Dynamically Adjustable Approach through Obfuscation Type Recognition

Notebook for PAN at CLEF 2015

Miguel A. Sanchez-Perez, Alexander Gelbukh, and Grigori Sidorov

Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico
masp1988@hotmail.com, gelbukh@gelbukh.com, sidorov@cic.ipn.mx

Abstract. The task of (monolingual) text alignment consists in finding similar text fragments between two given documents. It has applications in plagiarism detection, detection of text reuse, author identification, authoring aid, and information retrieval, to mention only a few. We describe our approach to the text alignment subtask of the plagiarism detection competition at PAN 2015. Our method relies on a sentence similarity measure based on a tf-idf-like weighting scheme and cosine and dice similarity measures. We used and extended our previous algorithm for clustering and introduced a new verbatim detection method and extended the decision making regarding which approach or output to use. We improve significantly the performance regarding our previous PAN 2014 approach and hence, our approach outperforms the best-performing system of the PAN 2014. Our system is available open source.

1 Introduction

Plagiarism detection, and more generally, text reuse detection, has become a hot research topic given the increasing amount of information being produced as the result of easy access to the Web, large databases and telecommunication in general, which poses a serious problem for publishers, researchers, and educational institutions [5]. Plagiarism detection techniques are also useful in applications such as content authoring systems, which offer fast and simple means for adding and editing content and where avoiding content duplication is desired [1]. Hence, detecting text reuse has become imperative in such contexts.

Text reuse detection is divided into two main tasks: First, searching related documents to a suspicious document, and then finding concrete evidence that passages of text were reused. This paper focused in the second task called Text Alignment as part of the PAN 2015 lab [10].

Text Alignment may be seen as a simple task but imply a huge load of work. Usually passages of text that were reused are not a verbatim copy, instead, some degree of obfuscation is introduced in order to make the new text seem like original work. Given the amount of obfuscation techniques that might be present in a reused passage, proposing a single method to addressed all of them is nearly impossible. Another issue to deal with is, how strict a method should be. A stricter approach could find cases of text reused more precisely but discard those cases with higher level of obfuscation. In the other hand, a more lenient approach could result in a lot of false positives. Finding this balance between precision and recall and creating a method capable of recognizing several obfuscation types, are the biggest challenges of a Text Alignment model.

In our approach, we used two different methods, one stricter based on string matching and other based on text similarity at sentence level. We combined both methods resulting in three possible outputs and deciding which is more suitable according to the type of obfuscation detected. Our system is available open source.[1]

2 Our Approach

We describe our approach using the strategy of the common building blocks used in text alignment algorithms in previous years [9]: seeding, extension, and filtering. Our approach is not solely built on this blocks, we also implemented a method based on string matching and a decision maker to determine which output to use, as will be explained in the Adaptive behavior section. The stages of seeding, extension and filtering were describe in our previous work in [12]. In order for this paper to be self-contained, we include a brief description of them. However there is a modification in the clustering algorithm since we made it recursive.

2.1 Seeding

Given a suspicious document and a source document, the task of the seeding stage is to construct a large set S of short similar passages called *seeds*. We extract our seeds segmenting the documents by sentences and computing the cosine and dice similarities between them. We used a tf-idf-like weighting scheme where sentences are consider documents as follows:

$$tf\left(t,s\right) = f\left(t,s\right), \tag{1}$$

$$isf\left(t,D\right) = \log \frac{|D|}{|\{s \in D : t \in s\}|}, \tag{2}$$

$$w\left(t,s\right) = tf\left(t,s\right) \times isf\left(t,D\right), \tag{3}$$

In Fig. 1, each dot represents a seed in the comparison of two documents that contain plagiarism cases with random obfuscation; a darker color indicates a greater degree of similarity between the two sentences.

