Predicting Type of Obfuscation to Enhance Text Alignment Algorithms

Fatemeh Mashhadirajab(✉) and Mehrnoush Shamsfard

NLP Research Lab, Faculty of Computer Science and Engineering,
Shahid Beheshti University, Tehran, Iran
f.mashhadirajab@mail.sbu.ac.ir, m-shams@sbu.ac.ir

Abstract. Plagiarism detection can be divided into source retrieval and text alignment subtasks. The text alignment subtask extracts all plagiarized passages from a given a pair of documents. The challenge is to identify passages of text that have been obfuscated. A given pair of documents could contain different types of obfuscation. Information about the type of obfuscation in a document pair could be useful for text alignment algorithms in plagiarism detection systems when choosing the most suitable algorithm for each type. The current paper describes a proposed approach to improve text alignment algorithms. The SVM neural network is used for classification of documents according to the type of obfuscation strategy used in the document pair. The parameter values in the proposed text alignment algorithm are set based on the type of obfuscation detected. The results of the proposed algorithm for Persian Plagdet corpus 2016 are shown. The proposed algorithm ranked first in the Persian Plagdet 2016 competition from among nine participant teams.

Keywords: Plagiarism detection · Text alignment · SVM neural network

1 Introduction

Because human beings use varied and sometimes sophisticated methods to conceal plagiarism, systems with highly-varied algorithms have been designed for automatic discovery of plagiarism types. Many plagiarism detection systems have been developed [1–3] and plagiarism detection has become a task in the PAN competition[1] held every year since 2009. At the PAN competition, the plagiarism detection task is divided into source retrieval and text alignment subtasks. The source retrieval task is to retrieve documents similar to the suspicious document from a set of source documents and the text alignment task is to extract all plagiarized passages from a given source-suspicious document pair [4]. Figure 1 shows different parts of a plagiarism detection system. At the PAN competition, the text alignment algorithms are assessed by evaluation corpora that contain different types of obfuscation. In the PAN 2013–2014 competitions, for example, the evaluation corpus consisted of the None, Random, Translation and Summary obfuscation types [5]. The Persian Plagdet 2016 evaluation corpus included the None (exact copy) and Artificial (random) types created by automatic paraphrasing

[1] http://pan.webis.de/.

P. Majumder et al. (Eds.): FIRE 2016 Workshop, LNCS 10478, pp. 80–93, 2018.
https://doi.org/10.1007/978-3-319-73606-8_6

technology for word addition, deletion and shuffling, semantic word variation and Simulated obfuscation created by crowdsourcing [6]. PAN corpus 2010 contained None, Artificial and Simulated obfuscation [7]. Different methods could be used to detect the type of obfuscation to in plagiarized passages and or different sets of values could be defined for each parameter set to improve performance of the text alignment algorithm. In PAN text alignment corpora, it is assumed that just one type of obfuscation is employed in each document pair [7]; thus, most participants try to predict the type of obfuscation strategy used in a document pair and detect similarities based on the predicted type [8–10].

The current paper describes the proposed algorithm, which was submitted to the Persian Plagdet 2016 competition[2]. Persian Plagdet 2016 is a subtask of the PAN Fire 2016 competition[3] that is held for the Persian language and the text alignment algorithms were evaluated on a Persian corpus [6]. The proposed approach includes four stages: preprocessing, seeding, extension and filtering. After preprocessing, a neural network is used to detect the type of obfuscation in each document pair. The parameters in the text alignment algorithm are set based on the detected type of obfuscation. In the seeding stage, similar sentence pairs are extracted from the given document pair. In extension, the seeds are extended to detect the longest similar passages from the suspicious and source documents. Small or overlapping passages are removed in the filtering stage. The rest of the paper is organized as follows. Section 2 reviews published work related to the text alignment task. Section 3 explains the proposed algorithm with a focus on the obfuscation type detection module. Section 4 describes the evaluation framework, training and test datasets and the performance measures used for evaluation. The experimental results are discussed in Sect. 4 and the conclusion and future work are reported in Sect. 5.

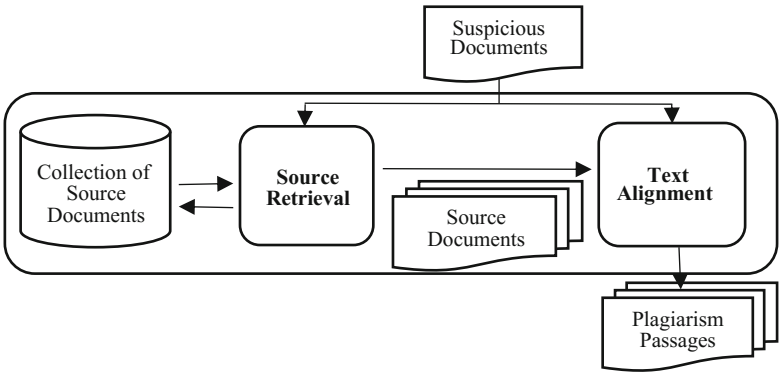


Fig. 1. Plagiarism detection systems.

