The Encoplot Similarity Measure for Automatic Detection of Plagiarism
Notebook for PAN at CLEF 2011

Cristian Grozea[1]* and Marius Popescu[2]

[1] Fraunhofer Institute FIRST, Berlin, Germany
[2] University of Bucharest, Romania

Abstract This paper describes the evolution of our method Encoplot for automatic plagiarism detection and the results of the participation to the PAN'11 competition. The main novelties are the introduction of a new similarity measure and of a new ranking method, which cooperate to rank much better the source–suspicious document pairs when selecting the candidates for the detailed analysis phase. We have obtained excellent results in the competition, ranking 1^{st} on the manually paraphrased cases, 2^{nd} overall in the external plagiarism detection task, and getting the best recall on the non-translated corpus.

1 Introduction

Encoplot is an automatic method for plagiarism detection developed by the authors. It was first introduced in the PAN'09 competition, where it has outperformed all other competing methods [3]. It has been since then enhanced, parallelized and used also for detecting the direction of plagiarism [6].

For this year's competition a new similarity measure for the candidate documents retrieval phase has been introduced, aiming to increase the quality of the ranking and implicitly allowing for an increased recall. Apart from the improvement of the recall, this new similarity measure increases the consistency of the encoplot method, the same strategy being now used both in the candidate documents retrieval phase and in the detailed analysis phase, making thus more probable that a correct find in the first phase will not be missed by the second phase and the other way around.

All details omitted here for brevity are given in the extended version of this paper, available online [4], along with figures suitable for viewing without rescaling.

2 Methods: Encoplot

Encoplot is both the name of the plagiarism detection system we have developed and used, and the name of the pairwise document comparison algorithm at its core. For the sake of completion, we describe these again here to the extent the available space allows for, marking also the changes to the version described in [3], where optimized C-language code is included.

* corresponding author – cristian.grozea@brainsignals.de

2.1 The Core Encoplot Algorithm for Comparing Two Documents

Input:

- A and B, two strings of length l_A and respectively l_B over some alphabet Σ.
- N, a natural number – the length of the N-grams.

Output:
$E \subseteq \{1 \dots l_A\} \times \{1 \dots l_B\}$, such that:

- For each $(u, v) \in E$ the N-grams A(u…u+N-1) and B(v…v+N-1) coincide.
- Each index appears at most once: $(u, v) \in E, (u, v') \in E \Rightarrow v = v'$; $(u, v) \in E, (u', v) \in E \Rightarrow u = u'$.
- For each N-gram $w \in \Sigma^N$, let $n_w A, n_w B$ be the number of the occurrences of w in A and B, respectively. Then $|\{(u, v) \in E; A(u \dots u + N - 1) = w\}| = min(n_w A, n_w B)$, where $|X|$ denotes the cardinal of the set X. Further more, if $a_w(i)_{i=1 \dots n_w A}$ are the ascendingly ordered positions on which w occurs in A and $b_w(i)_{i=1 \dots n_w B}$ the ones for B, then $(a_w(i), b_w(i)) \in E$, for $i = 1 \dots min(n_w A, n_w B)$.

Algorithm:

1. Extract and sort the N-grams in A, let the result be denoted by $S_A \subseteq \{1 \dots l_A\} \times \Sigma^N$. The serial implementation produces a sorting index with a radix sort specialized in sorting N-grams.
2. Do the same for B, let the result be denoted by $S_B \subseteq \{1 \dots l_B\} \times \Sigma^N$.
3. Intersect with a merging procedure the projections of S_A and S_B on their second component, while reporting the values in the first component for the matches.

Note that the set produced by encoplot for two documents is a subset of the set of “dots“ produced by the well-known method dotplot [2].

2.2 The Clustering Heuristic

The heuristic employed for clustering the “dots” produced by the encoplot core algorithm into passages is ommited here for brevity, being included in the extended version of this paper, available online [4].

2.3 The Encoplot Plagiarism Detection System

A run of the whole system consists in the following steps:

1. Preprocess the text, text normalization (including translation when needed).
2. Compute a similarity matrix with one value for each (source document, suspicious document) pair.
3. Rank the document pairs based on their similarity.
4. Analyze the highest ranked pairs in detail, with the following sub-steps:
   - Apply the core encoplot algorithm described above on the two texts and obtain the encoplot data.
   - Cluster the matches of N-grams into matches of passages, using the heuristic algorithm mentioned above.

