Journal of AI and Data Mining
Vol 4, No 2, 2016, 125-133

doi: 10.5829/idosi.JAIDM.2016.04.02.01

Plagiarism checker for Persian (PCP) texts using hash-based tree representative fingerprinting

Sh. Rafieian[1*] and A. Braani Dastjerdi[2]

1. Computer Engineering Department, Sheikh Bahaii University, Isfahan, Iran.
2. Computer Engineering Department, University of Isfahan, Isfahan, Iran.

Received 21 October 2015; Accepted 24 January 2016
**Corresponding author: shima.rafieian@gmail.com (Sh. Rafieian).*

Abstract
With due respect to the authors' rights, plagiarism detection is one of the critical problems in the field of text-mining, in which many researchers are interested. This issue has been considered as a serious one in high academic institutions. There exist language-free tools that do not yield any reliable results since the special features of every language are ignored in them. Considering the paucity of works in the field of Persian language due to the lack of reliable plagiarism checkers in Persian, there is a need for a method to improve the accuracy of detecting plagiarized Persian phrases. An attempt is made in this work to present the PCP solution. This solution is a combinational method, in which, in addition to the meaning and stem of words, synonyms and pluralization are dealt with by applying the document tree representation based on manner fingerprinting the text in the 3-grams words. The grams obtained are eliminated from the text, hashed through the BKDR hash function, and stored as the fingerprint of a document in fingerprints of the reference document repositories in order to check the suspicious documents. The proposed PCP method here is evaluated by eight experiments on seven different sets, which include the suspicions documents and the reference document from the Hamshahri newspaper website. The results obtained indicate that the accuracy of this proposed method in detecting similar texts, in comparison with the "Winnowing" localized method, has a 21.15% average improvement. The accuracy of the PCP method in detecting the similarities, in comparison with the language-free tool, reveals a 31.65% average improvement.

Keywords: *Text-Mining, Natural Language Processing, Plagiarism Detection, External Plagiarism Detection, Persian Language.*

1. Introduction

Nowadays plagiarism has become a cancer cell in the literary world. This important global issue is considered as a serious crisis for high academic institutions even in freelance writing. Accessibility of different digital documents in Worldwide Web makes it easy for the swindlers to copy explicit subjects from students and academicians by allowing them to be promoted to high academic levels or grades in life without any required scientific background [1].
Plagiarism may include:

- Replacing the original author's name
- Copying ideas, phrases, concepts, research proposals, articles, reports, computer program designs, websites, and the internet and other electronic resources without citing the author's name
- Lack of citation regarding quotation
- False referencing or referencing the non-existing resources
- Translation plagiarism, where the translated text is submitted without reference to the original text
- Artistic plagiarism, where different media including images and videos are used for other works without (a) proper reference(s) to the resource(s) [2,4].

There are two major methods that can be used to reduce literary pirating: plagiarism detection and plagiarism prevention [3,4]. An attempt is made in this work to adapt the detection method.

The path and status of this work are presented in figure 1, with their hierarchical sequence in gray boxes. According to this tree diagram, plagiarism detection methods include manual methods and software tools that are simple to be implemented, and can be applied in plagiarism [3].

Software plagiarism detection is categorized based on text homogeneity regarding monolingual plagiarism detection and cross-lingual plagiarism detection [2].

Detecting plagiarism in monolingual environments refers to a homogeneous and congruent environment like English to English, and nearly all systems that are developed to detect it and are divided into the inherent and external types [2,4].

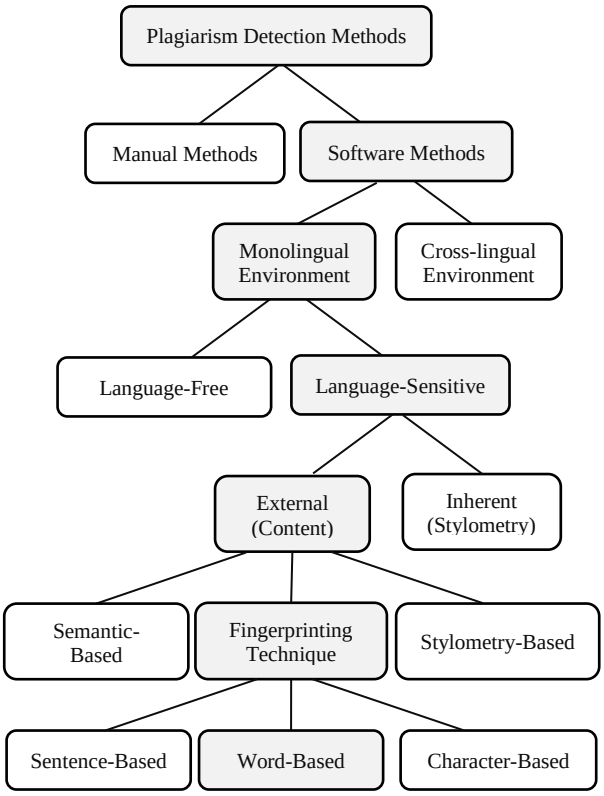

Figure 1. Methods of plagiarism detection.

