Plagiarism Alignment Detection by Merging Context Seeds

Notebook for PAN at CLEF 2014

Philipp Gross[1] and Pashutan Modaresi[2,3]

[1] pressrelations GmbH, Düsseldorf, Germany
{philipp.gross, pashutan.modaresi}@pressrelations.de
[2] Heinrich-Heine-University of Düsseldorf, Institute of Computer Science, Düsseldorf, Germany
modaresi@cs.uni-duesseldorf.de

Abstract We describe our submitted algorithm to the text alignment sub-task of the plagiarism detection task in the PAN2014 challenge that achieved a plagdet score 0.855. By extracting contextual features for each document character and grouping those that are relevant for a given pair of documents, we generate seeds of atomic plagiarism cases. These are then merged by an agglomerative single-linkage strategy using a defined distance measure.

1 Introduction

Given a pair of text documents, the problem of text alignment is the task of identifying all pairs of contiguous passages that are equal up to obfuscation. The various strategies of the latter pose a challenge for this task. They reach from randomized simple operations like word or sentence permutations to more sophisticated transformations like semantic word variation or translation cycles, or even manual paraphrasing.

The problem of text alignment is a sub-task of the *PAN 2014 Plagiarism Detection* task. It also contains the sub-task *source retrieval* that concerns he retrieval of source documents when a suspicious document is given. Our submission only deals with the problem of text alignment. We follow the common strategy as explained in [2]:

1. *Seed generation*: Given a suspicious document X and a source document Y select a set of likely plagiarism cases, which are pairs of small passages in X and Y that are very similar by a defined measure.
2. *Merging*: Merge two plagiarism cases whose passages in X and Y are close to each other. Repeat this step until there are no adjacent plagiarism cases left.
3. *Extraction and filtering*: Postprocess the remaining plagiarism cases, filter outliers, and generate output plagiarism cases.

In the following we describe our submitted algorithm in detail and give the evaluation results for various corpora.

2 Problem Statement

In this section we formally define the problem of *text alignment* as part of *PAN 2014 Plagiarism Detection* task. The precise objective of this task is, for a given set of plagiarism cases S, to find a set of detections R such that the score $\mathrm{plagdet}(S, R)$ is high.

We follow a slightly different terminology compared to [3] and recall all definitions for the readers convenience.

A *document* of length n_X is a finite totally ordered set $X = \{x_i \colon i = 0, \ldots, n_X\}$ of (positioned) characters $x_i = (c, i)$, $c \in \mathcal{C}$, where $\mathcal{C}$ denotes some finite set of symbols. A *passage* $P \subseteq X$ is a connected subset, thus either empty or of the form

$$P \;=\; \{\, x_i \,:\; 0 \;\leq\; a \;\leq\; i \;<\; b \;\leq\; n\}. \tag{1}$$

For brevity we use the notion of closed intervals and define $[x_a, x_b] = P$ for non-empty passages.

Given a pair of documents (X, Y) we define a *passage reference* r as a rectangular subset in the Cartesian product set $r \subseteq X \times Y$. Every non-empty passage reference is always of the form

$$r = [x_a, x_b] \times [y_c, y_d] = \{(x_i, y_j) \colon 1 \leq a \leq i < b \leq n_X, 1 \leq c \leq j < d \leq n_Y\} \tag{2}$$

for passages $[x_a, x_b] \subseteq X$ and $[y_c, y_d] \subseteq Y$. Each pair $(x, y) \in X \times Y$ gives rise to a passage reference by taking the singleton set $\{(x, y)\}$, called *seed*. It is a minimal passage reference. Note that every non-empty passage reference is the linear span of finitely many seeds.

We say that a passage reference r *detects* another passage reference s if both belong to the same product space $X \times Y$ and have non-empty intersection $r \cap s$. The latter is also a passage reference. By embedding a document into the disjoint union of all documents, the definition $r \cap s$ extends naturally to passage references with different pairs of parent documents (namely, by the empty intersection).

