An Enhanced Framework for Extrinsic Plagiarism Avoidance for Research Articles

S. Imran[1], M. U. G. Khan[2], M. Idrees[3], I. Muneer[4], M. M. Iqbal[5]

[1,2]*University of Engineering and Technology, Lahore, Pakistan*
[3,4]*University of Engineering and Technology, Lahore, Narowal Campus, Pakistan*
[5]*University of Engineering and Technology, Taxila, Pakistan*
[1]shamaz.imran@gmail.com

Abstract-Various approaches have been implemented for plagiarism detection used, for author's work and academic publication. There is a purpose of creating such reliable and effective plagiarism detection with increasing amount of publications. This is a serious offense where one author presents someone else's work as his ownership. Moreover, these algorithms don't consider similar sections for efficient comparison. The proposed framework performs efficient sections wise plagiarism detection and provides suggestions for improving documents. The precision, recall and accuracy based on different n-gram features are presented showing the strictness of higher level n-gram features.

Keywords-Plagiarism Avoidance, Extrinsic Plagiarism, Plagiarism Detection, Plagiarism Remover, Copy Detection, Self-plagiarism

I. INTRODUCTION

With the ever-increasing amount of data over the internet, verifying plagiarism in documents has become a challenging task. Many algorithms for matching two different textual sources have been developed, but such algorithms don't consider the author of the manuscript thus resulting in self-plagiarism.

The exponential growth of online resources [i], [ii] has encouraged plagiarism [iii], and many researchers have tried to set a precise definition of this type of misconduct.

Identifying plagiarism in the era of big data has become a challenging task [iv]. Existing textual data over the web is enormous, and researchers are contributing to it on a daily basis. Significant research on automated plagiarism detection is in progress [v], and the researchers usually use text comparison techniques to compare new documents with existing document collection. Due to the limited vocabulary of a person, one can repeat words in new publications and may cause self-plagiarism. However, terms like “copying” and “borrowing” can disguise the weightiness of the crime [vi].

To check and avoid plagiarism claims different software packages such as Turnitin [vii] have been developed which help a person to change or rephrase document contents identified as plagiarized. Such systems evaluate an incoming document against already published documents collection and suggest a person to remove or change the contents which are found to be copied from other sources

Current algorithms to detect plagiarism normally compare the entire contents of a document against the documents collection without considering the content sections. The plagiarized document contents which belong to a specific section are typically plagiarized from the same section of other documents. For example, if contents of literature review section is plagiarized, the contents will have been copied from the literature review section of some other article so performance evaluation can be done at section level instead of complete documents.

The propose partitioning the documents on the basis of different sections. We have developed an algorithm to analyze documents based on its section to perform partitioning. A document may contain many sections, and the author may choose to use different names for the same type of section. For example, to present her contributions, the author may use various section headings: our research work, research contribution etc. So, our partitioning algorithm is able to perform semantically.

Analysis and partitions, it into a uniform model. Once the document is partitioned, each section of the document will serve as a separate document unit. During plagiarism detection, we analyze each section of the document against the documents having the same category.

The paper is divided into following sections: next section describes the literature work in this area; the architecture of the proposed system is presented next followed by evaluation of the system. Finally, the last section concludes our work and presents future research prospects in this domain.

II. LITERATURE REVIEW

Plagiarism detection is a vast and well-studied field in both research and industry. Two techniques used to detect plagiarism are: intrinsic detection - source documents are not available and plagiarism detection is performed by examining the textual inconsistencies in the document, and extrinsic detection - a new document is examined against a collection of documents already submitted to the system. PAN has been organizing competitions in this field since 2009 and they have a productive contribution in developing standard and techniques in the field of plagiarism detection[viii].

Finding plagiarism from documents whose content is just copied from some other document is an easy task. Detecting plagiarism becomes challenging when word substitution or paraphrasing is used. Intrinsic detection deals with the syntactical nature of the document and requires knowledge of natural language processing and normally used to evaluate web contents. In extrinsic detection, a source document is evaluated against a document collection for possible plagiarism. Algorithms to evaluate extrinsic plagiarism can be classified into statistical and semantical categories [ix]. Statistical models focus on term frequency and are easy to implement. Fingerprinting – where the text is divided into n-grams minutiae - is the most commonly used example of this model. The semantic model emphasizes order and semantics of text. These models are complex and challenging to implement due to performance issues and involvement of lexical ontologies such as Wordnet.

Statistical methods such as Jaccard measure, overlap coefficient and dice measure are the most used techniques due to their simplicity and performance[x]. These methods have been proven very useful along with fingerprinting to detect extrinsic plagiarism. A comparison of statistical and semantically models is shown in Table I. The focus of the statistical model is on textual similarity whereas semantically model focuses on contextual similarity. A statistical model is simple and more efficient than the semantical model. For the importance of the purposed work, we have made a table showing the last decade work from 2009-2017. It is clear that most of the work is done during 2016. Graph no 1 also shows peak point between last three years.

