10th International Conference on Computer and Knowledge Engineering (ICCKE2020) October 29-30, 2020- Ferdowsi University of Mashhad- Iran

ParsiPayesh: Persian Plagiarism Detection based on Semantic and Structural Analysis

Soghra Lazemi*, Hossein Ebrahimpour-Komleh
Department of Computer Engineering
The University of Kashan, Kashan, Iran
*Soghra.lazemi@gmail.com

***Abstract*—In recent years, the rapid increase of Persian electronic resources and facility of access to them has seriously triggered the plagiarism problem of the Iranian scientific community. Despite the automatic systems of plagiarism detection, like Turnitin, Eve2, this problem has strongly remained due to lack of support from Persian. The main purpose of this article is to detect exact plagiarisms and re-writings in Persian science texts. In our proposed method, after the candidate retrieval based on the statistical characteristics, in the text alignment step, structural analysis and semantic analysis of expression has been performed to detect re-writing plagiarisms. Firstly, data-driven dependency parser has been improved with the help of a deep learning model for Persian language to analyze the structure of the expression, and then the degree of structural similarity of the expression is evaluated through the analysis of the dependency tree. In this paper, our suggestion to examine the semantic similarity of expression is to use the semantic role labeling obtained from the deep learning model presented. The experiments have been performed on the corpus prepared in the AAIC2015 and corpus of the PAN2015 competitions. The results indicate that structural and semantic information improves the performance of the proposed method. ParsiPayesh is available on http://www.parsipayesh.ir.**

Keywords— Plagiarism Detection, Semantic Role Labeling, Deep Learning, MSTparser, Persian

I. INTRODUCTION

Unfortunately, today in the education field in some countries, the quantity and level of degrees are considered to be paramount. When everyone is looking for academic degrees and not paying much attention to the growth of science, naturally, they will do anything to plagiarize (copying). The same factor has caused the spread of such matters such as the plagiarism of scientific articles, theses, tasks, reports, source codes…, at the level of universities, scientific centers or even commercial companies.

The ease of access to scientific-research journals by increasing communication networks and a huge amount of electronic resources, allows plagiarism for those profiteers who tend to achieve their goals in the shortest time and without any effort. , and made it possible for various scientific communities, including the scientific community of Iran, to be confronted with plagiarism.

Identifying and tracking plagiarism with the exponential growth and massive amount of electronic information is beyond the control of human judges; therefore, different countries have designed and implemented automatic systems like Turnitin, Eve2, iThenticate and PlagScan, to overcome this problem. Fortunately, the number of these systems has been daily increased and their quality and efficiency has dramatically increased. It's an important point that existing systems are usually only able to support one language (usually English) or languages other than Persian [1]. Of course, it must be noted that systems which are able to support multiple languages are not of good quality because of ignoring the peculiar features of the language. Regarding more than 2.21 trillion web pages in Persian and over one million electronic theses and articles, it is necessary to design and implement independent systems focusing on Persian language in this field, to specify the authenticity of scientific materials and actual connection of the work and the author [2]. Fig. 1 illustrates a simple categorization of Plagiarism in a number of different ways:

Language: cross-lingual or mono-lingual: In the cross-lingual environment, the source language and suspicious language are different, and the methods for detecting plagiarism are performed in two levels (the first level, two languages have the same grammatical systems, and the second level, grammatical system of languages are not the same) [3]. In mono-lingual environment, the source language and suspicious language are the same, and plagiarism detection is done lexically, grammatically and semantically based on the features of the text [4]. In a mono-lingual environment, the proposed method can be presented in a particular language considering the specific features of the language (language-sensitive), or can be presented in several languages by omitting and merging minor features (language-free).

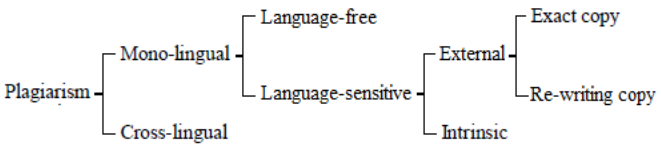

Fig. 1. Taxonomy of plagiarism.

978-1-7281-8566-8/20/$31.00 ©2020 IEEE

References: external or intrinsic [5]: In external references, the suspicious document is compared to documents in the local database or web pages, and documents or web pages similar to the suspicious document are reported as plagiarism sources. In intrinsic references, different parts of a document are compared with each other using different attributes (such as writing style), and parts that are not in harmony with each other are identified as stolen sections.

Types of plagiarism: exact or intelligent [5]: In exact or direct plagiarism, sentence or text is being used without any change. Intelligent plagiarism can be done in two ways, translated and re-writing. In translation plagiarism, a text from a source language is used in the suspicious language. In re-writing plagiarism, the deceitful attempts to create differences by making changes (such as changing the order of words, deleting or adding words, replacing words) in the source. Exact plagiarism and re-writing plagiarisms are done in a mono-lingual environment. Translation plagiarism takes place in a cross-lingual environment.

Our goal in this article is to discover exact plagiarism and intelligent re-writing plagiarism in a mono-lingual environment (Persian) with external references.

The 2009 international competitions on plagiarism detection (PAN) are conducted every year. In this competition, plagiarism of texts is divided into two distinct stages: candidate retrieval and text alignment. Some of the works heeded only one step and others took both stages into consideration. The focus of this paper is on both stages.

The rest of the paper is organized as what follows. In the second part, the conducted research works in the field of Persian plagiarism detection in mono-lingual environment are discussed. In the third section, our method is presented. The fourth section includes experiments and results. Finally, in the fifth section, a summary of the paper is presented with some future works.

