Detecting Urdu Text Plagiarism Using Similarity Matching Techniques

Anam Lodhi[1], Saad Razzaq[2] and Muqaddas Gull[3]
[1,2,3]Department of Computer Science, University of Sargodha, Pakistan

***Abstract*— Plagiarism Detection is emerging as an important part in various fields either it is educational sector or industrial. Stealing someone's work or ideas is now more common than before. There have been developed many tools for plagiarism detection dealing directly with English language. Huge part of academic content is based on Urdu language, which serves as a means of communication in many countries. Detecting plagiarism is really a strenuous task accomplished in any language but it is even harder in Urdu language because of its complex structure. The sentence formation and lexicon structure of Urdu language are totally different from the structure of English and other European languages. We are evaluating text matching techniques to detect plagiarism in Urdu language.**

***Keywords*— Urdu Language, Plagiarism, Pre-Processing Techniques, Detection System and Similarity**

I. INTRODUCTION

It is convenient to access information through internet whether it is in the form of text, image, audio or video. Due to the fact, one can easily use digital information in any way. This practice raises issues to content security, and decreases the complications of writing essays and documents along with their cost and effort. Nowadays, there is a huge trend of copying content of others and presenting them as your own intellectual property. It is an alarming situation for those who put their utter efforts in making original content.

Content theft and idea-stealing are found almost in every concerning field. For example, students find secure methods to escape writing efforts for their assignments, discussions and papers etc. Another dimension of this practice is translating original content into other language and presenting it as your own work.

Plagiarism includes duplication of content of others and presenting them as your own work. If there exits plagiarism, there should be a method to detect the quality of a document. Appliance of plagiarism detection techniques are one way to guarantee the quality of academic documents. Academic agencies like HEC have great interest in ensuring the quality and eligibility of academic content. So to ensure this, there is a need of better system which can possibly prohibit the misuse of accessible content. Typically, one cannot be accused of content fraud or theft without any evidence. A document is property of a particular person is to be proved by providing proper evidence. This could be proved by the comparison of original and suspected document.

There are many tools available for the detection of plagiarism like Aplag, Turnitin, EVE2 and Copyscape. All these tools are developed mainly for Arabic and English. Intellectual and academic documents are written almost in every language; therefore, there is a need of hardcore tools for detecting plagiarism in other languages as well.

Researchers are working on detecting plagiarism for Urdu language. To the best of our knowledge, we have found only one tool for detecting plagiarism for Urdu language [1]. The system undergoes all the necessary preprocessing steps before evaluating the percentage of copied material in the documents. A threshold value of 75% resemblance is set as yard stick for text classification being plagiarized [1]. At conclusion, it evaluates that tri-gram is found to be best fit for word extraction model as compared with other n-gram models. This comparison was made with bi-gram and four-gram [1]. This work becomes the basis of more research in this field.

II. RELATED WORK

Nowadays, there is a huge trend of incorporating online information in documents of any kind like articles, poetry, books and other educational materials. When it comes to study or evaluate number of documents written by multiple authors, there is a need of checking authenticity of these documents. The authenticity of the documents are checked on the basis of plagiarism, which is one of the most important factor verify the originality of any document. For this, information processing tools provide proper assistance. All these tools are able to measure similarity between multiple documents. Excessive use of internet information by organizations and individuals is the main reason behind the development of similarity measuring tools.

Plagiarism is a serious problem for all those individuals and educational institutes where originality of document is quite of concern. Use of online digital libraries and databases is advent

Anam Lodhi is with the Department of Computer Science, University of Sargodha, Pakistan
Saad Razzaq is with the Department of Computer Science, University of Sargodha, Pakistan
Muqaddas Gull is with the Department of Computer Science, University of Sargodha, Pakistan, (Email: muqaddasgull@yahoo.com)

cause of occurring plagiarism is documents. There is very thin border-line between research and plagiarism. For identifying a document for plagiarism, various factors are to be considered in text-based on grammatical structure of the language of that document.

Plagiarism detection is widely used in academic area for checking academic documents for plagiarism. These are helpful in identifying copied assignments of students. Vast and comprehensive work has been made in the field of plagiarism detection for natural languages. Especially for English language, a great volume of efforts have been put in this regard.