[2] http://ictrc.ac.ir/plagdet/.

[3] http://fire.irsi.res.in/fire/2016/home.

2 Related Work

The text alignment task is to extract all plagiarized passages from a given a pair of documents. The challenge with this task is to identify passages of text that have been obfuscated. Text alignment algorithms usually consist of three phases: seeding, extension and filtering [5]. Before seeding, some algorithms contain a preprocessing stage. In preprocessing, non-alphanumeric characters and stop words are removed and words are stemmed. The task of seeding is to extract small fragments from a source document and a suspicious document that are similar. At the end of this stage, a large collection of similar fragments will be constructed that are called "seeds". The small fragments could be character [11], word [8, 12] or skip word [13] n-grams or a combination of n-grams [11, 14]. Some algorithms use sentences as small fragments [15–19]. To extract similar fragments, some algorithms use exact matching [8, 11, 13, 14, 18]. Alvi et al. [11] used the Rabin-Karp algorithm and Sanchez-Perez et al. [15, 16] used cosine similarity and the Dice coefficient in a vector space model (VSM) for matching. In the extension phase, the set of seeds is extended to larger fragments that are reported as plagiarism cases. Plagiarism cases are fragments in the given source-suspicious document pair that are similar. Many algorithms have been applied in the extension stage. For example, for the text alignment task in the PAN competition, the extension algorithms were either rule-based, dynamic programming or clustering-based approaches [5]. Alvi et al. [11] used rule-based approaches to merge matching pairs. They divided matching pairs into four categories based on their proximity to one another. Two mapping pairs could be merged if they belong to a specific category. Glinos et al. [8] applied the Smith-Waterman [10] dynamic programming algorithm to find maximal length passages. Ehsan et al. [20] also used the dynamic algorithms to detect alignments between n-grams. Abnar et al. [21] and Gross et al. [13] applied clustering algorithms to detect maximally-aligned sequences of document pairs. The extension phase in the Miguel algorithm [15, 16] consisted of clustering and validation parts. In this approach, the seeds were clustered based on a distance *maxgap* and then the similarity in each cluster was computed. If the similarity in a cluster fell below a threshold value, it was discarded. In the filtering phase, some short [8, 11, 13, 15, 16, 19, 21] or overlapping [15, 16, 19] passages are removed. The PAN evaluation corpora contain the following: none obfuscation, random obfuscation, translation obfuscation and summary obfuscation. Participants at the PAN competition use various methods to predict the type of obfuscation and detect similarities based on the predicted type. At PAN 2014, the Glinos algorithm [8] divided all plagiarism documents into order-based and non-order based categories. The order-based plagiarism detects none and random obfuscations. The non-order-based plagiarism detects translation and summary obfuscation. The Smith-Waterman algorithm [10] is used to detect aligned sequences of document pairs and order-based plagiarism cases. If no aligned sequences are found, the document pairs are given to the clustering component to detect non-order-based plagiarism cases. Sanchez-Perez et al. [15, 16] categorized the document pairs of PAN 2014 corpus into verbatim, summary

and other plagiarism categories and set the parameters based on these categories. They used the longest common substring algorithm to find every single common sequence of word (th-verbatim). If at least one verbatim case has been found, the document pair is considered to be verbatim plagiarism. If no verbatim cases are found and the length of plagiarism fragment in the suspicious document is much smaller than the length of the source fragments, the document pair is considered to be summary plagiarism; otherwise the document pair is considered to be another plagiarism case. Palkovskii and Belov [9] used a graphical clustering algorithm to detect the type of plagiarism in a document pair. They classified the document pairs of the PAN 2014 text alignment corpus into verbatim, random, summary type and undefined plagiarism. They then set the parameters based on the detected type of plagiarism. Persian plagdet 2016 corpus has three types of obfuscation: none, random and simulated. In the proposed approach, the document pairs of the Persian plagdet 2016 corpus are classified into verbatim and simulated plagiarism categories. The SVM neural network was used to detect the type of plagiarism in the proposed algorithm and was trained by type of obfuscation in the Persian Plagdet 2016 training corpus. The parameters were then set based on the detected type of plagiarism. The proposed algorithm was entered into the Persian Plagdet 2016 competition and was evaluated using their evaluation corpus. Table 1 summarizes the other approaches used by participants in the Persian Plagdet 2016 competition using the four stages mentioned above.

