Adaptive Algorithm for Plagiarism Detection: The Best-performing Approach at PAN 2014 Text Alignment Competition

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 2014, which resulted in the best-performing system at the PAN 2014 competition and outperforms the best-performing system of the PAN 2013 competition by the cumulative evaluation measure Plagdet. Our method relies on a sentence similarity measure based on a tf-idf-like weighting scheme that permits us to consider stopwords without increasing the rate of false positives. We introduce a recursive algorithm to extend the ranges of matching sentences to maximal length passages. We also introduce a novel filtering method to resolve overlapping plagiarism cases. Our system is available as 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 [8]. 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.

Our approach outperforms the best-performing systems of both PAN 2013 [14] and PAN 2014 [13] competitions. PAN[1] is a CLEF Lab on uncovering plagiarism, authorship, and social software misuse. In 2013 and 2014, the PAN competition consisted of three tasks: plagiarism detection, author verification, and author profiling. The plagiarism detection task was divided into source retrieval and text alignment subtasks. In the text alignment subtask, the systems were required to identify all contiguous maximal-length passages of reused text between a given pair of documents. At the PAN 2014 competition, our approach

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

showed the best result out of ten participating systems. Our system is available open source.[2]

The rest of the paper is organized as follows. Section 2 explains the general steps to build a text alignment model with some related work, and the main problems to solve when building one. Section 3 describes in detail our approach. Section 4 discusses the experimental experiments. Finally, Section 5 gives conclusions and future work.

2 Text Alignment

The text alignment task consists in the following: given a pair of documents, to identify contiguous passages of reused text between them. Most of the text alignment models follow a three-step approach: seeding, extension, and filtering [13]. The first step consists in finding relations (so-called "seeds") between features extracted from the documents. At this stage, it is important to determine which type of features to use and what kind of relation to look for. For example, the features could be word n-grams with several implementations like Context n-grams [9,16–18], Context skip n-gram [18], Stopwords n-grams [16,17] and Named entity n-grams [16]; all of them looking for exact match. In our approach, we extracted sentences and compared them in a Vector Space Model (VSM) using the cosine similarity alike [6]. We also used the Dice coefficient as in [7] given that this measure look for a basic and equal distributions of the terms in the passages to compare.

Taking into account only the seeds extracted, some passages that do not show high similarity but are part of a plagiarism case could be missed. This due to the presence of noise and also because a specific type of feature or similarity measure does not necessarily identify all possible types of obfuscation techniques.[3]

Accordingly, the extension step consists in joining these seeds into larger fragments. This is the core of a text alignment model. The basic idea here is to cluster together nearby seeds. A plagiarism case will be defined by the edges of a cluster: if we draw a rectangle around the cluster, the plagiarism case is the fragment of text in the suspicious document and its corresponding counterpart in the source document, as shown in Fig. 1. Defining a cluster by its edges and not as a set of seeds allows for small gaps in the range of seeds that can be part of the plagiarism case even if our seeding process did not detect them; for example, see cluster 1 in the figure.

However, the greater the distance allowed between seeds in a cluster, the greater the probability of including passages that do not really belong to the plagiarism case. Measuring the quality of a plagiarism case includes computing the similarity between the two fragments of text. Thus, the challenge for an extension algorithm is to find a balance between the dispersion in a clusters and

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

[3] Obfuscation techniques refers to the changes done to the plagiarized passages like sentence reordering, changing words with synonyms, using summaries, among others.

