topic of each sentence, first the stop-words are removed from it. The list of all stop-words that we remove in this step is shown in table 3c.

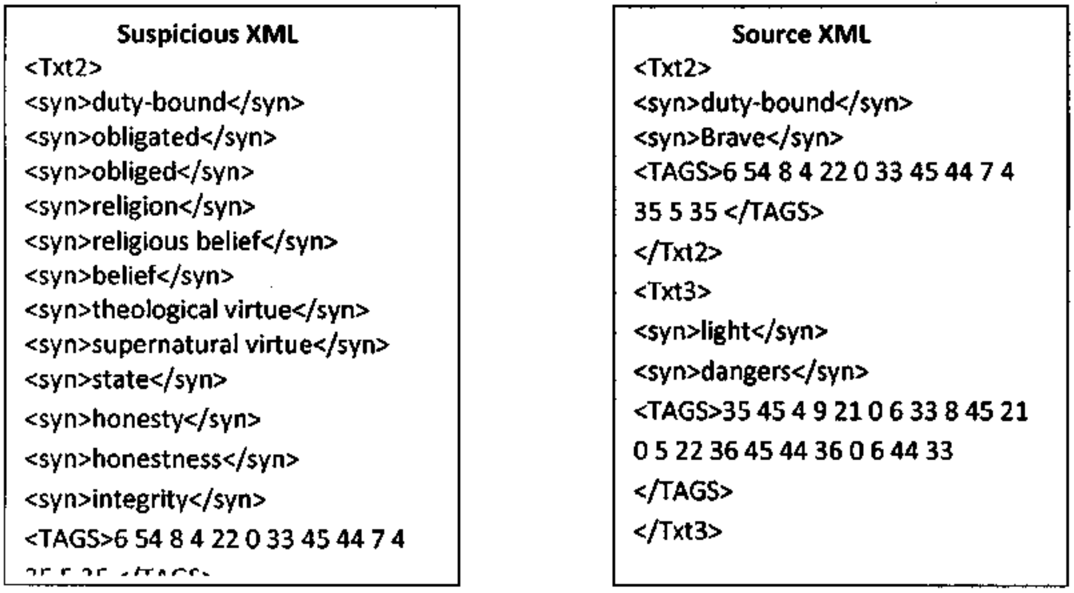
Table 3c: List of Stop-Words removed for sentence topic production

the	many	can't	will	on	my	his	was	be
your	more	isn't	would	how	which	she	were	there
mine	too	couldn't	shall	very	with	they	are	do
you've	haven't	shalln't	should	much	where	them	of	did
I've	shouldn't	don't	not	we	her	their	an	done
into	hasn't	there's	nor	I	him	is	also	so
being	hadn't	that's	no	you	a	this	in	all
must	wasn't	n't	yes	he	on	these	from	to
that	as	by	had	has	if	at	may	might
but	ought	n't	have	who	it	it's	what	and
been	or	for	„s	how	can	could	if	else

4. Then the sentence will be assigned a topic tag using the LDA which will be used while querying the suspicious sentences. The number of topics produced for each sentence depends on the threshold value α chosen by user; where α is the number of words chosen for each topic assignment. By default its value will be 1 (topic per sentence) for suspicious document and 2 (topics per sentence) for source documents.

5. The synonyms for each topic will also be generated in case of suspicious documents pre-processing.

6. Each sentence along with topics and POS tags will be written to an xml file along with its index number. This index number is used in case to back track the text from documents.



Suspicious XML

```
<Txt2>
<syn>duty-bound</syn>
<syn>obligated</syn>
<syn>obliged</syn>
<syn>religion</syn>
<syn>religious belief</syn>
<syn>belief</syn>
<syn>theological virtue</syn>
<syn>supernatural virtue</syn>
<syn>state</syn>
<syn>honesty</syn>
<syn>honestness</syn>
<syn>integrity</syn>
<TAGS>6 54 8 4 22 0 33 45 44 7 4
35 5 35 </TAGS>
```

Source XML

```
<Txt2>
<syn>duty-bound</syn>
<syn>Brave</syn>
<TAGS>6 54 8 4 22 0 33 45 44 7 4
35 5 35 </TAGS>
</Txt2>
<Txt3>
<syn>light</syn>
<syn>dangers</syn>
<TAGS>35 45 4 9 21 0 6 33 8 45 21
0 5 22 36 45 44 36 0 6 44 33
</TAGS>
</Txt3>
```

Figure 3a: A view of both Source and Suspicious XML files

Once the xml profiles are complete for each document, we are ready for the next step to query the xml documents for plagiarized sentences.

3.3.2 Post Processing

1. Each sentence topic T_S in source XML file $x \epsilon X_S$ (X_S is the set of all source files) is queried for the matching topic (this also includes topic synonyms) T_u in the Suspicious XML file X_u using LINQ.

2. Each matched result r is 3-tuple <*i, POS-Su, POS-Sr*> which includes index i of sentence S_i in suspicious document and the array of POS Tags decimal values *POS-Su* for suspicious sentence and *POS-Sr* for source sentence.

$$r \epsilon R \text{ where } R = (T_S, X_S) \cap (T_u, X_u) \qquad (5)$$

   Now the decision for the sentence being plagiarized or not depends on the following factors.

   a. Length L which is the longest sequence of matched indices in *POS-Su* and *POS-Sr*.

$$L=(|POS\text{-}Su \cap POS\text{-}Sr|) \qquad (6)$$

   b. The Length ratio L_r between *POS-Su* and *POS-Sr* where we assumed that if the length of one sentence is half or less than the length of other sentence then it should not be processed further.

$$L_r = abs(\frac{max\ (|POS-Su|,|POS-Sr|)}{min((|POS-Su|,|POS-Sr|)}) \qquad (7)$$

c. Indices of stop-word sequence matched *PS* in *POS-Su* and *POS-Sr*

d. Indices of nouns sequence matched *PN* in *POS-Su* and *POS-Sr*

e. Indices of verbs sequence matched *PV* in *POS-Su* and *POS-Sr*

f. Indices adverbs sequence matched *PA* in *POS-Su* and *POS-Sr*

g. Indices of pronouns sequence matched *PP* in *POS-Su* and *POS-Sr*

h. Indices of adjective sequence matched *PD* in *POS-Su* and *POS-Sr*

Moreover a set of threshold values or parameters are to be set in order to compare and adjust the values of matching POS tag values. These values are β_L, β_S, β_N, β_V, β_A, β_P and $\beta_D$which represent matched POS index sequence Length, percentage of matched stop-words sequence, percentage of matched nouns sequence, percentage of matched verbs sequence, percentage of matched adverbs sequence, percentage of matched pronouns sequence and percentage of matched adjectives sequence respectively. All these parameters are to be set by the user in order that these values best reflect the accuracy of our rules (discussed below) while comparing two POS arrays during post-processing step.

Combining equations 6 and 7 we get the basic rule of comparison shown in following equation. Where *pl* is the Boolean result of following equation.

$$pl = ((L \geq \beta_L) \wedge (L_r \leq 2)) \qquad (8)$$

3. In order to decide whether two sentences are syntactically related or not we made some rules, because just comparing the matching sequences of POS tags is not enough as this can lead to false positive results e.g. the two sentences "Dog loves to chew bones and play around" and "cats loves to sleep long hours of day and jump over", these two different sentences can be related to each other through a common topic i.e. "love", so, we are now left with POS based comparison to decide if the sentences are syntactically same or not. The first four highlighted words clearly have same POS tags i.e. "NN VBD TO VBD", so if we set β_L= 4, this will give us a false positive result in absence of other rules. The rules we made are dependent on assumptions that are stated below.

a. The Nouns are important part of speech and semantic content of sentences is borne mostly by nouns [31].

b. Along with Nouns, Verbs are also dominant parts of speech in English Text as Nouns tell us what is involved in a situation; verbs tell us what happens in a situation.

c. The classes of POS that we have grouped as stop-words are the bonds to combine verbs and nouns and express the real meaning of a sentence e.g. "Ali is playing and Hamad is sleeping", "the bird is flying", "the book fell from the shelf" and so on.

So along with the matching sequences of POS tags, we must also consider the matching sequences of any two of the above three POS categories i.e. Nouns and Verbs, stop-words and Nouns and Verbs and stop-words. This rule is depicted by the following equation, where *tl* is the Boolean value.

$$tl = (pl) \wedge \left(\left((|PS| \geq \beta_S) \wedge (|PN| \geq \beta_N) \right) \vee \left((|PS| \geq \beta_S) \wedge (|PV| \geq \beta_V) \right) \vee \left((|PV| \geq \beta_V) \wedge (|PN| \geq \beta_N) \right) \right) \qquad \textbf{(9)}$$

We made another rule depending on the other three classes of parts of speech i.e. Pronouns, Adverbs and Adjectives, assuming that in any grammatical sentence the sequence of these three categories may not play much important role in relating the two sentences, we derived the rule depicted by following equation, where *tf* is the Boolean value.

$$tf = \left((L \geq 2\beta_L) \wedge \left((|PA| \geq \beta_A) \vee (|PP| \geq \beta_P) \vee (|PD| \geq \beta_D) \right) \right) \qquad \textbf{(10)}$$

Combining the equations 9 and 10 we get the single rule over which we can compute the similarity among two sentences syntactically, where *TF* is the Boolean value which is true in case either *tl* or *tf* is true.

$$TF = (tl) \vee (tf) \qquad \textbf{(11)}$$

An example of how indices are matched in two suspicious and source arrays is as follows.

POS-Sr = {35 45 27 4 36 45 35 0 5 21 35 0 27 46 21 7 41 4 21 6 35}

POS-Su = {35 46 6 4 0 27 4 36 45 35 0 5 21 35 0 27 46 21 7 41 4 35 21 6 35 36 0 5 21 36 6 36}

From the above two arrays we have

Longest sequence match L_S = {27 4 36 45 35 0 5 21 35 0 27 46 21 7 41 4}

Length of L_S is L =16

Length Ratio among *POS-Sr* and *POS-Su* is L_r= *abs* (32 / 20)

Total nouns in *POS-Su* = 9

Matching Noun indices in *POS-Sr* and *POS-Su* is *PN* = {35 36 35 35 35}

Length of *PN* = 5

The matching percentage for nouns |*PN*| in *POS-Su* = (5 / 9) x 100

In this way we will calculate all the other parameters.

Once it's decided that S_i is a plagiarized sentence; the plagiarized indices P_i are retrieved where $P_i \subseteq$ *POS-Su*.

$$P_i = L_S \cup PN \cup PS \cup PV \cup PA \cup PD \cup PP \qquad (12)$$

4. After finding the set of plagiarized indices in *POS-Su*; we calculate the *lp* which is the percentage between the Length of P_i and Length of *POS-Su*.

$$lp = \frac{|P_i|}{|POS-Si|} \times 100 \qquad (13)$$

In case *TF* is true for a sentence *Si* and $lp \geq \beta_{LP}$, then all the indices of *POS-Su* are set as plagiarized which means the whole sentence S_i will be considered plagiarized otherwise only the words with indices P_i in S_i will be included in plagiarized set. Where β_{LP} is the parameter value for the percentage of length to be set by user.

5. Another rule that we call **Suspicious Neighbor Rule** was made on the basis of assumption that Nouns are most important parts of speech. This rule is made to gather potential suspicious sentences that fulfill the criteria depicted by following equation are assigned potential suspicious sentences *PSS*.

$$tfp = ((L \geq \beta_L) \wedge (|PN| \geq \beta_N)) \quad (14)$$

In case *tfp* is true for a sentence S_x and *TF* is false for a sentence X_i, we consider X_i as a potential suspicious sentence and a list of all *PSS* sentences is prepared $<X_1, X_2, \ldots X_i \ldots X_n>$

Finally when the set of plagiarized sentences $<S_1, S_2, \ldots S_i \ldots S_n>$ is retrieved, all *PSS* sentences that are neighbors of any of plagiarized sentences in the list are also considered as plagiarized.