II. Related Works

General methods of detecting plagiarisms can be classified into three categories: grammar-based methods, semantic-based methods and hybrid methods [6]. Grammar-based methods use string-based methods. In this method, two strings (for example, two sentences) are compared together. This method is suitable to detect exact plagiarisms. Semantic-based methods transfer text or sentence to vector-space models and measure their similarity. Keyword-based methods fall into this category. This method is suitable for re-writing copies. The third category is the combination of grammatical-based and semantic-based methods that can identify both exact and re-writing copies. Subsequently, the submitted works have been assessed to detect Persian plagiarism.

A. Gramer-Based Methods

The method presented by [7] focuses only on the text alignment phase and uses a string-based method to match plagiarized items. In their method, after performing routine pre-processes, two sequences of words, a sequence for the original text, and another sequence for the suspicious text, are compared with a combination of Jaccard and N-grams similarity methods. The corpus used is provided by the authors. In the proposed method by Mansoorizadeh et al. [8], after tokenizing, a similarity matrix is created by cosine similarity measure for all paired sentence of the suspicious document and the source document. Elements of the matrix with the maximum similarity represent the plagiarized sentence. Esteki and Esfahani [9], to identify fraud, have provided a machine learning based method. For each pair of sentence, the Jaccard similarity and Levenshtein distance measures have been calculated and used as a feature to train SVM. The achieved precision and recall on Persian PlagDet2016 corpus [10], which is not publicly available, with a threshold level of 0.5, is 82% and 92%, respectively.

B. Semantic-Based Methods

In the method presented by [11] after preprocessing, for the purpose of candidate retrieval, the keywords of suspicious document and their synonyms are extracted and the source documents that contain these words have been retrieved as candidate documents. Then, for each sentence from the suspicious document, the most similar blocks are recovered from the candidate's documents. In the text alignment step, the candidate blocks are converted to sentence and the similarity of sentences is calculated. Experiments were performed on three corpora, (IRANDOC, TMC and Prozhe.com) which unfortunately these corpora have not yet been publicized. The average of reported accuracy was 70%. Mahdavi et al. [12], have focused on both the steps of candidate retrieval and text alignment. They examined the documents as Bag of Words, and after the preprocessing, words with TP (Transition Point) and TF-IDF criterion were transferred to vector space. In the candidate retrieval phase, the Weighted cosine similarity is used and in the text alignment phase, the 3-grams and Overlap Similarity is used. The authors compiled a corpus of 41 source documents and 84 suspicious documents for evaluation and reported 95% of accuracy. Lazemi and Ebrahimpour-komleh [13] have proposed a three-step approach. In their proposed method, the keywords of document are extracted using Markov chain model for candidate documents retrieval, then the similarity of the suspicious document is calculated with other documents using the extracted keywords and similar documents have been retrieved as candidate documents. In the next step, by extracting the 19 structural and semantic defined features at the sentence level, the classification of the sentences of the suspicious document has been performed. Experiments were performed on the AAIC2015 (Amirkabir Artificial Intelligence competition, 1394) and PAN2015 [14] corpora. The mean of achieved precision and recall was 86.4% and 86.3% for first corpus, 82.9% and 76.4% for second corpus respectively. Lazemi et al. [15] have developed a CNN and machine learning based approach. In their proposed method, after performing the necessary preprocesses, using the CNN, a vectorial representation of the documents created and using the clustering algorithm, the candidate's documents are restored. For text alignment, using the sentence-embeddings matrix, features of n-grams are extracted at the sentence level by CNN, and sentences have been classified by the SVM algorithm. Experiments were performed on the AAIC2015 and PAN2015 [14] corpora. The mean of precision and recall was 84.3% and

80.6% for the first corpus and 83.3% and 82.6% for the second corpus respectively. The articles submitted in the Persian PlagDet2016 competition (held in August 2016) focused only on the text alignment phase. In this competition, a corpus is provided to conduct experiments that all articles have carried out their experiments based on it [10]. The provided corpus is not yet available to the public. Among researches done in this competition, we can mention [16] which has used the distribution vector of words, and for each sentence of the suspicious document, cosine similarity is calculated with all the source document expression. Momtaz et al. [17], also illustrate the source document graphically with a specific length, then each section of the suspicious document is compared with the source document.

C. Hybrid Methods

Ahangarbahan and Montazer [18] have proposed a mixed fuzzy similarity approach for the purpose of text alignment. In the first step, after pre-processing and removing stop word, a text was divided into two parts: general and domain-specific knowledge words. Then, the mixed lexical and semantic fuzzy inference system was designed to assess text similarity. The proposed method was evaluated on the corpus provided by the authors. The results indicated that the proposed method can achieve a rate of 79% in terms of precision and can detect 83% of the plagiarism cases.

III. Proposed Method

In this article, we intend to provide a framework for identifying plagiarism in Persian science texts that are capable of detecting exact and intelligent re-writing fraud with structural-semantic analysis of texts. In our proposed method, for each suspicious document, candidate documents are retrieved based on statistical features. In the discovery of structural plagiarism step, due to the lack of an appropriate structural parser for the Persian language, the MSTParser parser has been improved with the help of the deep learning model. Semantic plagiarism has been carried out with the help of the analysis of the semantic role labeling (SRL), due to the lack of SRL tools for the Persian, we have presented a deep learning based approach, in the following we give a description of each step. Our proposed system is available on http://www.parsipayesh.ir.