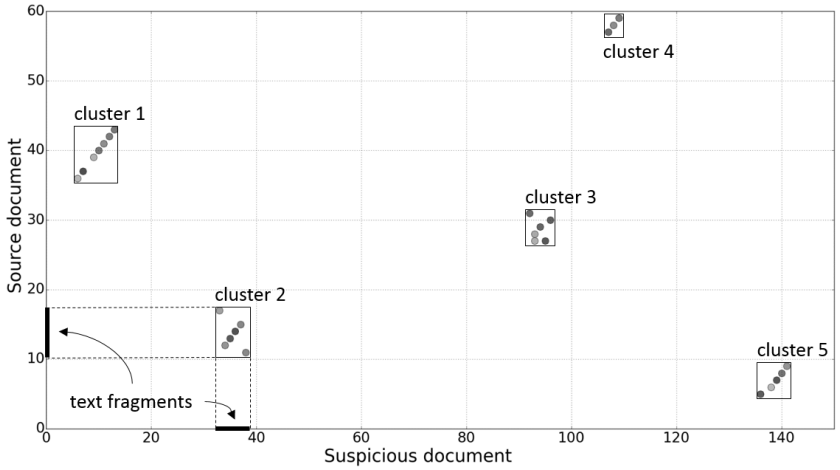

Fig. 1. Clusters obtained after the extension step. The fragments of text (ranges of sentences) corresponding to cluster 2 are shown

the similarity of the fragments of text this cluster correlates. A problem that arises in finding this balance in our approach is that the sentences do not necessarily have the same length, so a distance good for one cluster is not necessarily good for another cluster given the resulting similarity between the fragments of text. Therefore, balancing should be done for each cluster independently after the initial iteration of clustering is done.

Another important problem when building an extension method is to determine what type of measure of distance should be used, and this is not a trivial problem. From the dots in Fig. 1, it is expected to have clusters such as those represented, which relate a fragment of text in the suspicious document with a fragment of text in the source document. However, a Euclidean distance clustering algorithm as in [9] will fail to detect cluster 2, because two of its points are far from the rest of the group using this distance. These seeds in cluster 2 represent just a reordering of sentences: for instance, changing the last sentence in the source document to the first one in the suspicious document. Another way to compute distance could be using a function that returns the minimum distance in either dimension. This would result in correct detection of cluster 2, but also would join together clusters 2 and 5, because they are close on the source document axis. Given that the two measures mentioned above compute the distance taking into account both dimensions at the same time. We used a method that computes the distance in one dimension at a time, alternating between them until no more division is needed. Several participants used algorithms in this direction taking into consideration the distance in character [7, 16–18] or sentences [6].

The final step in the text alignment task is responsible for filtering out those clusters that do not meet certain criteria. Usually this includes removing too

Table 1. Main ideas used in the systems participating in PAN 2012 and 2013

Stage	Method	[6]	[18]	[17]	[16]	[9]	[7]	Our
Preprocessing	Special character removal	+	−	−	−	−	−	+
	Number removal	−	−	−	−	+	−	−
	Stopword removal	+	+	−	−	−	−	−
	Case folding	+	+	+	+	+	−	+
	Stemming	+	+	−	−	+	−	+
Seeding	Bag of words	+	−	−	−	−	+	+
	Context n-grams	−	+	+	+	+	−	−
	Context skip n-grams	−	+	−	−	−	−	−
	Stopword n-grams	−	−	+	+	−	−	−
	Named entity n-grams	−	−	−	+	−	−	−
Extension	Bilateral Alternating Sorting	+	−	−	−	−	−	−
	Distance between seeds	+	+	+	+	−	+	+
	Euclidean distance clusters	−	−	−	−	+	−	−
	Extension with multiple features	−	+	−	+	−	−	−
Filtering	Passage similarity	+	−	−	−	−	−	+
	Small passage removal	−	+	+	−	+	−	+
	Overlapping removal	−	−	+	+	−	−	+
	Nearby passage joining	−	−	−	+	−	−	−

short plagiarism cases or treating overlapping cases. The main problem we found using the PAN 2013 training corpus was that some plagiarism cases are contained inside larger cases in any of the two sides. To solve this problem we introduced a measure of quality that compares overlapped cases, to decide which one to keep and which one to discard.

Finally, given that the three-step model for text alignment uses many parameters, it is impossible to find one optimal setting for all types of obfuscation. Therefore, the model should be adaptive: it should use heuristics to decide which type of obfuscation it deals with in a given document and choose the corresponding settings optimized for each type of obfuscation.

