A Text Alignment Algorithm Based on Prediction of Obfuscation Types Using SVM Neural Network

Fatemeh Mashhadirajab
NLP Research Lab,
Faculty of Computer Science and Engineering,
Shahid Beheshti University
Iran
f.mashhadirajab@mail.sbu.ac.ir

Mehrnoush Shamsfard
NLP Research Lab,
Faculty of Computer Science and Engineering,
Shahid Beheshti University
Iran
m-shams@sbu.ac.ir

ABSTRACT

In this paper, we describe our text alignment algorithm that achieved the first rank in Persian Plagdet 2016 competition. The Persian Plagdet corpus includes several obfuscation strategies. Information about the type of obfuscation helps plagiarism detection systems to use their most suitable algorithm for each type. For this purpose, we use SVM neural network for classification of documents according to the type of obfuscation strategy used in a document pair. Then, we set the parameter values in our text alignment algorithm based on the detected type of obfuscation. The results of our algorithm on the test dataset and training dataset in the Persian Plagdet 2016 are shown in this article.

CCS Concepts

• Information systems → Near-duplicate and plagiarism detection•Text mining→ Paraphrase detection→ Plagiarism detection→ Support Vector Machine.

Keywords

Plagiarism detection; Text alignment; SVM neural network.

1. INTRODUCTION

In recent years, automatic discovery of plagiarism has been considered by many researchers, and many plagiarism detection systems have been developed [5, 6, 7]. Plagiarism detection has become a task of PAN competition[1] which is held every year since 2009 to evaluate participants' plagiarism detection algorithms. At the PAN competition the plagiarism detection task is divided into source retrieval and text alignment subtasks. The task of source retrieval is to retrieve documents similar to the suspicious document from the set of source documents, and the duty of text alignment task is to extract all the plagiarized passages from the given source-suspicious document pair. Figure 1 shows different parts of a plagiarism detection system.

As mentioned, in text alignment, the documents in the data sets used to evaluate similarity detector systems are divided into two categories of source documents and suspicious documents. Each Suspicious document contains one or more parts of a source document in its original or edited or rephrased form. The duty of text alignment -which is the focus of this paper- is to find the plagiarized parts of the source document in the suspicious document for each pair of source and suspicious document [8].

Persian Plagdet 2016 competition[2] which is a subtask of PAN Fire 2016 competition[3] is held for Persian language. It means that the text alignment algorithms are evaluated on a Persian corpus.

In this paper we discuss our proposed algorithm which has participated in Persian Plagdet 2016 and ranked first among participants. Our approach firstly uses a neural network for detecting the type of obfuscation in each document pair. Then it sets the parameters in the text alignment algorithm based on the detected type of obfuscation. The rest of the paper explains the proposed algorithm with the special focus on the obfuscation type detection module. Then the result of our evaluation of the system on Persian Plagdet corpus 2016 is discussed.

2. RELATED WORK

At the PAN competition, the text alignment algorithms are evaluated by the evaluation corpora that contain different types of obfuscation. For example in PAN 2013- 2014 competitions, the evaluation corpus consisted of the obfuscation types: none, random, translation and summary. In PAN text alignment corpora it is assumed that just one type of obfuscation is employed in each document pair. Based on this assumption most participants try to predict the type of obfuscation strategy used in a document pair and detect similarities based on the predicted type. At PAN 2014 competition in Glinos' algorithm [10] all of plagiarism documents are divided into two categories: order-based and non-order based. The order-based plagiarism involves none and random obfuscations. The non-order based plagiarism involves translation and summary obfuscation. They use Smith-Waterman algorithm [13] to detect aligned sequences of document pairs and so, detect the order-based plagiarism cases. If no aligned sequences have been found, then document pairs are given to the clustering component to detect non-order based plagiarism cases. Miguel et al [1, 9] categorize document pairs of PAN 2014 corpus into three categories: Verbatim, Summary and Other plagiarism cases and set the parameters in their algorithm based on the categories. They use the Longest Common Substring (LCS) algorithm to find every single common sequence of word (th-Verbatim). If at least one Verbatim case have been found, the document pair is considered as Verbatim plagiarism. If no Verbatim case have been found and the length of plagiarism fragments in the suspicious document is much smaller than the length of source fragments, the document pair is considered as Summary plagiarism, otherwise the document pair is considered as Other plagiarism cases. Also Palkovskii et al [11] in their algorithm use a graphical clustering algorithm to detect type of plagiarism in a document pair. They classify document pairs of PAN 2014 text alignment corpus into four categories: Verbatim Plagiarism, Random Plagiarism, Summary type Plagiarism and Undefined type. Afterward, they set the parameters based on the detected type of plagiarism. In Persian plagdet 2016 corpus there are three type of obfuscation:

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

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

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

none, random and simulated. In our proposed approach, the document pairs of the Persian plagdet 2016 corpus are classified into two categories: Verbatim plagiarism and Simulated plagiarism. We use SVM neural network to detect type of plagiarism. The SVM neural network has been trained by type of obfuscation in the Persian plagdet 2016 training corpus. Then we set the parameters based on the detected type of plagiarism.