Table 1. The summarizing of some submitted approaches at Persian Plagdet corpus 2016

			Our approach	Talebpour et al. [22]	Minaei and Niknam [12]	Ehsan and Shakery [19]	Momtaz et al. [23]	Esteki and Esfahani [24]	Gharavi et al. [25]	Mansoori and Rahgooy [26]	Gillam and Vartapetiance [27]
Pre-processing	Stop_words removal		✓	✓		✓	✓	✓	✓		
	Stemming		✓	✓				✓			
	POS tagging			✓							
	Special character removal		✓	✓		✓		✓			
	Tokenizing		✓	✓		✓					
	FarsNet		✓	✓				✓			
Seeding	Small fragment	Character-ngram									
		Word-ngram		✓	✓	✓	✓				✓
		Sentence	✓			✓	✓	✓	✓	✓	
		Bag of words							✓	✓	
	Matching method	W2V, cosin							✓		
		VSM, cosin	✓							✓	
		levenshtein						✓			
		Jaccard						✓	✓		
		Dice						✓			
		LCS						✓			
		SVM						✓			
		Graph matching		✓			✓				
Extension	Rule based		✓	✓	✓	✓	✓				✓
	Dynamic programming					✓					
	Clustering		✓					✓			
Filtering	Small passage removal		✓		✓	✓					✓
	Overlapping removal		✓		✓	✓	✓				

3 The Proposed Approach

The proposed text alignment algorithm, like many other text alignment algorithms [5], includes preprocessing, seeding, extension and filtering stages. Each stages is explained below. Figure 2 is an overall scheme of the proposed text alignment algorithm that shows these four stages.

3.1 Preprocessing

In the preprocessing stage, the text is first segmented into sentences and then tokenized using STeP_1 [28]. The stop words [29] are removed and inflectional and derivational stems of the tokens are extracted and restored by STeP_1. Preprocessing is done for a suspicious document and a source document. These sentences will be given in the seeding stage.

3.2 Seeding

The seeding stage extracts similar seed sentence pairs from source and suspicious documents. The proposed method was initially based on the method introduced by Sanchez-Perez et al. [15]. The method was then expanded using the SVM neural net to predict the obfuscation type and adjust the parameters to gain better results. Based on the VSM method, first the *tf-idf* vector is calculated for all sentences of the suspicious and source documents in which *tf* is the term frequency in the corresponding sentence and *idf* is the inverse sentence frequency. The similarity of each sentence pair in the suspicious and source documents is calculated using the cosine measure and Dice coefficient as in Eqs. (1), (2) and (3):

$$cosine\left(susp_i, src_j\right) = \frac{susp_i.src_j}{|susp_i||src_j|} \tag{1}$$

$$Dice\left(susp_i, src_j\right) = \frac{2\left|\delta(susp_i).\delta(src_j)\right|}{|\delta(susp_i)|^2 + \left|\delta(src_j)\right|^2} \tag{2}$$

$$\delta(x) = \begin{cases} 1 & if\ x \neq \varnothing \\ 0 & otherwise \end{cases} \tag{3}$$

where $susp_i$ is the vector of the ith sentence from the suspicious document, src_j is the vector from the jth sentence of the source and |.| is the Euclidean length.

The cosine measure and Dice coefficient are calculated for all pairs of sentences and if the similarity of $susp_i$ and src_j is greater than the threshold of 0.3 (chosen based on Sanchez-Perez et al. [15]), this pair of sentences are considered to be a seed. For pairs of sentences having a similarity of more than 0.1 and less than 0.3 (based on experimentation), the semantic similarity is computed. The SVM neural network[4] is used to

[4] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

predict the type of obfuscation strategy used in the document pairs. To create a SVM input vector, the cosine similarity measure between all sentence pairs was computed for the given suspicious and source documents and then all values were divided into 8 clusters. Values between 0.2 and 0.3 are placed in the first cluster[5], values between 0.3 and 0.4 are placed in the second cluster and so on. An 8-bit vector is considered for each document pair. Each bit of the vector is set as follows:

$$v_i = \begin{cases} 0 & \text{if } cluster_i = \varnothing \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

Where v_i is the ith bit of the input vector and $cluster_i$ is the ith cluster that contains values between $\frac{i+1}{10}$ and $\frac{i+2}{10}$. For example, for $i = 1$, if no cosine similarity value exists between 0.2 and 0.3 in all sentence pairs of a document pair, then $v_1 = 0$; otherwise, $v_1 = 1$. These vectors were created for all document pairs in the Persian Plagdet training dataset 2016 and then SVM neural network was trained by these vectors.