A. Preprocessing

The input of the preprocessing phase is a TXT file. In this phase, depending on the characteristics of the Persian language, a series of changes are made to the file. These changes take place at 2 levels. These 2 levels include:

Changes at the letter level: Some changes to the input file characters are required for text integration in Persian. These changes include:

- Replacing all the "ي" characters in the text with the character "ی":
- Replacing all the "ك، ڪ، ﻙ" characters in the text with the character "ک"
- Replacing all the " أ، إ، آ " characters in the text with the character "ا"
- Replacing all the "ؤ، ۇ" characters in the text with the character "و"
- Replacing all the "ۀ ، ة" characters in the text with the character "ه"
- Replacing all half-spaces with the half-space coded 0xEE in DOS Code Page 720.
- Removing all nunnations, intensification marks, etc.:

Changes at the word level: To avoid any problems in the next phase, we need to make changes at the words level. These changes include:

- Omission of stop words: Stop words are words that do not contain information and are used only to connect words in the sentence [19].
- Connecting words: In some documents, half-spaces are not used. For example, the verb "می‌باشد" is written as "می باشد".

B. Candidate Retrieval

The aim of this step is to decrease the search range from all the documents in the database to a limited number of documents, which are probable to be source of copying. This step dramatically reduces the time of Plagiarism's discovery. If the source document used is not among the recovered documents, plagiarism detection will fail. Hence, the accuracy of this step should be high. In this paper, a two-step procedure has been proposed for Candidate Retrieval.

Document Representation using Keywords: Keywords are a collection of important words in a document that are the core topic of the discussion [20]. Keywords can serve as a useful tool for searching in a short amount of time to help the user purposefully explore the massive amount of text information. In other words, using the keywords of a text, without having to study all of this, you can get the main theme of the text [21]. Therefore, displaying each of the documents by keywords can be helpful to us to purposefully explore massive amount of text data in a short time. This step can be very helpful in speeding up and reducing complexity.

We use the statistical method presented in our previous research [22] in order to extract the keywords of the documents. In this method, the feature vector is defined for each of the document words. By using the classification algorithms, the "key" or "none-key" tag is assigned to the word which is being examined.

Document Similarity: The Suspicious document keys are compared with the keywords of any other document in the database by using the Jaccard similarity, and if the similarity between the two document's keywords is greater than the defined threshold, then we can assume that the domain of the two documents is similar. As a result, the source document will be listed on the candidate list.

A. Exact Plagiarism Detection

Exact plagiarism is carried out without any change in the source document. The string matching technique seems proper to detect this type of fraud. At this step, the similarity between the two sentences is calculated using the equation 1:

$$Sim_{exact}(S_1, S_2) = \frac{2*|S_1 \cap S_2|}{|S_1|+|S_2|} \quad (1)$$

B. Structural Re-writing Plagiarism Detection

Typically, the deceitful person may change some parts of the text through altering the order of words by replacing the words with synonyms without changing the sentence structure, adding or deleting some words to inhibit plagiarism detection. In order to detect this type of plagiarism we scrutinize the structure of the sentence. We need to take sentence apart structurally to analyze the structure of sentence. Through structural analysis, the grammatical role of the words in the sentence is determined. It is apparent that the displacement of words does not change their grammatical role. Structural analysis is generally divided into phrase-structure and dependency classes based on grammatical theories. Phrase-structure parsers are not flexible in terms of words and by changing the order of words, create another parsing tree, while the dependency parsing is not sensitive to the order of the words. Therefore, by creating a dependency tree of sentence, it is possible to identify plagiarized expression concealed by changing the order of the words. According to previous research [25] and considering specific features of Persian language, MSTParser (dependency-based parser) is suitable for Persian language:

- Due to the SOV property, the head and the dependent are usually spaced far from each other, and MSTParser is appropriate to determine long-distance relationships due to the creation of a sentence graph.
- Due to the free word-order property, most Persian sentences produce non-projective trees, and MSTParser is able to produce non-projective trees.

Despite the traditional MSTParser method, in which each edge score is obtained through weighted sum of features, our proposed method extracts the score of each edge in four steps 1- A Bi-LSTM network that receives the distributive representation of words as inputs, 2- A LSTM network which receives binary data as input, 3- A hidden layer with a rectified activation function combining the two above-mentioned features and 4- An output layer for generating scores for each kind of dependency type between each pair of head and it's dependent.

Neighborhood information is very useful in graph-based structure parsing [26, 27], so we use the LSTM to extract foundation information. The LSTM network allows the use of information far from the current word. The bidirectional network allows us to access the information on the right and left side of the words. In other words, by using the Bi-directional network, we can obtain the information of surrounding neighbors and neighbors between head and dependent. To achieve this, at first, each word is represented

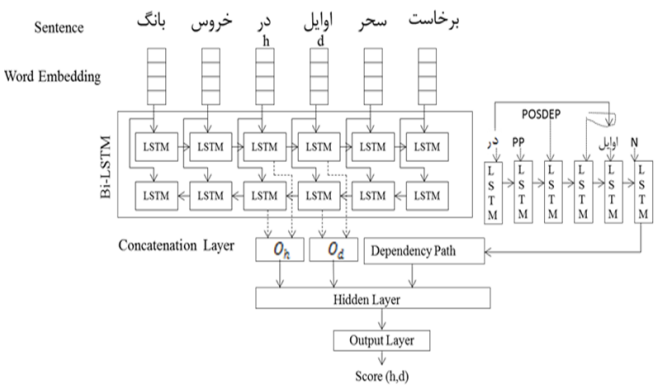

Fig. 2. An example of score computing for a pair of head and its dependent using our propose method.

by word embedding with a d-dimensional vector, $e_i^w \in \Re^d$. The generated vector for each word is used as an input to the LSTM network. The net amount reached to the output layer is the output concatenation of the forward and backward two-layer output.

[26, 27] have stated the dependency label as a useful feature. We extract this property as a route from head to depend by a LSTM network. For each pair of head and dependent, we have considered a LSTM network that takes a sequence of dependent to head as an input. The network input for each word in the dependent-head route, at any time step, includes a binary indicator for an edge direction to the next word, an edge label (syntactic role) to the next word, word POS tag, word embedding, next word embedding, and next word POS tag (see Fig. 2).