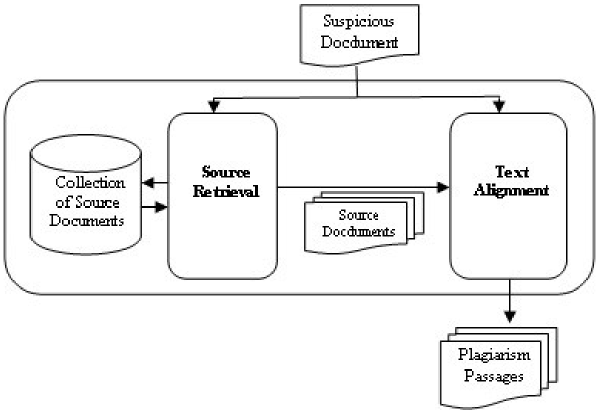


Figure 1. Plagiarism detection systems.

3. METHODOLOGY

Our proposed text alignment algorithm like many other text alignment algorithms [12] includes four stages of preprocessing, seeding, extension, and filtering. Each of these four stages will be explained in this section. In addition, Figure 2 is an overarching scheme of our text alignment algorithm that shows these four stages.

3.1 Preprocessing

In the preprocessing stage, first, the text is segmented into sentences and then tokenized by STeP_1 [2], and Stopwords [4] are removed, and inflectional and derivational stems of tokens are extracted and restored by STeP_1. Preprocessing is done for a pair of suspicious and source document and the sentences of suspicious and source document will be given the seeding stage.

3.2 Seeding

In this stage, the purpose is to extract the similar sentence pairs from source and suspicious documents that we call them seed. For seeding, our method is initially based on the method introduced by [1]. We expanded the mentioned method by using SVM neural net to predict the obfuscation type in order to adjust parameters to gain better results. In this approach, based on vector space model (VSM) method, first, tf-idf vector is calculated for all sentences of suspicious and source documents. Where tf is the term frequency in the corresponding sentence and idf is the inverse sentence frequency. Then, the similarity of each sentence pair of suspicious and source document is calculated using cosine measure and Dice coefficient according to Eq. 1, 2 and 3.

$$cosine(susp_i, src_j) = \frac{susp_i.src_j}{|susp_i||src_j|} \qquad (1)$$

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

$$\delta(x) = \begin{cases} 1 & if\ x \neq \emptyset \\ 0 & other\ wise \end{cases} \qquad (3)$$

Where susp$_i$ is the vector of ith sentence from suspicious document and src_j is the vector from jth sentence of source and |.| is the Euclidean length. Cosine measure and Dice coefficient are calculated for all pairs of sentences and if the similarity of two vectors of $susp_i$ and src_j is more than threshold of 0.3 (chosen based on [1]) based on the both criteria above, this pair of sentences are considered as seed, and for the pairs of sentences whose similarity is more than 0.1 and less than 0.3 (chosen based on our experiments), the similarity will be evaluated semantically. For this purpose, using SVM neural network[4], the type of obfuscation strategy used in the document pairs will be specified. We use cosine similarity percentage between all pairs of sentences of two suspicious and source documents to create our SVM input vector. SVM input vector for suspicious and source document pair is calculated as follows:

An 8-bit vector for each document pair is considered. The range of similarities is divided into 8 intervals. Each bit of the vector corresponds to one of these intervals and indicates if there are two sentences in the document pair whose similarity is in the corresponding interval. In other words, v_i indicates sentence similarity between the α value and β value. Where for $i = 1$:

$\alpha = 0.2$, $\beta = \alpha + 0.1$

If there are sentence pairs whose cosine similarity are between α and β in a document pair, then $v_i = 1$; otherwise, $v_i = 0$.