The threshold is set for semantic similarity based on the type of obfuscation. If the SVM predicts that the type of obfuscation is simulated, then the threshold will be equal 0.2; otherwise it will be equal to 0.3. To calculate the semantic similarity, FarsNet [30] is used to extract synsets of each term and STeP_1 is used to extract inflectional and derivational stems of each term. For each term, a set of words called $\varphi(\omega)$ is considered as shown in Fig. 3. For each w_i in vector $susp_i$, if $\varphi(w_i)$ overlaps $\varphi(\acute{w}_j)$ of vector src_j, w_i of vector $susp_i$ is replaced by $\acute{w}_j$ of vector src_j. The similarity of the cosine and Dice coefficients are calculated for the resulting vectors and the similarities are averaged between the results at this stage and the results of the cosine and Dice coefficients in the previous stage; if the average is greater than the threshold (0.2 for simulated and 0.3 for artificial and noun), the pair of $susp_i$ and src_j are considered to be seeds. The set of seeds obtained in this stage then enter the extension stage.

3.3 Extension

The purpose of the extension stage is the extraction of the longest similar passages from the suspicious and source documents. Figure 2 shows that the extension consists of clustering and validation parts. In the clustering stage, the seeds are clustered into passages that are not separated by more than a *maxgap* number of sentences. The *maxgap* is 4 in the proposed implementation. In the validation stage, those pairs of passages created in the clustering stage that are not similar are removed. The thresholds of all stages are shown in Table 2. For the extension stage, the method proposed by Sanchez-Perez et al. [15] was extended and enhanced. In the validation stage, the semantic similarity measure was used instead of the cosine measure to determine the similarity between pairs of passages.

[5] In clustering, the values between 0 and 0.1 are removed because almost all documents pairs contain at least one value between 0 and 0.1.