Finally all plagiarized sentences that exist in a continuous sequence are made a single plagiarism case and written in another XML file for evaluation purpose.

3.4 Pseudo Code

Following is the pseudo-code written to test the effectiveness of our methodology.

3.4.1 Pre Processing

```
Initialize $i$ to 0
Initialize $n$ to $\alpha$
while $i$ < Total number of documents $d_n$, do:
    Initialize $S_n$ to total number of sentences in document $d_i$
    Repeat $S_n$ times:
        Assign POS Tag to each word in Sentence $S_i$
        Remove Stop Words from $S_i$
        if $d_i$ is Suspicious Document, then:
            Find a topic with respective Synonyms for $n$ words in $S_i$
        else
            Find a topic for n words in $S_i$
            Write topics and POS Tag Arrays to XML file
    End repeat
Increment $i$
End while
```

3.4.2 Post Processing

```
Initialize $i$ to 0

While  $i <$ Total number of Topics $T$ in each node of Source XML  $X_s$, do:

          If  Source Topic $T_s$ matches Suspicious Topic $T_u$, then:

           Select POS Tags Arrays $P_s$ and $P_u$ for both $T_s$ and $T_u$ nodes

           Initialize $L_r$ to ratio of Lengths of $P_s$ and $P_u$
           Initialize $L$ to Maximum number of POS Tags Matched in $P_s$ and $P_u$
           Initialize $PS$ to Maximum stop-word sequence matched in $P_s$ and $P_u$
           Initialize $PN$ to Maximum nouns sequence matched in $P_s$ and $P_u$
           Initialize $PV$ to Maximum verbs sequence matched in $P_s$ and $P_u$
           Initialize $PA$ to Maximum adverbs sequence matched in $P_s$ and $P_u$
           Initialize $PP$ to Maximum pronouns sequence matched in $P_s$ and $P_u$
           Initialize $PD$ to Maximum adjective sequence matched in $P_s$ and $P_u$

         If $L \geq \beta_L$ and $L_r \leq 2$, then:
              Initialize $ta$ to Boolean result of ($L \geq 2\beta_L$ and $|PA| \geq \beta_A$ or $|PP| \geq \beta_P$ or $|PD| \geq \beta_D$)

              Initialize $tb$ to Boolean result of $|PS| \geq \beta_S$ and $|PN| \geq \beta_N$ or $|PS| \geq \beta_S$ and $|PV| \geq \beta_V$ or $|PN| \geq \beta_N$ and $|PV| \geq \beta_V$)

              Initialize $nb$ to Boolean result of ( $|PN| \geq \beta_N$)

              if $ta$ is true or $tb$ is true, then:

                   Initialize $Pi$ to $Ls \cup PN \cup PS \cup PV \cup PA \cup PD \cup PP$

                          If $|Pi| \geq lp$, then:

                          Mark whole Sentence as Plagiarized

                          else

                           Mark only the Matching indices in $Pi$ as Plagiarized

               else if $nb$ is true

                   Mark the Sentence as Suspicious Neighbor

Increment  $i$

End while
```

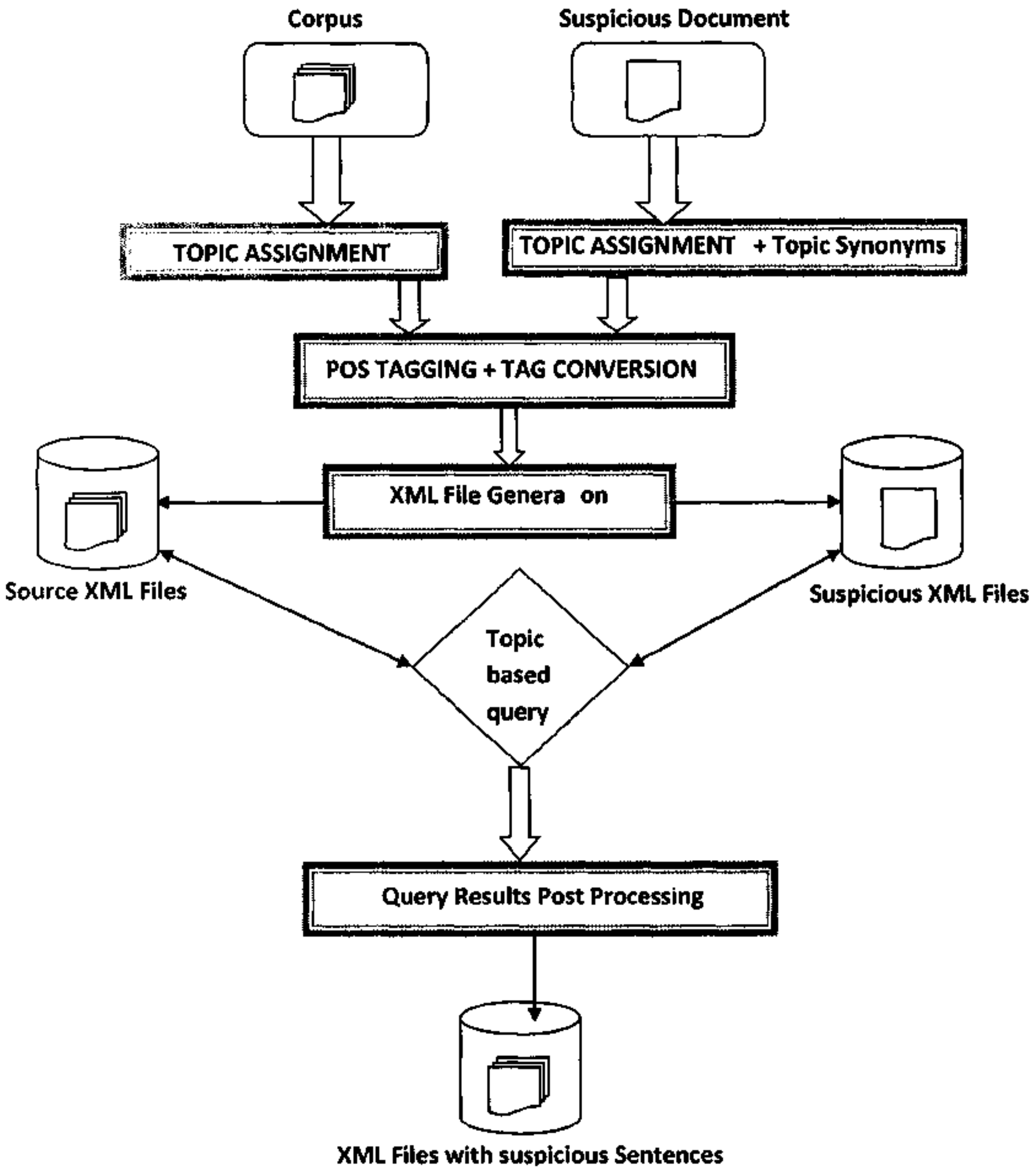

Corpus
Suspicious Document
TOPIC ASSIGNMENT
TOPIC ASSIGNMENT + Topic Synonyms
POS TAGGING + TAG CONVERSION
XML File Genera on
Source XML Files
Suspicious XML Files
Topic based query
Query Results Post Processing
XML Files with suspicious Sentences

Figure 3b: Showing the Overall Methodology of Proposed Framework

CHAPTER 4

EXPERIMENTS

4.1 Dataset

To get some real plagiarism cases, Ideally, we may have to study and monitor a large number of people who plagiarize and use their plagiarized text for verification and evaluation of proposed models; but there are certain aspects against this approach, one of which is distributing or using actual cases of plagiarism involve the permission from the plagiarist and real owner of text and a free text archive with real cases is questionable from an moral and lawful point of view. So this is more sensible for us to generate plagiarism cases by decided alteration, which is also called "simulated plagiarism". We will use this strategy to make a testing corpus for our model.

The dataset that we will use will be of two types. For trial purpose and testing our rules and parameter values, we used about 300 different documents over different topics like scientific, English literature, Political columns downloaded from the web [32][33]. We prepared a total of 52 suspicious documents by simulating different plagiarism cases of variable complexity. For this purpose we used passage summarization and compaction tools like Ginger, words to synonym replacer tools over the web [34]. Along with this, different types of challenging plagiarized passages over the web were also used. The types of cases created to test our model, are mentioned in table 4a. Moreover a combination of following methods was used while creating the plagiarized passages.

For final evaluation of our model and to test the performance of stop-word n-gram based model [18] and a simple VSM based n-gram model, we used the data-set of PAN-13 (Plagiarism, Authorship and Social Software Misuse) championship text alignment corpus-2 [35]. This dataset consists of about 3168 source documents and four types of plagiarized documents which are listed below i.e. No obfuscation, Random obfuscation, Translation obfuscation and Summary obfuscation. We however used first three types to test our model. Following are the details.

a. **No Obfuscation:** These plagiarized documents contain copied passages from different source files and no obfuscations were added. We used 166 suspicious files from this type for test purpose.

b. **Random Obfuscation:** Random text operations were applied to alter the source passages, such as shuffling, adding deleting and replacing of words and short phrases with synonyms. We used 170 suspicious files from this type of obfuscation.

c. **Translation Obfuscation:** Cyclic translation technique is used in order to simulate this type of obfuscation. In this type the passages go through the translation process of different languages like French, German, Italian, Arabic, Indo-European and Swedish, while the end result is always an English phrase. We used suspicious 161 files from this type for testing.

We also downloaded and used 50 documents from the PAN-13 trial to evaluate and further refine the parameter values that we adjusted over the self-built dataset.

Table 4a: Methods Used to for Simulated Plagiarism Cases

Type	**Description**
Cut	Different passages were copied from source and used
Synonyms	Words in copied passages were replaced with synonyms
Stop-words removal	Stop-words were removed or altered from the copied passages
Online text Paraphrasing	Online tools were used to paraphrased copied passages
Passage Slicing	Copied passages were sliced into smaller sentences and were paraphrased then pasted into suspicious documents
Sentence Slicing	Different sentences were paraphrased and combined to make a single passage
Online Examples[36][37]	Online plagiarism examples were used

4.2 Performance Evaluation

Before explaining the test measures in following subsections first look at some notations; let's represent a copied text from the set *S* of all plagiarized texts. Detection is represented by *r* from the universal detection set *R*. Let *SR* is a subset of *S* for which detected cases exists in *R*. Let |*s*|, |*r*| indicate the lengths of characters in *s*, *r* and let |*S*|, |*R*| and |*SR*| be the sizes of the relevant sets [38].