This vector is given to SVM neural network previously trained by Persian Plagdet training dataset 2016, and document pair obfuscation strategy is projected. We set maximum and minimum similarity threshold in semantic similarity measure based on the type of obfuscation and the amount of similarity between the pairs of sentences. To calculate the semantic similarity we use FarsNet [3] to extract synsets of each term and STeP_1 to extract inflectional and derivational stems of each term. Thus, for each term, a set of words called $\varphi(\omega)$ is considered as shown in Figure 3. Then, for each w_i from vector $susp_i$, if $\varphi(w_i)$ overlaps $\varphi(\acute{w}_j)$ of each $\acute{w}_j$ of vector src_j, w_i of vector $susp_i$ is replaced by $\acute{w}_j$ of vector src_j. Finally, the similarity of cosine and Dice is calculated for the two resulting vectors, and the similarity between the results at this stage and results of cosine and Dice in the previous stage are averaged; if the result is greater than the threshold, the pair of $susp_i$ and src_j are considered as seed. The set of seeds obtained in this stage 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. As shown in Figure 2, extension consists of two parts: clustering and validation. In the clustering stage, the document is clustered into pieces, so that each piece contains a number of seeds where the (similarity) distance between them does not exceed a threshold. In the validation stage, among the pair of passages created in the clustering stage; those that are not similar enough are removed. Again, for the extension stage, we adopt and enhance the method proposed by [1]. The difference is that in the validation stage, we use semantic similarity measure instead of cosine measure to determine the similarity between pairs of passages.

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

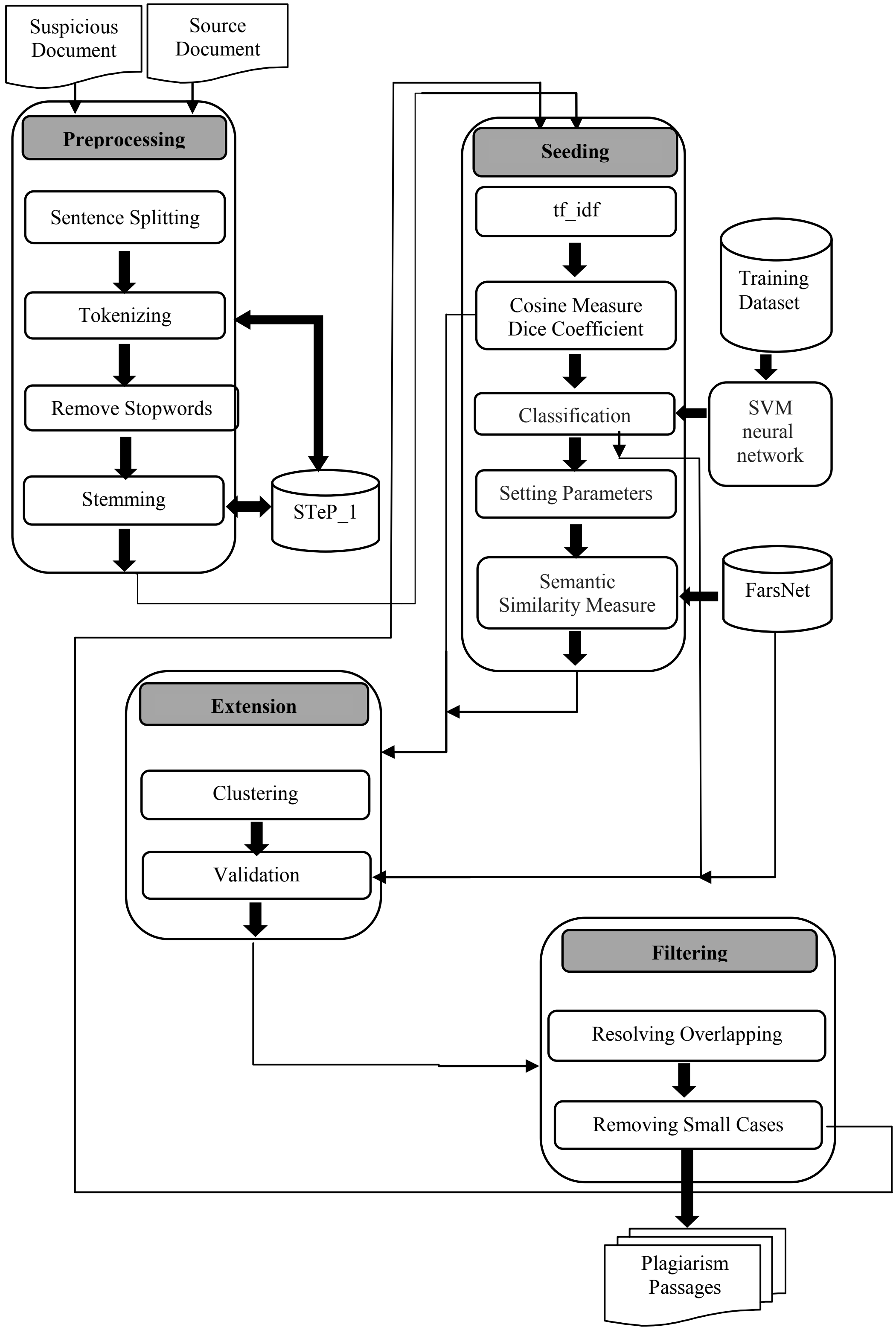
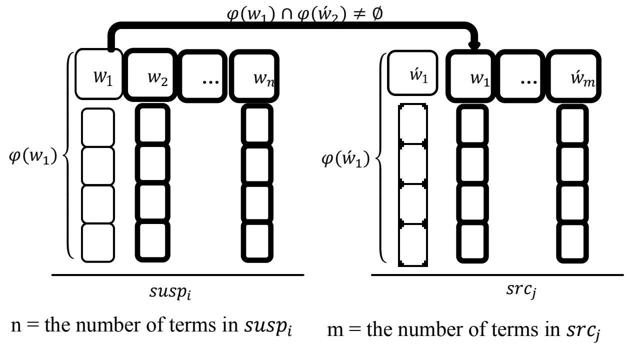