The output concatenation of the two networks goes into the hidden layer, including d_H number of nodes, and the rectified activation function of formula 2.

$$H = g(W_1(O_h \circ O_d \circ O_{dp}) + b_1)$$
$$score(h,d) = Softmax(W_2 H + b_2) \quad (2)$$

The output score $score(h,d) \in R^{|L|}$ is a score vector where |L| is the number of dependency types and each dimension of $score(h,d)$ is the score for each kind of dependency type of head-dependent.

The dependency tree is formed for the paired suspicious sentence and retrieved source sentence. We consider the structural similarity of the two sentences as the structural similarity of the common or synonymous words. We use the similarity of P-Rank [28] to calculate the structural similarity of the same words or synonyms. Same words or synonyms with the similarity more than threshold are considered to be similar in structural terms, and the similarity of the two sentences is calculated using formula 3.

$$Sim_{structure}(S_1, S_2) = \frac{2*|\text{same words or synonyms with the similarity more than threshold}|}{|S_1|+|S_2|} \quad (3)$$

C. Semantic Re-writing Plagiarism Detection

Plagiarism may also occur at the semantic level. As an example, consider the following two sentences:

تماشاگران، کردند میدان پیروز را تراکتورسازی تیم. *(The audience was the cause of Tractorsazi football team victory);* Source sentence شد پیروز تراکتورسازی تیم. *(The Tractorsazi football team won.);* Suspicious sentence

As it is visible, a suspicious sentence in another structure, with the same words (or synonyms), and some words added or deleted, has been rewritten. To identify this type of plagiarism, we have suggested the analysis of the SRL. The purpose of the SRL is to affixing semantic roles such as the *agent, patient, theme, source, goal and …* to each of the sentence's words or expression, regarding the considered predicate (mostly verbal predicate) [29]. For instance, in the example above, the *Tractorsazi* in the first sentence has the syntactic role of the "object" and in the second example, it has the syntactic role of the "subject", but in this case, in both expressions, its semantic role is "*patient*".

We have proposed a deep learning algorithm due to the lack of a SRL tool in the Persian language. The SRL, in terms search procedure, is similar to the dependency parser; both of them try to discover the relation between pairs of words. The main difference between them is the search range; the dependency parser tries to detect a connection between all the pairs of words while the search for the SRL is restricted to searching between paired predicates and arguments [25, 30].

Traditional methods have introduced structural analysis as a pre-requisite for SRL, and used the structural analysis tree (phrase-structure tree or dependency tree) to extract useful features and classify words into appropriate semantic categories. The recent presented methods provide SRL using deep learning methods to obviate the need to feature engineering. In this research, we have suggested a combination of both methods for SRL.

At this step, for the sentence, $S = \{x_1, x_2, \dots, x_n\}$, with arguments, $A = \{a_1, a_2, \dots, a_m\}$, and semantic roles, $R = \{r_1, r_2, \dots, r_m\}$, each word has been displayed as word embedding concatenation and the word dependency path to the predicate. Dependency path is extracted using the network provided in the previous section.

Our goal is to extract the contextual features of words. To achieve the goal, we create a meaningful representation of each word according to its context using the Bi-LSTM network. The LSTM network learns long dependencies and allows the use of information far from the current word for labeling. Regarding that in semantic role labeling both previous and past dependencies are important, hence access to the right and left side information is required. The Bi-directional network allows us to do this. The forward network learns the previous dependencies and backward network learns the past dependencies.

In anticipating SRL, because of the relation between semantic roles of words, examining each of the words independently would be inappropriate; therefore, it would be useful to consider the relationship between the labeled neighbors [31]. Therefore, we have considered SRL as a sequence-labeling problem and we have used the Linear-chain-Conditional Random Field (CRF) model [32]. In this model, having the sequence of words, $X = (x_1, \dots, x_t)$, and the sequence of labels, $R = (r_1, \dots, r_t)$, the goal is to calculate the conditional probability of the labels sequence, $P(r_1, \dots, r_t | x_1, \dots, x_t)$, and find the sequence with the highest probability, $R^* = argmax_r P(r_1, \dots, r_t | x_1, \dots, x_t)$.

The SRL carries out for the paired suspicious sentence and the retrieved source sentence. The semantic similarity of the two sentences is calculated using the formula 4 by counting the number of the synonymous and same words which have the same semantic role label.

$$Sim_{semantic}(S_1, S_2) = \frac{2*|\text{same words or synonymous which have the same semantic role label)}|}{|Argument(S_1)|+|Argument(S_2)|} \quad (4)$$

IV. Experiments and Results

A. Corpora and Evaluation Metrics

Persian belongs to low-resource languages and the available corpora for conducting experiments are very limited. To conduct experiments on extracting the dependency tree of sentences, there are only two corpora (Persian Dependency Treebank (PerDT) [33] and Uppsala Persian Dependency Treebank (UPDT) [34]) that we have used from both. PerDT is the first Persian dependency Treebank, and includes about 30,000 sentences annotated with syntactic roles and morpho-syntactic features and the corresponding dependency tree. There are 44 dependency relations, 17 types of coarse-grained, and 32 types of fine-grained POS tags. In UPDT, the syntactic relation of words is determined by the dependency grammar. This corpus contains 6000 sentences from the Uppsala Persian Corpus with a corresponding dependency tree. In this corpus, there are 48 types of dependency relations, 13 types of coarse-grained, and 18 types of fine-grained POS tags. Both corpora are prepared based on the CoNll template and the Stanford Typed. More information about the corpora used is given in Table I.