TABLE I
COMPARISON OF STATISTICAL AND SEMANTICAL MODELS

	Statistical Models	Semantical Models
Focus On	Textual similarities	Contextual similarities
Complexity	Low	High
Work Intensity	Low	High
Efficiency	Average	Good
Speed	Fast	Slow
Memory Usage	Low	High
Examples	Jaccard measure, overlap coefficient etc.	Wordnet based models

A human being has limited vocabulary[xi] and the writing style of an author is also a constant factor. If a person has a number of publications, then his/her new research work are more vulnerable to be claimed plagiarized by current plagiarism detection techniques from his already published work. Information analysis is very complicated process in Information Retrieval (IR) systems. Analyzing a text based on the part of speech tagging is very helpful to reduce the comparison cost because all parts of speech in a text are not equally important for plagiarism detection. Wordnet is an open source treasure having rich capabilities in English language processing and POS tagging [xii]. Porter stemmer is the most popular algorithm to normalize the search contents[xiii].

Full-text search - indexing and extracting keywords from shared documents - became a popular topic in information retrieval with the inventions of some great search engines on the internet[xiv]. Many open source software are available in the market for indexing documents such as Apache's Lucene [xv]. A document can have many sections or subtopics and analysis can be conducted to parse the entire document based on its subtopics [xvi]. We have used an open source library "Spire.Doc" to parse a word document [xvii].

The paper was written to propose an algorithm designed for near-copy and paraphrasing types of plagiarism[xviii].

Bi-gram and a graph structure based method [xix] to present that graph-based approach achieve better results in plagiarism detection in the Persian language.

The paper [xx] to propose a method for cross-lingual plagiarism detection based on a semantic approach. They revealed that the highly accurate translation has a significant impact on Intelligent plagiarism detection, compared its method without employing Google translation. 98.82% when employing highly accurate translation tools, 56.9%. Without accurate translation. It also showed that monolingual methods [xxi] literal document 2017 Arabic word stemming, Fingerprinting. A web-based plagiarism detection framework for Arabic documents [xxii].

TABLE II
EXTRACTED PAPERS BASED ON THE CRITERIA IN LITERATURE

Ref	Source/Target	Year	Techniques
[xxiii]	document	2017	A web-based plagiarism detection framework for Arabic documents.
[xxiv]	Text	2017	Similarity Techniques in Information retrieval
[xxv]	Text	2017	COUNTER: corpus of Urdu news text reuse
[xxvi]	Document and Text	2017	Conceptual Review of Literature on Student Plagiarism:
[xxvii]	Document and Text	2016	Bi-gram and a graph structure based method.
[xxviii]	Text	2016	sentence-level algorithm based on tf-idf features
[xxix]	Text	2016	An extrinsic SVM based

[xxx]	Document and Text	2016	Fingerprinting the text in the tri-grams words.
[xxxi]	Text	2016	N-gram
[xxxii]	Document and Text	2016	Obfuscation strategies to provide corpus
[xxxiii]	Document and Text	2016	An examination of the efficacy of the plagiarism detection software program Turnitin
[xxxiv]	Document and Text	2016	A systematic study of knowledge graph analysis for cross-language plagiarism detection.
[xxxv]	document	2015	Similarity technique in information retrieval
[xxxvi]	Document and Text	2015	Fuzzy approach
[xxxvi]	Text	2015	a mixed fuzzy inference system method
[xxxvii]	Text	2015	an artificial obfuscation strategy
[xxxviii]	Text	2014	NLP techniques and N-gram
[xxxix]	Authorship	2014	Two popular Classifiers: FT and SVM.
[xl]	Authorship	2014	MBNB technique Naıve Bayes classifiers
[xli]	Authorship	2013	Word N-Grams.
[xlii]	Text	2013	Examined the existing literal system
[xliii]	Text	2012	Understanding plagiarism linguistic patterns, textual features, and detection methods.
[xliv]	Text	2012	An improved plagiarism detection scheme based on semantic role labeling.
[xlv]	Text & document	2012	winnowing n-gram fingerprinting
[xlv]	Document	2011	A plagiarism detection tool for the Arabic language.
[xlvi]	Text	2011	n-gram model for word retrieval
[xlii]	Text	2011	Understanding plagiarism linguistic patterns, textual features, and detection methods.
[xlvii]	Text	2010	fingerprint matching
[xlviii]	E-learning	2009	Syntax Similarity based detection
[xliii]	Text	2009	based on interpolation of n-gram Probabilities techniques.
[xxx]	Text	2009	N-Gram Based Authorship Attribution in Urdu Poetry.
[xlv]	Document	2009	Fuzzy technique in information retrieval