Detecting cross-lingual plagiarism refers to detecting texts that encompass multi-languages like English and Arabic. In this method, the document recovery process is similar to the suspicious documents in a cross-lingual environment [5].

In detecting inherent plagiarism, named the stylometry-based method as well, there is no reference document, and just the suspicious document is controlled [2]. The objective of inherent plagiarism detection is to identify the potential pirate(s) with analyzing changes in writing style [6].

For detecting external plagiarism, named the content-based method, a suspicious document is compared with a number of documents, and the text contents are analyzed based on the logical structure and detection of similarities among texts [2]. In this method, a text investigation is made through textual features including removing stop words from the text [7].

The common techniques that act based on the content-based method rely on the explicit comparison of the document contents. Most detection methods use stop word deletions [3]. The objective of this work is to improve the accuracy of detecting the similarities among the pirated phrases in Persian texts through the stem of current words and document tree representation, and applying the fingerprinting technique according to the word-based 3-grams.

The innovation aspects of this proposed method consist of preprocessing operation(s) in more accuracy in comparison with the previous works, and replacement of pluralization or broken words.

Applying the document tree representing and its fingerprinting introduces a new tree-nodes with a key volume that contains the hash value of its children trio. Therefore, in copy detection, only branches with the same hash values are considered, which prevent excessive search.

The rest of this article is organized as follows: A literature review is presented in section 2. The solution and operation used for pre-processing the text and document tree representation, text fingerprinting, and detecting the suspected phrases are presented in section 3. The presented combinational method (PCP) is discussed in section 4, and, finally, the conclusion is presented in section 5.

2. Literature review

Using language-free plagiarism detection tools are inefficient on texts like Persian and Arabic, and the outputs of these tools are imprecise and unreliable because they do not consider their special features and structural complexities [3]. Hence, the language-sensitive tools should be used. Despite the endeavors in this field in the recent years, no updated and efficient tool has been presented for Persian texts.

ZiHayat and Basiri have presented a tool that makes the detection of copying scale of phrases possible in the Persian electronic documents through a native-user interface based on the grand algorithm "Winnowing" [8]. The average accuracy of this total is 64%, which is relatively low. It is possible to adapt more updated algorithms for document categorization and

natural language processing in order to improve the accuracy of this system [9].

Kamran et al. have also presented a tool for detecting plagiarism in Persian documents using "Simhash" algorithm which, despite its low accuracy, is fast in detecting pirates in a large collection of texts. There exist 300 reference articles and 25 suspicious articles as the inputs of the system, which are used to detect phrase similarities of the word-based grams and "Simhash" and "Shingling" algorithms. The developers have concluded that in large sets of Persian documents, using the "Simhash" algorithm (despite its low accuracy) is a more proper method [10].

Mahmoodi and MahmmodiVarNamkhasti has proposed another tool for plagiarism detection; a precise tool for detecting plagiarism in short paragraphs [1]. It is impossible to detect plagiarism in documents with multiple paragraphs because the inputs of this tool are both a suspicious document and a reference document, where each one of them includes one paragraph by itself. Assuming the high level of accuracy in the plagiarism detection for short paragraphs, it is not possible to detect plagiarism in multiple paragraphs, and if either of these documents contain more than one paragraph considered as an input, the results would be of low accuracy, and unreliable.

Mahdavi et al. have adapted the vector space model to detect external plagiarism in Persian texts. In their article, 41 reference documents and 84 suspicious documents were created by the developers, and using the vector space model and cosine similarity among them, more accurate document processings s were selected as the candidates. Next, the similarity coefficient shows the overlapping features of 3-grams comprising each document, where the probable similarities are discovered. For every feature, the vector of a document requires both more memory and a long time in the processing of finding similar documents. Therefore, the size and number of features of this vector depend on the length and expression of the documents [11].

Rakian at al. have used the new method of a fuzzy algorithm to consider the different levels of a hierarchical text and use the synonyms necessary in determining the degree of similarity between two sentences in Persian texts, and hence, the external plagiarism detection in Persian texts. Here, 1,000 reference documents and 400 suspicious documents were established, where the structural change in sentences and then being rewritten are recognizable. In order to select the candidate documents related to the keywords of the text offer recovery and divide their constituent sentences, the potential similarities are detected by the fuzzy methods [12]. An increase in the sentence divisions can slow down the processing time and accelerate the memory consumption.