For evaluation, the Unlabeled Attachment Score and the Labeled Attachment Score are used [40] and defined as formula 5 and formula 6.

$$UAS = \frac{Number\ of\ identical\ edges\ in\ two\ trees\ regardless\ of\ the\ label}{Total\ number\ of\ two\ tree\ edges} \quad (5)$$

$$LAS = \frac{Number\ of\ identical\ edges\ in\ two\ trees\ with\ label}{Total\ number\ of\ two\ tree\ edges} \quad (6)$$

For the SRL experiments, *The First Semantic Role Corpus in Persian Language* [35] has been used. It includes 29983 sentences in contemporary Persian language, which are manually annotated *based on the concept of thematic roles of Fillmore*, with 27 semantic roles in three stages. This corpus has added a semantic layer to the Persian syntactic dependency treebank [33]. The used semantic roles include two groups of *thematic roles and functional tags*, such as: *agent, patient, theme, experiencer, instrument, location, source, goal, cause, productive*. For evaluation, the precision, recall and F_1-measure are used and defined as formulas 7, 8 and 9.

$$Precision = \frac{number\ of\ correctly\ labelled\ arguments}{number\ of\ detected\ arguments} \quad (7)$$

$$Recall = \frac{number\ of\ correctly\ labelled\ arguments}{number\ of\ arguments} \quad (8)$$

$$F_1 - measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (9)$$

For the plagiarism detection experiments, the prepared corpus in the Amirkabir Artificial Intelligence competition (AAIC2015) and the prepared corpus in the PAN2015 competition [14] have been used. Another corpus called the Mahak Samim [36], which is presented in the Persian PlagDet2016 competition, and the articles presented in the competition, have done their experiments using this corpus, which unfortunately this corpus has not yet been publicized. The corpus presented in the AAIC2015 is an altered Corpus PAN2015. Both corpora include source documents, suspicious documents, the marked files of the exact location of each plagiarism in suspicious documents and the related source document. Also, the type of obfuscation (none, random, simulated) and its degree (low, high) for fraud section is specified. The statistical information of the corpora are reported in Table II.

We have used the presented measures in the 3rd PAN plagiarism detection to evaluate the proposed system [37]. The used evaluation measures are recall (as formula 10), precision (as formula 11) and plagdet (as formula 12). These measures have also been used in subsequent PAN competitions.

$$Recall(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R}(s\sqcap r)|}{|s|} \quad (10)$$

$$Precision(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(s\sqcap r)|}{|r|} \quad (11)$$

$$where \quad s\sqcap r = \begin{cases} s\cap r & if\ r\ detects\ s \\ \emptyset & otherwise \end{cases}$$

$$Plagdet(S,R) = \frac{F_1}{log_2(1+gran(S,R))} \quad (12)$$

$$where \quad gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_S|$$

All corpora are individually split into train, development and test sets by using of 10-fold cross validation method. 80% of the data is used for training, 10% is used for developing and 10% is used for testing.

B. Implementation and Hyper-Parameter Tuning

We implement the neural network using the UKPLab (https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf) and the CRF using the MADlib (https://madlib.apache.org/docs/v1.10/group__grp__crf.html). Training is performed with mini-batch stochastic gradient descent (SGD) with a fixed learning rate. Also, we explored AdaGrad, AdaDelta, RMSProp, Adam and Nadam optimization algorithms, but they did not improve upon SGD.

In order to reducing overfitting, we apply the dropout on output vector of each LSTM layer.

TABLE I. STATISTICAL PROPERTIES OF PERDT CORPUS AND UPDT CORPUS.

	Persian Dependency Treebank	**Uppsala Persian Dependency Treebank**
Number of sentences	29982	6000
Number of words	498081	151671
Number of distinct words	37618	15692
Number of verbs	9200	-
Number of distinct verbs	62889	-
Average sentence length	16.61	25.28
Coarse-grained POS tags	17	13
Fine-grained POS tags	32	18
	is freely available in CoNLL format	is freely available in CoNLL format

TABLE II. STATISTIC ABOUT THE PLAGIARISM DETECTION CORPORA.

	AAIC2015	**PAN2015**
Number of source documents	1524	1057
Number of suspicious documents	1501	1054
Number of plagiarism with no-obfuscation	184	259
Number of plagiarism with Low-obfuscation	940	301
Number of plagiarism with High-obfuscation	777	228

The MADlib implementation uses limited-memory BFGS (L-BFGS), a limited-memory variation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update, aquasi-Newton method for unconstrained optimization. In order to create word embeddings, we use the word2vec [38] algorithm. We used FarsNet [39] to extract all the synonyms of the word in Persian.

We tune the hyper-parameters on the development sets by random search. We evaluated 300 hyper-parameter setting. Table III summarizes the chosen hyper-parameters for all experiments.

C. Results and Discussion

MSTParser is used with four settings: projective-first-order, projective-second-order, non-projective-first-order, and non-projective-second-order. MIRA is used to estimate the weight vector. Based on results, non- projective-second-order settings for both corpora have brought good results.

Table IV, illustrate the results for both corpora. To perform comparisons, the experiments have been carried out in two conditions: once without any change and with MSTParser settings itself and again with our proposed method. The results show that our proposed method has been performing better. Our suggested method, although does not indicate any significant improvement in comparison with our previous research, has however been able to achieve some good results in some cases. On the other hand, the use of the distributive representation of words has solved the problem of sparsity mentioned in [40].

TABLE III. HYPER-PARAMETERS OF OUR MODEL.

		Dependency Path		Syntactic Phrase		Semantic Phrase
	Ranges	PerDT & SRL Corpus	UPDT	PerDT	UPDT	SRL Corpus
POS tags	-	17	13	-	-	17
Syntactic label	-	44	48	-	-	44
LSTM layers	-	1	1	2	2	2
LSTM state size	[100,400]	200	150	300	200	275
Learning rate	$[10^{-4},10^{-1}]$	0.010	0.007	0.041	0.053	0.030
Dropout rate	[0,1]	0.68	0.50	0.46	0.42	0.32
Mini-batch size	[5,14]	9	9	10	10	6
Neuron	[50,200]	-	-	150	100	-
Threshold-Candidate retrieval	0.71					
Threshold-Structure similarity	0.83					

TABLE IV. NON PROJECTIVE-SECOND ORDER.