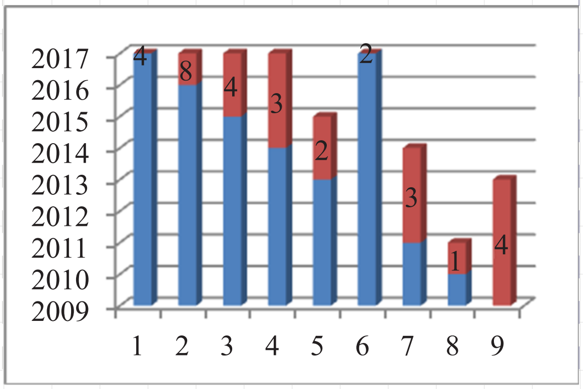

Fig. 2. Publications per year

For better understanding and make our literature review more clear, we generate a bar chart of publications per year as shown in. Graph 1 shows 30 papers per year between 2009 and 2017, the result of a bar chart is the publication of continual plagiarism detection growth. In 2009, only three papers were found, but in 2016 there were 8 papers, with most publications between 2014 and 2016. However, the result for 2017 was much better with four publications. Hence these results show that this area is a new and active area, which means that in the last decade the researchers have focused on this area in publications, especially in the last three years.

III. Design of Projected System

This is the simulation of the paper to be submitted to UET Plagiarism in the simple definition is representing other's works and thoughts as one's own original work or using the words and ideas of someone else as own work without authorization [xvii]. The below-given model has two approaches for the purpose of the operation: admin type and user type. In the admin type, the user builds the allocated space. In user type, the user submits their documents to the system for fraud recognition and after that for removal of that detection. Fig. I show the design of the proposed structure and the detail of each sub-module is described in the next sections.

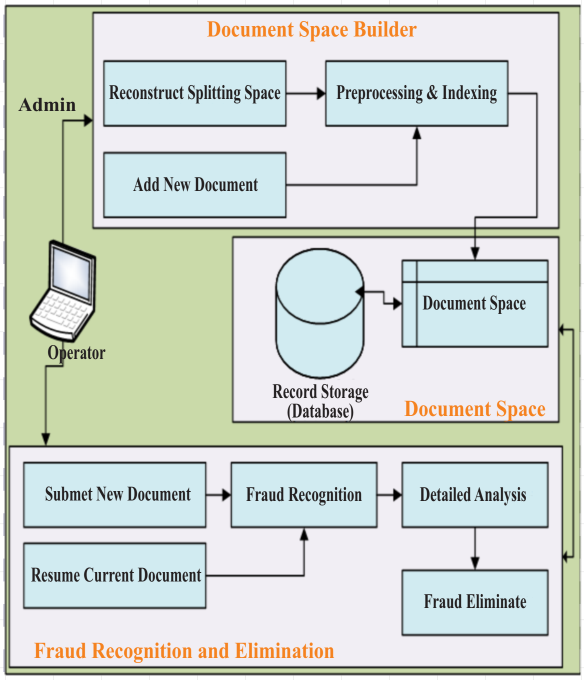

Fig. 1. Architecture of the proposed system

A. Partitioned Space

Partitioned space will have all the files which have been acquired by our proposed system for the purpose of indexing the files. Two kinds of storage strategy are used: first, divide the document on the disk and then store in the database to map them clearly. The mapping happens between the document and their sections clearly.

Fig. 2 shows the Design hierarchy of the proposed system. *Userspace* in the hierarchy is used to store the files submitted by end operator and *system space* have the records which have been delivered to the system as source contents. Both the user and the system space have two additional folders. First one is for the original documents and other is used for the storage of the fragments extracted from these documents. For the ease of retrieval both of the spaces have been indexed.

Rebuild the Partition Method

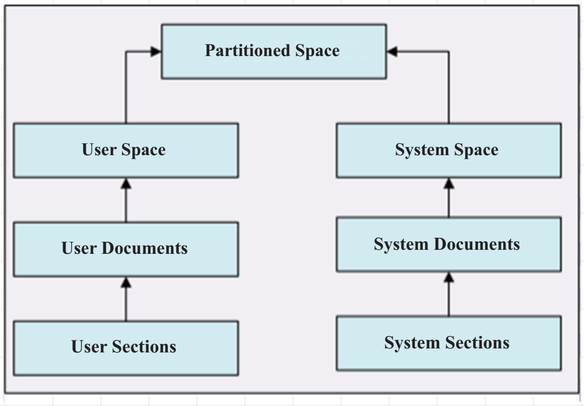

Fig. 3. Architectural Hierarchy

The document space that is indexed on the disk has a relationship like (parent-child), in that case, the parent documents are analyzed, the sections are an excerpt and the mapping is stored in the database. Mapping is of two type in this case when data is storing in database one is reserved for user space and the second one is the document space.