3. Proposed combinational method (PCP)

Implementing this combinational method includes text preprocessing, document tree representation, text fingerprinting, and copy detection (see Figures 2).

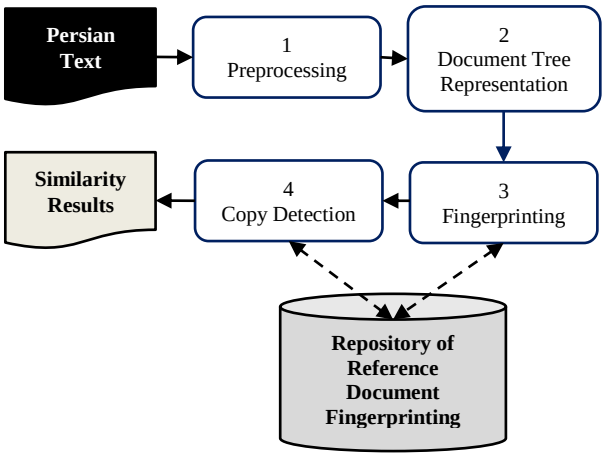

Figure 2. Steps of detecting similarities in PCP method.

In this study, the fingerprints were based upon 3-grams of the text created by different levels of the document tree representation. This representation can be obtained by traversing the bottom-up tree. The final fingerprint of a document created by the hash of the paragraph level will be less than the volume of the hash made at the level of the 3-grams words. The fingerprint of a document is compressed and improves the fault memory-consumption presented in [3, 7] and similar works with respect to another language. Since in their fingerprint idea, the hashes in the level of words were copied into their father, they created a high volume of hash word levels in the fingerprint of a document. Moreover, the fingerprint idea in the PCP method causes a difference in the similarity detection approach towards the proposed method in [3,7].

3.1. Text preprocessing measure

Text preprocessing is run in order to clean and delete useless information from the text, causing a rise in the accuracy and a reduction in the time required for a possible similarity detection.

According to figure 3, this measure includes the following steps:

1. Text segmentation: here, the text is separated into its constituent paragraphs.

2. Sentence tokenization: here, the constituent sentences of a paragraph are separated by the punctuation marks "?, ! , .", and the excess spaces in each paragraph are removed and replaced by one empty space; therefore, it is assumed that all sentences are separated with an empty space.

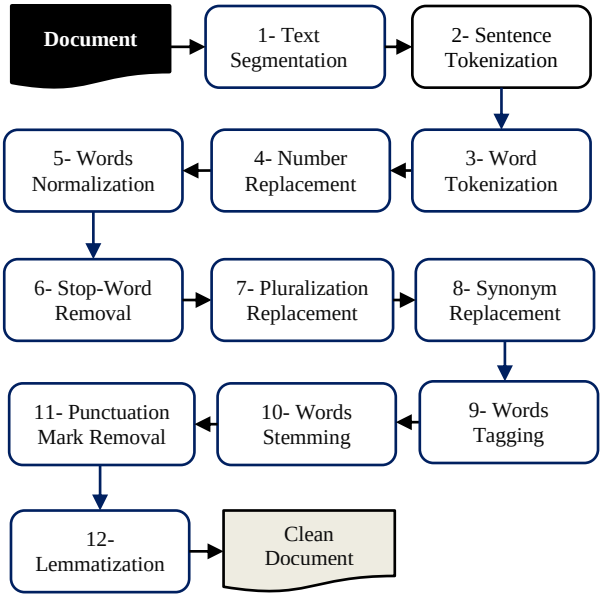

Figure 3. Text preprocessing measure.

3. Word tokenization: here, for every specified sentence in the previous step, word ranges and punctuation marks are determined in a sense that each sentence would be broken into its constituent words.

4. Number replacement: here, the number character is replaced by the "#" sign, which makes finding similarity among the number in the text independent.

5. Words normalization: here, operations like removing three points from the text, putting half-space between the prefixes and postfixes including "می،نمی،تر،ترین،ها", and finally, replacing the excessive spaces with one space are applied to the normalized words.

6. Stop word removal: words like relation words including "و، از، را، ولی، اما، به" are among the frequent words in the Persian language, which are applied to all texts, and must be ignored in order to assess the similarity in texts because they have no special meaning weight.

7. Fragmented pluralization or broken word replacements: in the Persian language, there are words that have the same stem but their pluralization is irregular, like the word "اخبار", which is a pluralization summation of the word "خبر". It is worth mentioning that this step is being presented for the first time in the Persian language.