Table 1 summarizes the main ideas employed by the systems participating in PAN 2012 and 2013 [4, 6, 7, 9, 16–18], classified by the four main stages of a typical alignment process as suggested in [14].

3 Our Approach

We describe our approach using the three-steps model: seeding, extension, and filtering. Before these steps, we pre-process the documents applying sentence splitting and tokenization, removing all tokens ("words") that do not start in a letter or digit, reducing all letters to lowercase, applying stemming, and joining small sentences (shorter than $minsentlen = 3$ words) with the next one (if the new joint "sentence" was still small, we join it with the next one, etc.). In the following sections, we describe our processes of seeding, extension, and filtering.

3.1 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*. Each seed is a pair that consists of a small fragment of the suspicious document and a small fragment of the source document that are in some sense similar. In our case, the fragments to form the pairs were sentences, which may be joined as described above. Constructing these pairs required to measure similarity between sentence vectors, for which we had to choose a weighting scheme.

To measure the similarity between two sentences, we represented individual sentences with a tf-idf vector space model (VSM), as if each sentence were, in terminology of VSM, a separate "document" and all sentences in the pair of original document formed a "document collection." The idf measure calculated in this way is called *isf measure* (inverse sentence frequency) to emphasize that it is calculated over sentences as units and not documents:

$$\begin{aligned} tf\left(t,s\right) &= f\left(t,s\right), \\ isf\left(t,D\right) &= \log \frac{|D|}{|\{s \in D : t \in s\}|}, \\ w\left(t,s\right) &= tf\left(t,s\right) \times isf\left(t,D\right), \end{aligned}$$

where for term frequency $tf(t,s)$ we simply used the number of occurrences $f(t,s)$ of the term t in the sentence s; D is the set of all sentences in both given documents, and $w(t,s)$ is the final weight of a term t of the sentence s in our VSM representation.

After we defined the weighting scheme and transformed all sentences into vectors in both documents we compared each sentence in the suspicious document to each sentence in the source document.

Now we construct the desired set S as

$$S = \{(i,j) \mid \cos\left(susp_i, src_j\right) > mincos \wedge dice\left(susp_i, src_j\right) > mindice\},$$

where the two sentences are represented as vectors, cos is the cosine similarity, *dice* is the Dice coefficient:

$$\begin{aligned} \cos\left(susp_i, src_j\right) &= \frac{susp_i \cdot src_j}{|susp_i| \times |src_j|}, \\ dice\left(susp_i, src_j\right) &= \frac{2\left|\delta\left(susp_i\right) \cap \delta\left(src_j\right)\right|}{\left|\delta\left(susp_i\right)\right| + \left|\delta\left(src_j\right)\right|}, \end{aligned}$$

$\delta(x)$ is the set of non-zero coordinates of a vector x, $|*|$ is the Euclidean length of a vector or the cardinality of a set, respectively, and *mincos* and *mindice* are some thresholds determined experimentally.

3.2 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

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.

We divide the extension process into two steps: (1) Clustering and (2) Validation. In the clustering step we create text fragments grouping seeds that are not separated by more than a *maxgap* number of sentences. In our implementation, an easier way to proceed is to sort and cluster the set of seeds by i (left or suspicious document) such that $i_n - i_{n+1} \leq maxgap$. Then, for each of the resulting clusters, sort and cluster by j (right or source document), and thereby alternate by i and j until no new clusters are formed. Each cluster should have at least *minsize* seeds or will be discarded. Since we use the parameter *maxgap* to cluster seeds into larger text fragments, some sentences in these fragment may have no similarity to any of the sentences in the corresponding fragment. Therefore in order to avoid adding to much noise in the clustering step we validate that the similarity between the text fragments of the remaining clusters exceed some threshold. If the similarity is less than the given threshold we apply the extension stage using $maxgap - 1$ for this particular cluster. We will reduce *maxgap* at most to a *min_maxgap* value. If the *min_maxgap* value is reached and the validation condition is not met then the cluster is discarded.