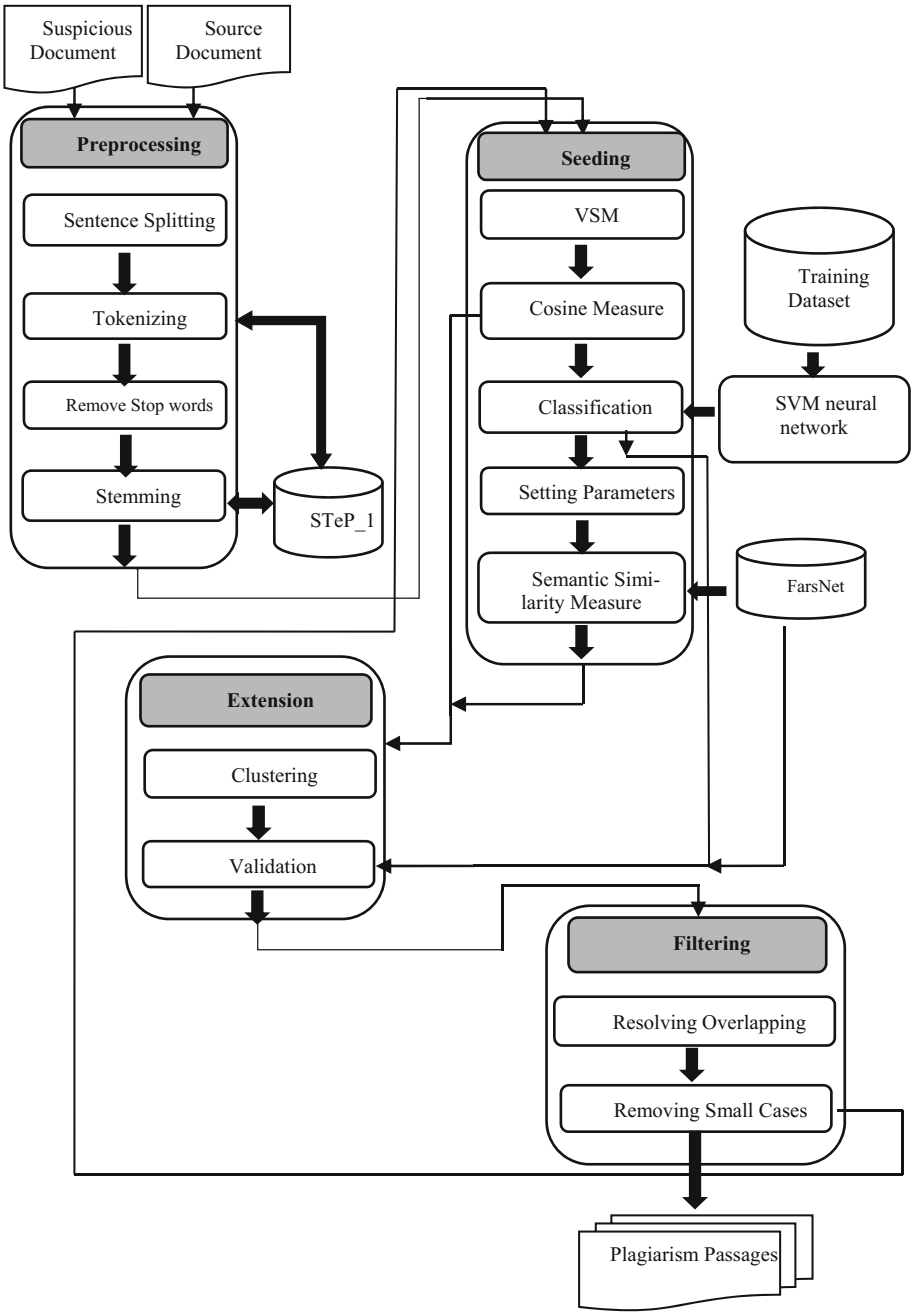


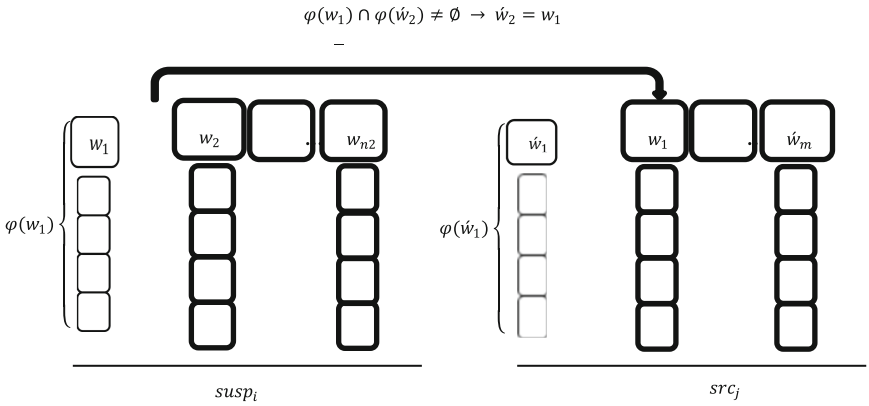
Fig. 2. The proposed text alignment algorithm.

3.4 Filtering

The filtering stage removes passages that either overlap or are too short. To remove overlapping passages the method proposed by Sanchez-Perez et al. [15] is used. To remove passages that are too short, a recursive algorithm is used. If the length of a passage

Table 2. Parameters settings

Parameter		Value
Threshold-cosine similarity		0.3
Threshold-dice similarity		0.3
MaxGap		4
Min PlagPassage length		100
Simulated	Threshold-semantic similarity	0.2
	Threshold-validation	0.2
Artificial and noun	Threshold-semantic similarity	0.3
	Threshold-validation	0.3
Conditions to calculate the semantic similarity of two sentences		0.1 < cosine similarity < 0.3



$$for\ (w_i = 1\ \ to\ \ n\ \ \ in\ \ \ susp_{terms})$$
$$for\ (\acute{w}_j = 1\ \ to\ \ \ m\ \ in\ \ \ src_{terms})$$
$$if\ \ \ (\varphi(w_i)\ \cap\ \varphi(\acute{w}_j))\ \neq \emptyset$$
$$\acute{w}_j = w_i$$
$$Where\ \varphi(\omega) =\ the\ set\ of\ \ inflectional\ and$$
$$derivational\ stems\ and\ Synsets\ of\ \omega.$$

Fig. 3. New vectors for semantic similarity.

is less than a 100 characters, it is first assumed that other seeds exist in the passage, but have not been identified. The semantic similarity threshold is thus reduced to −0.06 and goes back to the seeding stage. The new seeds are then extracted based on the new threshold (old threshold - 0.06); and all stages are repeated to remove passages that are too short. If the passage is less than 100 characters again, this passage will be removed.

4 Experiments

4.1 Dataset

The Persian Plagdet evaluation corpus 2016 includes the none (exact copy), artificial (random) and simulated obfuscation categories. Artificial obfuscation consists of word additions, deletions and shuffling, semantic word variation and POS-preserving word shuffling. A crowdsourcing approach was used to create simulated obfuscation [6]. This dataset contains 5830 documents organized by the ICT Research Institute, ACECR, under the partial support of Vice Presidency for Science and Technology of Iran. Table 3 shows some of the statistics of this corpus.

Table 3. Corpus statistics [6]

Corpus statistics		
Entire corpus	Number of documents	5830
	Number of plagiarism cases	4118
Document purpose	Source documents	48%
	Suspicious documents	52%
Document length	Short (1–500 words)	35%
	Medium (500–2500 words)	59%
	Long (2500–21000 words)	6%
Plagiarism per document	Small (5%–20%)	57%
	Medium (21%–50%)	15%
	Much (51%–80%)	18%
	Entirely (>80%)	10%
Obfuscation	Number of cases	1628
	None	11%
	Artificial	81%
	Simulated	8%