	PerDT		UPDT	
	UAS	LAS	UAS	LAS
Baseline-Features	85.16	82.99	85.31	83.80
Lazemi et al. [40]	89.17	85.83	88.96	86.25
Our Proposed Method	88.34	86.22	89.51	86.96

TABLE V. ACCURACY OF FEATURE SETS.

		Precision	Recall	F1
SRL Corpus	Base features	79.84	82.39	81.09
	+Dependency Path features	86.41	80.14	83.15

TABLE VI. ACCURACY OF SEMANTIC ROLE LABELING.

		Precision	Recall	F1
Used Semantic Role Labels by [41]	Base	80.18	79.95	80.01
	Our model	86.07	82.31	84.14
All semantic role Labels	Our model	80.47	79.15	79.80

Table V illustrates a comparison between the features. This comparison has been done to determine the effect of the "dependency path" feature on expression. The results indicate that by adding the feature derived from the dependency tree has significantly improved the results.

In [41], a number of semantic labels have not been considered. We tested our suggested method once with semantic labels in [41] and again with the semantic labels considered in this paper. In accordance with Table VI, our method is better than [41]. By adding other semantic labels, the results have slightly declined. The reason for this may be the inadequacy of class-related train data related to certain labels.

We conducted the experiments in two steps to evaluate the efficiency of the suggested method in the plagiarism detection phase. In the first step, the accuracy of the suggested method in candidate retrieval is calculated to see whether the source document used in the suspicious document is in the list of retrieved candidate documents for the suspicious document, is examined. Table VII illustrates the mean of accuracy obtained for both used corpora.

We manually reviewed the candidates' retrieved documents for each suspicious document, the reviews indicate that the documents that have been retrieved were similar in content to the suspicious document; hence we looked for the reason for the poor results in creation of corpora. In both corpora the documents are clustered according to the topic and the documents having the same topic have been classified in one cluster. The source document and the suspicious document both have been randomly selected from one cluster. We know that in the real world, the source document that is used is semantically related to the suspicious document; however, topic clustering and random selection have eliminated the interrelation of contents and meaning in the source document and the suspicious document, and as a result, the recovery results have reduced.

For the purpose of performance evaluation in the second step, text alignment, by adding the used source document in the suspicious document in the list of retrieved candidate documents of the suspicious document, Table VIII shows the results. As you can see, the study of the structure and semantic of expression did not have a significant impact on none-obfuscation plagiarism. But when the obfuscation is added to sentences, structural and semantic analysis has greatly improved the results. It is worth to mention that the effect of the accuracy of the dependency tree extraction and SRL on the final results should not be overlooked. If better results were achieved in the earlier steps, the results of the plagiarism could certainly be improved.

We conducted the experiments in two steps to evaluate the efficiency of the suggested method in the plagiarism detection phase. In the first step, the accuracy of the suggested method in candidate retrieval is calculated to see whether the source document used in the suspicious document is in the list of retrieved candidate documents for the suspicious document, is examined. Table VII illustrates the mean of accuracy obtained for both used corpora.

We manually reviewed the candidates' retrieved documents for each suspicious document, the reviews indicate that the documents that have been retrieved were similar in content to the suspicious document; hence we looked for the reason for the poor results in creation of corpora. In both corpora the documents are clustered according to the topic and the documents having the same topic have been classified in one cluster. The source document and the suspicious document both have been randomly selected from one cluster. We know that in the real world, the source document that is used is semantically related to the suspicious document; however, topic clustering and random selection have eliminated the interrelation of contents and meaning in the source document and the suspicious document, and as a result, the recovery results have reduced.

TABLE VII. ACCURACY OF CANDIDATE RETRIEVAL.

	AAIC2015					PAN2015				
	CSI	EB	MF	TF-ISF	Our method	CSI	EB	MF	TF-ISF	Our method
Naïve Bayes	78.00	80.01	87.26	82.39	88.48	80.09	82.87	86.39	85.41	89.02
Support Vector Machines	73.60	79.00	86.19	80.67	87.35	76.61	79.98	80.06	82.11	87.36
Logistic Regression	76.28	79.56	86.99	80.77	86.21	77.30	81.37	84.00	81.25	87.90
Random Forest	78.33	83.33	87.78	84.22	88.52	80.86	83.44	86.69	86.04	89.64

CSI: co-occurrence statistical information, **EB:** eccentricity-based keyword extraction, **MF:** most frequent, **IF-ISF:** term frequency-inverse sentence frequency

TABLE VIII. THE RESULTS OF THE TEXT ALIGNMENT.

		AAIC2015				PAN2015			
		Prec	Rec	Gran	PlagDet	Prec	Rec	Gran	PlagDet
Sim_{exact}	None-obfuscation	0.95	0.89	1.00	0.92	0.87	0.96	1.01	0.90
	Low-obfuscation	0.77	0.61	1.02	0.67	0.90	0.52	1.04	0.64
	High-obfuscation	0.94	0.43	1.04	0.57	0.85	0.29	1.00	0.44
$Sim_{exact} + Sim_{Structure}$	None-obfuscation	0.96	0.92	1.00	0.94	0.96	0.89	1.00	0.93
	Low-obfuscation	0.93	0.80	1.00	0.86	0.86	0.91	1.00	0.89
	High-obfuscation	0.84	0.82	1.00	0.83	0.90	0.80	1.00	0.84
$Sim_{exact} + Sim_{Structure} + Sim_{semantic}$	None-obfuscation	0.98	0.96	1.00	0.96	0.97	0.97	1.00	0.97
	Low-obfuscation	0.94	0.88	1.01	0.91	0.92	0.91	1.00	0.92
	High-obfuscation	0.87	0.91	1.00	0.89	0.96	0.84	1.04	0.88

For the purpose of performance evaluation in the second step, text alignment, by adding the used source document in the suspicious document in the list of retrieved candidate documents of the suspicious document, Table VIII shows the results. As you can see, the study of the structure and semantic of expression did not have a significant impact on none-obfuscation plagiarism. But when the obfuscation is added to sentences, structural and semantic analysis has greatly improved the results. It is worth to mention that the effect of the accuracy of the dependency tree extraction and SRL on the final results should not be overlooked. If better results were achieved in the earlier steps, the results of the plagiarism could certainly be improved.