2.2 Extension

Given the set of seeds S, defined as the pairs (i,j) of similar sentences, the task of the extension stage is to form larger text fragments that are similar between two

[1] http://www.gelbukh.com/plagiarism-detection/PAN-2014

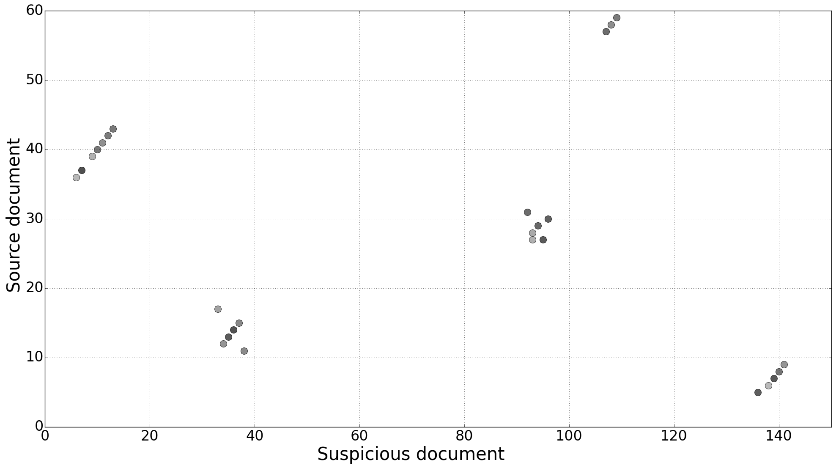

Fig. 1. Seeds representation: similarity between sentences of the suspicious document and the source document

documents. For this, the sentences i are joint into maximal contiguous fragments of the suspicious document and sentences j into maximal contiguous fragments of the source document, so that those large fragments be still similar.

The extension stage is divide into two recursive procedures: (1) Clustering and (2) Validation. In the clustering step we first cluster the seeds according to a distance $maxgap$ looking only to the suspicious side and then we divide these clusters alternating sides. The validation step consist in assessing the quality of resulting clusters from the clustering step. The quality is given by the cosine similarity from the text fragments on each document represented by a cluster and the amount of seeds in it. If the similarity in a cluster is less than a given threshold $th\_validation$, it goes to the clustering stage using a distance of $maxgap - 1$ and up to $maxgap\_least$. In the case a cluster has less than $minsize$ seeds, then it is discarded.

We measured the similarity between text fragments F_{susp} and F_{src} computing the cosine between theirs vectors. The vector representation of the fragments is done adding together the vectors corresponding to all sentences of F_{susp} and F_{src} respectively.

$$similarity\left(F_{susp}, F_{src}\right) = \cos\left(\sum_{v \in F_{susp}} v, \sum_{v \in F_{src}} v\right) \tag{4}$$

Following with the example from Fig. 1 the expected result and output from our extension method is shown in Fig. 2.

The output of the Extension stage is a set of pairs of similar text fragments $\{(F_{susp}, F_{src}), \dots\}$. The diagram of the extension algorithm is shown in Fig. 3.

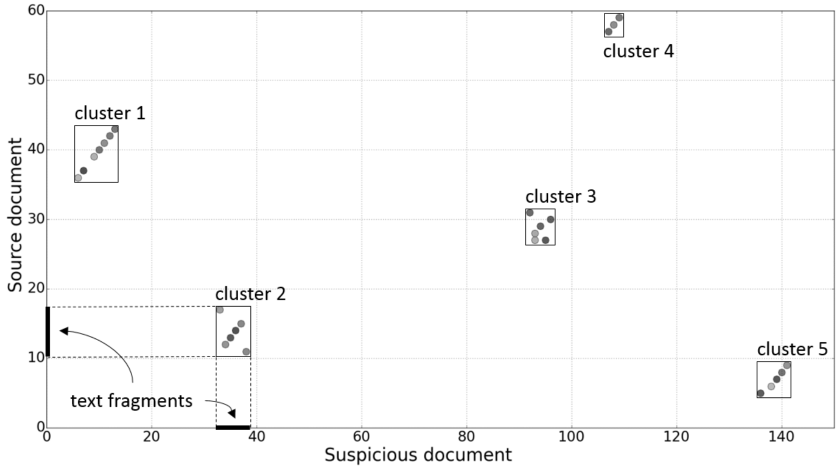

Fig. 2. Clusters and corresponding fragments representing plagiarism cases

2.3 Filtering

Given the set $\{(F_{susp}, F_{src}), \dots\}$ of plagiarism cases, the task of the filtering stage is to improve precision (at the expense of recall) by removing some "bad" plagiarism cases. We did the filtering in two stages: first, we resolved overlapping fragments; then, we removed too short fragments (in what follows we only refer to fragments that represent plagiarism cases, not to arbitrary fragments of the documents).