A text fragment is defined as the collection of all the sentences comprised in the seeds of a particular cluster. Given a cluster integrated by seeds of the form (i, j), then the text fragment in the suspicious document F_{susp} is the collection of all the sentences from the smallest i to the largest i in the cluster, similarly the corresponding text fragment in the source document F_{src} is the collection of all the sentences from the smallest j to the largest j in the cluster.

We measured the similarity between text fragments F_{susp} and F_{src} computing the cosine between theirs vectors:

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

where the vector representation of the fragments is done adding together the vectors corresponding to all sentences of F_{susp} and F_{src} respectively.

For details of our method, see Algorithm 1. The variable *side* indicates by which side the pairs are clustered: $+1$ means clustering by sentences of the suspicious document (i) and -1, by sentences of the source document (j). The output of the Extension stage is a set of pairs of similar text fragments $\{(F_{susp}, F_{src}), \dots\}$ taken from the resulting clusters.

3.3 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

Algorithm 1: Extension algorithm

```
const minsize, minsim
Function extension(seeds, maxgap)
1   clusters ← clustering(seeds,maxgap,+1)
2   clusters ← validation(clus,maxgap)
3   return clusters
Function clustering(seeds, maxgap, side)
1   clusters ← clusters of seeds such that in each cluster, side-hand sentences
    form in the document fragments with at most maxgap-sentence gaps
2   discard all c ∈ clusters such that |c| < minsize
3   if |clusters| ≤ 1 then
4     return clusters
    else
5     result ← ∅
6     foreach c ∈ clusters do
7       result ← result ∪ clustering(c,maxgap,−side)
8   return result
Function validation(clusters, maxgap)
1   result ← ∅
2   foreach c ∈ clusters do
3     if similarity(F_susp(c), F_src(c)) < minsim then
4       if maxgap > min_maxgap then
5         result ← result ∪ extension(c,maxgap − 1)
        else
6         result ← result ∪ { c }
7   return result
```

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 (1)
   (c) Find the maximum value among all obtained $q_y(x)$
   (d) Discard all cases in $\Psi(P) \cup \{P\}$ except the found x

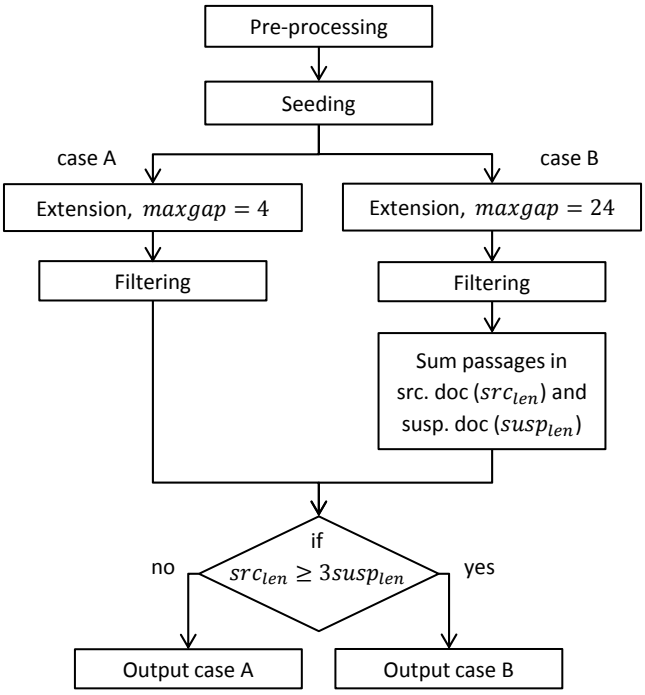

Fig. 2. Adaptive behavior

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 $x = \left(F^x_{susp}, F^x_{src}\right)$ 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

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

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):

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

Formula (1) 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.