4.2 Performance Measures

The evaluation measures used in this competition are the recall, precision, granularity and plagdet scores [5]. These criteria are defined in accordance with the following equations:

$$prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(s\prod \mathrm{r})|}{|r|} \tag{5}$$

$$rec(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R}(s\prod \mathrm{r})|}{|s|} \tag{6}$$

$$\text{Where} \quad s\prod \mathrm{r} = \begin{cases} s\cap r & \text{if r detects s,} \\ \varnothing & \text{otherwise.} \end{cases} \tag{7}$$

Where S is the set of plagiarism cases in the corpus and R is the set of cases of plagiarism detected by the algorithm. Each plagiarism case is defined by $s = s_{plg}, d_{plg}, s_{src}, d_{src}$, where s ∈ S, d_{plg} is the suspicious document, d_{src} is the source document, s_{plg} is the plagiarized segment, s_{src} is the source segment and r ∈ R is the discovered plagiarism case.

Granularity addresses overlapping or multiple detection for one plagiarism case and is defined as:

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_s| \tag{8}$$

Where $S_R = \{s|s \in S \bigwedge \exists r \in R : r \text{ detects } s\}$ and $R_s = \{r|r \in R \bigwedge r \text{ detects } s\}$.

These measures combine into a single plagdet score as:

$$plagdet(S,R) = \frac{F_1}{\log_2(1+grand(S,R))} \tag{9}$$

Where F1 is the average harmonic weight of precision and recall [5].

All these metrics are computed at character level. The plagdet measure used to evaluate the software submitted to Persian Plagdet 2016 is a combination of character level precision, recall and granularity. The plagiarism detection systems were ranked according to their character level performance. Detection performance could be also measure at the case and document levels. Potthast et al. [5] measured detection performance at the case level and the document level. The case level fixes the minimum precision and recall with which a plagiarism case must be detected. The document level

Table 4. The proposed algorithm performance at case level and document level in Persian Plagdet training dataset 2016

	Parameter	Value
Case_level	Recall	0.9769
	Precision	0.9574
	F-measure	0.9671
Document_level	Recall	0.9801
	Precision	0.9705
	F-measure	0.9753

disregards whether or not all plagiarism cases in a document have been detected as long as a significant portion of one has been detected. Table 4 shows the results of the proposed algorithm for the training Persian plagdet corpus at the case and document levels.

4.3 Results

The proposed algorithm was submitted to the Persian Plagdet 2016 competition and compared with other participants approaches on Persian PlagDet corpus 2016 [6] based on the PAN evaluation setup [7, 31, 32]. Table 5 shows the overall performance and runtimes of the nine submitted text alignment approaches. As seen, the proposed approach achieved the highest PlagDet score for the complete corpus and ranked first. Gharavi et al. [25] completed the entire corpus in only 00:01:03 min and the proposed approach required 02:22:48. Table 6 shows the results of the proposed algorithm on all obfuscation strategies in the training dataset. Table 6 column P_1 shows the results of the proposed algorithm for type of obfuscation in the training dataset where semantic similarity measure is not used. The P_2 column shows the algorithm results using the semantic similarity measure. Column P_3 shows the results of the proposed algorithm after adding the criterion of semantic similarity and adjusting the parameters based on detected type of obfuscation using a neural network. As seen in column P_2, adding the semantic similarity criteria improved the recall for the types of obfuscation in training corpus, but the precision declined in some cases. In column P_3, adding a neural network to the system for the diagnosis of type of obfuscation and parameter settings based on the type of obfuscation improved precision and recall dramatically for all types of obfuscation.

Table 7 shows the results of the submitted algorithms on obfuscation types in the test corpus. Gharavi et al. [25] ranked first for no obfuscation and the proposed approach ranked first for both artificial and simulated plagiarism. The proposed approach ranked first for best recall across all parts of the corpus.

Table 5. The text alignment algorithms performance on Persian Plagdet corpus 2016 [6]

Rank/team	Runtime (h:m:s)	Recall	Precision	Granularity	F-measure	PlagDet
1 Mashhadirajab	02:22:48	**0.9191**	0.9268	1.0014	**0.9230**	**0.9220**
2 Gharavi	**00:01:03**	0.8582	0.9592	**1**	0.9059	0.9059
3 Momtaz	00:16:08	0.8504	0.8925	**1**	0.8710	0.8710
4 Minaei	00:01:33	0.7960	0.9203	1.0396	0.8536	0.8301
5 Esteki	00:44:03	0.7012	0.9333	**1**	0.8008	0.8008
6 Talebpour	02:24:19	0.8361	**0.9638**	1.2275	0.8954	0.7749
7 Ehsan	00:24:08	0.7049	0.7496	**1**	0.7266	0.7266
8 Gillam	21:08:54	0.4140	0.7548	1.5280	0.5347	0.3996
9 Mansourizadeh	00:02:38	0.8065	0.9000	3.5369	0.8507	0.3899