Resolving overlapping cases. We call two plagiarism cases $\left(F^{'}_{susp}, F^{'}_{src}\right)$ and $\left(F^{''}_{susp}, F^{''}_{src}\right)$ overlapping if the fragments $F^{'}_{susp}$ and $F^{''}_{susp}$ share (in the suspicious document) at least one sentence. We assume that the same source fragment can be used several times in a suspicious document, but not vice versa: each sentence can be plagiarized from only one source and thus can only belong to one plagiarism case. To simplify things, instead of re-assigning only the overlapping parts, we simply discarded whole cases that overlapped with other cases. Specifically, we used the following algorithm:

```
1. While exists a case $P$ ("pivot") that overlaps with some other case
   (a) Denote $\Psi(P)$ be the set of cases $Q \neq P$ overlapping with $P$
   (b) For each $Q \in \Psi(P)$, compute the quality $q_Q(P)$ and $q_P(Q)$; see (5)
   (c) Find the maximum value among all obtained $q_y(x)$
   (d) Discard all cases in $\Psi(P) \cup \{P\}$ except the found $x$
```

In our implementation, at the first step we always used the first case from the beginning of the suspicious document. We compute the quality function $q_y(x)$ of the case x with respect to an overlapping case y as follows. The overlapping cases

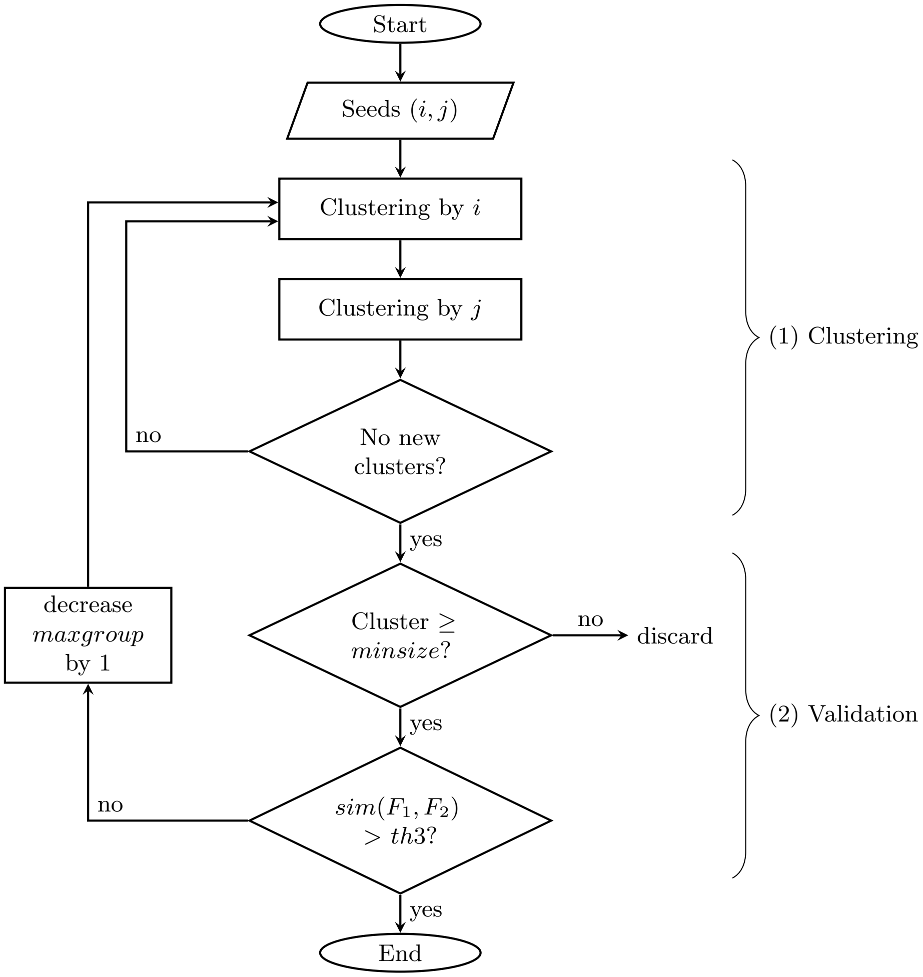

Fig. 3. Diagram of the extension algorithm

$x = \left(F^x_{susp}, F^x_{src}\right)$ and $y = \left(F^y_{susp}, F^y_{src}\right)$ are pairs of corresponding fragments. Let $O = F^x_{susp} \cap F^y_{susp}$ be the overlap and $N = F^x_{susp}/O$ be the non-overlapping part. Then the quality

$$q_y\left(x\right) = sim_{F^x_{src}}\left(O\right) + \left(1 - sim_{F^x_{src}}\left(O\right)\right) \times sim_{F^x_{src}}\left(N\right), \tag{5}$$

where sim is a non-symmetric similarity of a fragment F_{susp} (in the suspicious document) to a reference fragment F_{src} (in the source document):

$$sim_{F_{src}}\left(F_{susp}\right) = \frac{1}{|F_{susp}|} \sum_{s \in F_{susp}} \max_{r \in F_{src}} \left(\cos\left(s, r\right)\right). \tag{6}$$

Formula (5) combines the similarity of the overlapping part and of the non-overlapping part of suspicious fragment to the source counterpart.

Removing small cases. We also discard the plagiarism cases that relate too small fragments: if either suspicious or source fragment of a case has the length in characters less than *minplaglen*, then the case is discarded.

2.4 Adaptive behavior