4.2.1 Micro Averaged Precision

Precision measures the proportion between the correctly detected plagiarized passages and the total amount of detected plagiarized passages including the ones that were detected as plagiarism but in fact were not plagiarism.

$$\text{Precision} = \frac{\text{true Positives}}{\text{true positives} + \text{false positives}} = \frac{|R \cap S|}{|R|} = \sum_{i=1}^{|S|} \frac{\#\, of\ detected\ \ chars\ in\ ri}{|ri|} \quad \textbf{(15)}$$

4.2.2 Micro Averaged Recall

Recall measures the proportion between the correctly detected plagiarized passages and the total amount of plagiarized passages including the ones that were not detected.

$$\text{Recall} = \frac{\text{true Positives}}{\text{true positives} + \text{false negatives}} = \frac{|R \cap S|}{|S|} = \sum_{i=1}^{|R|} \frac{\#\, of\ detected\ \ chars\ in\ si}{|si|} \quad \textbf{(16)}$$

In Micro-average method, you sum up the individual true positives, false positives, and false negatives of the system for different sets and the apply them to get the statistics. Suppose there are n number of true positive TP, false positive FP and false negative FN cases then equation 15 and 16 can be expressed as

$$\text{Micr-Average Precision} = \frac{TP1+TP2+TP3+\cdots\ldots+TPn}{(TP1+TP2+\cdots TPn)+(FP1+FP2+\cdots FPn)} \qquad \textbf{(17)}$$

$$\text{Micr-Average Recall} = \frac{TP1+TP2+TP3+\cdots\ldots+TPn}{(TP1+TP2+\cdots TPn)+(FN1+FN2+\cdots FNn)} \qquad \textbf{(18)}$$

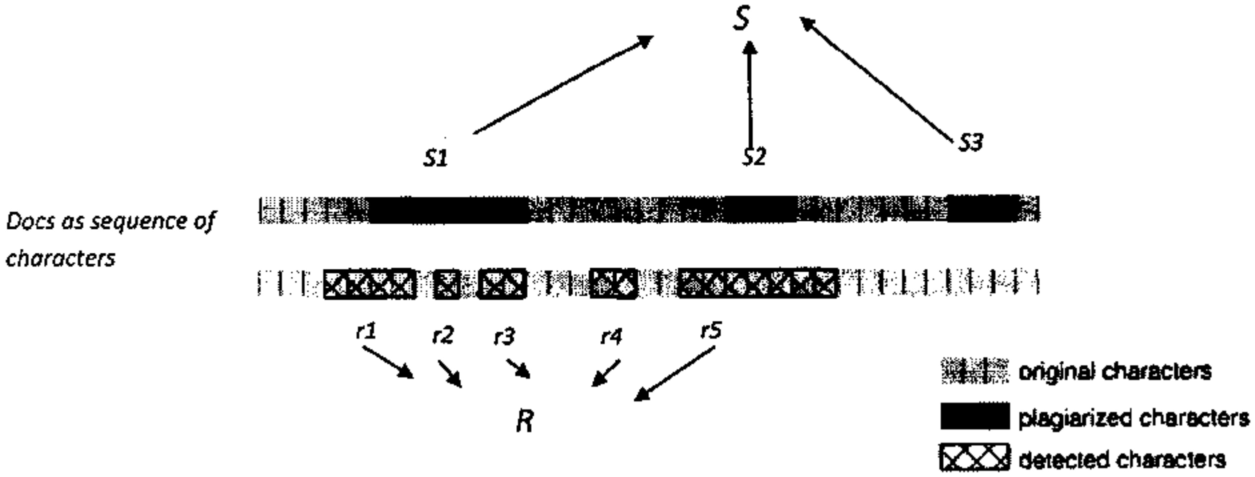


Figure 4a: A document D as a sequence of characters with plagiarized sections S and detected cases R [38]

For the situation shown in Figure 5 the micro-averaged precision is 8/16, likewise, the micro-averaged recall is 8/13.

4.2.3 Macro Averaged Precision and Recall

Macro-averaged Precision and Recall are simply the average of all precisions P and recalls R, and is useful when you want to know how the system performs overall across the sets of data. You should not come up with any specific decision with this average.

$$\text{Macro-Avg Precision} = \frac{P1+P2+P3+\cdots...+Pn}{n} \quad (19)$$

$$\text{Macro-AvgRecall} = \frac{R1+R2+R3+\cdots...+Rn}{n} \quad (20)$$

4.2.4 F-measure

The F-measure is a composite measurement that tries to capture both Precision and Recall, and is defined as the harmonic mean between them.

$$F\text{-measure} = \frac{2.\text{Precision} \ .\text{Recall}}{\text{Precision} \ +\text{Recall}} \quad (21)$$

4.2.5 Granularity

The granularity measures the number of times a part of the text is detected as plagiarism and introduces a way to penalize overlapping plagiarism detections. If n is the number of true positive detections then.

$$\text{Granularity} = \sum_{1}^{n} \frac{\#\text{ Of times a true positive case is reported}}{\text{Precision+Recall}} \qquad (22)$$

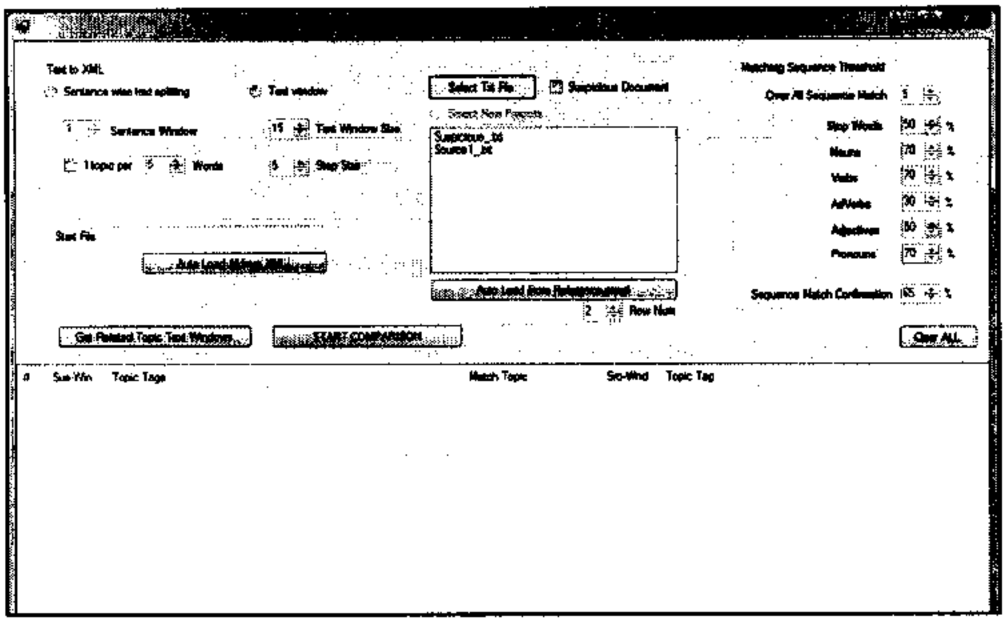
4.2.6 Overall

The overall measurement is, as the F-measure, a way to combine the other measurements; it combines precision, recall, and granularity.

$$\text{Overall} = \frac{\text{F−Measure}}{\text{Log } 2(1+\text{Granualarity })} \qquad (23)$$

4.3 Tests, Results and Discussion

This section provides the detail comparison of different parameter settings that we have tested in order to adjust our method settings for final comparison with baseline methods. Once the parameter values are adjusted, we will compare the results of our method with the base line methods. We implemented the whole network using a C# based GUI which is shown in following figure. This application generates XML files containing alleged plagiarized passages in order to be compared with the XML files that contain actual plagiarized passages from suspicious files.



Text to XML

○ Sentence wise text splitting ● Text window

1 Sentence Window 15 Text Window Size

☐ 1 topic per 5 Words 5 Step Size

Start File

☑ Suspicious Document

○ Select New Reports

Suspicious_txt
Source1_txt

2 Row Num

Matching Sequence Threshold

Over All Sequence Match 5

Stop Words 50 %

Nouns 70 %

Verbs 70 %

AdVerbs 30 %

Adjectives 50 %

Pronouns 70 %

Sequence Match Confirmation 65 %

Figure 4b: Showing the GUI based Application for Plagiarism Detection

In order to compare the xml files containing original plagiarized passages with those originated by our application, we implemented another comparison application that compare two files and calculates Precision, Recall and Granularity for each individual file and macro-averaged Precision, Recall, F-measure and Overall score for full provided dataset. The output of this application is an Excel file containing detailed results.

4.3.1 Parameter Adjustment Tests

As discussed in section 3.3.1 and 3.3.2 different parameters in our model performs a vital role in detection of plagiarized passages. Hence, comprehensive tests were conducted over the self built dataset in order to find out the best values of these parameters for which the detection results (precision, recall, F-measure) are best. These parameters are shown in following table.

Table 4b: Parameters with range of values and the relevant description

Name	**Description**	**Range**
α	This is the maximum number of words chosen for each topic assignment. **Where α = 1 means one topic for each sentence**	1-10
β_L	Max Sequence matched for source and suspicious sentences.	4-15
β_S	Percentage of stop-words sequence match.	20%-100%
β_N	Percentage of nouns sequence match	20%-100%
β_V	Percentage of Verbs sequence match	20%-100%
β_A	Percentage of adverbs sequence match.	20%-100%
β_P	Percentage of pronouns sequence match.	20%-100%
β_D	Percentage of adjectives sequence match.	20%-100%
lp	Maximum percentage of matched indices for a whole sentence to be marked as plagiarized.	30%-100%

We tested the software with different parameter settings and those with best results are shown in following graphical figures. The tests were conducted over the five different values of α i.e. 1 (1 topic per sentence), 5, 6, 7 and 8.