B. Paper Space Constructor

This component is responsible for two type things, i.e. constructing and elevating the partitioned space. The user has the choice of either to reconstruct current space or to upload a new document to space. In the process, the foremost step is to analyze all the documents content and then divide into small portions (sections) based on the primary title (heading). On the other side, some documents like research have a clear and constant format, e.g. IEEE which is the main issue. In this case, we must have known about the required sections which appear in the document just like the heading text may change for the sections that have same semantic meaning in different sections. We rid out of such a problem by making a bond matrix. The matrix contains the parent keys section headings provided by IEEE in the research document and the keys point pointers to the heading that represent the same sections in others documents. For example, the recommended work has pointers to keys "ENQUIRY CONTRIBUTION" and "PROJECTED TECHNIQUE". The administrator can update the table for getting the future key. The operator can replace with a new document in the existing space. The document is divided based on its sections and then added to the current document space. The process is functionally divided and run at the "Reconstructing Partitioning Space" process which runs for the entire space only once for a particular document.

There are two types of entities associated when the document is added to space. The first one includes various properties of a document such as status, path and other related entity (a section which appears in a document). The document comes with several sections that are mapped using a table association of the document at the time of parsing. This child module is responsible for parsing a document and identifying its different sections. Fig. 3 shows the pseudo code to rebuild the portioning space.

```
Procedure RebuildPartition
  Lookup[]=GetdocumentSetions()
  SS=Getsystemspace()
  n1=lenght(ss)
 for i=0 to n-1 do
  section_text=Φ
  p[]=GetdocumentParagraphs()
  n2=legnth(p)
  for j=0 to n2-1 do
  if is_new_section then
  section_name=GetSectionNameFromLookup()
           Normalize(section_text)
           CreateFile(Section_name)
           section_text=Φ
           else
           section_text+=GetText(j)
           end if
           end for
           CreateFile(LastSection)
end for
IndexPartitionSpace()
end Procedure
```

C. Plagiarized Content Detection

The main responsibility of this child module is to identify doubtful contents inside the document. Once the contents are recognized, the user is in control to change the contents of the document. The author can update documents in two ways: replacing a single word with a suggested word or the whole sentence. To become a user of the system, the author shall create a free account in the portal and he/she will be able to submit a document for plagiarism detection. The documents are processed on a section by section basis. Once the

Plagiarism Detection Method

```
Procedure PLAGRIAMDETECTION($doc_i$)
ds[]=GetSectionsFromCurrentDocument()
dn=GetLengthds()
for i=0 to dn-1 do
  dn_gram[]=n_gram of dn(i)
  ss[]=GetSameNameSection(dn(i))
  sn=length(ss)
  cns=configured size of the n_gram
cda=configured detection algorithm
threshold=tolerated plagrirasm cofficent
for j=0 to sn-1 do
  sn_gram[]=n_gram of ss(j)
  common_ngram=sn_gram ^dn_gram
  union_ngram=sn_gram Ųdn_gram
  score=2(common_ngram)/ union_ngram
  if score>threshold then
  plagrirasm_set.add[ss(j)]
    end for
end for
return plagriasm_set
end Procedure
```

documents are processed, they are stored in the user space. The document contains many suspicious sections and it's not possible to precise all of them in one run, so user saves his/her work and can resume them after some time when required.

The system used a procedure called fingerprinting along with some statistical method for data processing. The fraud algorithm is configurable and the base of the algorithm named dice coefficient of n-gram size of four shown in Fig. 5. The selection of the method was based on our observations during the evaluation. The results were more precise with dice coefficient as shown in the evaluation part. The other measures which were tried include Jaccard coefficient and overlap coefficient. The algorithm returns the probability score within a range of 0 to 1. The threshold value is configurable in the system which is compared with the probability score. If the score value is higher than the threshold value, the section selected mark as copied from the compared section. The sections in the document are then compared with the sections in document space and the list of the possible plagiarized sections is presented to the user.