Figure 2. The proposed text alignment algorithm.

3.4 Filtering

The filtering stage removes some passages that either overlap or are too short. To remove overlapping passages we use the proposed method in [1].To remove too short passages, we use a recursive algorithm. If the length of a passage is less than a threshold, we first assume that other seeds should have been existed in this passage, but we had not identified them. So we decrease the threshold on semantic similarity measure and go back to the seeding stage, and we extract the seeds based on the new threshold; and repeat all the stages to remove the too short parts. If the part was not big enough this time, the part will be removed.



$\varphi(w_1) \cap \varphi(\acute{w}_2) \neq \emptyset$
w_1
w_2
...
w_n
$\acute{w}_1$
w_1
...
$\acute{w}_m$
$\varphi(w_1)$
$\varphi(\acute{w}_1)$
$susp_i$
src_j
n = the number of terms in $susp_i$
m = the number of terms in src_j

```
$for\ (w_i = 1\ \ to\ \ n\ \ in\ \ susp_{terms})$
  $for\ (\acute{w}_j = 1\ \ to\ \ m\ \ in\ \ src_{terms})$
    $if\ \ (\varphi(w_i) \cap \varphi(\acute{w}_j)) \neq \emptyset$
      $\acute{w}_j = w_i$
```

Where $\varphi(\omega) =$ the set of inflectional and derivational stems and Synsets of ω.

Figure 3. New vectors for semantic similarity.

4. RESULTS

We implemented our algorithm in C#.Net and evaluated it based on the PAN evaluation setup [15, 16, 17]. In the evaluating stage we ran our algorithm on the Persian Plagdet 2016 training and test dataset [14]. The results of this evaluation also are shown in Table 1. As can be seen, the results of our algorithm on both training and test corpora are very close. Training corpus in this competition includes a variety of obfuscation strategies including None, Random and simulated obfuscation category. Table 2 shows the results of our algorithm on any of the obfuscation strategies in training dataset. In Table 2, column P_1 shows the results of our algorithm on the types of obfuscation in training dataset where the semantic similarity measure is not used. P_2 column shows the algorithm results using semantic similarity measure. Column P_3 shows the results of our algorithm after adding the criterion of semantic similarity, and also adjusting the parameters based on the detected type of obfuscation using neural network. As can be seen, in column P_2, by adding semantic similarity criteria, the recall for the types of obfuscation in training corpus is improved, but the precision has been declined in some cases while in the column P_3, it is seen that by adding a neural network to the system for the diagnosis of type of obfuscation and parameter settings based on the type of obfuscation, precision and recall have been improved dramatically on all types of obfuscation.