Table 4c: Parameter Range

Parameter	Value
β_L	5
β_S	40%
β_N	60%
β_V	60%
β_A	50%
β_D	40%
β_P	40%
lp	55%

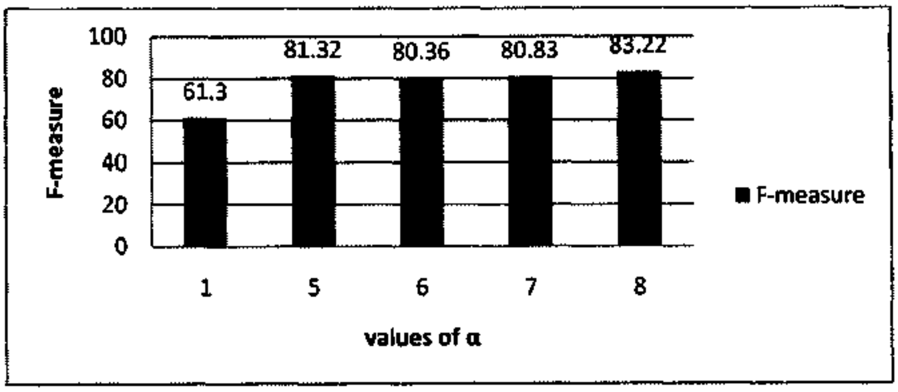


Figure 4c: F-Measure results for all five values of α

Table 4d: Parameter Range

Parameter	Value
β_L	7
β_S	30%
β_N	50%
β_V	40%
β_A	30%
β_D	50%
β_P	70%
lp	45%

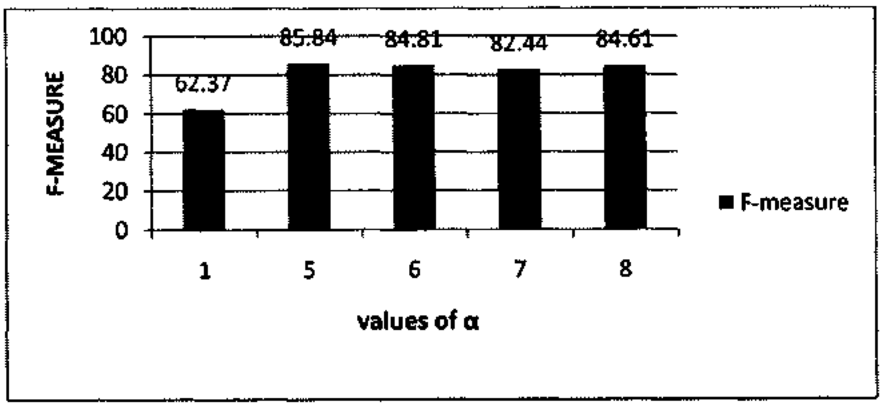


Figure 4d: F-Measure results for all five values of α

Table 4c: Parameter Range

Parameter	Value
β_L	8
β_S	30%
β_N	50%
β_V	50%
β_A	30%
β_D	50%
β_P	70%
lp	50%

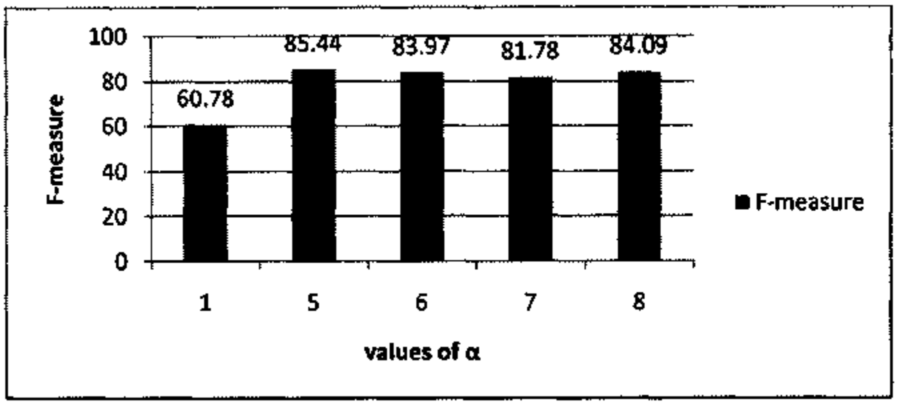


Figure 4e: F-Measure results for all five values of α

From the above results in figure 4d, we observed that the second parameter configuration with α = 5 gave the highest overall results of 85.84. The reason for this is obvious because 5 is the least number of words for which a topic will be produced; the more number of topics in each sentence makes it difficult even for heavily paraphrased sentences to escape from matching query which depends on topic equivalence of sentences. So, starting from the parameter values shown in table 4d, we tested over 30 files from each category of trial dataset of PAN-13 competition, the categories include No obfuscation, Random obfuscation and Translation obfuscation. Following are the results for different parameter settings in tabular and graphical results.

Table 4f: Parameter Range

Parameter	Value
β_L	7
β_S	30%
β_N	50%
β_V	40%
β_A	30%
β_D	50%
β_P	70%
lp	45%

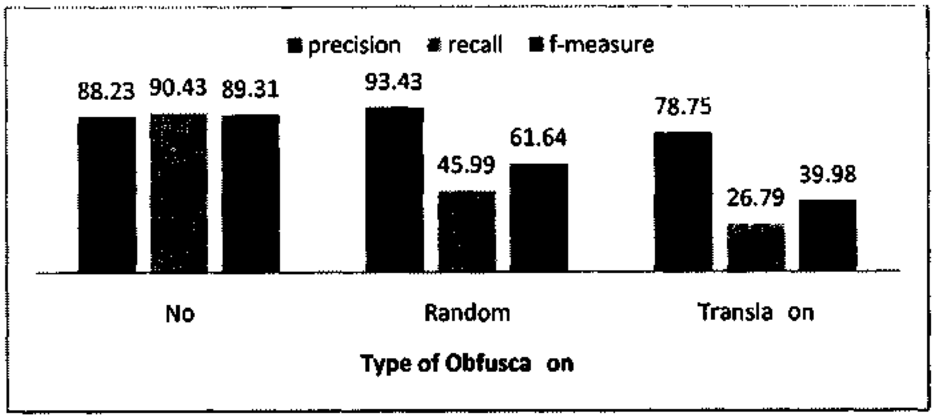


Figure 4f: Results for all three trial datasets for α=5

The figure 4f clearly shows that the parameter values with α value 5 generated good results for both precision and recall in case of No obfuscation types, but for other two types, the recall is much low which means we missed many suspicious cases as the numbers of topics were not enough to match the suspicious and source sentences and hence a high percentage of false negative results caused low recall rate, which means that different types of datasets need different parameter adjustments. Hence, we further tested the other parameter values by further lowering the value of α to 4 over the datasets of Random and Translated categories. We did not further tested the "No obfuscation" type dataset because lowering the value of α can also result in a high number of false positive results, which could affect the precision for this category.

Following table shows the some parameter values that were set in order to carry out further trials over the datasets and showed notable results.

Table 4g: Parameter values tested for Random Obfuscation trial dataset with $\alpha = 4$

	β_L	β_S	β_N	β_V	β_A	β_D	β_P	*lp*
Serial #								
1	6	30	50	40	30	50	70	35
2	6	30	50	40	30	50	70	30
3	5	30	40	30	30	50	70	35
4	5	30	50	40	40	40	50	35
5	5	30	40	40	40	40	40	40

Table 4h: Parameter values tested for Translation Obfuscation trial dataset with $\alpha = 4$

	β_L	β_S	β_N	β_V	β_A	β_D	β_P	*lp*
Serial #								
1	4	20	30	30	20	20	20	25
2	4	30	50	40	40	40	40	35
3	5	30	40	30	30	30	30	35
4	5	30	30	30	30	30	30	30
5	6	30	50	40	30	50	70	45

Following graphs shows the results for parameter values shown in tables 4g and 4h.

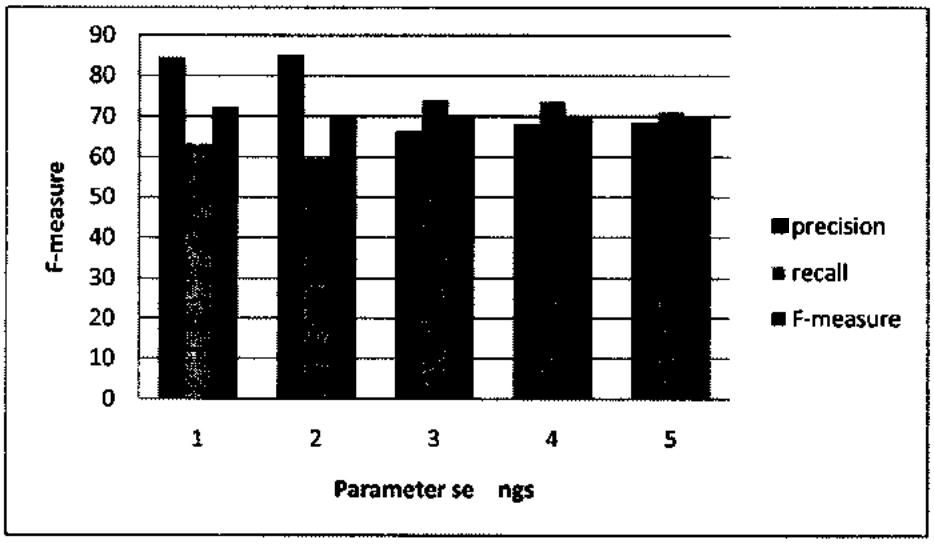


Figure 4g: Results for all sets of Parameter values shown in table 4g

We can clearly see in the figure 4g that the parameter settings of type 1 and 2 have highest precision while types 3, 4 and 5 have high recall, which shows that how the overall matching length β_L parameter is directly proportional to precision and inversely proportional to recall, however the parameter settings of set 1 have shown best F-measure results.

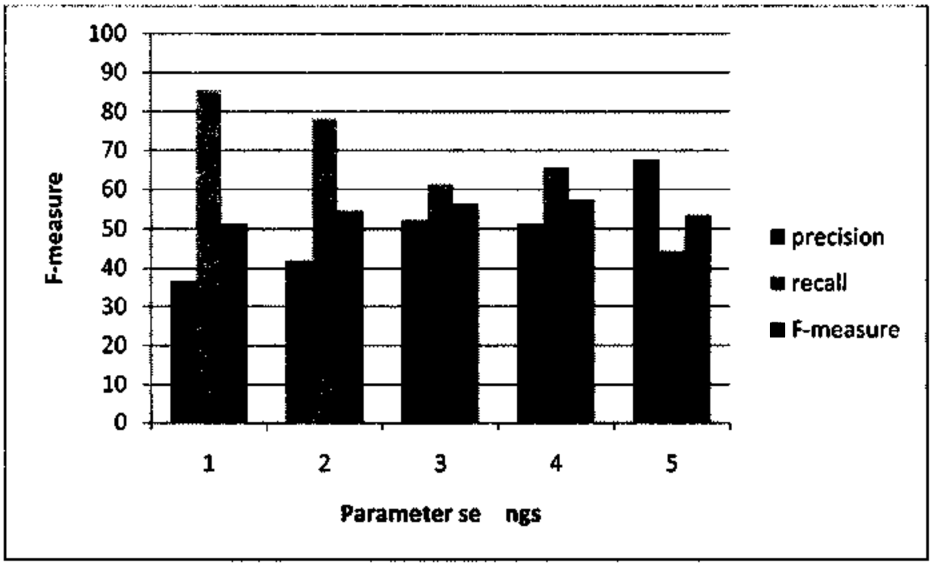


Figure 4h. Results for all sets of Parameter values shown in table 4h

Here again in figure 4h the parameter settings of type 1 and 2 have highest recall having lower value of β_L i.e. 4, while types 3, 4 and 5 have high precision with greater values of β_L, however the parameter settings of set 4 have shown best F-measure results for this category.

4.3.2 5-Fold Cross Validation

After the parameters adjustment testing over self-built and trial datasets, we got best values for our parameters and hence we conducted the 5-fold cross validation tests over the PAN-13 dataset discussed in section 4.1. These tests were performed with the parameter configurations shown in tables 4g and 4h for which the F-measure was best. Following are the detailed results for all the three categories of obfuscation.

Table 4i: 5-fold cross validation results for No-Obfuscation Dataset

K Value	Precision	Recall	F-measure
1	90.32	92.81	91.55
2	91.02	91.49	91.26
3	90.57	90.08	90.32

4	88.9	92.51	90.67
5	92.22	91.5	91.86

Table 4i: 5-fold cross validation results for Random-Obfuscation Dataset

K Value	Precision	Recall	F-measure
1	79.85	64.33	71.25
2	82.05	60.44	69.61
3	86.59	64.75	74.1
4	79.59	67.04	72.77
5	80.51	64.06	71.35

Table 4j: 5-fold cross validation results for Translation-Obfuscation Dataset

K Value	Precision	Recall	F-measure
1	48.19	69.26	56.83
2	48.18	64.48	55.15
3	51.04	60.74	55.47
4	51.91	67.01	58.5
5	50.51	66.78	57.52

4.3.3 Baseline Methods

We implemented two baseline methods in order to compare our results with each of them over the same dataset. These two methods use stop-words n-gram and word n-grams based string similarity matching. We implemented these methods in the same way as we implemented our method i.e. the detected plagiarized passages were written to an XML file for further evaluation.

4.3.3.1 Stop-Word N Gram based method

We skipped the initial steps of document retrieval as this was not our aim in this project. We implemented the stop-word n-gram based method [18] (discussed earlier) into two steps i.e. first finding the suspicious passages from pre-nominated suspicious files and then applying the final criteria (equation 24) for identifying a passage as suspicious or not. We used the same threshold parameter values for this parameter that were mentioned by the author in paper that gave best results, however θ_l was kept low in order to detect the shorter sentences, which was an implementation requirement as well.

$$Sim(t_x, t_s) = \frac{|P_c(n_c,t_x) \cap P_c(n_c,t_s)|}{max\,(|P_c(n_c,t_x)|,|P_c(n_c,t_s)|)} \quad (24)$$

Table 4k: the Parameter Values used for Comparison

Parameter	Value	Description
θ_c	0.5	Lower threshold of the similarity measure to keep a detection
θ_l	20	Lower limit (in characters) of the detected passage length
n_c	3	Character n-gram length to measure similarity between passages
n	5	The stop words 5-grams profiles will be generated

4.3.3.2 Word N-Gram based method

The detection methods that we discussed in section 2.1.3.3 are based on word n-gram based matching of n-gram strings. Hence we implemented a general n-gram based model that suits the purpose of external plagiarism detection. Following steps are applied for this purpose.