We define the *perimeter* of a passage reference r as

$$\pi(r) = 2(b - a) + 2(d - c) \text{ if } r = [x_a, x_b] \times [y_c, y_d], \quad \text{and} \quad \pi(\emptyset) = 0. \tag{3}$$

The union of passage references is in general not a passage reference. But the perimeter extends in a natural way for such sets by taking the (one-dimensional) volume of the boundary. The upshot of the perimeter is that a passage reference r detects another passage reference s if and only if $\pi(r \cap s) > 0$.

A *set (or corpus) of plagiarism cases* S is just a set of passage references for varying document pairs (X, Y), $X, Y \in \mathcal{D}$ and some set of documents $\mathcal{D}$.

The quality of a set of detections R is evaluated by the numerical $\mathrm{plagdet}$ score. It is a composition of the micro precision, micro recall and granularity. For the sake of completeness, we recall their definitions. With $S \cdot R = \{s \cap r \mid s \in S, r \in R\}$ let

$$\mathrm{prec}(S, R) = \frac{\pi(S \cdot R)}{\pi(R)}, \qquad \mathrm{rec}(S, R) = \frac{\pi(S \cdot R)}{\pi(S)}. \tag{4}$$

Precision and recall give rise to the classical F_1-measure, i.e. the harmonic mean of precision and recall. In order to penalize fragmented passage references one weights the F_1 measure with the granularity,

$$\operatorname{gran}(S, R) = \frac{\sum_{s \in S_R} |R_s|}{|S_R|} \in [1, |R|], \tag{5}$$

where $S_R = \{s \mid s \in S, \exists r \in R \text{ detects } s\}$ and $R_s = \{r \mid r \in R, r \text{ detects } s\}$. Then the $\operatorname{plagdet}$-score is defined as the weighted F_1-measure:

$$\operatorname{plagdet}(S, R) = \frac{F_1(\operatorname{prec}(S, R), \operatorname{rec}(S, R))}{\log_2(1 + \operatorname{gran}(S, R))} \in [0, 1]. \tag{6}$$

3 Feature Extraction and Seed Generation

In this section we describe our approach to extract seeds of passages from $X \times Y$ for some pair of documents X and Y. We decided to apply feature extraction on a per document basis. Hence, this step can be done in a preprocessing phase before actually considering pairs of documents.

3.1 Extraction of contextual features for documents

Throughout the paper, F denotes the set of all features. As seen below, we use skip word ngrams as features, but for the sake of clarity we keep it general first.

Given a document $X = \{x_i\}$ we map each character to a finite set of binary features $x_i \mapsto \varphi(x_i) = \{f_{i1}, \ldots, f_{id}\}$, $d = d(i)$, $f_{ij} \in F$. Recall that the power set $\mathcal{P}(F)$ is the set of all subsets of F. Therefore, we defined a *feature map*

$$f_X \colon X \to \mathcal{P}(F). \tag{7}$$

In return, by mapping $F \ni f \mapsto \{x_i \mid f \in \varphi(x_i)\} \in \mathcal{P}(X)$ we get an *index map*

$$\iota_X \colon F \to \mathcal{P}(X). \tag{8}$$

It tells us at which character positions in the document a feature f is present.

In our approach we first tokenized the document X into a sequence of lowercased stemmed words $w_0, w_1, \ldots$ by omitting whitespaces, non-alphanumeric characters and stop words, and used skip-bigrams of length 1 to 4:

$$\varphi(x_i) = \begin{cases} \{w_\beta w_\alpha\}_{\beta=\alpha-4,\ldots,\alpha} & \text{if } x_i \text{ is the beginning character of a word } w_\alpha. \\ \emptyset & \text{otherwise.} \end{cases} \tag{9}$$

Table 1 illustrates the feature extraction for the example phrase *The quick brown fox jumps over the lazy dog*.