Table 2. Our results on PAN 2013 training corpus

Obfuscation	PAN 2013 training corpus				PAN 2013 test corpus			
	Plagdet	Recall	Precision	Granul.	Plagdet	Recall	Precision	Granul.
None	0.8938	0.9782	0.8228	1.0000	0.9003	0.9785	0.8336	1.0000
Random	0.8886	0.8581	0.9213	1.0000	0.8841	0.8606	0.9101	1.0008
Translation	0.8839	0.8902	0.8777	1.0000	0.8865	0.8895	0.8846	1.0008
Summary	0.5772	0.4247	0.9941	1.0434	0.5607	0.4127	0.9991	1.0588
Entire	0.8773	0.8799	0.8774	1.0021	0.8781	0.8790	0.8816	1.0034

3.4 Adaptive behavior

At PAN-2014, the methods were evaluated on four different corpora: no obfuscation, random obfuscation, translation obfuscation, and summary obfuscation, the final result being averaged over those four corpora. We observed that the optimal parameters of our method are different for such different types of plagiarism. Therefore, we introduce adaptive selection of parameters: we detect which type of plagiarism case we are likely dealing with in each specific document pair, and adjust the parameters to the optimal set for this specific type.

Our implementation of this approach is shown in Fig. 2. After initial preprocessing and seeding, we applied the same processes twice, with different *maxgap* values: one value that we found to be best for the summary obfuscation subcorpus (variant B) and one that was best for the other three corpora (variant A). After we obtain the plagiarism cases using these two different settings, we decide whether those cases are likely to represent summary obfuscation or not, judging by the relative length of the suggested suspicious fragments with respect to the source fragments, and depending on this, choose to output the results of one of the two variants. Our results at PAN-2014 were obtained with:

$$mincos = 0.33 \quad minsim = 0.4 \quad minsize = 1 \quad minsentlen = 3$$
$$mindice = 0.33 \quad minplaglen = 150 \quad min\_maxgap = 2$$

We used equal values for *mincos* and *mindice*; however, later we obtained better results (not reported here) when their values were different.

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 dealing with summary obfuscation.

4 Experimental Results

The evaluation framework for plagiarism detection referring to the Precision, Recall, Granularity and Plagdet measures on this specific task was introduced by Potthast in [15]. We trained our system using the corpus provided for PAN 2014 competition (pan13-text-alignment-training-corpus-2013-01-21) [13]. We also evaluated our model on the test corpus of PAN 2013 (pan13-text-alignment-test-corpus2-2013-01-21) in order to compare our approach with existing approaches.

Table 3. Comparative results according to the Plagdet measure on PAN 2013 test corpus. Performance of the systems was published in [14]

Team	Year	**Entire corpus**	None	Random	Translation	Summary
Sanchez-Perez (our)	–	**0.8781**	0.9003	**0.8841**	**0.8865**	0.5607
Torrejón	2013	0.8222	0.9258	0.7471	0.8511	0.3413
Kong	2013	0.8189	0.8274	0.8228	0.8518	0.4339
Suchomel	2013	0.7448	0.8176	0.7527	0.6754	**0.6101**
Saremi	2013	0.6991	0.8496	0.6566	0.7090	0.1111
Shrestha	2013	0.6955	0.8936	0.6671	0.6271	0.1186
Palkovskii	2013	0.6152	0.8243	0.4995	0.6069	0.0994
Nourian	2013	0.5771	0.9013	0.3507	0.4386	0.1153
Baseline	2013	0.4219	**0.9340**	0.0712	0.1063	0.0446
Gillam	2013	0.4005	0.8588	0.0419	0.0122	0.0021
Jayapal	2013	0.2708	0.3878	0.1814	0.1818	0.0594

Table 2 shows our results on the training corpus of PAN 2014, which was the same as the training corpus of PAN 2013, and on the test corpus of PAN 2013. Table 3 compares our results using the cumulative Plagdet measure with those of the systems submitted to PAN 2013. Column shows the system results on each sub-corpus built using different types of obfuscation.