The input function is a word processing of the document (see Figure 4). If this word is pluralized and replaced by its singular term, then the homogenization of this class of words is accomplished. This step requires pluralized lexicon in the Persian language. For this purpose, the Persian Gate 6.0 plug-in, which is applied in natural language processing in [13], is applied.

```
Pluralization Replacing Algorithm
1.  Input: The word of Document
2.  Output: The Singular_word of Clean Document
3.  Begin
4.     While (! Pluralization _Lexicon.EndOfFile)
5.        If (word = = Pluralization _Token)
6.        {
7.           Singular_word = Singular_Token
8.           Break
9.        }
10. End
```

Figure 4. Pseudo-code of pluralization replacement.

8. Synonym replacement: in the Persian language, there are words that have the same meaning but different stems such as the word "پند" that has the synonyms "اندرز، موعظه، رهنمون، عبرت، نصیحت، وعظ". If there are such words in a sentence, all of them are replaced with their stems, The word "پند" is followed by homogenization of this kind of words in the text.

The input function is a word-processing of the document (see Figures 5). If this word is in a series of synonymous words, replaced by their root words, this class of words is homogenized. This step is required to be lexicon synonyms in the Persian language. Here, a comprehensive synonymous and antonyms lexicon in the Persian language has used name as Raghoumi version [14].

```
Synonym replacement Algotithm
1.  Input: The word of Document
2.  Output: The Root_word of Clean Document
3.  Begin
4.     While (! Synonym_Lexicon.EndOfFile)
5.        If (word = = Syn_Token)
6.        {
7.           Root _word = Root_Token
8.           Break
9.        }
10. End
```

Figure 5. Pseudo-code of synonym replacement.

9. Part-of-speech (POS) tagging: here, the reminded basic words of the text are tagged, and their types are specified on grammatical parts like the noun, verb, adverb, adjective, and punctuation marks [15]. This step is impressive in determining the stem of the words.

10. Stemming: here, the words are stemmed based on a specified tag given to them in the previous

step followed by removal of prefixes, postfixes, and infixes from the word, respectively.
In the manner, different derivative and inflectional states of words in similarity detection are not affected. For example, the words "می‌رود", "رفته بود" are verbs with the stems "رفت" in past and "رو" in the present. This process becomes possible through the trained model in NHazm [16], which is a tool for processing Persian natural language in Visual Studio environment.
11. Punctuation removal: in this step, ignore all the writing signs and available punctuation marks in the text.
12. Lemmatization: in the final step, words are replaced with stems in their dictionaries. This step proceeds with each word tag and its stem.

3.2 Fingerprinting

A document tree representation is applied in order to fingerprint a text. The PCP approach is to determine the fingerprint of the document at words level in the text, which is divided into 3-grams, and after applying the hash function on them, a fingerprint of the document is generated in the 3-grams words. In the next step, to produce a fingerprint of the document in sentences, the generated hashes in the 3-grams are broken into the next 3-grams, as well, where the hash function would be applied on them. Finally, to create the final fingerprint of the document (at the paragraph level), the hashes generated in sentences are broken into the 3-grams again, and then the hash function is applied to them. The final fingerprint of a document created based on tree representation and applied the hash function would generate the hashed 3-grams at each level, whose volume is smaller than the approach presented in [3,7].
As shown in figure 6, the stem consists of the tree basic document, the second level consists of all refined text paragraphs, and the third level of the tree encompasses the sentences of the paragraph.

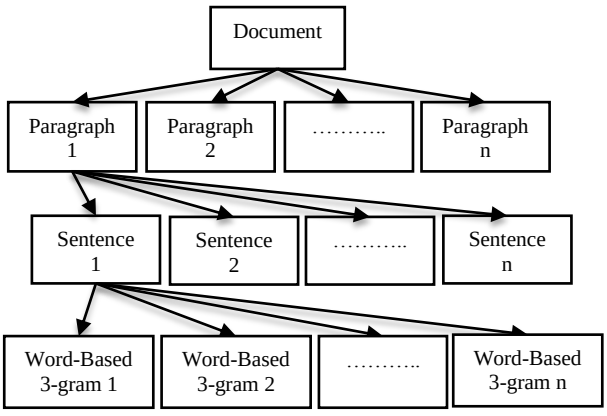

Figure 6. Document tree representation.

Then sentences are divided into word-based 3-grams, and using a proper hash function, they are converted into a number. In this manner, the processing speed is increased in the copy detection operation.
In figure 7, there is a tree representation of the single sentence paragraph " امروز هوای اصفهان ابری و بارانی است.".

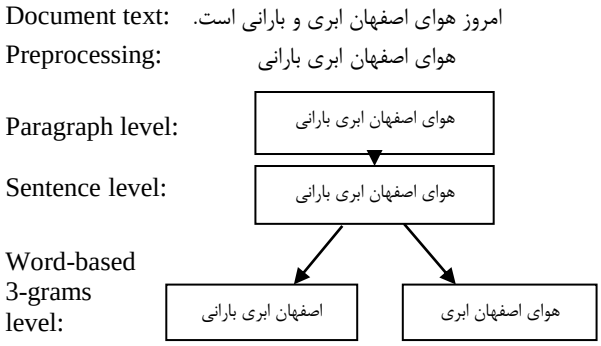

Figure 7. An example of a document tree representation.