D. Comprehensive Review

Once the plagiarism in a section has been recognized, the user is to manually compare both documents to view the similar content for further analysis. The text between two sections will be highlighted if they are to be found similar. It empowers the user to perform further analysis by using Robin Karp [xi], gst, or Knuth Morris Pratt [xii] algorithms. The user will be able to recognize words or sentences which have been copied and may require any further updates. Fig V shows the detailed analysis algorithm:

Detailed plagiarism analysis

```
Procedure PLAGRIAMDETECTION($section_i$)
  xiW=GetWord($section_i$)
  xiY=GetWord($section_i$)
  xn=length(xiW)
  yn=length(xiW)
  for i=0 to xn-1 do
  occurances[]=KMPratt(xiW[i], yiW)
  o_count=length(occurances)
  for j=0 to o_count  1  do
    xiW[j].highlight=TRUE
    if score>threshold then
    plagrirasm_set.add[ss(j)]
   end if
   end for
end for
highlighted_text=convertToText
return  highlighted_text
end Procedure
```

E. Plagiarism Eliminator

The user can eliminate fraud in the given two ways: replace a word by synonym or rephrasing. If a single word is replaced by the user when he/she will be afforded a variety of all the matching synonyms sorted on the frequency base used in the current Documents collection in which user can pick any of them, or he/she can replace by picking any of them in the existing word. We then retrieve all the available rephrased sentences which can be used to substitute the selected sentence. The order of the list based on analyzing score and resemblance with the sentence. Stanford parser is used to parse the score which determines the strength of a sentence whereas similarity score can be found using Levenshtein distance and Kuhn-Munkres algorithms [xiii].

The average of these scores is used to sort the paraphrase collection. When there is no further doubtful sentences and words have been substituted, and the process will be continuing on until all the signs are entirely eliminated by reassessing the document again and again.

We present the Pseudo code for our fraud eliminator in Fig. 6.

Plagiarism Removal Method

```
Procedure PLAGRIAMDETECTION($section_i$)
 ss[]=GetSubTitles($section_i$)
 dn= length(ss)
 weight=Φ
  for i=0  to  dn-1  do
   s_score=similarity_score([i])
   lm_score=language_model_score([i])
   weights[i]=s+lm_score/2
   end for
   started_options[]=sort(c,weights)
 subtitles=Top(started_options,5)
return  subtitles
end procedure
```

IV. Evaluation

We performed the evaluation by conducting some experiments to assess the primary prototype of our system. The aim of these experiments is to study the efficiency of the current system in fraud recognition and eliminator. The framework is constructed from three significant operations, i.e. to fraud recognition, fraud evaluation and eliminating questionable content from the document. Plagiarism analysis shows questionable content in the document to help the user identify which part of the document is weak and needs further alterations. Or we can say that it's all about comparison, in which two strings are compared on the bases of similar words by using the existing method in this segment so that the calculation is not considered too much. The last component of our proposed framework is to notice plagiarized content which is done automatically (no manual intervention is required). The document is partitioned and then we applying fingerprinting algorithms to discover the doubtful segments in a document. Here due to space limitation, we only present three examples as shown in next subsections.

V. Corpus Creation

During the creation of mass, we selected only four documents. Two of them were selected whose author was same and they were used to detect self-fraud, and the other two documents are from different authors.

TABLE III
Test Documents for Evaluation

Paper	Piracy Notes	Author
A1	Not Copied	KLN
B1	Not Copied	ABC
C1	Not Copied	ABC
D1	Not Copied	DHQ
A2	Mockup of A	NIL
B2	Mockup of B	NIL
A2B2	Fully Copied From A1 & B1	NIL
A2B2C2	Fully Copied From A1, B1 & C1	NIL
B2C2	Fully Copied From B1 & C1	NIL
A2D2	Fully Copied From A1 & D1	NIL
A3B3	Moderately Copied From A1 & B1	NIL
A3B3C3	Moderately Copied From A1, B1 & C1	NIL
B3C3	Moderately Copied From B1 & C1	NIL
A3D3	Moderately Copied From A1 & D1	NIL
A3B3C3D3	Moderately Copied From A1, B1, C1 & D1	NIL
E1	Not Copied	NIL
F1	Not Copied	NIL
G1	Not Copied	NIL
H1	Not Copied	NIL
I1	Not Copied	NIL

From the user space three sets of documents were selected: completely plagiarized from the system space, moderately (50% approx.) derived from system space, and original content. Fig. 7 shows the document space for evaluation.

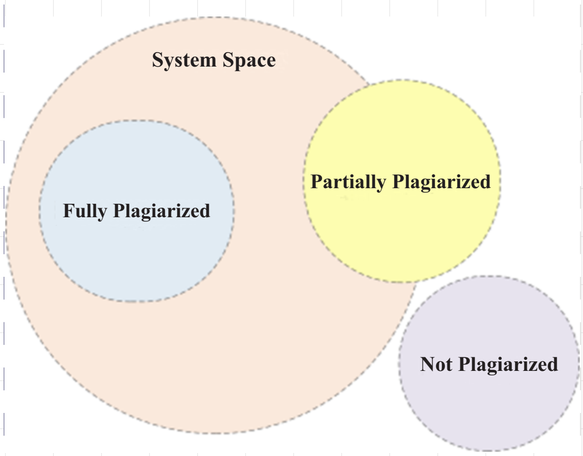

Fig. 4. Document space for evaluation

VI. Test Results

We have configured dice coefficient as default plagiarism detection algorithm in our system. There are 16 experiments which are conducted in each case. Following table shows the documents which were selected for evolution.