1. Removal of all stop-words (depicted in table 5) from a suspicious document *D* and stemming all the words to basic form.
2. Constructing the grams vocabulary *V* of all unique stemmed words in *D* of size n.
3. Creating n sized binary inverted index array for each sentence s in *D*, where each non-zero entry in array depicts the presence of same unique word present in both s and *D*.

4. Suppose there are *k* numbers of sentences present in *D*, hence a binary matrix *M* of *k* rows and *n* columns is prepared.

5. Each sentence in the source documents *S* is passed through the steps 1 and 3 and a binary array *S* of length n is created.

6. The array *S* is queried with each row of matrix *M* in order to find the similarity measure. The type of similarity measure we used is Jaccard's similarity. The Jaccard's similarity is shown in following example

   Suppose we have two binary arrays A and B of size 13

   A = [1 1 0 0 1 0 1 0 1 0 0 0 1]

   B = [1 0 0 1 1 1 0 0 1 0 0 0 1]

   Total number of 00 in two arrays = 5

   Total number of 11 in two arrays = 4

   Other indices = 13 – 5 (Total number of 00 in two arrays)

   $$\text{Jaccard's Similarity} = \frac{\text{Total number of 11 in two arrays}}{\text{Other indices}} = \frac{4}{8} \quad \textbf{(25)}$$

   We have set the lower threshold value for **Jaccard's similarity= 0.5** and **n =1** for our comparison purpose.

7. In case the Jaccard's similarity is satisfactory according to the threshold value, we further investigate the two source and suspicious sentences with same criteria as was described in equation 24 with n = 3.

4.3.4 Qualitative Comparison

In this section we will present a few examples to from our dataset that will show how our method has achieved good results.

4.3.4.1 Example 1

Table 4l: Example of Text Paraphrasing

Source Sentence	Suspicious Sentence
First then, it is	initial then, it is
thought that every name has, or ought to	notion that each name has, or must to have,
have, one only precise and	one only exact and
settled signification, which inclines men to	established meaning, which persuade guy t
think there are certain abstract	believe there are sure summary

a) Stop-Words N-Gram Method

The 5-gram stop-word profile for source passage

{ it is that has or | is that has or to | that has or to have |has or to have and |

or to have and which |to have and which to | have and which to there |

and which to there are }

The 5-gram stop-word profile for suspicious passage

{it is that has or | is that has or to | that has or to have | has or to have and |

or to have and which | to have and which to | have and which to there |

and which to there are}

These profiles match each other, now we apply the final condition (eq. 4) for deciding whether or not two texts are related.

Number of distinct words 3-grams in suspicious sentence = 144

Number of distinct words 3-grams in suspicious sentence = 144

Number of matching 3-grams in both sentences = 68

$$\text{PlagDet}=\frac{68}{\text{MAX}\,(144{,}144)} = 0.47 < \theta_c$$

Hence this method **failed** to detect plagiarism.

b) Words N-Gram Method

Unique n-gram vocabulary of suspicious sentence with stop-words removed

A = {initial notion each name one only exact established meaning persuade guy believe sure summary }

As this is the only sentence, thus the binary array for this will be

a = {1 1 1 1 1 1 1 1 1 1 1 1 1 1}

Unique n-gram vocabulary of source sentence with stop-words removed

B = { first thought every name one only precise settle signification incline men think certain abstract}

The binary array that we get as result of applying binary query is over matching words in A and B

b = {0 0 0 1 1 1 0 0 0 0 0 0 0 0}

Total number of 00 in two arrays = 0

Total number of 11 in two arrays = 3

Other indices (10 and 01) = 11

$$\text{Jaccard's Similarity} = \frac{\text{Total number of 11 in two arrays}}{\text{Other indices}} = \frac{3}{11} = 0.27 < 0.5$$

Hence this method also **failed** to detect plagiarism.

c) POS-LDA Method

Topics and synonyms produced for suspicious sentence with POS tags

{initial, one, lone, sole, first, single........ }

A = {21 8 0 33 46 35 6 4 35 46 0 5 48 7 41 0 2 8 21 5 44 35 0 27 42 35 7 41 3 45 21 35 }

Topics produced for source sentence with POS tags

{certain, abstract, precise, men, first}

B = {8 8 0 33 46 44 6 4 35 46 0 5 48 7 41 0 2 8 21 5 42 21 35 0 27 46 36 7 41 3 45 21 35}

The best adjusted parameters values from section 4.3.1 b are

β_L= 7, β_S= 30%, β_N= 50%, β_V= 40%, *lp*= 45%, β_A= 30%, β_D= 50%, β_P= 70%

After applying the matching criteria we get the following results

Max Sequence Match Length =	14	$>\beta_L$
Stop-Words Sequence Match =	100%	$> \beta_S$
Nouns Sequence Match =	40 %	$<\beta_N$
Verbs Sequence Match =	37%	$<\beta_V$
Adverbs Sequence Match =	100%	$>\beta_A$
Adjectives Sequence Match =	43%	$<\beta_D$
Pronouns Sequence Match =	43%	$<\beta_P$
Length Ratio L_r=	33 / 32< 2	

By inserting these values in equations 9 and 10

$tf = ((14 \geq 2\beta_L) \wedge ((100 \geq \beta_A) \vee (43 \geq \beta_P) \vee (43 \geq \beta_D)))$

tf = *true*

$tl = ((14 \geq \beta_L) \wedge (L_r \leq 2)) \wedge (((100 \geq \beta_S) \wedge (40 \geq \beta_N)) \vee ((100 \geq \beta_S) \wedge (37 \geq \beta_V)) \vee (40 \geq \beta_N) \wedge (37 \geq \beta_V)))$

We get a detection *TF* = true

Hence this method **managed** to detect plagiarism.

4.3.4.2 Example 2

Table 4m: A More Complex Example of Text Paraphrasing

Source Sentence	Suspicious Sentence
A drink driver who ran into the Queen Mother's official Daimler was fined £sever and banned from driving for 2 years.	A DRUNK driver who crashed into the bac of the Queen Mum's limo was forbidden fc two years yesterday.

a) Stop-Words N-Gram Method

The 5-gram stop-word profile for both source passages

{ a who the was and | who the was and from | the was and from for}

The 5-gram stop-word profile for both suspicious passages

{a who the of the | who the of the was | the of the was for }

We get no stop-word 5-gram match for suspicious passage

Hence this method **failed** to detect plagiarism.

b) Words N-Gram Method

Unique n-gram vocabulary of suspicious sentence with stop-words removed

A = {drunk driver crash back queen Mumlimo forbidden two year yesterday}

As this is the only sentence, thus the binary array for this will be

a = {1 1 1 1 1 1 1 1 1 1}

Unique n-gram vocabulary of source sentence with stop-words removed

B = {drink driver ran queen Mother official Daimler fine seven banned drive year}

The binary array that we get as result of applying binary query is over matching words in A and B

b = {0 1 0 0 1 0 0 0 0 1 0 }

Total number of 00 in two arrays = 0

Total number of 11 in two arrays = 3

Other indices (10 and 01) = 8

$$\text{Jaccard's Similarity} = \frac{\text{Total number of 11 in two arrays}}{\text{Other indices}} = \frac{3}{8} = 0.37 < 0.5$$

Hence this method also **failed** to detect plagiarism.

c) POS-LDA Method

Topics and its synonyms produced for suspicious sentence with POS tags

{period, class, twelvemonth, year........ }

A = {4 35 35 25 42 6 4 35 35 35 35 42 44 8 5 44 6 43 6 2 36}

Topics produced for source sentence with POS tags

{daimler, drink, seven, year, banned }

B = { 4 35 35 25 42 6 4 35 6 4 35 35 35 42 44 6 2 36 35}

The best adjusted parameters values from section 4.3.1 b are

β_L= 7, β_S= 30%, β_N= 50%, β_V= 40%, *lp*= 45%, β_A= 30%, β_D= 50%, β_P= 70%

After applying the matching criteria we get the following results

Max Sequence Match Length =	8	$>\beta_L$
Stop-Words Sequence Match =	42%	$>\beta_S$
Nouns Sequence Match =	87 %	$>\beta_N$
Verbs Sequence Match =	100%	$>\beta_V$
Adverbs Sequence Match =	0%	$<\beta_A$
Adjectives Sequence Match =	42%	$<\beta_D$
Pronouns Sequence Match =	42%	$<\beta p$
Length Ratio Lr =	21 / 19	< 2

By inserting these values in equations 9 and 10

$tf = ((8 \geq 2\beta_L) \wedge ((0 \geq \beta_A) \vee (42 \geq \beta p) \vee (42 \geq \beta_D)))$

$tf = true$

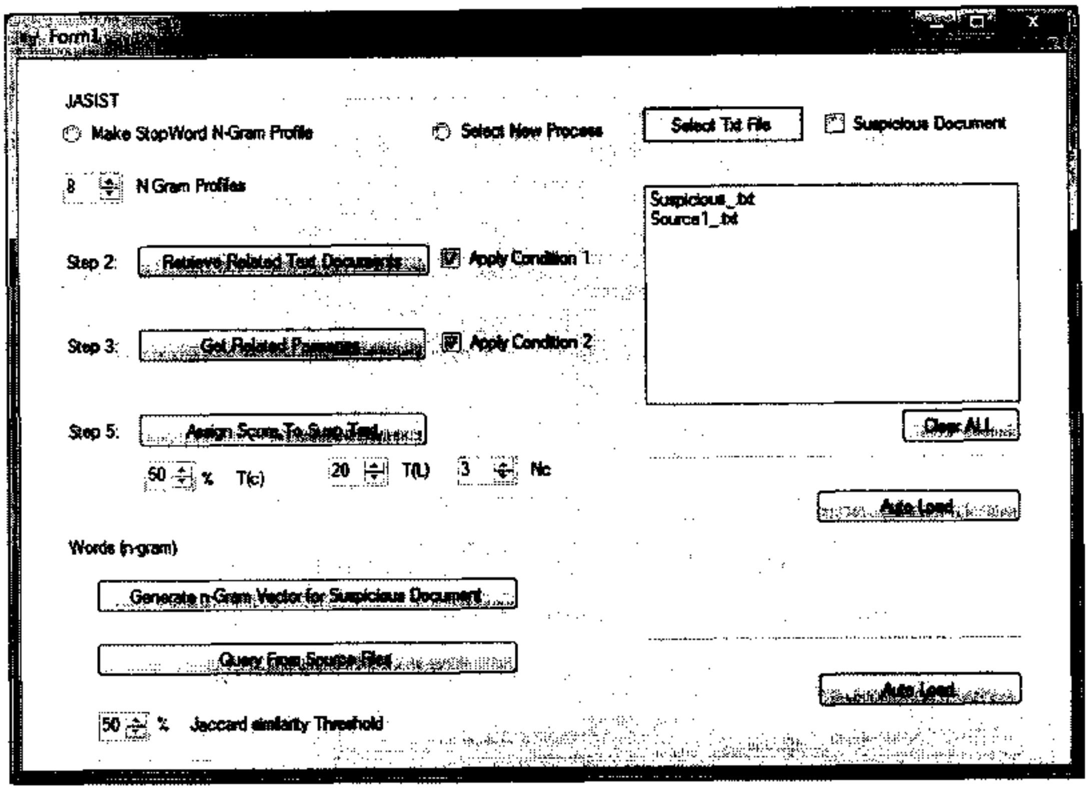
$tl = ((8 \geq \beta_L) \wedge (L_r \leq 2)) \wedge (((42 \geq \beta_S) \wedge (87 \geq \beta_N)) \vee ((42 \geq \beta_S) \wedge (100 \geq \beta_V)) \vee (87 \geq \beta_N) \wedge (100 \geq \beta v)))$

We get a detection *TF = true*

Hence this method **managed** to detect plagiarism.