It is important to select a hash function that minimizes the collisions due to mapping different chunks to the same hash [6, 10]. In this implementation, the BKDR hash function is used. This function is the sum of each character's multiplication in a certain value named "seed" that usually has the value of 31. The seed value must be an odd number because odd numbers are unique, and multiplication of a number in an odd number creates a unique hash value [6, 10].
The steps for the above example of fingerprinting are shown in figure 8. The fingerprint of this single sentence paragraph is 25319069.

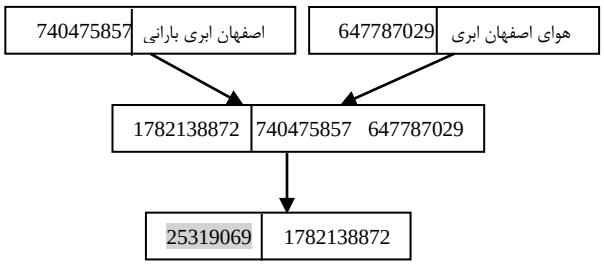

Figure 8. A fingerprinting example.

According to figure 8, after breaking all the words contained in sentences into 3-grams, it is time to hash operations at sentence-level. Through this procedure, the hashes obtained from words-based 3-grams are broken into 3-grams in tree sentence-level, and a hash operation is run on them.
In the final step, the hashed 3-grams will be converted from sentence-level into paragraph-level 3-grams. Therefore, the document fingerprints obtained contain paragraph-level hashes of the document.

3.3 Copy detection

The main objective of the document tree representation is time-saving during similarity investigation and preventing excessive comparisons.

In the PCP method, the similar detection approach is based upon the membership fingerprint in each level of suspicious document fingerprint and the corresponding level of reference document fingerprint. For example, if a hash value of fingerprint (at paragraph level) in the suspicious document exists in the hash fingerprint collection (at paragraph level) of the reference document, each one of the 3-grams that here created this hash (each one of the three hash of manufacturer this hash) at the sentence levels is checked separately. Similarly, if existed similarity in sentences, the hashes of the suspicious document at sentences level checked more precisely at the 3-grams words. In other words, the generated hashes of sentences in the 3-grams words level are examined separately. Therefore, if there exists a similarity, the 3-grams words in the tree leaf are displayed to the user for final decisions.

According to the pseudo-code in figure 9, a tree is surveyed by a top-down traverse, and the fingerprints of two texts in the document level are evaluated. Due to the lack of injective hash function and generation of equal hashes for different phrases with these, in order to ensure the final result, the star-tagged parts are added to the code that can generally be deleted from the algorithm.

```
Copy Detection Algorithm
1.    Input: Fingerprinting of Doc suspect,
      Fingerprinting of Doc source,
2.    Output: Similarity
3.    Begin
4.      For each hash_ paragraph _suspect
5.        If (suspect_ paragraph  in source_ paragraph)
6.          For each hash_ sentence _suspect
7.    *       If (suspect_ sentence  in source_ sentence)
8.                For each hash_ words _suspect
9.    *             If (suspect_ words  in source_ words)
10.                       Similarity = True
11.   *             Else
12.                       Similarity = False
13.   *       Else
14.               Similarity = False
15.       Else
16.         Similarity = False
17.   End
```

Figure 9. Pseudo-code of copy detection.

1. The fingerprints of the reference document and the suspicious document are considered as the algorithm inputs.
2. If there is any similarity/dissimilarity in each one of the steps, the algorithm output or "Similarity" variable is determined by “True” or “False”.
3. Similarity detection operation begins.
4. Following steps will continue for all the current document paragraph-level hashes.
5. If the following paragraph-level hashes of the suspicious document are the subsets of paragraph-level hashes of the reference document, evaluate the comparison process in sentence-level.
6. For each hash in sentence-level of the suspicious document, the comparison process continues at the level of current word.
7. If sentence-level hashes of the suspicious documents are the subsets of the sentence-level hashes of the reference document, then the comparison process continues in their word-level.
8. For each hash at word-level of the suspicious document, the comparison process continues at their 3-grams level.
9. If the 3-grams level hashes of the suspicious document are the subsets of the 3-grams level hashes of reference document,
10. Possible similarity is detected.
11, 12. Otherwise, the comparison process continues at the sentence-level hashes of the suspicious document.
13, 14. According to line 9, if the sentence-level hashes of the suspicious document are not the subsets of sentence-level hashes of the reference document, then the comparison operation continues at the paragraph-level hashes of the suspicious document.
15, 16. According to line 7, if the paragraph-level hashes of the suspicious document are not a subset of paragraph-level hashes of the reference document, then the comparison operation stops.
17. Operation of similarity detection ends.