Table 6. The proposed algorithm on types of obfuscation in Plagdet training dataset 2016

Obfuscation	Parameter	P_1	P_2	P_3
None	Plagdet	0.94	0.96	0.97
	Recall	0.96	0.98	0.99
	Precision	0.92	0.95	0.94
	Granularity	1	1	1
Artificial	Plagdet	0.81	0.84	0.94
	Recall	0.78	0.84	0.93
	Precision	0.85	0.84	0.94
	Granularity	1	1	1
Simulated	Plagdet	0.55	0.69	0.86
	Recall	0.41	0.61	0.83
	Precision	0.84	0.80	0.91
	Granularity	1	1	1

Table 7. The algorithm submitted based on types of obfuscation in Persian Plagdet test dataset 2016 [6]

Team	No obfuscation				Artificial obfuscation				Simulated obfuscation			
	Recall	Precision	Granularity	PlagDet	Recall	Precision	Granularity	PlagDet	Recall	Precision	Granularity	PlagDet
Mashhadirajab	**0.9939**	0.9403	**1**	0.9663	**0.9473**	0.9416	1.0006	**0.9440**	**0.8045**	0.9336	1.0047	**0.8613**
Gharavi	0.9825	0.9762	**1**	**0.9793**	0.8979	0.9647	**1**	0.9301	0.6895	**0.9682**	**1**	0.8054
Momtaz	0.9532	0.8965	**1**	0.9240	0.9019	0.8979	**1**	0.8999	0.6534	0.9119	**1**	0.7613
Minaei	0.9659	0.8663	1.0113	0.9060	0.8514	0.9324	1.0240	0.8750	0.5618	0.9110	1.1173	0.6422
Esteki	0.9781	0.9689	**1**	0.9735	0.7758	0.9473	**1**	0.8530	0.3683	0.8982	**1**	0.5224
Talebpour	0.9755	**0.9775**	**1**	0.9765	0.8971	**0.9674**	1.2074	0.8149	0.5961	0.9582	1.4111	0.5788
Ehsan	0.8065	0.7333	**1**	0.7682	0.7542	0.7573	**1**	0.7557	0.5154	0.7858	**1**	0.6225
Gillam	0.7588	0.6257	1.4857	0.5221	0.4236	0.7744	1.5351	0.4080	0.2564	0.7748	1.5308	0.2876
Mansourizadeh	0.9615	0.8821	3.7740	0.4080	0.8891	0.9129	3.6011	0.4091	0.4944	0.8791	3.1494	0.3082

5 Conclusions and Future Work

The current work described the proposed text alignment approach and compared it to the eight other approaches submitted to the Persian Plagdet 2016 competition. The detection performance for all nine approaches has been provided. The proposed method consists of four stages used to aligned the passages of a given document pair. The SVM neural network was used to identify the type of obfuscation and set the parameters on the basis of obfuscation. The results showed that this was effective for improving precision and recall. Although the proposed approach ranked first for performance compared with other participants, the runtime should be decreased to make it much shorter than two hours. Future study will focus on improving the runtime and the semantic similarity measure in the seeding stage. The most of runtime is spent on the preprocessing and seeding stages to detect similar sentences. So, in order to improve runtime, we intend to use deep learning techniques instead of using multiple tools to discover similarities.

