UPD: A Plagiarism Detection Tool for Urdu Language Documents

M. Hassaan Rafiq[1], Saad Razzaq[2] and Tanzella Kehkashan[3]

[1]Department of Computer Science, Lahore Garrison University, Lahore, Pakistan
[2]Department of Computer Science & IT, University of Sargodha, Pakistan
[3]Department of Computer Science & IT, University of Lahore, Sargodha, Pakistan
[1]hassaan.rafq@lgu.edu.pk, [2]saadrazzaq@uos.edu.pk, [3]tanzella.kehkashan@yahoo.com

***Abstract*---In literature, various tools and techniques for plagiarism detection in natural language documents are developed, particularly for English language. In this article, we have proposed a tool for plagiarism detection in Urdu documents. The tool is based on the techniques of tokenization, stop word removal, chunking (trigram) and hashing (absolute hashing) of suspected documents for the detection of plagiarism. For performance evaluation, we have developed a prototype in Java and the performance of proposed tool is evaluated on five datasets of Urdu documents. Furthermore, T test is used to check the validity of our data sets.**

***Keywords*---Plagiarism Detection, Urdu Language, Tokenization, Chunking and Hashing**

I. INTRODUCTION

Plagiarism is considered the most serious scholastic misconduct [5]. It has been around since humans started producing research and art work. The easy access to digital information especially through Internet has made plagiarism an easy task for teachers, researchers and students. Plagiarism can be found both in free text (written in natural language) and in source code. Several types of plagiarism include; using other's idea or work as your own, copying of passages from a published text without citation, translation of content to another language and the use of program code without permission [4]. However, Plagiarism is not always intentional; it can be unintentional or accidental and may consist of self-stealing [5]. No best system is available till now that can help to stop or limit the misuse of the digital data.

Two main methods that are used to reduce plagiarism are plagiarism prevention methods and plagiarism detection methods [4], [6].

II. RELATED WORK

In plagiarism detection methods, three approaches are generally used to find plagiarism in suspected documents. In first approach, suspected document is compared with sets of documents and comparison is carried out on a word by word basis. In second approach, a paragraph from a suspected document is searched with a good search engine such as Google. In third approach, plagiarism in the document is found by writing style analysis. In this analysis, writing style of the author is compared with previous written documents by the same author. This technique is known as Stylometry [4].

Content-based plagiarism detection methods depend on the explicit comparisons of the document contents that are written in a specific representation. In content based, texts in the documents are analyzed on the basis of logical structure in order to find similarity in documents.

Fingerprinting [9] is the most popular technique used in content based plagiarism detection methods. In fingerprinting, similarity in documents is compared on the basis of fingerprints. A fingerprint is a set of integers build by hashing subsets of a document to show the key content of the document. Techniques that are used for the generation of fingerprints are mostly based on *n*-grams. In text categorization, n-gram model was first used on the statistical information collected from the usage of characters sequence [2]. Fingerprints are selected according to various schemes that include 0 mod *p* hash, *i*th hash and Winnowing [9].

Khan *et al.* [3] presented a copy detection technique for Urdu documents that is based on the N-gram model. They have used trigram model for text representation. Developed model finds the plagiarism in two Urdu documents. Resemblance measure *R* is used to calculate the probability of matching text in two documents. Taking the work done in [3] as a starting point, we have developed a content-based plagiarism detection tool, UPD (Urdu Plagiarism Detection), for Urdu documents.

III. PROPOESED PLAGIARISM DETECTION TOOL

When developing a plagiarism detection tool for natural languages, following properties must be satisfied [9].

- Insensitivity to capitalization, extra whitespace and punctuation.
- Insensitivity to small matches (a match should be big enough to suggest plagiarism).
- Insensitive towards permutations found in the contents of the document.

UPD satisfies the mentioned three properties. Preprocessing process that includes tokenization and stop word removal takes care of the first property. Second property is satisfied if

value of chunk parameter *n* is large enough to avoid common idioms that are found in Urdu language. Third property is demonstrated later in Section 4 by the performance of the UPD. Fig. 1 shows the main components of UPD which are explained in detail next.