2.4 Changes to the System

Translation and text normalization are new features for our system.For selecting the candidate pairs for detailed analysis, we have introduced both a new similarity measure and a new ranking method.

The new similarity measure leverages the ideas of core encoplot algorithm, by computing the encoplot set for the pair of documents of which the similarity is evaluated, followed by a projection of it on the source axis, and a counting of in how many positions a moving window of fixed size (256 characters) contains a number of N-grams matches above a fixed threshold (64). This count is taken to be the similarity of the two documents.

Once the similarity measure matrix is obtained (11093x11093 in this case), we need to rank the pairs based on those values. In the previous years we have considered and contrasted ranking all sources for a given suspicious document and ranking all suspicious documents for a given source. Which one is best, depends much on the dataset ([5]), therefore we have decided to combine the two rankings into a single ranking that guarantees that whichever pair was selected by either one of those, will be selected by the combined ranking as well. For this we built **the min ranking** $r_{min}(pair) = min(r_{sources}(pair), r_{suspicious}(pair))$, where $r_{sources}(pair)$ is the rank of the pair in its column in the similarity matrix and $r_{suspicious}(pair)$ is its rank in the row containing it in the same matrix.

The detailed analysis remained almost unchanged, still we have had to do a change related to a serious problem the 2010 competition corpus had, namely some of the passages from the source were copied multiple times into the destination suspicious document – a substantial amount: out of 55723 external plagiarism instances, 10694 ($> 19\%$) had the multiplicity at least 2, 3483 multiplicity at least 3. The maximum multiplicity of a single passage was 17 (!).

This probably explains our suspiciously low recall in the 2010 competition on the non-obfuscated cases (and other subcorpora). As a side effect of the speed and space optimizations the core encoplot algorithm offers over dotplot, for the simple case when there is no obfuscation at all and just verbatim copying multiple times, only the first copy of a passage is matched. To understand why, remember that each position in the source is paired with at most one position in the suspicious document. Therefore a full match of the source passage fully "consumes" it, and it cannot match any of the subsequent copies. Having a second copy of the same passage in the source would allow for a second match and so on. To cope with that, we have concatenated each source with itself 4 times before analyzing the pair in detail with our heuristic, creating thus 4 copies in the source of each passage previously there. The number 4 has been chosen as a compromise, balancing the effort and the expected increase in recall.

2.5 Software Infrastructure

For parallelizing the computation of the similarity matrix we have switched from OpenMP to the framework StarSs [7], developed at the Barcelona Supercomputing Center (BSC). The advantage of this task-based parallelization framework is that with simple annotations, tasks can be defined that are executed then in parallel on different execution threads. The necessary tasks dependency detection, locking and thread-to-thread communication and data transfers are performed automatically by the framework. We have developed this version of the code during a HPC Europa2 virtual visit to the BSC, where the first author had the chance to work together on this with Jesus Labarta, Rosa M. Badia and Aislan Gomide Foina and to run the code on machines with up to 256 cores (article in preparation).

3 Evaluation

3.1 Processing Speed

The main machine used was a 12-core machine: 2xAMD 6-core Opteron 2427 64bit 2200 MHz in our multi-core lab.

Translation was prohibitively slow, and there was not enough time for it (just 48h), see the Discussion section in [4] for more. Other preprocessing took less than 1 hour on this machine. The computation of the similarity matrix took 24h on the same machine. Ranking took a few minutes. The detailed analysis took 24h on the local Condor cluster (34 cores).

3.2 Dataset

This year's competition test data consisted of 11093 source documents in English, German (about 500) and Spanish (about 200).

There were 49621 plagiarism cases, more than 10% of which involved translation. The distribution of their types is given in Table 1.