At the PAN competition, the methods are evaluated on a corpus that contain plagiarism cases created using four different types of obfuscation: none, random, translation and summary. In the training dataset we have the option to test our approaches in sub-corpus divided according the type of obfuscation used to create the plagiarism cases. We observed that the optimal parameters of our method are different to detect such diverse types of obfuscated plagiarism cases. Therefore, we introduced three alternative paths and decided which output to use according to the type of obfuscation we are likely dealing with in each specific document pair.

The final set up of our approach is shown in Fig. 4. After initial preprocessing and seeding steps, we follow two separate path with different *maxgap* values: one value (*maxgap_summary*) that we found to be best for the summary obfuscation sub-corpus and one that was best for the other three corpora (*maxgap*). After we obtain the plagiarism cases using these two different settings, we applied a verbatim detector method to the non-summary approach resulting in three possible results. We named these results as: Verbatim plagiarism cases (cases V), summary plagiarism cases (cases S) and other plagiarism cases (cases O).

Verbatim detector The verbatim detection method is based on the Longest Common Substring (LCS) algorithm. We modify the LCS algorithm in order to use words instead of characters and to find every single common sequence of words above a certain threshold measured in characters (*th_verbatim*).

Output selector The decision of which of the three outputs (cases V, cases S and cases O) report as the final result of our approach follows a decision cascade where the verbatim plagiarism cases have priority, them summary and finally other cases. All three possible outputs are mutually exclusive.

If there is at least one Verbatim case, the pair of documents is consider as a none obfuscation pair and the Verbatim output is reported. If none Verbatim cases were reported, we decide whether cases S are likely to represent summary obfuscation or not, judging by the relative length of the suggested suspicious fragments with respect to the source fragments. Specifically, the decision is made based on the variables src_{len} and $susp_{len}$, which correspond to the total length of all passages, in characters, in the source document and the suspicious document, respectively: when $susp_{len}$ is much smaller than src_{len}, then we are likely