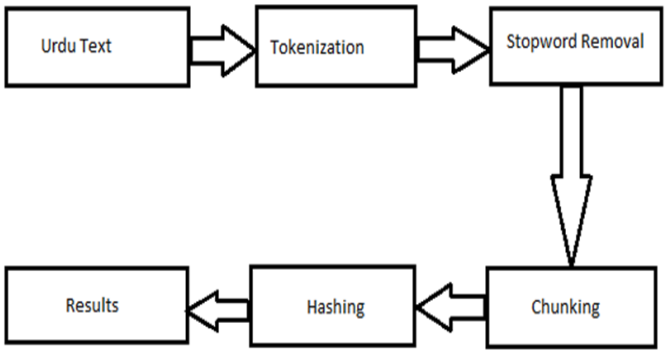

Fig. 1: Main components of UPD

A) Tokenization and Stop Word Removal

The first step in the content based plagiarism detection method is the preprocessing phase where a text is broken into tokens (pieces) and stop words are removed. Preprocessing step is usually done to transform a text into representation that is more suitable for the process of plagiarism detection. In Urdu language, stop words are those set of words that have no integral useful meaning or information. If stop words are not removed from the document, it creates problems in the identification of key words and concepts from textual bases. Removal of stop words from documents also reduces false positives. Therefore, in any text categorization technique, it is important to remove the stops words. Fig. 2 shows an example of preprocessing stepson a sentence from Urdu document.

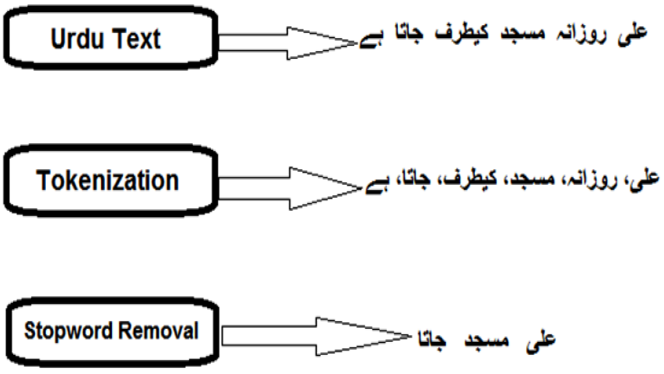

Fig. 2: Preprocessing steps on Urdu sentence

B) Chunking

Chunking is a technique in which a document is divided into smaller pieces called chunks [7]. A sentence or a word from a document can be considered as a chunk unit. Two types of chunking are sentence-based chunking and word-based chunking. In sentence-based chunking, the document is partitioned into different chunks on the basis of chunk parameter *n*, which associates each sequence that contain *n* sentences into a chunk. Sentence-based chunking is explained with a simple example. Suppose the value of $n = 3$ and the document contains the following sentences: $s_1s_2s_3s_4s_5$. The chunks obtained with $n = 3$ are $s_1s_2s_3$, $s_2s_3s_4$ and $s_3s_4s_5$. Same is also the case for word-based chunking. For word-based chinking, suppose $n = 3$ and the document contains the words: $w_1w_2w_3w_4w_5$. For $n = 3$, obtained chunks are: $w_1w_2w_3$, $w_2w_3w_4$ and$w_3w_4w_5$. On comparison, word-based chunking offers high rate of similarity detection than sentence-based chunking [6].

UPD is based on word-based chunking method and we have used the trigram model which means that each token contains three words. The hash values of these tokens are computed next with a hash function.

C) Hashing

In hashing, it is critical to choose a hash function that will not cause any collisions due to mapping of different chunks to the same hash value. For example, it is easy to develop a hash function that will map each chunk to the sum of the integer values of characters in the chunk. However, this is not a right hash function because the chunks that have the same characters in different order will be represented by same hash values, which will cause collisions. In order to avoid collisions, we have used the absolute hash function. In absolute hash function, value of each letter in the word is first calculated by multiplying it with integer that represents its location. Then we add all the values of the letters to get the value of the single word. Each words value is then summed up to get the hash value for that chunk. The process of hashing is shown in Fig. 3.

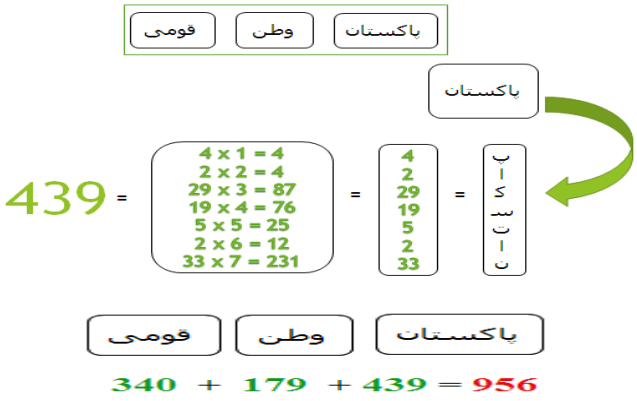

Fig. 3: Example of hashing

IV. EXPERIMENTAL EVALUATION

We developed a prototype of UPD in Java and evaluated its applicability and performance on data test set of 150 Urdu documents. We have calculated the plagiarism percentage between documents by resemblance measure. The chunk parameter $n = 3$. The resemblance measures (R) [1] can be calculated by formula:

$$R = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|}$$

Where S(A) in the formula represents the set of trigram from document A and S(B) represents the set of trigram from document B. Matched trigrams in two documents are calculated as:

$$M = |S(A) \cap S(B)|$$

Finally, we calculate the total number of trigram by formula: $N = |S(A) \cup S(B)|$

From original documents, 5 data sets are generated as follows:

Data Sat – 0 % Similarity: In first dataset, we have 30 different pairs of Urdu documents. Each pair contains documents that are 0 % similar. The results obtained are shown in Fig. 4.