According to Lancaster (2003), there are several classifications of plagiarism detection approaches. Multiple factors are responsible for this classification such as type of detection method, number of documents that are to be processed by metrics, complexity of metrics and availability of document [2].

A. NLP Approaches

NLP approaches involve machines for processing human languages. Pre-processing techniques are used to improve plagiarism detection accuracy. This includes removing punctuations, lemmatization, and removal of irrelevant words. Some heuristics had positive impact on accuracy but NLP did not exhibit significant improvement with respect to basic approach. NLP based on simple pre-processing techniques such as tokenization, stop-word removal, stemming had given feasible results for detection of duplication between documents [2].

B. Influence of Text Preprocessing

Detail study shows that use of text pre-processing techniques for plagiarism detection can seemingly produce improved results as compared to no pre-processing used. There are several pre-processing techniques used in multiple combinations to gain benefit of each one. Techniques include tokenization, lemmatization, stop-word removal, synonymy recognition, number replacement and word generalization [3].

The certain experiments have been employed against documents of Czech corpus. Results showed that experimental text preprocessing cannot significantly improve the accuracy of plagiarism. NMR, SYR and WG showed a slighter good performance. If execution efficiency is the issue, then LM and STR should be considered. Taking punctuation in account left negative impact on performance. Closure note is that 4th generalization level approaches the best results for text preprocessing techniques [3].

C. Detecting Plagiarism Using Aplag

A tool has been developed to detect plagiarism in Arabic documents which uses exact heuristics to compare suspected documents while avoiding unnecessary comparisons [4]. There are many traditional ways to detect plagiarism as detecting copy-past and detecting writing style within a document. But the proposed tool known as Aplag is suggested as language independent tool which focuses on character, length of sentences and frequency of special words instead. Aplag (Arabic Plagiarism Detection) tool is based on content based methodology. Three proposed phases for Aplag are preprocessing of input text, processing & removing common idioms in Arabic language and evaluating performance results. Aplag is basically suggested as a prototype for Arabic language. This works for synonyms replication and some hidden forms of plagiarism. A heuristic based algorithm for plagiarism has been described, and finally a series of experiments on a large number of Arabic documents have been presented. It is concluded that Aplag is capable to detect change in sentence structure, synonym replacement and exact copy in documents [4].

Using Iqtebas1.0 for Arabic Language: There are many other techniques used for detecting plagiarism in Arabic documents. Iqtebas 1.0 is one of them which is a Fingerprinting based plagiarism detecting method. The purpose is to propose a search engine based on the factor of fingerprinting which reduces pairwise similarity, index size and yielding a maximum recall value with robust results. The proposed technique in this paper focuses on text based documents. Fingerprinting is basically like generating a unique numerical value for a text based document which uniquely identifies the document. Comparing fingerprints of suspected documents will generate a plagiarism result. Generally approaches for fingerprinting go through multiple steps which are: non overlap approach, overlap approach, winnowing and post processing. Iqtebas 1.0 is a robust and complete plagiarism detector in Arabic text based documents which is built around search engine based mechanism. It does not execute pairwise comparison, hence it yield results through ranked Boolean queries. Winnowing fingerprinting technique has been working on reducing index size and increasing the efficiency of search engine performance [5].

Various N-grams For Various Plagiarism Cases: Plagiarism has become a crucial task as the volume of the available information on the internet has breached the extensive level. Many techniques have been employed in this regard. Each technique/method suits a specific problem and plagiarism nature. Text can be plagiarized in many ways, copy and pasting, replicating synonyms or diving into someone's written style. The relevant paper on this subject proposes a plagiarism detection system which aims at detecting the plagiarism method adopted either verbatim or obfuscated. The proposed method explains three techniques for detecting various plagiarism methods: Stopword, n-grams and n-grams with at least one named entity.

The proposed system aims at detecting plagiarism in passages with different levels of obfuscation. Three different types of n-grams have been used to detect obfuscation differently from suspected passages. A closure note is left that these methods should be combined in such a way that could not hurt or harm the integrity of the detection, and single & integrated system is possible to detect different levels of obfuscations indeed [6].