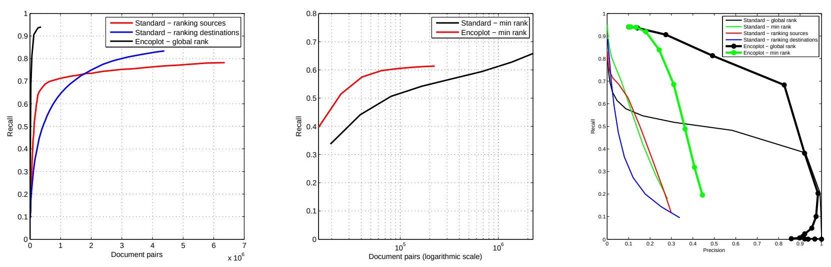
At document level, out of all 120 million document pairs, only 17674 contained plagiarism cases.

3.3 Analysis

As our method consists of two distinct phases, ranking the document pairs followed by detailed analysis of the selected ones, it is useful to measure the performance in each phase, in order to be able to contrast the alternative approaches.

The purpose of the ranking is to identify the pairs of documents for which there is a high chance that there is at least one plagiated passage from the source to the suspicious document. An ideal ranking should put first the pairs where such a relation exists and last the other pairs. A good ranking should be an approximation of this. It should enable the achievement of a good recall at document level without having to select too many document pairs. Note that all rankings (even a random one) will achieve a recall of 100%, the latest after selecting all pairs. What is important is how fast the recall is increasing when the selection is extended to include more and more of the document pairs, as ranked.

For examining the performance of the ranking, we have considered two types of graphs: effort versus recall – where the recall is shown together with the number of pairs needed to achieve that recall – and a standard precision versus recall.



(a) 2010 (b) 2011 (c) Precision-Recall

Figure 1. (a) Comparative recall at document level on the 2010 competition corpus of the newly introduced encoplot similarity measure and of the two ranking variants we have employed in 2010, based on a standard string kernel. The encoplot similarity measure graph ends when all document pairs with non-null similarity values are included. (b) Same for the 2011 competition corpus, using the ranking which performed best for the standard string kernel. (c) Precision-Recall plots for several ranking types for the standard and the encoplot based similarity measures, on the 2010 competition corpus data.

On the 2010 competition data, the effect of the new similarity ranking is impressive. Both on the “effort versus recall” plots (Figure 1(a)) and in the “precision versus recall” plots (Figure 1(c)), the new similarity method outperforms the standard one consistently by a great margin.

On the 2011 competition data, while it still dominates in the “affordable effort” range, it is eventually outperformed by the standard kernel coupled with min rank, Figure 1(b).

3.4 Results

The results on the 2011 competition data are summarized in Table 1.

Table 1. Results on 2011 Competition Data

Subset	Size	Recall	Precision	F-score	Granularity	Plagdet score
Entire corpus	49,621	0.34	0.81	0.48	1.22	0.42
No paraphrasing	976	0.90	0.84	0.86	1.02	0.85
Manual paraphrasing	4,609	**0.36**	**0.96**	**0.53**	1.06	**0.50**
Automatic paraphrasing low obfuscation	19,779	**0.58**	0.90	0.71	1.27	0.60
Automatic paraphrasing high obfuscation	19,115	**0.08**	0.64	0.14	1.19	0.13
Manual translation	433	0.08	0.25	0.12	1.01	0.12
Automatic translation	4,709	0.23	0.40	0.29	1.07	0.28

Our team has ranked 2^{nd} on the external plagiarism detection task.

We have had the best score among all teams on the category that mimics best the real human plagiarism: manual paraphrasing. Also we have had the best recall values on manual paraphrasing and on both subcategories of automatic paraphrasing.

By using the annotations provided we have been able to compute for all teams the recall on the non-translated cases subcorpus. We lead also there, with 0.3512, followed by the team that ranked first with 0.3468. It was impossible to compute the precision on the same subcorpus without having access to the answer files of the other teams.

We have tested the new method also on the 2010 competition data, on which it had a plagdet score 0.72, with recall 66% and precision 86%. These results are very encouraging, they would have ranked our method on the second place, without even handling the translated cases (14%) - whereas the team which ranked first in 2010 did handle those too.