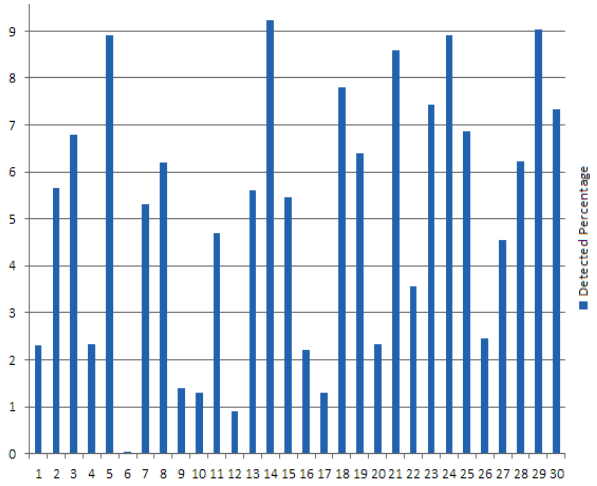

Fig. 4: UPD performance on first data set

UPD showed reasonable performance on first data set. Maximum detected similarity was found in pair 14 while the minimum similarity was found in pair 6 documents.

Data Set - 30 % Similarity: In this dataset, we have 30 different pairs of Urdu documents. Each pair contains documents that have low similarity between them (30 % similarity). Fig. 5 shows the results.

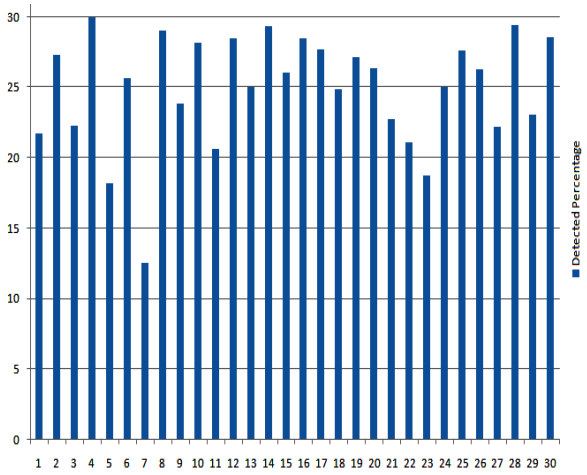

Fig. 5: UPD performance on second data set

Data Set - 50 % Similarity: In this dataset, we have 30 different pairs of Urdu documents. Each pair contains documents that are half similar to each other meaning that both of the documents in a pair have 50 % similarity. The results obtained are shown in Fig. 6.

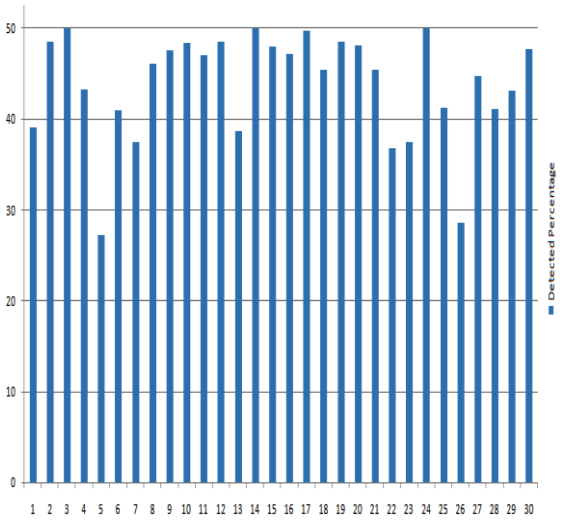

Fig. 6: UPD performance on third data set

Data Set - 70 % Similarity: In this dataset, we have 30 different pairs of Urdu documents. Each pair contains documents that are 70 % similar to each other. Obtained results are shown in Fig. 7.