III. PROPOSED COPY DETECTION SYSTEM

Our proposed plagiarism (copy) detection system for Urdu text documents is based on three similarity detection algorithms. We have also incorporate n-gram techniques combined with recommended methodology. Before applying similarity algorithms, Urdu text documents undergo preprocessing steps. Combination of our applied pre-processing techniques is - tokenization, punctuation removal, stop word removal, chunking and hashing.

Brief explanation of mentioned pre-processing techniques explains how these steps are performed on Urdu text document. Fig. 1 presents a model diagram that shows mentioned steps applied on Urdu text.

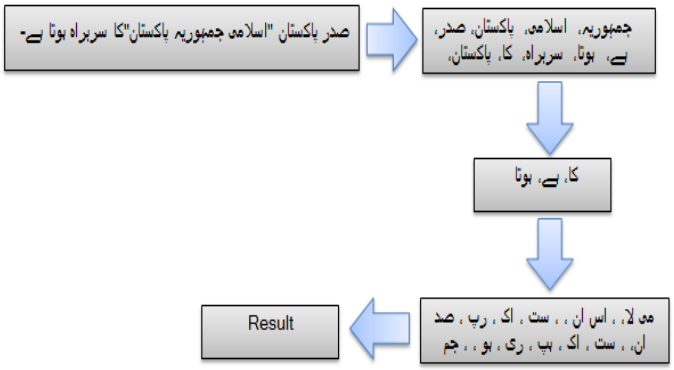


Fig. 1: Supporting Steps of Proposed Model

A. Preprocessing Techniques

Tokenization: The process of defragmenting a series of textual data into small words, phrases or any other meaningful components is known as tokenization [7]. Each piece of word or phrase in a document which undergoes this process is termed as a token. Main idea of using this process is to identify meaningful words out of complete document. It explores all words in a document.

Punctuation Removal: Punctuations are an effective part of any speech, and these are used in forming a proper sentence structure. Punctuations are meaningless when used separately. While applying text classification or text processing methods, punctuations never heal any meanings, and it is better to remove all punctuation marks from the document which is undergoing the process of plagiarism detection [8].

Stop Word Removal: Stop words are the part of textual speech, but they make no sense when taken alone. There are multiple categories of stop words such as auxiliary verbs, prepositions and pronouns etc. Stop words always have high frequency in corpse, so it is necessary to remove them before applying plagiarism detection methods [8]. Removal of stop words minimizes the size of indexing file. This process ensures the overall efficiency and effectiveness of retrieved results as suggested by authors [9].

Chunking: Process of dividing the complete document into small pieces is known as chunking. Chunking is beneficial for applying plagiarism detecting algorithms on suspected textual document. Words are taken as small unit of chunk in a document. Out of several types of chunking methods, we have selected N-gram model.

The technique of making chunks using N-grams apply following steps.

For example, a given document consist of series of words like w1 w2 w3 w4 w5 w6, if value of N is set to 3, every next chunk will contain words from the preceding chunk. Resultant will be like w1 w2 w3, w2 w3 w4, w3 w4 w5, w4 w5 w6.

B. Similarity Matching Algorithm

Our proposed copy detection system for Urdu text is based on implementation of cosine, Jaccard and dice similarity matching algorithms. These algorithms have served for English language multiple times but our proposed system evaluates their possible performance for Urdu text.

Cosine Similarity Coefficient: It is a vector based similarity measure. Cosine similarity is the measure of cosine angles between two vectors (strings). The result appears in the range of -1 and 1. 1 corresponds to similarity and -1 to dissimilarity. Zero resultant value indicates that both documents are dissimilar and unrelated to each other. The resultant value will be derived by calculating Euclidian distance between vectors. It can be derived by using Euclidean dot product formula.