Six of the test documents were entirely copied. Five documents were partially plagiarized. Five test cases did not have any contents which were copied and also the text has been taken from the unique domain for each of the document. The final output of these experiments was a confusion matrix which is shown in Table III.

TABLE IV
Confusion Matrix for N-Grams

N-gram Size	True +ve	True -ve	False +ve	False -ve
3	7	7	4	0
4	6	8	2	0
5	7	7	0	2

We take a different reading while continuously changing the size of n-gram. We see N-gram value that records which is less than three is too much strict and the algorithms warrant that all document is copied, and the values greater than five output in none of the document as copied. On the bases of this statement, n-gram values between 3 and 5 have been presented in Table IV.

TABLE V
SIMULATIONS & RESULTS

N-gram	Sensitivity	Specificity	Precision	Accuracy
3	1	0.6364	0.6364	0.7778
4	1	0.8	0.75	0.875
5	0.7778	1	1	0.875

Trigram: The size affects the overall system behavior we see the result form the size value n-gram when it was three the system behavior is harsh. In the system space, the result was imitative, but we found that the four documents were not copied from any document. Look at Table IV, it will realize that all the real cases were calculated correctly and the value of understanding is 1, but in case of definite, the value is 0.6364 that result many documents which were not imitate were marked as derivative because of many common trigrams between the source and destination sections.

4-gram: The effect doesn't fall on the system precision on the sensitivity point of view even if the value of n-gram equal to 4, also we can see and compare the values from the above-mentioned chart that the 4-gram is better in other parameters than trigram. From the system performance point of view, this one is the best-case scenario.

5-gram: When the value of n-gram is when to adjust to 5 then all the documents which are not imitative were weighed correctly by the system, but in this picture what we can is that some document which was imitative are selected as not copied by the prototype. For detection of coping purpose, this mode is also not useful.

Plagiarism elimination is also partially-automatic, in such process user should maintain all of the actions. We work on the classification of the search results, but still, the user can decide the choice to ignore all the options suggested by our prototype. The substitutions of a sentence are drawn from Microsoft rephrasing web service. We also compute language model score and resemblance score of each sentence. Language modeling score is calculated by using Stanford parser and resemblance score can be calculated by using the Edit-distance algorithm.

$$w_i = \frac{X_i}{\sum X} + \frac{Y_i}{\sum Y} \qquad (1)$$

$$Ranks = Sort(w_1, w_2, \ldots , w_n) \qquad (2)$$

Below are the step given to calculate the ranking using formula. The formula filtered and return 5 top results to the operator (user) and the remaining are discounted as shown in Table V.

TABLE VI
SIMILARITY SCORE / SENTENCES RANKING

LM Scoring	Similarity	Ranking
-210.9883	0.8356	1
-212.0762	0.827	2
-213.2161	0.8205	3
-207.9625	0.7567	4
-210.1902	0.7341	5
-209.0503	0.7368	6

Input: This paper output the implementation of a reliable network file system on a delegated server

Replacements:

The present article indicates a way to contrivance a trusted "network file system" on some assigned server.

This paper defines an implementation technique over a confidential network file system for an allocated server

This paper help determine a way to implement a reliable network file system on a particular server

This article presents the implementation of dependable network file system for an assigned server

The given paper demonstrates how to contrivance a dependable network file system for a particular assigned server

This paper also designates how to devise a reliable and efficient network file system on some dispersed server

VII. CONCLUSIONS

Exterior plagiarism detection is a most popular way to compare the new publications and already published research articles. The techniques used to identify plagiarism include syntactic and grammatical resemblance which is difficult for humans to compare manually because of the lack of terminology set. Natural Language Processing (NLP) techniques as well as supervised machine learning algorithms, are combined to detect plagiarized texts. Here, the primary emphasis is on to construct a framework which detects. For successfully detecting the plagiarism, n-gram frequency comparison approach has been implemented to construct the model framework. N-gram frequency comparison approach has been implemented to construct the model framework for successfully detecting the plagiarism, the presented system facilitates a person to update his research work by utilizing synonym replacement feature and reshaping the doubtful textual content. Filter metrics have applied to select most relevant characteristics and then supervised classification learning algorithm is being used to classify the documents in different levels of plagiarism. Confusion matrix was built to estimate the false positives and false negatives rates. We have further proposed algorithms to perform section wise plagiarism detection and efficient indexing techniques for our knowledge base and segment based document indexing. The estimation of our system shows satisfactory results.