Table 1. Features extracted for the characters $x_0, \dots x_{43}$ of the example string *The quick brown fox jumps over the lazy dog*. Characters that were not at the beginning of a contiguous alphanumeric substring have no features and are omitted from the table. The symbol $\star$ is a placeholder for the empty word.

Offset	Token	f_1	f_2	f_3	f_4
4	quick	⋆_quick	⋆_quick	⋆_quick	⋆_quick
10	brown	quick_brown	⋆_brown	⋆_brown	⋆_brown
16	fox	brown_fox	quick_fox	⋆_fox	⋆_fox
20	jump	fox_jump	brown_jump	quick_jump	⋆_jump
35	lazy	jump_lazy	fox_lazy	brown_lazy	quick_lazy
40	dog	lazy_dog	jump_dog	fox_dog	brown_dog

3.2 Selection of relevant features

Clearly, features which are not present at all, or appear at almost every character are useless for the text alignment task. In order to reduce the number of generated features, we apply a simple feature selection strategy. If $\iota_X(f)$ is non-empty and has low cardinality, then we consider f as meaningful. We say that $f \in F$ is a *relevant feature (for X)*, if the cardinality satisfies the following estimation:

$$1 \leq |\iota_X(f)| \leq \varrho \tag{10}$$

for some given threshold parameter ϱ. The latter is also called *relevance threshold*. The subset of all relevant features is denoted as $F_X \subseteq F$.

In our approach we use a constant relevance threshold $\varrho = 4$. For time constraints we kept this simple approach because it already worked quite well.

3.3 Feature extraction of document pairs and seed generation

The feature extraction for documents carries over to feature extraction to pairs of documents. Given a suspicious document X and a source document Y their index maps give rise to a natural index map of $X \times Y$:

$$\iota_{XY} \colon F \to \mathcal{P}(X \times Y), \quad f \mapsto \{(x_i, y_j) \mid f \in \varphi(x_i) \text{ and } f \in \varphi(y_i)\}. \tag{11}$$

It maps a feature f to the set of all pairs (x_i, y_j) of characters such that f is simultaneously present at x_i and y_i. Now let $F_X \cap F_Y \subseteq F$ be the subset of features which are relevant for X and Y. The union

$$\sigma(X, Y) = \bigcup_{f \in F_X \cap F_Y} \iota_{XY}(f) \subseteq X \times Y \tag{12}$$

is the *seed set* of plagiarism cases between X and Y. These atomic plagiarism cases are the starting point of a merge process, which is explained in the next section.

We can deduce an estimation of the cardinality in terms of the relevance threshold. Namely, for each $f \in F_X \cap F_Y$ holds the estimation

$$1 \leq |\iota_{XY}(f)| = |\iota_X(f)| \cdot |\iota_Y(f)| \leq \varrho^2. \tag{13}$$

Hence, $|\sigma(X,Y)| \leq \varrho^2 |F_X||F_Y|$. Consequently, the number of seeds can be estimated just in terms of X and Y without having to inspect the pair (X,Y).

4 Merging

In this section we describe the process of merging passage references in the product space $X \times Y$ for two documents X and Y. The merge criterion will be given in terms of a distance function and the merge process is then the agglomerative single-linkage clustering with an additional termination condition.

4.1 Merge criteria

In order to define a distance between two passage references, let us first introduce further notation.

For two non-empty passages $P_1 = [x_{a_1}, x_{b_1}]$ and $P_2 = [x_{a_2}, x_{b_2}]$ in X let their *distance* be $\operatorname{dist}(P_1,P_2) = \min\{|u_1 - u_2| \colon a_1 \leq u_1 \leq b_1, a_2 \leq u_2 \leq b_2\}$. It is positive if and only if P_1 and P_2 are disjoint. Now, for two non-empty passage references $P_1 \times Q_1, P_2 \times Q_2 \subseteq X \times Y$ their *distance* is