Similarity score is calculated by comparing the difference of angles between set of strings. It is done by comparison of intersection and union of set of strings i.e., the union of binary occurrence vector and frequency occurrence vector. Resultant frequency occurrence vector is a set representation of strings. If the vectors are parallel, they are similar and if they are orthogonal then there is no similarity between them.

$$cos(\overrightarrow{t1}, \overrightarrow{t2}) = \frac{\overrightarrow{t1} \cdot \overrightarrow{t2}}{\|\overrightarrow{t1}\| \, \|\overrightarrow{t2}\|} \quad (1)$$

Jaccard Similarity Coefficients: The Jaccard similarity coefficient is measured by calculating the relation between intersection and union of two sets. It is formulated as following:

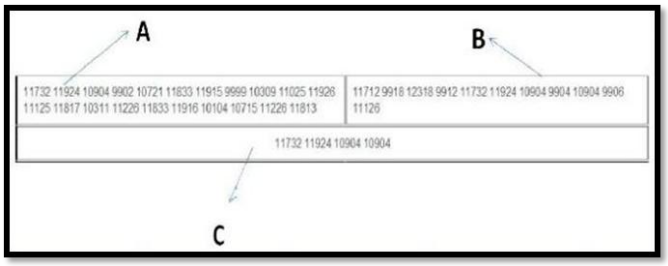
$$Sj = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Let us consider an example for better understanding of this similarity algorithm.

Where, ‘A’ and ‘B’ are two sets of words belonging to two document each. This measure results in estimated value of likelihood of content among documents.

t_1 = مجھے اردو اچھی لگتی ہے۔ t_2 = اردو سیکھنی چاہیے۔

Perform pre-processing steps preceding with evaluating hash value for words in a document. As a result we will get:



Evaluate : $J(A,B) = \frac{C}{A+B-C}$

A = number of total hash values in document 1, B = number of total hash values in document 2
C = total number of hash intersection between both documents

The resultant value is 0 when the two sets are disjoint, 1 when they are equal, and strictly between 0 and 1 otherwise. Two sets are more similar (i.e. have relatively more members in common) when their Jaccard index is closer to 1. The process of cutting down strings from a complete document is termed as hashing. Actual result of hashing produces a hash values associated with chunks of text in a document. After the successful chunking/tokenization, hashing functions are applied to extract integer values corresponding to every token. These values are used for similarity matching and identifying percentage of plagiarized text.

C. Dice Similarity Coefficient

Dice similarity measurement model uses n-gram approach to measure similarity between pair of words. It is formulated by the relation of common entity (n) by two entities as total ($n_x + n_y$)

$$S = \frac{2nt}{nx+ny} \qquad (3)$$

Where, n_t is total number of n-grams found in both documents and n_x, n_y are number of n-grams found in each of the document.

Algorithm for computing:

D1 ⟵ original document

D2 ⟵ suspicious document

Remove punctuations from D1 and D2

Remove Stop words from D1 and D2

strGram1 ⟵ D1 converted into n-grams

strGram2 ⟵ D2 converted into n-grams

Total Matched ⟵ 0

for (i ⟵ 0 to size of strGram1)

if (strGram2 contains strGram1[i])

Total Matched ⟵ Total Matched + 1

diceSimiliarity ⟵ 2*TotalMatched /(sizeof strGram1+ sizeof strGram2)

dicePercent ⟵ diceSimiliarity * 100

IV. RESULT

This section contains evaluated results of our proposed methodology. We collect dataset of Urdu documents from multiple sources like Urdu blogs, news portals, Urdu websites etc. Major portion of our tested datasets is gathered from Urdu Wikipedia and Emille corpus. We have analyzed our prototype by testing it for multiple data sets. Example datasets are shown below in Fig. 2 and Fig. 3.

Fig. 2. Example of Urdu Dataset

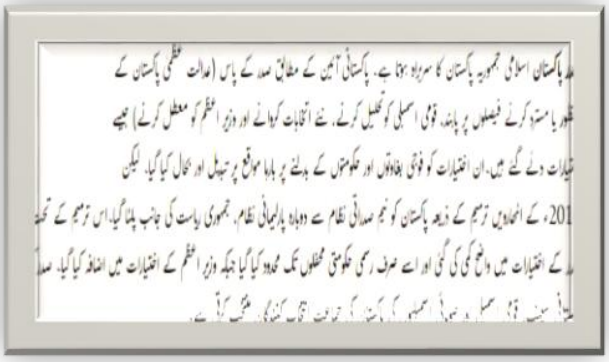


Fig. 3. Example of Urdu Dataset

Similarity Index: It shows the range of similarity values starting from least up till the largest similarity value of each dataset.