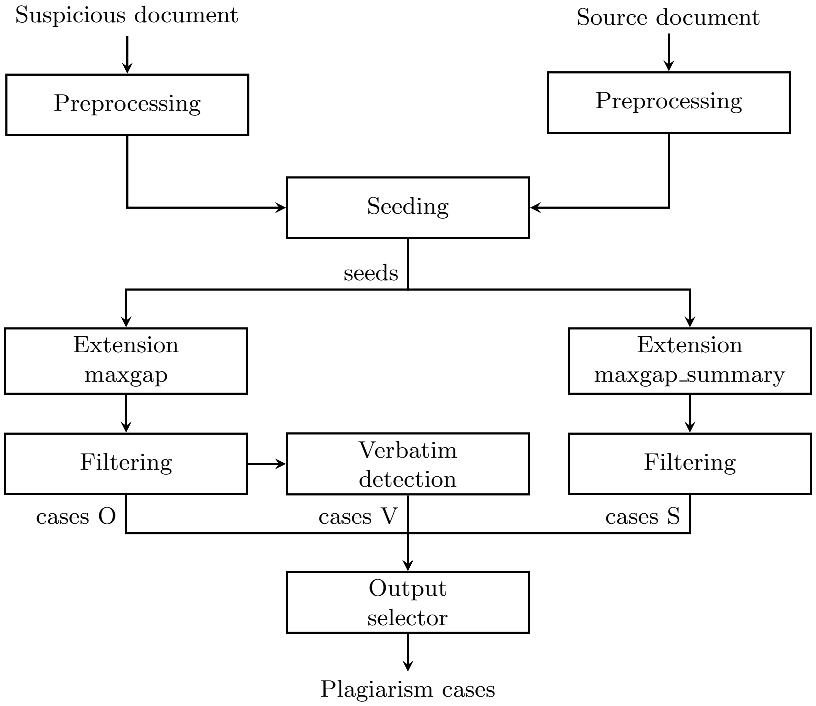

Fig. 4. Diagram of the final approach

dealing with summary obfuscation. If both, Verbatim and Summary cases where discarded then the reported output is Other plagiarism cases.

In table Table 1 we show the final setting of the parameters used in our system.

Table 1. Parameters settings

Parameter	Value	Parameter	Value
minsentlen	3	maxgap_summary	24
th_cos	0.30	maxgap_least	0
th_dice	0.33	minsize	1
th_validation	0.34	minplaglen	150
maxgap	4	th_verbatim	256

3 Experimental Results

We trained our system using the corpus provided for PAN 2014 competition and using the performance measures introduced in [11]. We compared this approach to our previous one in all of the sub-corpus and measurements. We tested our system in the PAN 2014 training corpus using the TIRA platform [4]. As observed in Table 2 the performance in the none obfuscated sub-corpus was increased dramatically because of the new verbatim detector. We also observed a increased in the summary obfuscated sub-corpus because of the new parameters settings. These improvement came without affecting significantly the other sub-corpus and hence, the final result was increased as well.

Table 2. Comparison between our result from PAN2015 and PAN2014 approaches on PAN 2014 training corpus

Obfuscation	PAN 2015 approach				PAN 2014 approach			
	Plagdet	Recall	Precision	Granul.	Plagdet	Recall	Precision	Granul.
None	0.9812	0.9761	0.9933	1.0048	0.8938	0.9782	0.8228	1.0000
Random	0.8847	0.8699	0.8999	1.0000	0.8886	0.8581	0.9213	1.0000
Translation	0.8792	0.9128	0.8481	1.0000	0.8839	0.8902	0.8777	1.0000
Summary	0.6304	0.4862	0.9739	1.0404	0.5772	0.4247	0.9941	1.0434
Entire	0.9025	0.8937	0.9164	1.0036	0.8773	0.8799	0.8774	1.0021

The results showed that recall is the measure where we excel but need to improve the precision of the model by identifying and adjusting to other types of obfuscation rather than just summary and verbatim obfuscation. Regarding the system runtime, even our goal is not aiming at efficiency, our software performed in an average level.

In Table 3 we present a comparison between our approach and 2014 participants showing a remarkable improvement.