References

1. Fiedler, R., Kaner, C.: Plagiarism detection services: how well do they actually perform. IEEE Technol. Soc. Mag. **29**, 37–43 (2010)
2. Alzahrani, M., Salim, N., Abraham, A.: Understanding plagiarism linguistic patterns, textual features, and detection methods. IEEE Trans. Syst. Man Cybern.—Part C Appl. Rev. **42**(2), 133–149 (2012)
3. Ali, A.M.E.T., Abdulla, H.M.D., Snasel, V.: Survey of plagiarism detection methods. In: IEEE Fifth Asia Modelling Symposium (AMS), pp. 39–42 (2011)
4. Potthast, M., Göring, S.: Towards data submissions for shared tasks: first experiences for the task of text alignment. In: Working Notes Papers of the CLEF 2015 Evaluation Labs, CEUR Workshop Proceedings (2015). ISSN 1613-0073
5. Potthast, M., Hagen, M., Beyer, A., Busse, M., et al.: Overview of the 6th international competition on plagiarism detection. In: Working Notes for CLEF 2014 Conference, Sheffield, UK, 15–18 September, CEUR Workshop Proceedings, vol. 1180, pp. 845–876 (2014). CEUR-WS.org
6. Asghari, H., Mohtaj, S., Fatemi, O., Faili, H., et al.: Overview of the PAN@FIRE2016 shared task on persian plagiarism detection and text alignment corpus construction. In: Notebook Papers of FIRE 2016, FIRE-2016 (2016). CEUR-WS.org
7. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. In: 23rd International Conference on Computational Linguistics (COLING 2010), pp. 997–1005 (2010)
8. Glinos, D.: A hybrid architecture for plagiarism detection. In: Notebook for PAN at CLEF 2014. CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers, Sheffield, UK, 15–18 September (2014). CEUR-WS.org. ISSN 1613-0073
9. Palkovskii, Y., Belov, A.: Developing high-resolution universal multi-type n-gram plagiarism detector. In: Notebook for PAN at CLEF 2014. CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers, Sheffield, UK, 15–18 September (2014). CEUR-WS.org. ISSN 1613-0073
10. Smith, T., Waterman, M.: Identification of common molecular subsequences. J. Mol. Biol. **147**(1), 195–197 (1981)
11. Alvi, F., Stevenson, M., Clough, P.: Hashing and merging heuristics for text reuse detection. In: Notebook for PAN at CLEF (2014)
12. Minaei, B., Niknam, M.: An n-gram based method for nearly copy detection in plagiarism systems. In: Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
13. Gross, P., Modaresi, P.: Plagiarism alignment detection by merging context seeds. In: Notebook for PAN at CLEF (2014)
14. Torrejón, D.A.R., Ramos, J.M.M.: CoReMo 2.3 plagiarism detector text alignment module. In: Notebook for PAN at CLEF (2014)
15. Sanchez-Perez, M.A., Gelbukh, A.F., Sidorov, G.: Dynamically adjustable approach through obfuscation type recognition. In: Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, 8–11 September 2015, CEUR Workshop Proceedings, vol. 1391 (2015). CEUR-WS.org
16. Sanchez-Perez, M., Sidorov, G., Gelbukh, A.: The winning approach to text alignment for text reuse detection at PAN 2014. In: Notebook for PAN at CLEF 2014, Sheffield, UK, 15–18 September, CEUR Workshop Proceedings, vol. 1180, pp. 1004–1011 (2014). CEUR-WS.org. ISSN 1613-0073

17. Kong, L., Han, Y., Han, Z., Yu, H., Wang, Q., Zhang, T., Qi, H.: Source retrieval based on learning to rank and text alignment based on plagiarism type recognition for plagiarism detection. In: Notebook for PAN at CLEF (2014)
18. Shrestha, P., Maharjan, S., Solorio, T.: Machine translation evaluation metric for text alignment. In: Notebook for PAN at CLEF (2014)
19. Ehsan, N., Shakery, A.: A pairwise document analysis approach for monolingual plagiarism detection. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2014). CEUR-WS.org
20. Ehsan, N., Tompa, F.W., Shakery, A.: Using a dictionary and n-gram alignment to improve fine-grained cross-language plagiarism detection. In: Proceedings of the 2016 ACM Symposium on Document Engineering, pp. 59–68. ACM (2016)
21. Abnar, S., Dehghani, M., Zamani, H., Shakery, A.: Expanded n-grams for semantic text alignment. In: Notebook for PAN at CLEF (2014)
22. Talebpour, A., Shirzadi, M., Aminolroaya, Z.: Plagiarism detection based on a novel trie-based approach. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
23. Momtaz, M., Bijari, K., Salehi, M., Veisi, H.: Graph-based approach to text alignment for plagiarism detection in Persian documents. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
24. Esteki, F., Esfahani, F.S.: A plagiarism detection approach based on SVM for Persian texts. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
25. Gharavi, E., Bijari, K., Zahirnia, K., Veisi, H.: A deep learning approach to Persian plagiarism detection. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
26. Mansoorizadeh, M., Rahgooy, T.: Persian plagiarism detection using sentence correlations. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
27. Gillam, L., Vartapetiance, A.: From English to Persian: conversion of text alignment for plagiarism detection. In: Working Notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, 7–10 December 2016, CEUR Workshop Proceedings (2016). CEUR-WS.org
28. Shamsfard, M., Jafari, H.S., Ilbeygi, M.: STeP-1: a set of fundamental tools for Persian text processing. In: LREC 2010, Malta (2010)
29. Davarpanah, M.R., Sanji, M., Aramideh, M.: Farsi lexical analysis and stop word list. Libr. Hi Tech **27**, 435–449 (2009)
30. Shamsfard, M., Hesabi, A., Fadaei H., et al.: Semi automatic development of FarsNet; the Persian WordNet. In: Proceedings of 5th Global WordNet Conference (2010)
31. Gollub, T., Stein. B., Burrows, S.: Ousting Ivory tower research: towards a web framework for providing experiments as a service. In: 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2012), pp. 1125–1126. ACM (2012). ISBN 978-1-4503-1472-5
32. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the reproducibility of PAN's shared tasks: plagiarism detection, author identification, and author profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) CLEF 2014. LNCS, vol. 8685, pp. 268–299. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11382-1_22