V. CONCLUSION

In this paper, a method based on structural and semantic analysis for the discovery of plagiarism in Persian texts is presented. The focus of this article is the discovery of exact and re-writing plagiarism. In the candidate retrieval step, statistical features are used, and in the text alignment step, for each paired sentence of the suspicious document and the retrieved source document, the amount of commonality of the two sentences in three phases of string based, structure-based and semantic-based has been calculated. Considering the features of Persian language, data-driven dependency parser and semantic role labeling have been developed and utilized using a deep learning model in order to determine the structural and semantic similarity. The results of the experiments indicate that the use of structural similarity and semantic similarity has had a significant effect on the detection of rewriting plagiarism. Our proposed system is available on http://www.parsipayesh.ir.

Unfortunately, in the area of plagiarism detection, there is a huge gap between researches carried out for Persian and done for other languages; the main reason for this is the dearth of labeled corpora, lack of tools such as structural parsers, semantic role labeling and ... .by the expansion of Persian language processing tools, hopefully this field will attract many researchers.

As the candidate retrieval phase plays a significant role in detecting of plagiarism texts, for future works, providing corpora that conform to standard criteria for assessing candidate retrieval methods is strongly recommended. Also regarding the fact that the deceitful attempts to conceal plagiarism by making changes, then in text alignment, the use of string-based methods alone cannot be used as an efficient method, but also machine is required to have a comprehensive understanding of texts to extract the concealed parts . Hence, it is suggested to use methods that can be helpful to machine to understand meaning.

References

[1] R. Lukashenko, V. Graudina, and J. Grundspenkis, "Computer-Based Plagiarism Detection Methods And Tools: An Overview", in Proceedings of the International Conference on Computer Systems and Technologies, 2007, pp. 40.

[2] B. Schwartz, "Google's Search Knows About Over 130 Trillion Pages. Available. https://searchengineland.com/googles-search-indexes-hits-130-trillion-pages-documents-263378.

[3] A. Barrón-Cedeño, P. Gupta, and P. Rosso, "Methods For Cross-Language Plagiarism Detection", Knowledge-Based Systems, vol. 50, 2013, pp. 211-217.

[4] R. M. A. Nawab, "Mono-Lingual Paraphrased Text Reuse And Plagiarism Detection", University of Sheffield, 2012.

[5] F. Safi-Esfahani, S. Rakian, and M. Nadimi-Shahraki, "English-Persian Plagiarism Detection Based On A Semantic Approach", Journal of AI and Data Mining, vol. 5, 2017, pp. 275-284.

[6] A. M. E. T. Ali, H. M. D. Abdulla, and V. Snasel, "Overview And Comparison Of Plagiarism Detection Tools", in 11th Annual International Workshop on Databases, Texts, Specifications, and Objects, Písek, Czech Republic, 20-22 April, 2011, pp. 161-172.

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

[8] M. Mansoorizadeh, T. Rahgooy, and I. Hamedan, "Persian Plagiarism Detection Using Sentence Correlations", in Forum for Information Retrieval Evaluation, India, 7-10 December, 2016, pp. 163-166.