REFERENCES

[i] M. Mansoorizadeh, "Persian Plagiarism Detection Using Sentence Correlations," in FIRE (Working Notes), pp. 163-166, 2016.

[ii] M. Momtaz, "Graph-based Approach to TextAlignment for Plagiarism Detection in Persian Documents," in FIRE (Working Notes), pp.176-179, 2016.

[iii] F. Safi-Esfahani, "English-Persian Plagiarism Detection based on a Semantic Approach," Journal of AI and Data Mining vol. 5, pp. 275-284, 2017.

[iv] Y. Abdelrahman, "A Method For Arabic Documents Plagiarism Detection," International Journal of Computer Science and Information Security, vol. 15, p. 79, 2017.

[v] S. Rafieian, "Plagiarism checker for Persian (PCP) texts using hash-based tree representative fingerprinting," Journal of AI and Data Mining, vol. 4, pp. 125-133, 2016.

[vi] L. Gillam and A. Vartapetiance, "From English to Persian: Conversion of Text Alignment for Plagiarism Detection," PAN@ FIRE2016 Shared Task on Persian Plagiarism Detection and Text Alignment Corpus Construction. Notebook Papers of FIRE 2016, 2016.

[vii] M. Sharifabadi and S. Eftekhari, "Mahak Samim: A Corpus of Persian Academic Texts for Evaluating Plagiarism Detection Systems," in FIRE (Working Notes), pp. 190-192, 2016.

[viii] K. Walchuk, (2016). An examination of the efficacy of the plagiarism detection software program Turnitin, 2016.

[ix] S. Orim, “Conceptual Review of Literature on Student Plagiarism: Focusing on Nigerian Higher Education Institutions”. World Journal of Educational Research, 4(1), 216, 2017.

[x] F. Salvador, M. Rosso, and M. Gómez, A systematic study of knowledge graph analysis for cross-language plagiarism detection. Information Processing & Management, 52(4), 550-570, 2016.

[xi] A. Raza, A. Athar, and S. Nadeem, N-Gram Based Authorship Attribution in Urdu Poetry. In Proceedings of the Conference on Language & Technology pp. 88-93, 2009.

[xii] M. Sharjeel, R. Nawab, and P. Rayson, COUNTER: corpus of Urdu news text reuse. Language Resources and Evaluation, 51(3), 777-803, 2017.

[xiii] I. H. Khan, M. A. Siddiqui, and K. Mansoor, "A Framework FOR PLAGIARISM DETECTION IN ARABIC DOCUMENTS." pp. 01–09, 2015.

[xiv] S. Rakian, "A Persian Fuzzy Plagiarism Detection Approach," Journal of Information Systems and Telecommunication (JIST), vol. 3, pp. 182-190, 2015

[xv] H. Ahangarbahan and G. Montazer, "A Fuzzy Approach for Ambiguity Reduction in Text Similarity Estimation (Case Study: Persian WebContents)," Information Systems &Telecommunication, p. 216, 2015.

[xvi] I. H. Khan, M. A. Siddiqui, and K. Mansoor, "AFramework FOR PLAGIARISM DETECTION IN ARABIC DOCUMENTS." pp. 01–09, 2015.

[xvii] M. Mahmoodi and M. Varnamkhasti, "Design a Persian Automated Plagiarism Detector (AMZPPD)," arXiv preprint arXiv:1403.1618, 2014.

[xviii] A. Otoom, "Towards author identification of Arabic text articles," in Information and Communication Systems (ICICS), 2014 5th International Conference, pp. 1-4, 2014.

[xix] A. Altheneyan and M. Menai, "Naïve Bayes classifiers for authorship attribution of Arabic texts," Journal of King Saud University Computer and Information Sciences, vol. 26, pp. 473-484, 2014.

[xx] S. Ouamour and H. Sayoud, "Authorship attribution of short historical arabic texts based on lexical features," in Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference, pp. 144-147, 2013.

[xxi] L. Ramya and R. Venkatalakshmi, "Intelligent plagiarism detection," International Journal of Research in Engineering & Advanced Technology (IJREAT), vol. 1, pp. 171-174, 2013.

[xxii] S. Alzahrani, M., Salim & A. Abraham, Understanding plagiarism linguistic patterns, textual features, and detection methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),42(2), 133-149, 2012.

[xxiii] S. Alzahrani, M. Salim, and A. Abraham, Understanding plagiarism linguistic patterns, textual features and detection methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, (99), 1. 2011.