A. Jaccard Coefficient Similarity Performance Chart

We have tested 10 different pair of Urdu documents. Each pair comprises of an original and a copied document. Chart shows the performance of the mentioned technique based on

multiple percentage of similarity. We evaluated our data set of 10 different documents on our prototype. Table I represents the Jaccard performance on 10 documents.

Table I: Jaccard performance on 10 documents

Jaccard Similarity Performance Chart						
Sr.no	**Name of Doc**	**10%**	**30%**	**50%**	**70%**	**90%**
1	محمد علی جناں	10	30	50	70	90
2	ویکیپیڈیا سارفین	10	27	49	70	88
3	صدر پاکستان	10	28	48	66	87
4	جاویداختر کے گیت	7.12	26	49	69	69
5	اردو مضمون	10	36	48.5	67	87
6	بی بی سی ریڈیو	8	26.7	45	70	87
7	جنرل ایوب خان	9.23	30	50	69	77.5
8	ال انڈیامسلم لیگ	6.98	26.5	50	68.5	90
9	ویکیپیڈیا کا اجراء	8.32	26	49.5	69	86
10	پاکستان کی سیاست	10	22.5	50	68	79.5

Above chart shows good performance of Jaccard similarity coefficient on different similarity percentages. According to evaluated results maximum and minimum value of similarity for each percentage value is as following:

- For 10% similarity – max 10, min 6.98
- For 30% similarity – max 30, min 22.5
- For 50% similarity – max 50, min 45
- For 70% similarity – max 70, min 66
- For 90% similarity – max 90, min 69

B. Cosine Coefficient Similarity Performance Chart

We have tested 10 different pair of Urdu documents. Each pair comprises of an original and a copied document. Chart shows the performance of the mentioned technique based on multiple percentage of similarity. We evaluated our data set of 10 different documents on our prototype. The Table II represents the cosine performance.

Table II: Cosine performance on 10 documents

Cosine Similarity Performance Chart						
Sr.no	**Name of Doc**	**10%**	**30%**	**50%**	**70%**	**90%**
1	محمد علی جناں	8	30	50	70	90
2	ویکیپیڈیا سارفین	9	30	49.7	69	89
3	صدر پاکستان	7.7	29	49	69.2	85
4	جاویداختر کے گیت	8	27.45	43.5	67	88
5	اردو مضمون	10	28	47	70	90
6	بی بی سی ریڈیو	8.89	27	50	70	89
7	جنرل ایوب خان	9.89	30	48.44	69.04	88.4
8	ال انڈیامسلم لیگ	10	27.5	49	69.5	87
9	ویکیپیڈیا کا اجراء	7	28	47.84	70	90
10	پاکستان کی سیاست	8.54	29.5	48	66.7	82

Above chart shows impressive performance of Cosine similarity coefficient on different similarity percentages. According to evaluated results maximum and minimum value of similarity for each percentage value is as following:

- For 10% similarity – max 10, min 7.7
- For 30% similarity – max 30, min 27.45
- For 50% similarity – max 50, min 43.5
- For 70% similarity – max 70, min 66.7
- For 90% similarity – max 90, min 82

C. Dice Coefficient Similarity Performance Chart

We have tested 10 different pair of Urdu documents. Each pair comprises of an original and a copied document. Chart shows the performance of the mentioned technique based on

multiple percentage of similarity. We evaluated our data set of 10 different documents on our prototype. Table III represents the performance chart

Table III: Dice performance on 10 documents

Dice Similarity Performance Chart						
Sr.no	**Name of Doc**	**10%**	**30%**	**50%**	**70%**	**90%**
1	محمد علی جناں	6.11	38.1	46.55	72.13	90.8
2	ویکیپیڈیا سارفین	3.57	24.49	50	67.68	91.75
3	صدر پاکستان	8.13	23.83	40.02	56.89	74
4	جاویداختر کے گیت	1.26	33.09	51.11	68.93	83.87
5	اردو مضمون	7.02	19.89	31.12	54.51	89.93
6	بی بی سی ریڈیو	5.21	14.93	43.18	62.22	80
7	جنرل ایوب خان	1.69	34.1	40.21	70	88.44
8	ال انڈیامسلم لیگ	1.1	28.73	38.17	54	72
9	ویکیپیڈیا کا اجراء	7.3	25.72	49	67.1	81.19
10	پاکستان کی سیاست	3.28	32.78	47	63.9	90