4.3.5 Quantitative Comparison

In order to compare the efficiency of our model with SWNG and general n-gram based models we implemented both using C# .net based GUI shown below in figure 4i. We executed both detection models over the whole PAN-13 test dataset-2 and compared the detection efficiency with our model.



8 N Gram Profiles

☑ Apply Condition 1

☑ Apply Condition 2

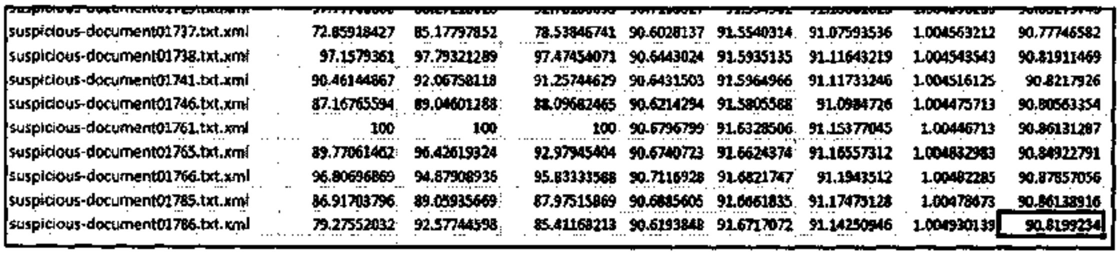
50 % T(c) 20 T(L) 3 Nc

50 % Jaccard similarity Threshold

Suspicious_txt
Source1_txt

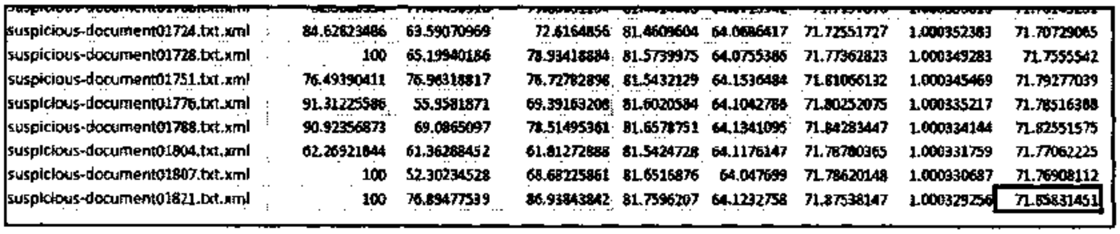
Figure 4i: C# based GUI for testing purpose of N-gram and SWNG Models

Following are the screen-shots of results generated by all the three models compared by us, first three are the results for POS-LDA then next three are the results for n-gram and last three are the results for SWNG based plagiarism detection models.



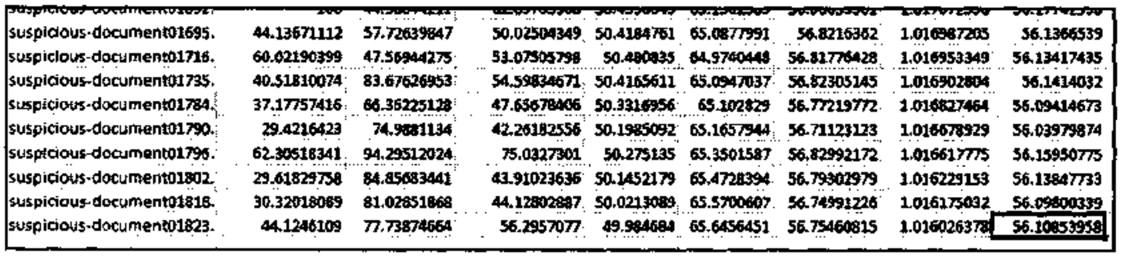
suspicious-document01737.txt.xml	72.85918427	85.17797852	78.53846741	90.6028137	91.5540314	91.07593536	1.004563212	90.77746582
suspicious-document01738.txt.xml	97.1579361	97.79321289	97.47454071	90.6443024	91.5935135	91.11643219	1.004543543	90.81911469
suspicious-document01741.txt.xml	90.46144867	92.06758118	91.25744629	90.6431503	91.5964966	91.11733246	1.004516125	90.8217926
suspicious-document01746.txt.xml	87.16765594	89.04601288	88.09682465	90.6214294	91.5805588	91.0984726	1.004475713	90.80563354
suspicious-document01761.txt.xml	100	100	100	90.6796799	91.6328506	91.15377045	1.00446713	90.86131287
suspicious-document01765.txt.xml	89.77061462	96.42619324	92.97945404	90.6740723	91.6624374	91.16557312	1.004832983	90.84922791
suspicious-document01766.txt.xml	96.80696869	94.87908936	95.83333588	90.7116928	91.6821747	91.1943512	1.00482285	90.87857056
suspicious-document01785.txt.xml	86.91703796	89.05935669	87.97515869	90.6885605	91.6661835	91.17475128	1.00478673	90.86138916
suspicious-document01786.txt.xml	79.27552032	92.57744598	85.41168213	90.6193848	91.6717072	91.14250946	1.004930139	90.8199234

Figure 4j: POS-LDA Results for No-Obfuscation dataset



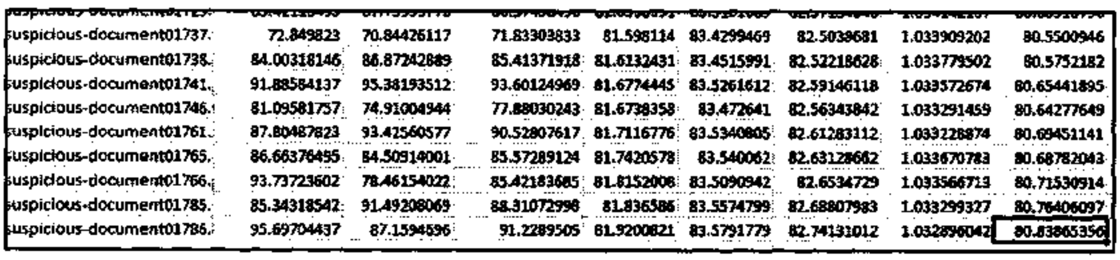
suspicious-document01724.txt.xml	84.62823486	63.59070969	72.6164856	81.4609604	64.0686417	71.72551727	1.000352383	71.70729065
suspicious-document01728.txt.xml	100	65.19940186	78.93418884	81.5739975	64.0755386	71.77362823	1.000349283	71.7555542
suspicious-document01751.txt.xml	76.49390411	76.96318817	76.72782898	81.5432129	64.1536484	71.81066132	1.000345469	71.79277039
suspicious-document01776.txt.xml	91.31225586	55.9581871	69.39163208	81.6020584	64.1042786	71.80252075	1.000335217	71.78516388
suspicious-document01788.txt.xml	90.92356873	69.0865097	78.51495361	81.6578751	64.1341095	71.84283447	1.000334144	71.82551575
suspicious-document01804.txt.xml	62.26921844	61.36288452	61.81272888	81.5424728	64.1176147	71.78780365	1.000331759	71.77062225
suspicious-document01807.txt.xml	100	52.30234528	68.68225861	81.6516876	64.047699	71.78620148	1.000330687	71.76908112
suspicious-document01821.txt.xml	100	76.89477539	86.93843842	81.7596207	64.1232758	71.87538147	1.000329256	71.85831451

Figure 4k: POS-LDA Results for Random-Obfuscation dataset



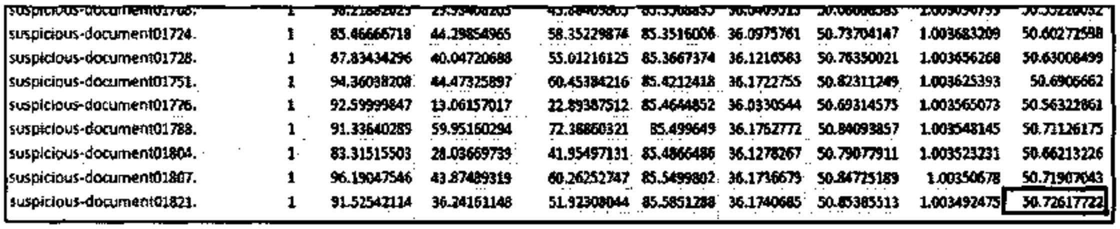
suspicious-document01695.	44.13671112	57.72639847	50.02504349	50.4184761	65.0877991	56.8216362	1.016987205	56.1366539
suspicious-document01716.	60.02190399	47.56944275	53.07505798	50.480835	64.9740448	56.81776428	1.016953349	56.13417435
suspicious-document01735.	40.51810074	83.67626953	54.59834671	50.4165611	65.0947037	56.82305145	1.016902804	56.1414032
suspicious-document01784.	37.17757416	66.36225128	47.65678406	50.3316956	65.102829	56.77219772	1.016827464	56.09414673
suspicious-document01790.	29.4216423	74.9881134	42.26182556	50.1985092	65.1657944	56.71123123	1.016678929	56.03979874
suspicious-document01796.	62.30518341	94.29512024	75.0327301	50.275135	65.3501587	56.82992172	1.016617775	56.15950775
suspicious-document01802.	29.61829758	84.85683441	43.91023636	50.1452179	65.4728394	56.79302979	1.016229153	56.13847733
suspicious-document01818.	30.32018089	81.02851868	44.12802887	50.0213089	65.5700607	56.74991226	1.016175032	56.09800339
suspicious-document01823.	44.1246109	77.73874664	56.2957077	49.984684	65.6456451	56.75460815	1.016026378	56.10853958

Figure 4l: POS-LDA Results for Translation-Obfuscation dataset



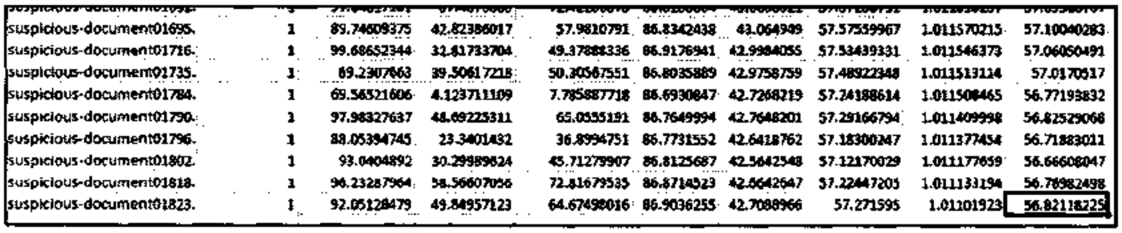
suspicious-document01737.	72.849823	70.84426117	71.83303833	81.598114	83.4299469	82.5039681	1.033909202	80.5500946
suspicious-document01738.	84.00318146	86.87242889	85.41371918	81.6132431	83.4515991	82.52218628	1.033779502	80.5752182
suspicious-document01741.	91.88584137	95.38193512	93.60124969	81.6774445	83.5261612	82.59146118	1.033572674	80.65441895
suspicious-document01746.	81.09581757	74.91004944	77.88030243	81.6738358	83.472641	82.56343842	1.033291459	80.64277649
suspicious-document01761.	87.80487823	93.42560577	90.52807617	81.7116776	83.5340805	82.61283112	1.033228874	80.69451141
suspicious-document01765.	86.66376495	84.50914001	85.57289124	81.7420578	83.540062	82.63128662	1.033670783	80.68782043
suspicious-document01766.	93.73723602	78.46154022	85.42183685	81.8152008	83.5090942	82.6534729	1.033566713	80.71530914
suspicious-document01785.	85.34318542	91.49208069	88.31072998	81.836586	83.5574799	82.68807983	1.033299327	80.76406097
suspicious-document01786.	95.69704437	87.1594696	91.2289505	81.9200821	83.5791779	82.74131012	1.032896042	80.83865356