4. Method evaluation

The implementation is run using the C# programming language, where the features, functions, and classes are used.

The evaluation process proceeds once with similarity parameters and their comparison with the native algorithm in "Winnowing" [9], and once, by using the Duplicate Content Checker tool, which implements text similarity detection, and is placed in the language-free categories [17].

4.1. Datasets

Evaluation of the performance of the proposed PCP method requires a standard textual dataset. Therefore, seven sets of texts consisting of one suspicious and one reference text in each are collected from the standard dataset of Persian language and Hamshahri newspaper sources [18].

The specification of these texts is tabulated in table 1.

Table 1. Randomly-created document sets.

Document Number	Word counts	Construction type
1	119	Random
2	117	Document 1
3	537	Random
4	907	Random + Document 3
34	575	Document 3 + Document 4
5	827	Random
6	497	Random + Document 3 + Document 5
7	5392	Document 1 + Document 3 + Document 4 + Random
8	3003	Document 7

4.2. Parameters

Evaluation through Recall, Precision, and F-measure scales are the three important measures in the efficiency of the plagiarism detection algorithms in addition to Jaccard Similarity Coefficient (1) to (4), and all of these algorithms are calculated as follow [10]:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{F-measure} = 2 . \frac{\text{Recall.Precision}}{\text{Recall+Precision}} \quad (3)$$

$$\text{Jaccard Similarity Coefficient} = \frac{TP}{TP+FP+TN} \quad (4)$$

where, TP is the number of cases that are detected True as a copy, FN is the number of cases that are detected False as the original, and FP is the number of cases that is detected False as a copy [10].

3.4. Evaluation results

With respect to table 2, this proposed combinational method is examined through seven random datasets created by documents, tabulated in table 1 with eight tests.

To illustrate the improved accuracy in similarity detection in Persian phrases, the similarity rate of each pair in the tested document is assessed by the "Winnowing" algorithm and PCP method, and hence, the desired parameters are provided.

Table 2. Suspicious and reference created document sets.

Document Set	Suspicious Document	Reference document	Used in
I	Doc2	Doc1	Test 1 , 2
II	Doc34	Doc3	Test 3
III	Doc34	Doc4	Test 4
IV	Doc6	Doc3	Test 5
V	Doc6	Doc5	Test 6
VI	Doc5	Doc6	Test 7
VII	Doc8	Doc7	Test 8

Then to compare the proposed combinational method with the language-free tools, the similarity rate of any suspicious and reference document acquired using the Duplicate Content Checker tool are calculated. The results obtained for these tests are tabulated in table 3.

The results shown in figure 10 show that by using this combinational method, where the meaning of each word and replacement of proper pluralization and synonyms are of concern, the average values for Recall, Precision, and F-measure are improved in the order of 19.26%, 23.61% and 20.58%, respectively, and according to the accuracy in the plagiarism detection evaluated by these parameters, the improved accuracy average is 21.15%. The similarity coefficient improvement of the two texts by 21.13% has gained more safety factor. Since the PCP method is used as a combination of word stems and the tree representation of documents, the effectiveness of all the hashes generated in the fingerprint of any document, which can increase accuracy in the similar detection process, is improved.

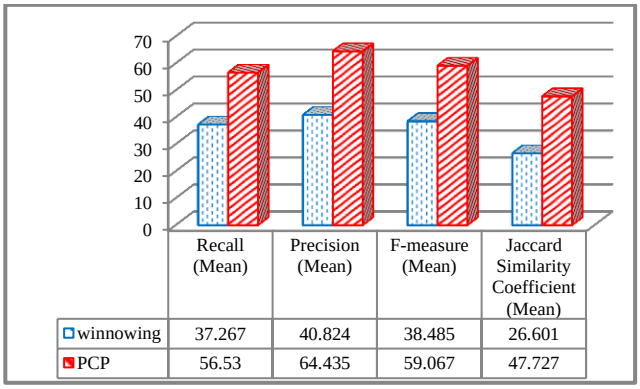

Figure 10. Comparison of PCP method and localized Winnowing algorithm.

This similarity scale in comparison with the similarity that is obtained from language-free tools is reliable by 31.65% (see Figures 11).