Above chart shows poor performance of Dice similarity coefficient on different similarity percentages. According to evaluated results maximum and minimum value of similarity for each percentage value is as following:

- For 10% similarity – max 8.13, min 1.1
- For 30% similarity – max 38.1, min 14.93
- For 50% similarity – max 50, min 31.12
- For 70% similarity – max 70, min 54
- For 90% similarity – max 90, min 72

All three proposed techniques exhibit variations in their results for different set of documents. The Table IV and Table V show the comparison these techniques.

Table IV: Comparison chart of Jaccard, Cosine and Dice Similarity

	10% Similarity			**30% Similarity**			**50% Similarity**		
Doc no.	**Jaccard**	**Cosine**	**Dice**	**Jaccard**	**Cosine**	**Dice**	**Jaccard**	**Cosine**	**Dice**
1	10	8	6.11	30	30	38.1	50	50	46.55
2	10	9	3.57	27	30	24.49	49	49.7	50
3	10	7.7	8.13	28	29	23.83	48	49	40.02
4	7.12	8	1.26	26	27.45	33.09	49	43.5	51.11
5	10	10	7.02	30	28	19.89	48.5	47	31.12
6	8	8.89	5.21	26.7	27	14.93	45	50	43.18
7	9.23	9.89	1.69	30	30	34.1	50	48.44	40.21
8	6.98	10	1.1	26.5	27.5	28.73	50	49	38.17
9	8.32	7	7.3	26	28	25.72	49.5	47.84	49
10	10	8.54	3.28	22.5	29.5	32.78	50	48	47

Table V: Comparison chart of Jaccard, Cosine and Dice Similarity

	70% Similarity			**90% Similarity**		
Doc no.	**Jaccard**	**Cosine**	**Dice**	**Jaccard**	**Cosine**	**Dice**
1	70	70	72.13	90	90	90.8
2	70	69	67.68	88	89	91.75
3	66	69.2	56.89	87	85	74
4	69	67	68.93	69	88	83.87
5	67	70	54.51	87	90	89.93
6	70	70	62.22	87	89	80
7	69	69.04	70	77.5	88.4	88.44
8	68.5	69.5	54	90	87	72
9	69	70	67.1	86	90	81.19
10	68	66.7	63.9	79.5	82	90

V. CONCLUSION

We have separately evaluated Cosine, Jaccard and Dice similarity matching techniques along with comparing their results with each other. Our two proposed techniques Jaccard and Cosine perform better, which is vibrant in the light of above mentioned results. We have tested many different Urdu text datasets taken from multiple sources like Urdu Wikipedia, Urdu news websites, BBC Urdu, Urdu blogs, news portals & Urdu sports websites. Finally, we have presented and discussed performance results of our proposed techniques on multiple data sets of Urdu documents. The results show that our proposed techniques have the ability to precisely detect plagiarism. At the end we presented an accuracy comparison chart of all our techniques. We have also discussed their utility on national and international level in the field of education. Tool could be further optimized by incorporating additional testing parameters, such as thresholds and chunk values.

VI. FUTURE WORK

There could be some improvements made in working of our developed prototype. Following additional features could be added to enhance the usability and performance of the proposed technique.

A. Stemming

It is the process of excluding affixes from words and minimizing the words to their roots. For example: read is a root word for reading, reader or reads. Look at another example: speak is a root word for speaking, speaker or speaks.

Advantages of Stemming

After transforming a word into its stem, all stemmed words can be used in the technique like compression, reducing the size of dictionary, content & text searcher & also for analyzing the text.

- *Compression* – This technique is helpful for reducing the size of large documents. Words can be transformed into their roots which minimizes the whole size of document. Grammar and context of document will help in determining the originality of the words.
- *Reducing the dictionary size* – Traditional approach demands us to search complete word in dictionary, this may require much time, so stemmer could be used to accomplish this task in better way. Instead of searching whole word, it could search for its stem. This will eventually reduce the dictionary in size.
- *Text Search* – Text searching is one of the key components of information retrieval. It plays its vital role in document processing methods. Web search engines provides best platform for text searching. Stemmer provides wide range of search if we search of root words.
- *Text Analyzing* – Stemmer is precisely useful is mapping grammatical variations of in statistical text analysis.