Table 1. The results of the proposed text alignment algorithm on Persian Plagdet corpus 2016

Corpus	Plagdet	Recall	Precision	Granularity
Test	0.92	0.92	0.93	1.00
Training	0.94	0.94	0.93	1.00

Table 2. The proposed algorithm on types of obfuscation in Persian Plagdet training dataset 2016

Obfuscation		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
Random	*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

5. Conclusions and Future Work

We described our algorithm for the task of text alignment, and presented the results of the evaluation of this algorithm on test and training dataset in Persian Plagdet 2016, that it was the best result compared with the results of other participants. In our method, we used SVM neural network to identify the type of obfuscation and then to set the parameters on the basis of obfuscation; the results showed that this is effective in improving the precision and recall. In the future, we are going to improve the semantic similarity measure in the seeding stage of our system. We want to use the neural network to estimate the semantic similarity of pair of sentences. We also want to use methods such as genetic algorithms to automatically adjust the parameters.

6. REFERENCES

[1] Sanchez-Perez, M. A., Gelbukh, A. F., Sidorov, G. 2015. Dynamically adjustable approach through obfuscation type recognition. In: *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*, (Toulouse, France, September 8-11, 2015). CEUR Workshop Proceedings, vol. 1391. CEUR-WS.org.

[2] Shamsfard, M., Kiani, S. and Shahedi, Y. STeP-1: standard text preparation for Persian language, *CAASL3 Third Workshop on Computational Approaches to Arabic Script-Languages.*

[3] Shamsfard, M. 2008. Developing FarsNet: A lexical ontology for Persian. *proceedings of the 4th global WordNet conference*.

[4] Davarpanah, M. R., sanji, M. and Aramideh, M. 2009. Farsi lexical analysis and StopWord list. *Library Hi Tech*, vol. 27, pp 435–449.

[5] FIEDLER, R. and KANER, C. 2010. Plagiarism Detection Services: How Well Do They Actually Perform. *IEEE Technology And Society Magazine,* pp. 37-43.

[6] Alzahrani, M., Salim, N. and Abraham, A. 2012. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Trans. SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS*, vol. 42, no. 2.

[7] Ali, A. M. E. T., Abdulla, H. M. D. and Snasel, V. 2011. Survey of plagiarism detection methods. *IEEE Fifth Asia Modelling Symposium (AMS)*, pp. 39_42.

[8] Potthast, M., Göring, S. 2015. Towards data submissions for shared tasks: first experiences for the task of text alignment. *Working Notes Papers of the CLEF 2015 Evaluation Labs, CEUR Workshop Proceedings*, ISSN 1613-0073.

[9] Sanchez-Perez, M., Sidorov, G., Gelbukh, A. 2014. The Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014. In: *Notebook for PAN at CLEF 2014*. (15-18 September, Sheffield*, UK*). CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 1180, CEUR-WS.org, pp. 1004–1011.

[10] Glinos, D. 2014. A Hybrid Architecture for Plagiarism Detection—Notebook for PAN at CLEF 2014. *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers,* (15-18 September, Sheffield*, UK*). CEUR-WS.org. ISSN 1613-0073.

[11] Palkovskii, Y. and Belov, A. 2014. Developing High-Resolution Universal Multi-Type N-Gram Plagiarism Detector—Notebook for PAN at CLEF 2014. *CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers,* (15-18 September, Sheffield, UK). CEUR-WS.org. ISSN 1613-0073.

[12] Potthast, M., Hagen, M., Beyer, A., Busse, M., Tippmann, M., Rosso, P., Stein, B. 2014. 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. CEUR-WS.org.

[13] Smith, T., Waterman, M. 1981. Identification of common molecular subsequences. *Journal of molecular biology*. Vol. 147(1), pp. 195–197.

[14] Asghari, H., Mohtaj, S., Fatemi, O., Faili, H., Rosso, P., and Potthast, M., 2016. Algorithms and Corpora for Persian Plagiarism Detection: Overview of PAN at FIRE 2016. In *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation,* Kolkata, India, December 7-10, 2016, CEUR Workshop Proceedings, CEUR-WS.org.

[15] Potthast, M., Stein, B., Barrón-Cedeño, A., and Rosso, P. 2010. An Evaluation Framework for Plagiarism Detection. In *23rd International Conference on Computational Linguistics (COLING 10)*, pp. 997-1005.

[16] Gollub, T., Stein. B. and Burrows, S. 2012. 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 12)*, pp. 1125-1126. ACM. ISBN 978-1-4503-1472-5.

[17] Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., and Stein, B. 2014. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization*. *5th International Conference of the CLEF Initiative (CLEF 14)*, pp. 268-299, Berlin Heidelberg New York. Springer. ISBN 978-3-319-11381-4.