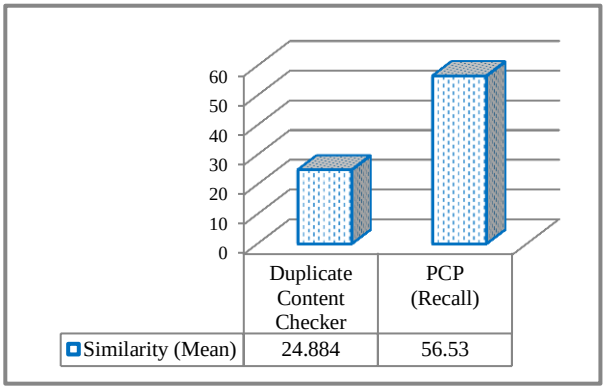

Figure 11. Comparison of PCP method and Duplicate Content Checker tool.

Since language-free tools do not consider the appearance of words and the specific characteristics of the Persian language in the text, they are not accurate enough in detecting similarity or dissimilarity in the Persian texts. To the contrary, this proposed method makes it possible to obtain more accurate results in relation to the language-free method.

Since there exists a direct relation between the document-length and the time-consumed, and since accurate preprocessing and tree representations are applied in this method, naturally, the time-consumption is increased, and this might be considered as a drawback, something that no new method can be without.

Table 3. Evaluating the proposed combinational method using eight tests.

		Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Suspicious Document		Doc 2	Doc2	Doc34	Doc34	Doc6	Doc6	Doc5	Doc8
Reference document		Doc 1	Doc1	Doc3	Doc4	Doc3	Doc5	Doc6	Doc7
Recall	**Winnowing**	76.67	53.55	38.64	27.56	44.7	9.74	15.45	32.06
	PCP	100	100	52.26	36.98	57.14	22.53	36.93	46.4
Precision	**Winnowing**	76.67	55.17	38.93	47.33	47.9	15.45	9.74	55.41
	PCP	100	100	50.73	61.68	63.07	36.93	22.53	80.54
F-measure	**Winnowing**	76.67	54.24	38.78	34.83	46.27	11.59	11.95	40.61
	PCP	100	100	51.48	46.24	59.96	27.99	27.99	58.88
Jaccard coefficient Similarity	**Winnowing**	62.16	37.21	24.06	21.09	30.1	6.35	6.35	25.48
	PCP	100	100	34.66	30.07	42.82	16.27	16.27	41.72
Duplicate Content Checker		56.56	36.59	23.54	20.64	29.63	4.9	4.9	22.31
PCP (Recall)		100	100	52.26	36.98	57.14	22.53	36.93	46.4

With respect to the nature of fingerprinting, this needs a repository for reference. The bigger the repository, the bigger is the memory storage. This issue, on its own, can be considered as a drawback.

In addition, due to the nature of the fingerprinting technique, the restructuring of the text and the changes thereof, word ordering is not possible.

5. Conclusion and future work

A combinational method based on the semantic of current words in text and tree representation of the document, accompanied with the fingerprinting technique according to words-based 3-grams and improvements made in similarity detection accuracy of plagiarized phrases in Persian texts is proposed in the PCP method.

The results obtained indicate that this combinational method improved the similarity coefficient of two texts by 21.13% because the word meanings and replacing proper pluralization and synonyms are of concern.

The calculated similarity scale has an improved rate of 31.65%, and is more reliable, in comparison with the similarity obtained from the language-free tool. This indicates the lack of accuracy in the language-free tools in relation to the language-sensitive methods, especially the proposed combinational method.

The data-mining algorithms in categorizing documents automatically are among the proposals to improve this method, which prevents excess-comparison between texts with different themes.

References

[1] Mahmoodi, M. & Mahmoodi Varnamkhasti, M. (2014). Design a Persian Automated Plagiarism Detector (AMZPPD). International Journal of Engineering Trends and Technology (IJETT), vol. 8, no. 8, pp. 465-467.

[2] Alzahrani, S. M., Salim, N. & Abraham, A. (2012). Understanding Plagiarism linguistic patterns, textual features, and detection methods. IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 42, no. 2, pp. 133-149.

[3] Menai, M. E. B. (2012). Detection of Plagiarism in Arabic Documents. International Journal of Information Technology and Computer Science, vol. 4. no.10, pp. 80-89.

[4] Bin-Habtoor, A. S. & Zaher, M. A. (2012). A Survey on Plagiarism Detection Systems. International Journal of Computer Theory and Engineering, vol. 4, no. 2, pp. 185-188.

[5] Ceska, Z. (2008). Plagiarism Detection Based on Singular Value Decomposition. 5th International Conference on Advances in Natural Language Processing. Berlin, Heidelberg, New York: Springer-Verlag.

[6] Potthast, M., Barron-Cedeno, A., Stein, B. & Rosso, P. (2011). Cross-language Plagiarism Detection. Language Resources and Evaluation, vol. 45, no. 1, pp. 45-62.

[7] Menai, M. E. B. & Bagais, M. (2011). APlag: A Plagiarism Checker for Arabic Texts. 6th International Conference on Computer Science & Education (ICCSE), SuperStar Virgo, Singapore, 2011.