B. How to Implement Stemmer

Step#1- Remove the longest suffix and the longest prefix
Step#2- Match the remaining part of the word with the verbal and noun patterns to obtain the root

C. Problems related Stemming

During derivation of root words, if any root contains weak letters like (ا ، و ، ی); word may change. Stemmer has to check correct form of weak letters. Another additional check for a stemmer is to take care of words with no roots. A stemmer should do nothing if it encounters such type of words.

Replace Synonyms: Synonym replacement is among those techniques which allow us to detect concealed plagiarism. Using synonym replacement all words are transformed into their frequent synonyms. Synonyms are extracted from UWN (Urdu Word Net).

Urdu Word Net: Urdu word net is among useful resources available, which describes multiple relations of word such as synonyms, hyponyms and anonyms. The dataset of Urdu word net contains nouns, adjectives, verbs and adverbs. Urdu word net is used for information retrieval and similarity.

Database Development and Linking: Our proposed technique could be linked with database in future for enhanced performance. Numbers of steps are required to design a database. Initial step is to study the whole system and develop list of all pre requests to develop database. Identify relationship among different sections of system. Understand the flow of information between modules. All these initial steps are necessary to clearly understand what processing is performed at each stage of the system. Next step if the analyze the requirement and working of supporting sections, such as results of one section may be input of another section. After requirement gathering next step is to design database. It is a technical phase, which requires smart skills. In this phase logical design of database is created.

Next step is to transform the logical design of database into physical design using DBMS. Perfect design depends of selection of DBMS. Next step is to implement the complete database system. Application programs are written to satisfy requirements. Maintenance is the final and ongoing step of developing and implementing a system.

D. Data Collection

This section is diverse for every system respectively, due to respective requirements of each system. Data may be collected from multiple resources such as work done in Urdu language related to multiple departments. Research work done by scholars and students as well can be added in the database repository. Additionally, all Urdu language related data available on web can be added in database.

REFERENCES

[1] M. A. Khan, A. Aleem et al, "Copy Detection in Urdu Language Documents using N-grams Model", in: Proceedings of IEEE, International Conference on Computer Networks and Information Technology, pp. 263-266, 2011.

[2] M Chong et al. "Using Natural Language Processing for Automatic Detection of Plagiarism". Proceedings of the 4th International Plagiarism Conference, UK, 2010.

[3] Ceska, Zdenek and Fox, Chris (2011) "The Influence of Text Pre-processing on Plagiarism Detection" in International Conference on Recent Advances in Natural Language Processing, Association for Computational Linguistics, 2009, pp. 55-59.

[4] Mohamed El Bachir Menai, "Detection of Plagiarism in Arabic Document", International Journal of Information Technology and Computer Science, 2012, pp. 80-89.

[5] A. Jadalla and A. Elnagar "Iqtebas 1.0: A Fingerprinting-Based Plagiarism Detection System for Arabic Text-Based Documents", in Proceddings of IEEE, 8th International conference on Computing Technology and Information Management (ICCM) , Vol. 2, No. 2, 2012.

[6] P. Shrestha and T. Solorio, "Using a Variety of n-grams for the Detection of Different Kinds of Plagiarism: Notebook for PAN at CLEF 2013", Journal of the American Society for Information Science and Technology, Jan 2013.

[7] T. Verma et al. "Tokenization and Filtering Process in Rapid Miner", International Journal of Applied Information Systems, Vol. 2, No. 7, pp. 16-18, 2014.

[8] R. Jayanthi and C. Jeevitha, "An Approach for Effective Text Pre-Processing Using Improved Porters Stemming Algorithm", International Journal of Innovative Science, Engineering & Technology, Vol. 2, No. 7, July 2015.

[9] V. Singh and B. Saini, "An Effective Pre-Processing Algorithm for Information Retrieval System", International Journal of Database Management Systems (IJDMS), Vol. 6, No. 6, December 2014.