Figure 4m: N-gram Results for No-Obfuscation dataset



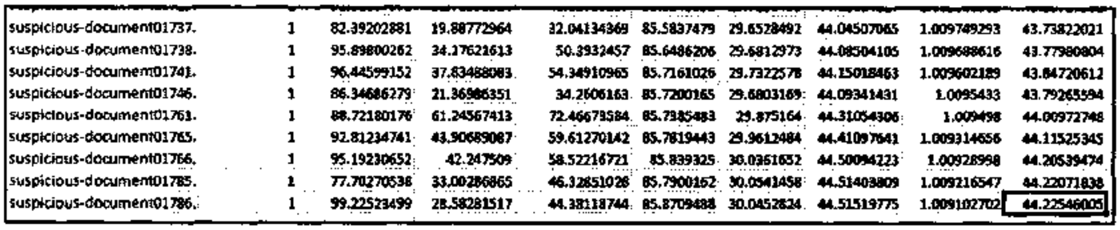
suspicious-document01724.	1	85.46666718	44.29854965	58.35229874	85.3516006	36.0975761	50.73704147	1.003683209	50.60272598
suspicious-document01728.	1	87.83434296	40.04720688	55.01216125	85.3667374	36.1216583	50.76350021	1.003656268	50.63008499
suspicious-document01751.	1	94.36038208	44.47325897	60.45384216	85.4212418	36.1722755	50.82311249	1.003625393	50.6906662
suspicious-document01776.	1	92.59999847	13.06157017	22.89387512	85.4644852	36.0330544	50.69314575	1.003565073	50.56322861
suspicious-document01788.	1	91.33640289	59.95160294	72.38860321	85.499649	36.1762772	50.84093857	1.003548145	50.71126175
suspicious-document01804.	1	83.31515503	28.03669739	41.95497131	85.4866486	36.1278267	50.79077911	1.003523231	50.66213226
suspicious-document01807.	1	96.19047546	43.87489319	60.26252747	85.5499802	36.1736679	50.84725189	1.00350678	50.71907043
suspicious-document01821.	1	91.52542114	36.24161148	51.92308044	85.5851288	36.1740685	50.85385513	1.003492475	50.72617722

Figure 4n: N-gram Results for Random-Obfuscation dataset



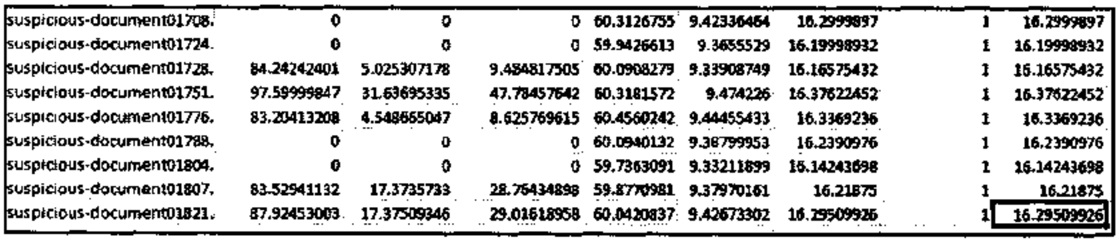
suspicious-document01695.	1	89.74609375	42.82386017	57.9810791	86.8342438	43.064949	57.57559967	1.011570215	57.10040283
suspicious-document01716.	1	99.68652344	32.81733704	49.37888336	86.9176941	42.9984055	57.53439331	1.011546373	57.06050491
suspicious-document01735.	1	89.2307663	39.50617218	50.30567551	86.8035889	42.9758759	57.48922348	1.011513114	57.0170517
suspicious-document01784.	1	69.56521606	4.123711109	7.785887718	86.6930847	42.7268219	57.24188614	1.011508465	56.77193832
suspicious-document01790.	1	97.98327637	48.69225311	65.0555191	86.7649994	42.7648201	57.29166794	1.011409998	56.82529068
suspicious-document01796.	1	88.05394745	23.3401432	36.8994751	86.7731552	42.6418762	57.18300247	1.011377454	56.71883011
suspicious-document01802.	1	93.0404892	30.29989824	45.71279907	86.8125687	42.5642548	57.12170029	1.011177659	56.66608047
suspicious-document01818.	1	96.23287964	58.56607056	72.81679535	86.8714523	42.6642647	57.22447205	1.011133194	56.78982498
suspicious-document01823.	1	92.05128479	49.84957123	64.67498016	86.9036255	42.7088966	57.271595	1.01101923	56.82118225

Figure 4o: N-gram Results for Translation-Obfuscation dataset



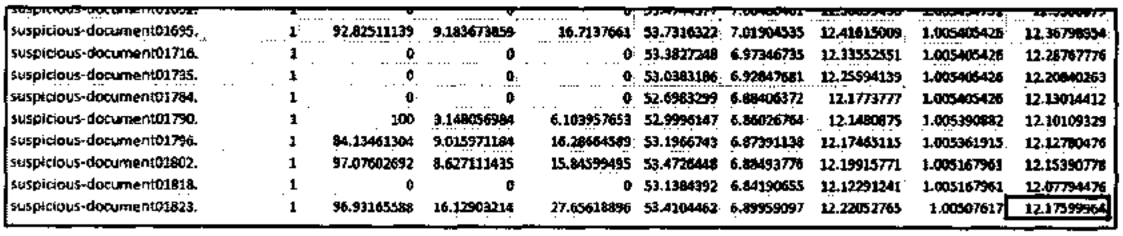
suspicious-document01737.	1	82.39202881	19.88772964	32.04134369	85.5837479	29.6528492	44.04507065	1.009749293	43.73822021
suspicious-document01738.	1	95.89800262	34.17621613	50.3932457	85.6486206	29.6812973	44.08504105	1.009688616	43.77980804
suspicious-document01741.	1	96.44599152	37.83488083	54.34910965	85.7161026	29.7322578	44.15018463	1.009602189	43.84720612
suspicious-document01746.	1	86.34686279	21.36986351	34.2606163	85.7200165	29.6803169	44.09341431	1.0095433	43.79265594
suspicious-document01761.	1	88.72180176	61.24567413	72.46673584	85.7385483	29.875164	44.31054306	1.009498	44.00972748
suspicious-document01765.	1	92.81234741	43.90689087	59.61270142	85.7819443	29.9612484	44.41097641	1.009314656	44.11525345
suspicious-document01766.	1	95.19230652	42.247509	58.52216721	85.839325	30.0361652	44.50094223	1.00928998	44.20539474
suspicious-document01785.	1	77.70270538	33.00286865	46.32851028	85.7900162	30.0541458	44.51403809	1.009216547	44.22071838
suspicious-document01786.	1	99.22523499	28.58281517	44.38118744	85.8709488	30.0452824	44.51519775	1.009102702	44.22546005

Figure 4p: SWNG Results for No-Obfuscation dataset



suspicious-document01708.	0	0	0	60.3126755	9.42336464	16.2999897		1	16.2999897
suspicious-document01724.	0	0	0	59.9426613	9.3655529	16.19998932		1	16.19998932
suspicious-document01728.	84.24242401	5.025307178	9.484817505	60.0908279	9.33908749	16.16575432		1	16.16575432
suspicious-document01751.	97.59999847	31.63695335	47.78457642	60.3181572	9.474226	16.37622452		1	16.37622452
suspicious-document01776.	83.20413208	4.548665047	8.625769615	60.4560242	9.44455433	16.3369236		1	16.3369236
suspicious-document01788.	0	0	0	60.0940132	9.38799953	16.2390976		1	16.2390976
suspicious-document01804.	0	0	0	59.7363091	9.33211899	16.14243698		1	16.14243698
suspicious-document01807.	83.52941132	17.3735733	28.76434898	59.8770981	9.37970161	16.21875		1	16.21875
suspicious-document01821.	87.92453003	17.37509346	29.01618958	60.0420837	9.42673302	16.29509926		1	16.29509926

Figure 4q: SWNG Results for Random-Obfuscation dataset



suspicious-document01695.	1	92.82511139	9.183673859	16.7137661	53.7316322	7.01904535	12.41615009	1.005405426	12.36798954
suspicious-document01716.	1	0	0	0	53.3827248	6.97346735	12.33552551	1.005405426	12.28767776
suspicious-document01735.	1	0	0	0	53.0383186	6.92847681	12.25594139	1.005405426	12.20840263
suspicious-document01784.	1	0	0	0	52.6983299	6.88406372	12.1773777	1.005405426	12.13014412
suspicious-document01790.	1	100	3.148056984	6.103957653	52.9996147	6.86026764	12.1480875	1.005390882	12.10109329
suspicious-document01796.	1	84.13461304	9.015971184	16.28664589	53.1966743	6.87391138	12.17465115	1.005361915	12.12780476
suspicious-document01802.	1	97.07602692	8.627111435	15.84599495	53.4726448	6.88493776	12.19915771	1.005167961	12.15390778
suspicious-document01818.	1	0	0	0	53.1384392	6.84190655	12.12291241	1.005167961	12.07794476
suspicious-document01823.	1	96.93165588	16.12903214	27.65618896	53.4104462	6.89959097	12.22052765	1.00507617	12.17599964

Figure 4r: SWNG Results for Translation-Obfuscation dataset

Following is the detailed graphical result of external plagiarism detection for all the three models over the PAN-13 dataset.

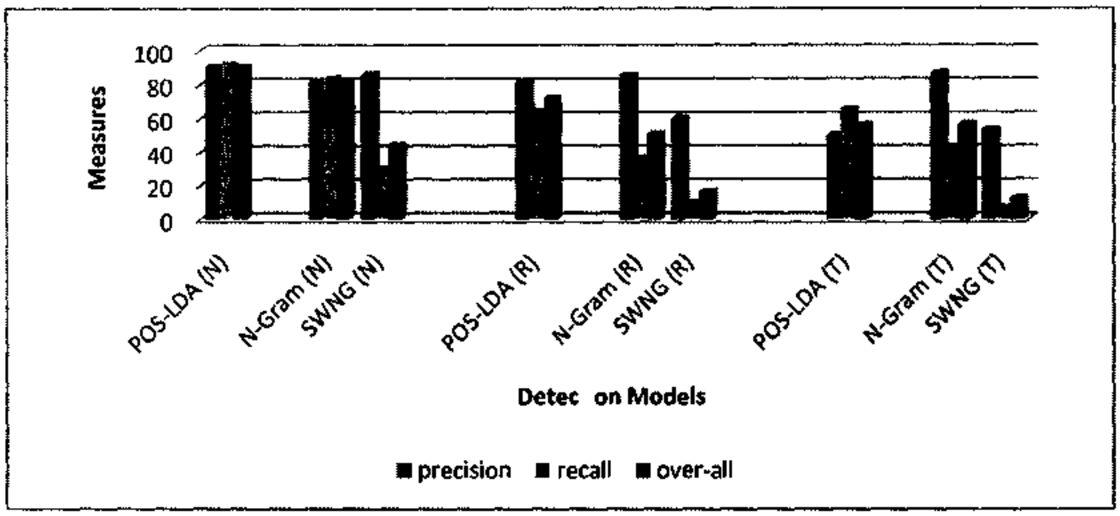


Figure 4s: Showing the Overall results for all three Models where N stands for No-obfuscation, R stands for Random-obfuscation and T stands for Translation-obfuscation Datasets

4.4 Results Analysis

Although all three methods were compared over same dataset but only the POS-LDA method showed most accurate results of the three models. The results of SWNG were the least because of the two reasons, one because in most of the cases the order of stop-words was either changed or stop-words were removed or replicated, hence SWNG method failed to identify plagiarized passages in preprocessing step, the other reason are the cases where there was heavy paraphrasing of passages both SWNG and WNG (word n-gram) methods showed low results in post processing steps because both methods ignore the matching synonyms of words in the step. Also for shorter sentences and passages where the number of stop-word n-grams is lesser than those present in set C, would be rejected at the pre-processing step.