[8] Schleimer, S., Wilkerson, D. S. & Aiken, A. (2003). Winnowing: Local Algorithms for Document

Fingerprinting. The 2003 ACM SIGMOD International Conference Management of Data, New York, 2003.

[9] Zihayat, M. & Basiri, J. (2009). Controlling Similarities and Copying in Persian E-documents Received from Danesh Pajohan for Virtual Lessons. International Conference of E-learning, Tehran, Iran, 2009.

[10] Kamran, K., Mohammadi, A. & Mohsenzadeh, M. (2012). Plagiarism Detection in Persian Texts with SimHash Algorithm. 11th Conference of Intelligent Systems, Iran science and industry-university, Iran, 2012.

[11] Mahdavi, P., Siadati, Z. & Yaghmaee, F. (2014). Automatic External Persian Plagiarism Detection Using Vector Space Model. 4th International Conference on Computer and Knowledge Engineering (ICCKF), Ferdowsi University of Mashhad, Iran, 2014.

[12] Rakian, Sh., Safi Esfahani, F. & Rastegari, H.(2015). A Persian Fuzzy Plagiarism Detection Approach. Journal of Information Systems and Telecommunication (JIST), vol. 3, no. 3, pp. 182-190.

[13] Academic Homepage of Majid Sazvar (2015), Available: http://sazvar.student.um.ac.ir/index.php.

[14] The Marjae Dadegan website (2015), Available: http://dadegan.ir/catalog/D3911124a.

[15] Pakzad, A. & Minaei Bidgoli, B.(2016). An improved joint model: POS tagging and dependency parsing. Journal of AI and Data Mining (JAIDM), vol. 4, no. 1, pp. 1-8.

[16] The Github website (2015), Available: https://github.com/mojtaba-khallash/NHazm.

[17] The Search Engine Optimization tool (2015), Available: http://www.seomastering.com/similar-text-checker.php.

[18] The Hamshahri Collection website (2015), Available: http://ece.ut.ac.ir/dbrg/hamshahri.

نشریه هوش مصنوعی و داده‌کاوی

جستجوگر سرقت ادبی برای متون فارسی (PCP) با استفاده از هش انگشت‌نگاری درختی

شیما رفیعیان[1]* و احمد براآنی دستجردی[2]

[1] دانشکده مهندسی کامپیوتر، دانشگاه شیخ بهایی، بهارستان، اصفهان، ایران.

[2] دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران.

ارسال ۲۰۱۵/۱۰/۲۱؛ پذیرش ۲۰۱۶/۰۱/۲۴

چکیده:

یکی از مسائل مهمی که در زمینه متن‌کاوی مورد استقبال پژوهشگران قرار گرفته است، تشخیص سرقت ادبی به منظور رعایت حقوق نویسندگان است. این مسئله به عنوان یک بحران جدی در دانشگاه‌ها به چشم می‌خورد. ابزارهای مستقل از زبان وجود دارند که به دلیل در نظر نگرفتن ویژگی‌های خاص هر زبان، نتایج قابل اعتمادی ایجاد نمی‌کنند. با توجه به معدود کارهای انجام شده در زبان فارسی، که از دقت قابل قبولی نیز برخوردار نیستند، نیاز به روشی برای بهبود دقت کشف عبارات سرقتی فارسی می‌باشد. در این مقاله راه‌کار PCP ارائه شده است. این راه‌کار به صورت ترکیبی بوده و علاوه بر در نظر گرفتن ریشه و معنی کلمات در تعیین کلمات مترادف و مکسر، به کمک نمایش درختی سند به انگشت‌نگاری متن بر اساس ۳-گرام‌های کلمات می‌پردازد. در نهایت گرام‌های به‌دست آمده از متنِ پاکسازی شده، به کمک تابع هش BKDR، هش شده و به عنوان اثرانگشت سند در مخزن اثرانگشت‌های اسناد مرجع، برای بررسی اسناد مشکوک ذخیره می‌گردد. روش ارائه شده PCP با اعمال هشت آزمایش متفاوت بر روی هفت مجموعه اسناد ایجاد شده از سایت روزنامه همشهری شامل سند مشکوک و سند مرجع ارزیابی گردید. نتایج نشان می‌دهد، میزان دقتِ تعیین کشف شباهت متون به کمک روش پیشنهادی نسبت به روش "Winnowing" بومی‌سازی شده، به طور متوسط ۲۱,۱۵ درصد بهبود داشته است. دقت کشف شباهت روش PCP نسبت به ابزار مستقل از زبان به طور میانگین ۳۱,۶۵ درصد بهبود یافته است.

کلمات کلیدی: متن‌کاوی، پردازش زبان طبیعی، سرقت ادبی، کشف سرقت ادبی بیرونی، زبان فارسی.