[xxiv] A. Osman, N. Salim, M. Binwahlan, R. Alteeb and A. Abuobieda,. An improved plagiarism detection scheme based on semantic role labeling. Applied Soft Computing, 12(5), 1493-1502, 2012.

[xxv] A. Jadalla and A. Elnagar, "A fingerprinting-based plagiarism detection system for Arabic text-based documents," in Computing Technology and Information Management

(ICCM), 2012 8th International Conference on, pp. 477-482, 2012.

[xxvi] M. Menai and M. Bagais, "APlag: A plagiarism checker for Arabic texts," in Computer Science & Education (ICCSE), 2011 6th International Conference on, pp. 1379-1383, 2011.

[xxvii]M. Khan, "Copy detection in Urdu language documents using n-grams model," in Computer Networks and Information Technology (ICCNIT), International Conference on, 2011, pp. 263-266, 2011.

[xxviii]C. Kent and N. Salim, "Features based text similarity detection," arXiv preprint arXiv:1001.3487, 2010.

[xxix] S. Alzahrani, "Work in progress: Developing Arabic plagiarism detection tool for elearning systems," in Computer Science and Information Technology-Spring Conference, IACSITSC'09. in Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09. International Association of. IEEE pp. 105-109, 2009.

[xxx] A. Raza, "N-Gram Based Basic format for periodicals: Authorship Attribution in Urdu Poetry," in Proceedings of the Conference on Language & Technology, pp.88-93, 2009

[xxxi] S. Alzahrani and N. Salim, "On the use of fuzzy information retrieval for gauging similarity of arabic documents," in Applications of Digital Information and Web Technologies, 2009. ICADIWT'09.Second International Conference, pp. 539 544, 2009.

[xxxii]M. Potthast, A. Eiselt, and L. Cedeño, "Overview of the 3rd international competition on plagiarism detection," CEUR Work., 2011.

[xxxiii]R. Lukashenko and V. Graudina, "Computer-based plagiarism detection methods and tools: an overview," Proc. 2007, 2007.

[xxxiv]V. Thada and D. Jaglan, "Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm," Int. J. Innov. Eng., 2013.

[xxxv]D. Buss, Evolutionary psychology: The new science of *the mind*. 2015.

[xxxvi]G. Miller, "WordNet: a lexical database for English," Commun. ACM, "Porter Stemmer." 1995.

[xxxvii]A. Broder, N. Eiron, M. Fontoura, and M. Herscovici, "Indexing shared content in information retrieval systems," *Extending Database* , 2006.

[xxxviii]A. Lucene, A. Shukla, "Sentiment Analysis of Document Based on Annotation," arXiv Prepr. arXiv1111.1648, 2011.

[xxxix]G. Gonnet and R. Baeza-Yates, "An analysis of the Karp-Rabin string matching algorithm," Inf. Process. Lett., 1990.

[xl] M. Abdellatif, "Accéleration des traitements de la sécurité mobile avec le calcul parallèle." PhD diss., École de technologie supérieure, 2016.

[xli] W. Heeringa, "Measuring dialect pronunciation differences using Levenshtein distance," 2004.

[xlii] R. Iqbal, A. Grzywaczewski, J. Halloran, F. Doctor, and K. Iqbal, "Design implications for task-specific search utilities for retrieval and reengineering of code", *Enterprise Information Systems Taylor and Francis*, pp. 1751-7575, 2015.

[xliii] O. Alhabashneh, R. Iqbal, F. Doctor and S. Amin, "Adaptive information retrieval system based on fuzzy profiling", Proc. of Intl. Conf. on Fuzzy Systems, pp. 1-8, 2015.

[xliv] M. Alsallal, R. Iqbal, S. Amin, and A. James, "Intrinsic plagiarism detection using latent semantic indexing and stylometry", Proc. 6th Intl. Conf. on Developments in eSystems Engineering (DeSE), pp. 145-150, 2013.

[xlv] M. Alsallal, R. Iqbal, S. Amin, and A. James, "Intrinsic plagiarism detection using latent semantic indexing and stylometry", Proc. 6th Intl. Conf. on Developments in eSystems Engineering (DeSE), pp. 145-150, 2013.

[xlvi] M. Alzahrani, "Work in progress: Developing Arabic plagiarism detection tool for elearning systems," in Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09. International Association, pp. 105-109 , 2009.

[xlvii] A. Raza, "N-Gram Based Basic format for periodicals: Authorship Attribution in Urdu Poetry," in Proceedings of the Conference on Language & Technology, pp.88-93, 2009.

[xlviii]M. Alzahrani and N. Salim, "On the use of fuzzy information retrieval for gauging similarity of arabic documents," in Applications of Digital Information and Web Technologies, 2009.