The word n-gram based method on the other hand showed good results for copy-paste type plagiarism detection model which is obvious because this model is made to show performance for this kind of plagiarism, but here also our method outperformed the WNG method because our method is more flexible in post-processing stage and while comparing different suspicious sentences with each other, the model also checks the neighboring sentences through its rule called "suspicious neighbor" that we have discussed above. On the other hand WNG only process each suspicious sentence individually and suspicious plagiarized sentence has no effect over its neighboring sentences. Also our model does not have a single criterion of deciding whether a sentence in suspicious document is plagiarized or not, as in the case of WNG where the only text matching criteria depends on cosine or Jaccard's similarity indexes.

Another thing to note is that the precision for both N-gram based methods was much higher in all the three cases but recall remained lower which means high percentage of false negative results. This clearly proves our point that the n-gram based detection models fail in case of high obfuscation. Here, the point which proves our case that in cases of higher obfuscation where the words are replaced by respective synonyms or are rewritten with same context. As we did not use "Word-Net" while implementing WNG model, which is used now a days in order to cater this biggest drawback of N-gram model. We ourselves however used to produce limited numbers of synonyms for each topic generated by LDA model in order to find similar topics while querying the related source XML files for each suspicious XML file.

In case of obfuscation and translated plagiarism the SWNG model has least recall which clearly shows that this model failed to find many true positive cases. The main reason behind this is the fact that the SWNG model first ignores many suspicious sentences in its preprocessing step where it reduces the passages into stop-words in order of appearance, then in post processing step where this model compares all matching text passages on the basis of n-gram based similarity measure more suspicious plagiarized passages escape from the criterion and hence this results in high precision and a very low recall rate.

In case of our proposed model i.e. "POS-LDA", the recall was relatively low in case of Random-obfuscation dataset, which means that we missed many true positive cases. This is probably because of the topics that were not matched in source and suspicious passages and hence escaped from post processing step.

In case of Translation-obfuscation detection, the precision was much low because of which the WNG model beat our approach. The reason for low precision was the lowering of parameter values which in turn made the rules for plagiarism detection of passages more flexible and hence we got more false positive results. This can be improved by reducing the number of parameters in the model over which the decision depends. Also the values of some parameters like Adverbs, Adjectives and Pronouns.

One observation while running the tests over whole dataset was the time taken by all three models. We observed that the fastest model to produce results was SWNG model, while the POS-LDA model took the most time to produce results which is because of two following reasons.

One reason is due to the implementation of the model, where we used three third party softwares for POS tagging, for topic generation and for synonyms generation. Hence the execution of these three applications for each sentence in suspicious and source documents made the preprocessing step slower. The second reason for slow performance in the post processing step could be the LINQ query to find out related passages in source and suspicious XML files.

In order to find out the overall execution time for plagiarized text detection in all three models we carried out a small test over small dataset of 40 source and suspicious files, ran all three models over it and calculated the time taken by all three models to produce final results.

Table 4n: Execution time of POS-LDA model in seconds for different values of α

Values of α	Time in seconds
4	49.863
5	46.176
6	46.340

Table 4o: Execution time of SWNG and WNG models in seconds

Model	Time in seconds
SWNG	13.180
WNG	20.233

CHAPTER 5

CONCLUSIONS

We have applied our method (POS-LDA) over all types of simulated plagiarism cases of self built and PAN-13 datasets and our model showed comparatively good results from the other two models based on n-gram based approach which means that the Stylometric approaches can also play an efficient role in case of external plagiarism detection even without the assistance of n-gram based approach in order to confirm a suspicious phrase as plagiarized. We also observed that how different classes of parts of speech can play a vital role along with the phrase topics in order to find plagiarism.

The reason that some of the plagiarism cases that escaped from detection of our POS-LDA method was because our method failed to match topics and their synonyms among source and suspicious passages. The reason why these topics did not matched is obvious e.g. we have a topic word "skilled" in source passage which is copied in suspicious passage as "skilled"; Now if the word "skilled" is also chosen topic of suspicious phrase then it may have its synonym list like "capable, able, trained, skillful, talented..."; none of which would match the source topic. Another main reason for a failure in detection would be in case the POS patterns of a suspicious passage are heavily revised and the order of words is changed.

5.1 Future Work

We compared pre-nominated source documents for each suspicious document in local archive and have used topic matching based approach in order to select suspicious passages for further processing and evaluation, which was the task of text alignment; however this approach can be further utilized to choose the related source documents from an archive of documents over the network i.e. for the task of source documents retrieval.

Also the model can be improved by following points that we observed after the results.

1. We need to work more over the importance of POS classes and should first find which POS tags are more dominant and frequent in suspicious files. The parameter values should be set depending on the frequency of POS classes.

2. The rules to detect a plagiarism case should be made more flexible according to type of plagiarism and hence more rules are required.

3. Phrasal tagging of sentences should also be done.

4. WordNet can be used in order to tag the sentences and also for synonyms generation.

Another observation is the time taken by our method in order to produce suspicious topics and related synonyms, which should be reduced. This can be done if we only include suspicious files in first sentence and query the topics from source files rather than to preprocess the source files also.

References:

[1] [Online] http://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=65284248, [Accessed: 2014].

[2] H. Maurer, F. Kappe and B. Zaka, "Plagiarism - A Survey," in Journal of Universal Computer Science, vol. 12, no. 8, 2006, pp. 1050-1084.

[3] [Online] http://turnitin.com, [Accessed: 2015].

[4] [Online] https://www.doccop.com, [Accessed: 2015].

[5] [Online] https://www.coremodetector.com, [Accessed: 2015].

[6] A. Broder, S. Glassman and M. Manasse, "Syntatic Clustering of the Web," in Sixth International Web Conference, Santa Clara, California USA. n.a

[Online] http://decweb.ethz.ch/WWW6/Technical/Paper205/paper205.html

[7] H.G. Molina and N. Shivakumar, "Building a Scalable and Accurate Copy Detection Mechanism," in Proceedings of 1st ACM International Conference on Digital Libraries, Bethesda Maryland, 1996.

[8] K. Monostori, et al., "Comparison of Overlap Detection Techniques", ICCS, Lecture Notes in Computer Science Volume 2329, 2002, pp. 51-60

[9] B. Stein and S.M. Eissen, "Near Similarity Search and Plagiarism Analysis", in 29th Annual Conference of the German Classication Society (GfKl) Magdeburg, ISBN 1431-8814, 2006, pp. 430-437.

[10] A. Jadalla and A. Elnagar, "A Fingerprinting-Based Plagiarism Detection System for Arabic Text-Based Documents", in International Conference on Computing Technology and Information Management, ISBN 978-1-4673-0893-9, 2012, pp 477-482.

[11] N. Carnahan, et al., "Plagiarism Detection", in Carleton College Computer Science Comps Gala, 2014.

[12] Dreher and Heinz, "Automatic Conceptual Analysis for Plagiarism Detection", Information and Beyond: The Journal of Issues in Informing Science and Information Technology, 2007, pp. 601–614.

[13] [Online] http://en.wikipedia.org/wiki/N-gram, [Accessed: 2015].

[14] O. Uzuner, B. Katz and T. Nahnsen, "Using Syntactic Information to Identify Plagiarism", in Proceedings of the 2nd Workshop on Building Educational Applications Using NLP, 2005, pp. 37–44.

[15] W.Y. Lin, et al.,"Online Plagiarism Detection Through Exploiting Lexical, Syntactic, and Semantic Information", in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Republic of Korea, 2012, pp. 145–150.

[16] S. Avram, D. Caragea and T. Borangi, "NLP Applications in External Plagiarism Detection", U.P.B. Sci. Bull., Series C, Vol. 76, Iss. 3, 2014.

[17] V.S. Ram, E. Stamatatos and S.L Devi, "Identification of Plagiarism using Syntactic and Semantic Filters", in Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science Volume 8404, 2014, pp. 495-506.

[18] E. Stamatatos, "Plagiarism Detection Using Stopword n-grams", Journal of the American Society for Information Science and Technology, 2011, pp. 2512-2527.

[19] M. Zechner, M. Muhr and R. Kern, "External and Intrinsic Plagiarism Detection Using Vector Space Models", Stein, Rosso, Stamatatos, Koppel, Agirre (Eds.): PAN'09, 2009, pp. 47-55.

[20] S.L Devi, et al., "External Plagiarism Detection", in Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse. Notebook Papers of CLEF 10 Labs and Workshops, 2010.

[21] S. Rao, et al., "External & Intrinsic Plagiarism Detection: VSM & Discourse Markers based Approach", in Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse. Notebook Papers of CLEF 11 Labs and Workshops, 2011.

[22] G. Parth, R. Sameeer, and P. Majumdar, "External Plagiarism Detection: N-Gram Approach using Named Entity Recognizer", in Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse. Notebook Papers of CLEF 10 Labs and Workshops, 2010.

[23] C. Vania and M. Adriani, "Automatic External Plagiarism Detection Using Passage Similarities", in Proceedings of the 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse. Notebook Papers of CLEF 10 Labs and Workshops, 2010.

[24] M. Jiffriya, et al., "AntiPlag: Plagiarism Detection on Electronic Submissions of Text Based Assignments", in 8th IEEE International Conference on, 2013, pp. 376 - 380.

[25] [Online] http://en.wikipedia.org/wiki/N-gram [Accessed: 2015]

[26] Y. Palkovskii, A. Belov and I. Muzyka, "WordNet-based semantic similarity measurement in External Plagiarism Detection", in Proceedings of the 5th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse. Notebook Papers of CLEF 11 Labs and Workshops, 2011.

[27] A. Schmid, et al., "A Concept for Plagiarism Detection Based on Compressed Bitmaps", IARIA, ISBN: 978-1-61, 2014, pp. 208-334.

[28] B. Santorini, "Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision)", 1990, [Online] repository.upenn.edu/cis reports/570 [Accessed: 2014].

[29] D.M. Blei, Andrew and M.I. Jordan, "Latent Dirichlet Allocation", Berkely, CA 94720, 2003.

[30] [online] http://gibbslda.sourceforge.net/, [Accessed: 2015]

[31] J. Algeo, "Having a look at the expanded predicate", in B. Aarts and C. Meyer (1995) The verb in contemporary English: theory and description. Cambridge: Cambridge University Press, 1995, pp. 203-217.

[32] [Online] http://www.shortstoryarchive.com/, [Accessed: 2014]

[33] [Online] https://archive.org/details/textfiles-dot-com, [Accessed: 2014]

[34] [Online] http://articlerewritertool.com/, [Accessed: 2014]

[35] [Online] http://www.uni-weimar.de/medien/webis/research/events/pan-13/pan13-web/plagiarism-detection.html, [Accessed: 2015]

[36] [Online] http://www.princeton.edu/pr/pub/integrity/pages/plagiarism/, [Accessed: 2015]

[37] [Online] http://examples.yourdictionary.com/examples/examples-of-plagiarism.html, [Accessed: 2014].

[38] M. Potthast, et al., "An Evaluation Framework for Plagiarism Detection", Poster Volume, Beijing, 2010, pp. 997–1005.