$$\operatorname{dist}(P_1 \times Q_1, P_2 \times Q_2) = \frac{2\operatorname{dist}(P_1,P_2) + 2\operatorname{dist}(Q_1,Q_2)}{\sigma + \pi(P_1 \times Q_1) + \pi(P_2 \times Q_2)}, \tag{14}$$

where $\sigma > 0$ denotes a constant smoothing parameter that is defined empirically. The distance is zero if and only if $P_1 \cap P_2 \neq \emptyset$ and $Q_1 \cap Q_2 \neq \emptyset$, or equivalently $P_1 \times Q_1 \cap P_2 \times Q_2 \neq \emptyset$, and thus reflects the fact that two passages are directly adjacent. If the distance is positive, but lesser than a given threshold τ, the passages have empty intersection but are relatively close.

The *merge* of two passage references in $X \times Y$ is the smallest passage reference that contains both. It always contains their union but is in general strictly larger.

4.2 Agglomerative clustering

Having defined criteria for merging two passage references, we apply agglomerative single-linkage clustering. That is, in each step we merge a pair of passage references that have minimal distance. If there is no pair whose distance is lesser or equal than a given constant $\tau > 0$, the process terminates.

5 Filtering and passage extraction

At the end of the merge process we remove all passage references where the suspicious passage has less than 15 words. The remaining passage references are the detected plagiarism cases.

6 Evaluation Results

We evaluated the algorithm on the data sets of the previous competition PAN2013 [3] and the current competition PAN2014 using *Tira*[1]. For completeness, we also state the runtime, although it does not affect the ranking of the algorithm in the competition. See Table 2 for the development results and Table 3 for the end results.

We also ran smaller experiments restricted to a fixed obfuscation strategy (Table 4). To no surprise the algorithm underperforms for summary obfuscated plagiarism cases because we use no synonym dictionaries.

Table 2. Text alignment results with retrieval performance and runtime for the data sets of PAN2013. The experiments were executed on an 8-core system.

Corpus	Pairs	PlagDet	Precision	Recall	Granularity	Runtime
pan13-training-corpus	4007	**0.825**	0.935	0.743	1.00485	48s
pan13-test-corpus1	399	**0.837**	0.930	0.760	1.00000	5s
pan13-test-corpus2	4042	**0.831**	0.939	0.750	1.00421	4ls

Table 3. Text alignment results with retrieval performance and runtime for the data sets of PAN2014. The experiments were executed on an 1-core system, provided by the host.

Corpus	Pairs	PlagDet	Precision	Recall	Granularity	Runtime
pan14-training-corpus	5164	**0.821**	0.928	0.763	1.02805	183s
pan14-test-corpus2	?	**0.826**	0.933	0.766	1.02514	180s
pan14-test-corpus3	?	**0.855**	0.925	0.818	1.02187	169s

Table 4. Text alignment *plagdet* scores with respect to obfuscation strategies

Corpus	None	Random	Cyclic translation	Summary
training-corpus-2013-01-21	0.903	0.812	0.824	0.299
test-corpus1-2013-03-08	0.901	0.791	0.854	0.412
test-corpus2-2013-01-21	0.913	0.811	0.835	0.316

7 Conclusion

The simple heuristics we used in our approach, already worked quite well and comparable to the state of the art text alignment algorithms for random and translation cycle obfuscations. In order to tackle the problem of summary obfuscation in the future, we intend to incorporate more semantic knowledge into the feature extraction stage.

References

1. Gollub, T., Potthast, M., Beyer, A., Busse, M., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Recent trends in digital text forensics and its evaluation. In: Forner, P., MÃijller, H., Paredes, R., Rosso, P., Stein, B. (eds.) Information Access Evaluation. Multilinguality, Multimodality, and Visualization, Lecture Notes in Computer Science, vol. 8138, pp. 282–302. Springer Berlin Heidelberg (2013)
2. 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: CLEF 2013 Evaluation Labs and Workshop - Working Notes Papers (2013)
3. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 997–1005. COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010), http://dl.acm.org/citation.cfm?id=1944566.1944681