[9] F. Esteki and F. S. Esfahani, "A Plagiarism Detection Approach Based On SVM For Persian Texts", in Forum for Information Retrieval Evaluation, India, 7-10 December, 2016, pp. 149-153.
[10] H. Asghari, S. Mohtaj, O. Fatemi, H. Faili, P. Rosso, and M. Potthast, "Algorithms And Corpora For Persian Plagiarism Detection", in Forum for Information Retrieval Evaluation, India, 7-10 December, 2016, pp. 61-79.
[11] S. Rakian, E. F. SAFI, and H. Rastegari, "A Persian Fuzzy Plagiarism Detection Approach", Journal of Information Systems and Telecommunication, Vol. 3, No. 3, July-September 2015, pp. 182-190.
[12] P. Mahdavi, Z. Siadati, and F. Yaghmaee, "Automatic External Persian Plagiarism Detection Using Vector Space Model", in 4th International Conference on Computer and Knowledge Engineering, Iran, 29-30 October, 2014, pp. 697-702.
[13] S. Lazemi and H. Ebrahimpour-komleh, "Persian Plagiarism Detection Using Structural-Semantic Features", in 3Rd International Conference on Pattern Recognition and Image Analysis, Iran, 19 March, in Farsi, https://www.civilica.com/Paper-IPRIA03-IPRIA03_006.html, 2017.
[14] K. Khoshnavataher, V. Zarrabi, S. Mohtaj, and H. Asghari, "Developing Monolingual Persian Corpus For Extrinsic Plagiarism Detection Using Artificial Obfuscation",in Notebook for PAN at Conference and Labs of the Evaluation forum, Toulouse, France, 8-11September, 2015, pp.7.
[15] L. Soghra, H. Ebrahimpour-Komleh, and N. Noroozi, "Persian Plagiarism Detection Using CNNs", in 8th International Conference on Computer and Knowledge Engineering, Iran, 25 October, 2018, pp. 171-175.
[16] E. Gharavi, K. Bijari, K. Zahirnia, and H. Veisi, "A Deep Learning Approach To Persian Plagiarism Detection", in Forum for Information Retrieval Evaluation, India, 7-10 December, 2016, pp. 154-159.
[17] M. Momtaz, K. Bijari, M. Salehi, and H. Veisi, "Graph-Based Approach To Text Alignment For Plagiarism Detection In Persian Documents", in Forum for Information Retrieval Evaluation, India, 7-10 December, 2016, pp. 176-179.
[18] H. Ahangarbahan and G. A. Montazer, "A Mixed Fuzzy Similarity Approach To Detect Plagiarism In Persian Texts", in International Work-Conference on Artificial Neural Networks, Cham, 10 Jun ,Springer, 2015, pp. 525-534.
[19] K. Taghva, R. Beckley, and M. Sadeh, "A List Of Farsi Stopwords", vol. 7, 2003.
[20] Y. Matsuo and M. Ishizuka, "Keyword Extraction From A Single Document Using Word Co-Occurrence Statistical Information", International Journal on Artificial Intelligence Tools, vol. 13, 2004, pp. 157-169.
[21] D.-Y. Lee, K.-R. Kim, and H.-G. Cho, "A New Extraction Algorithm For Hierarchical Keyword Using Text Social Network", in Information Science and Applications, Springer, 2016, pp. 903-912.
[22] S. Lazmi, H. Ebrahimpour-Komleh, and N. Noroozi, "PAKE: A Supervised Approach For Persian Automatic Keyword Extraction Using Statistical Features", The International Conference on Contemporary Issues in Data Science, Iran, 6-8 March, 2019, Accepted, All Accepted Short Papers in CiDaS 2019 Will be Published in the Springer's SN Applied Sciences.
[23] D. Jurafsky and H. James, "Speech And Language Processing An Introduction To Natural Language Processing, Computational Linguistics, And Speech", 2000.
[24] S. Kübler, R. McDonald, and J. Nivre, "Dependency Parsing", Synthesis Lectures on Human Language Technologies, vol. 1, 2009, pp. 1-127.
[25] S. A. M. Falavarjani and G. Ghassem-Sani, "Advantages Of Dependency Parsing For Free Word Order Natural Languages", in International Conference on Current Trends in Theory and Practice of Informatics, Czech Republic, 24-29 January, 2015, pp. 511-518.
[26] R. McDonald, K. Crammer, and F. C. Pereira, "Spanning Tree Methods For Discriminative Training Of Dependency Parsers", Technical Reports, 2006, p. 55.
[27] R. McDonald, F. Pereira, K. Ribarov, and J. Hajič, "Non-Projective Dependency Parsing Using Spanning Tree Algorithms", in Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Canada, 6 -8 October, 2005, pp. 523-530.
[28] P. Zhao, J. Han, and Y. Sun, "P-Rank: A Comprehensive Structural Similarity Measure Over Information Networks", in Proceedings of the 18th ACM Conference on Information and Knowledge Management, China, 2-6 November, 2009, pp. 553-562.
[29] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, "Deep Semantic Role Labeling: What Works And What's Next", in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Canada, 30 July-4 August, 2017, pp. 473-483.
[30] J. D. Choi and M. Palmer, "Transition-Based Semantic Role Labeling Using Predicate Argument Clustering", in Proceedings of the ACL Workshop on Relational Models of Semantics, USA, 23 June, 2011, pp. 37-45.
[31] T. Cohn and P. Blunsom, "Semantic Role Labelling With Tree Conditional Random Fields", in Proceedings of the Ninth Conference on Computational Natural Language Learning, 2005, USA, 29-30 June, pp. 169-172.
[32] C. Sutton and A. McCallum, "An Introduction To Conditional Random Fields", Foundations and Trends® in Machine Learning, vol. 4, 2012, pp. 267-373.
[33] M. S. Rasooli, M. Kouhestani, and A. Moloodi, "Development Of A Persian Syntactic Dependency Treebank", in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, 9- 14 Jun, 2013, pp. 306-314.
[34] M. Seraji, C. Jahani, B. Megyesi, and J. Nivre, "A Persian Treebank With Stanford Typed Dependencies", in The 9th International Conference on Language Resources and Evaluation, Iceland 26-31 May, 2014, pp. 796-801.
[35] A. Mirzaei and A. Moloodi, "First Persian Semantic Role Labeling Corpus", Language Science, vol. 3, 2015.
[36] M. R. Sharifabadi and S. A. Eftekhari, "Mahak Samim: A Corpus of Persian Academic Texts For Evaluating Plagiarism Detection Systems", in Forum for Information Retrieval Evaluation, India, 7-10 December, 2016, pp. 190-192.
[37] M. Potthast, A. Eiselt, L. A. Barrón Cedeño, B. Stein, and P. Rosso, "Overview Of the 3rd International Competition On Plagiarism Detection", in CEUR workshop proceedings, 2011.
[38] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors For 157 Languages", arXiv preprint arXiv:1802.06893, 2018.
[39] M. Shamsfard, A. Hesabi, H. Fadaei, N. Mansoory, A. Famian, S. Bagherbeigi, et al., "Semi Automatic Development Of farsnet; The Persian Wordnet", in Proceedings of 5th Global WordNet Conference, India, 31 Jan-4 February, 2010.
[40] S. Lazmi, H. Ebrahimpour-Komleh, "Feature Engineering In Persian Dependency Parser", Journal of AI and Data Mining, 2018.
[41] S. Lazmi, H. Ebrahimpour-Komleh, and N. Noroozi, "Improving Persian Dependency-Based Semantic Role Labeling Using Semantic And Structural Relations", The International Conference on Pattern Recognition and Image Analysis, Iran, 6-7 March, 2019.