Table 3. Our approach compared to the PAN 2014 Official results reported in [9]

Team	PlagDet	Recall	Precision	Granularity	Runtime
Sanchez-Perez15	0.9010	0.8957	0.9125	1.0046	–
Sanchez-Perez14	0.8781	0.8790	0.8816	1.0034	00:25:35
Oberreuter	0.8693	0.8577	0.8859	1.0036	00:05:31
Palkovskii	0.8680	0.8263	0.9222	1.0058	01:10:04
Glinos	0.8593	0.7933	0.9625	1.0169	00:23:13
Shrestha	0.8440	0.8378	0.8590	1.0070	69:51:15
R. Torrejón	0.8295	0.7690	0.9042	1.0027	00:00:42
Gross	0.8264	0.7662	0.9327	1.0251	00:03:00
Kong	0.8216	0.8074	0.8400	1.0030	00:05:26
Abnar	0.6722	0.6116	0.7733	1.0224	01:27:00
Alvi	0.6595	0.5506	0.9337	1.0711	00:04:57
Baseline	0.4219	0.3422	0.9293	1.2747	00:30:30
Gillam	0.2830	0.1684	0.8863	1.0000	00:00:55

4 Conclusions and Future Work

We have described our approach to the task of text alignment in the context of PAN 2015 competition showing great improvement. The additions from our previous work are the verbatim detector, applying clustering recursively and optimization of parameters. We also tested other methods to detect paraphrase but need further improvement and testing, and given the resources available at PAN so far, it is not possible.

In our future work, we plan to use linguistically motivated methods to address possible paraphrase obfuscation [2] and test it on the P4P corpus.[2] We also plan to build a meta-classifier that would guess which obfuscation type of plagiarism case we deal with at each moment and dynamically adjust the parameters. Finally, we plan to apply concept-based models for similarity and paraphrase detection [6–8].

Acknowledgements Work done under partial support of FP7-PEOPLE-2010-IRSES: Web Information Quality – Evaluation Initiative (WIQ-EI) European Commission project 269180, Government of Mexico (SNI, CONACYT), and Instituto Politécnico Nacional, Mexico (SIP 20144274, 20150028, BEIFI, COFAA).

References

1. Bär, D., Zesch, T., Gurevych, I.: Text reuse detection using a composition of text similarity measures. In: Kay, M., Boitet, C. (eds.) COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8–15 December 2012, Mumbai, India. pp. 167–184. Indian Institute of Technology Bombay (2012)
2. Barrón-Cedeño, A., Vila, M., Martí, M.A., Rosso, P.: Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. Computational Linguistics 39(4), 917–947 (2013)
3. Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (eds.): Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15–18, 2014, CEUR Workshop Proceedings, vol. 1180. CEUR-WS.org (2014)
4. Gollub, T., Stein, B., Burrows, S.: Ousting Ivory Tower research: Towards a web framework for providing experiments as a service. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). pp. 1125–1126. ACM (Aug 2012)
5. Maurer, H., Kappe, F., Zaka, B.: Plagiarism – A survey. Journal of Universal Computer Science 12(8), 1050–1084 (Aug 2006)
6. Poria, S., Agarwal, B., Gelbukh, A., Hussain, A., Howard, N.: Dependency-based semantic parsing for concept-level text analysis. In: Gelbukh, A.F. (ed.) Computational Linguistics and Intelligent Text Processing, 15th International Conference, CICLing 2014, Kathmandu, Nepal, April 6–12, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8403, pp. 113–127. Springer (2014)

[2] http://clic.ub.edu/corpus/en/paraphrases-en

7. Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP). pp. 28–37. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (August 2014)
8. Poria, S., Cambria, E., Winterstein, G., Huang, G.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowl.-Based Syst. 69, 45–63 (2014)
9. Potthast, M., Hagen, M., Beyer, A., Busse, M., Tippmann, M., Rosso, P., Stein, B.: Overview of the 6th International Competition on Plagiarism Detection. In: Cappellato et al. [3], pp. 845–876
10. Potthast, M., Hagen, M., Göring, S., Rosso, P., Stein, B.: 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, CLEF and CEUR-WS.org (Sep 2015)
11. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. In: Huang, C., Jurafsky, D. (eds.) COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China. pp. 997–1005. Chinese Information Processing Society of China (2010)
12. Sanchez-Perez, M.A., Sidorov, G., Gelbukh, A.: The winning approach to text alignment for text reuse detection at PAN 2014. In: Cappellato et al. [3], pp. 1004–1011