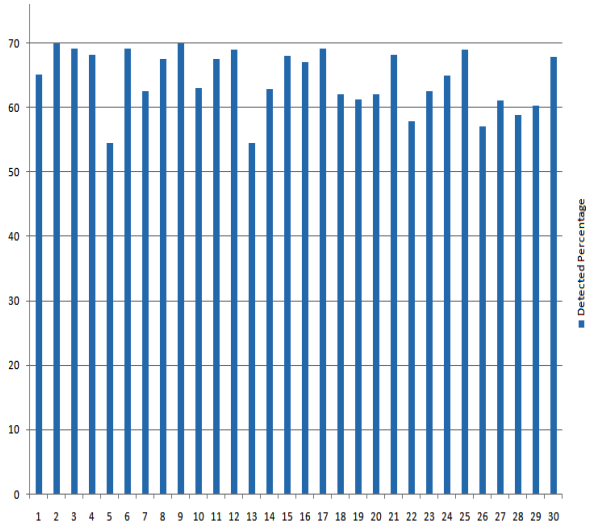

Fig. 7: UPD performance on fourth data set

Data Set -1000 % Similarity: In this dataset, we have 30 different pairs of Urdu documents. Each pair documents are exactly same to each other. UPD detected 100 % similarity in all the pairs of this data set.

The average similarity detected by UPD on all five data sets is shown in Fig 8. From Fig. 8, it is clear that our developed UPD prototype performs well in detecting plagiarism in Urdu documents.

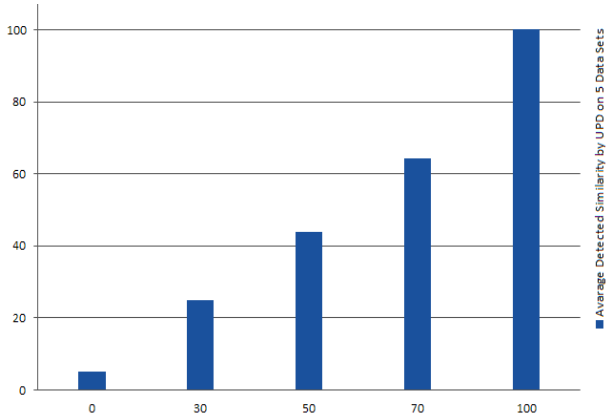

Fig. 8: Average performance of UPD on all data sets

In order to check the validity of our data sets, we performed the T test [8] in SPSS tool. In the test, the hypothesis was that our developed prototype is giving the significant results. We set the critical value to 0.05. After performing test, we get the probability (p) value of 0.01, which is less than our critical value. The p-value indicates that our hypothesis was true and we can say that our developed prototype is giving significant results. The summary of the T test result is shown in Table I.

Table I: T-test performance sheet

Data sets	t-value	df	Lower	Upper	Decision
30 %	26.937	28	12.5	30	Significant
50 %	38.93	28	27.27	50	Significant
70 %	30.785	28	62.86	70	Significant

V. CONCLUSION

In this article, we have proposed a plagiarism detection tool named UPD for Urdu documents. For similarity detection we used resemblance measure *R*. In order to show the effectiveness of UPD, we carried out number of experiments where performance of UPD is evaluated on large sets of Urdu documents. The results show that UPD has the capability to precisely detect plagiarism in Urdu documents. Finally, we have performed the T test to validate the significance of data sets that we have used in our experiments. T test gives more confidence to obtained results and the performance of UPD. Our tool can be linked with web or a database in future to facilitate educational institutes especially.

REFERENCES

[1]. Barr'on-Cedeno, A, and P. Rosso, (2009). On Automatic Plagiarism Detection Based on n-grams Comparison. *Advances in information Retreival*, pp. 696-700.

[2]. Canvar, W. B, and J. M. Trenkle, (1994). N-gram-Based Text Categorization, Ann Arbor, MI 48113(2): 161-175.

[3]. Khan, M. A., A. Aleem., A. Wahab, and M. N. Khan. (2011). 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.

[4]. Lukashenko, V., Graudina, and J. Grundespenkis, (2007). Computer-based plagiarism detection methods and tools: an overview [C]. *In: Proceedings of the ACM International Conference on Computer Systems and Technologies*, Bulgaria, pp. 1-6.

[5]. Maurer, H. A., F. Kappe, and Bilal Zaka, (2006).Plagiarism-A Survey.*Journal of Universal Computer Science*, 12(8): 1050-1084.

[6]. Menai, M. E., (2012). Detection of Plagiarism in Arabic Documents. *International Journal of Information Technology and Computer Science*, 4(10): 80-89.

[7]. Pataki, M., (2003). Plagiarism detection and document chunking methods. *In: Proceedings of the 12th International WWW Conference*, Budapest, Hungaria, May 20-24.

[8]. Prel, J. B., B. Röhrig., G. Hommel, and M. Blettner, (2010). Choosing Statistical Tests. *Deutsches Arzteblatt Internationa*,107(19): 343-348.

[9]. Schleimer, S., D. S.Wilkerson, and A. Aiken, (2003). Winnowing: local algorithms for document fingerprinting. *In: Proceedings of the ACM SIGMOD International Conference on Management of Data*, San Diego, California, USA, June 9-12, pp. 76-85.