4 Discussion

The distribution of the plagiarism types in this year's test data is far from matching their distribution in the training corpus, PAN-PC-10 [9]. Most notably, the amount of passages copied without changes (no obfuscation, no translation) was heavily reduced, from 40% to less than 2%. Therefore, tuning on the training corpus could have been to the disadvantage of the participants. We were consistent with our previous approach of not tuning our method to a particular and still mostly artificial training corpus. This may explain to some extent our top performance in the cases of human plagiarism, which is the only one of practical interest. We expect that the next corpora will contain more and more such human and manually simulated plagiarism to the detriment of the artificially generated using questionable choices (see the repeated insertion of the same passage from one source into a single destination suspicious document mentioned above).

4.1 The Issue with the Translation

Automatic translation is a rather separate problem in NLP. The idea to translate first, followed by the same language plagiarism detection is neither a scientific contribution, nor a distinguishing feature for a plagiarism detection system. In the extended version of this paper [4] you can find our entire treatment of this issue, including an example of how much obfuscation is introduced by a mismatch of the translation engines employed.

4.2 The Issue with the Granularity Correction in the Plagdet Score

The granularity has been introduced for plagiarism detection in [8]. It was meant to correct the standard F-score for excessive splitting of the plagiarized passages retrieved. It is an ad-hoc correction that divides the F-score by $log_2(1 + granularity)$. It exhibits unwanted behavior in certain cases. For example, let's assume we compare with plagdet two methods, one having recall 33.33%, precision 100% and granularity 1 with another method having both precision and recall 100% and granularity 3. The two methods will obtain the very same plagdet score, 0.5, as a result of applying the granularity correction, although the second method is obviously to be preferred. It has 100% recall and precision, it finds everything and nothing more and even the splitting is very far from excessive. No user will ever prefer a software that fails to find two thirds of the cases to a software that finds them all and even displays each as one block (when colouring text blocks, the adjacent parts will visually join).

More thought should be spent in finding a reasonable plagiarism detection score.

5 Conclusion

In this paper the newest changes to the plagiarism detection method Encoplot have been presented, as well as the results in the competition PAN'11. The main change was replacing the standard, kernel based pairwise document similarity measure by a new similarity measure that includes some of the encoplot core ideas. We have shown that this new similarity measure is to be preferred when only a small part of the whole set of document pairs can be analyzed in detail, as the recall at document level increases much faster with the here proposed similarity measure. The results in the competition were excellent, our team obtained the best scores for the manual paraphrasing subcorpus and ranked 2nd overall on the external plagiarism detection task, obtaining also the best recall on the subcorpus of non-translated plagiarism cases.

Acknowledgment: The parallelization of Encoplot has been performed under the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission - Capacities Area - Research Infrastructures.

References

1. Braschler, M., Harman, D., Pianta, E. (eds.): CLEF 2010 LABs and Workshops, Notebook Papers, 22-23 September 2010, Padua, Italy (2010)
2. Clough, P.: Old and new challenges in automatic plagiarism detection. National Plagiarism Advisory Service (2003)
3. Grozea, C., Gehl, C., Popescu, M.: ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In: 3rd PAN WORKSHOP. UNCOVERING PLAGIARISM, AUTHORSHIP AND SOCIAL SOFTWARE MISUSE. p. 10 (2009)
4. Grozea, C., Popescu, M.: The Encoplot Similarity Measure for Automatic Detection of Plagiarism - Extended Technical Report. `http://brainsignals.de/encsimTR.pdf` (Aug 2011)
5. Grozea, C., Popescu, M.: Encoplot - Performance in the Second International Plagiarism Detection Challenge - Lab Report for PAN at CLEF 2010 . In: Braschler et al. [1]
6. Grozea, C., Popescu, M.: Who's the Thief? Automatic Detection of the Direction of Plagiarism. In: Gelbukh, A.F. (ed.) CICLing. Lecture Notes in Computer Science, vol. 6008, pp. 700–710. Springer (2010)
7. Planas, J., Badia, R.M., Ayguadé, E., Labarta, J.: Hierarchical task based programming with StarSs. International Journal of High Performance Computing 23(3), 284–299 (August 2009)
8. 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. Association for Computational Linguistics (2010)
9. Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., Rosso, P.: Overview of the 2nd International Competition on Plagiarism Detection. In: Braschler et al. [1]