We experimented with each one of our improvements separately and verified that they do boost the cumulative Plagdet measure. Both the use of the tf-isf measure and our recursive extension algorithm considerably improved recall without a noticeable detriment to precision. On the other hand, resolution of overlapping cases improved precision without considerably affecting recall. Finally, the dynamic adjustment of the gap size improved Plagdet on the summary sub-corpus by 35%, without considerably affecting other corpora.

We participate in the Text Alignment task of the PAN 2014 Lab outperforming all 10 participants as shown in Table 4. The official 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 obfuscation. Regarding the system runtime, even our goal is not aiming at efficiency, out software performed at an average level.

5 Conclusions and Future Work

We have described our approach to the task of text alignment in the context of PAN 2014 competition, with which our system showed the best result of ten participating systems, as well as outperformed the state-of-art systems that participated in PAN 2013 on the corresponding corpus [14]. Our system is available open source.

Our main contributions are: (1) the use of the tf-isf (inverse sentence frequency) measure for "soft" removal of stopwords instead of using a predefined stopword list; (2) a recursive extension algorithm, which allows to dynamically adjust the tolerance of the algorithm to gaps in the fragments that constitute

Table 4. PAN 2014 official results reported in [13] using TIRA [5]

Team	**PlagDet**	Recall	Precision	Granularity	Runtime
Sanchez-Perez (our)	**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

plagiarism cases; (3) a novel algorithm for resolution of overlapping plagiarism cases, based on comparison of competing plagiarism cases; (4) dynamic adjustment of parameters according to the obfuscation type of plagiarism cases (summary vs. other types). Each of these improvements contributes to improve the performance of the system.

In our future work, we plan to use linguistically motivated methods to address possible paraphrase obfuscation [2] and test it on the P4P corpus.[4] 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 [10–12].

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 20152100, 20151406, 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. Forner, P., Navigli, R., Tufis, D., Ferro, N. (eds.): Working Notes for CLEF 2013 Conference, Valencia, Spain, September 23–26, 2013, CEUR Workshop Proceedings, vol. 1179. CEUR-WS.org (2013)

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

4. Gillam, L.: Guess again and see if they line up: Surrey's runs at plagiarism detection notebook for PAN at CLEF 2013. In: Forner et al. [3]
5. 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)
6. Kong, L., Qi, H., Du, C., Wang, M., Han, Z.: Approaches for source retrieval and text alignment of plagiarism detection notebook for PAN at CLEF 2013. In: Forner et al. [3]
7. Küppers, R., Conrad, S.: A set-based approach to plagiarism detection. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17–20, 2012. CEUR Workshop Proceedings, vol. 1178. CEUR-WS.org (2012)
8. Maurer, H., Kappe, F., Zaka, B.: Plagiarism – A survey. Journal of Universal Computer Science 12(8), 1050–1084 (Aug 2006)
9. Palkovskii, Y., Belov, A.: Using hybrid similarity methods for plagiarism detection notebook for PAN at CLEF 2013. In: Forner et al. [3]
10. 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)
11. 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)
12. 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)
13. 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, 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, pp. 845–876. CEUR-WS.org (2014)
14. Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., Stein, B.: Overview of the 5th International Competition on Plagiarism Detection. In: Forner et al. [3]
15. 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)
16. Shrestha, P., Solorio, T.: Using a variety of n-grams for the detection of different kinds of plagiarism notebook for PAN at CLEF 2013. In: Forner et al. [3]
17. Suchomel, S., Kasprzak, J., Brandejs, M.: Diverse queries and feature type selection for plagiarism discovery notebook for PAN at CLEF 2013. In: Forner et al. [3]
18. Torrejón, D.A.R., Ramos, J.M.M.: Text alignment module in CoReMo 2.1 plagiarism detector notebook for PAN at CLEF 2013. In: Forner et al. [3]