Encoplot – Performance in the Second International Plagiarism Detection Challenge

Lab Report for PAN at CLEF 2010

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

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

Abstract Our submission this year is generated by the same method Encoplot that we have developed for the last year competition. There is a single improvement, we compare in addition each suspicious document with each other and flag the passages most probably in correspondence as intrinsic plagiarism.

1 Introduction

Our method Encoplot[3] has won the last year competition in the external plagiarism detection task and in the overall ranking. Since PAN'09[5] we have tested it on some other tasks and it proved good even for the detection of the direction of the plagiarism: on the PAN'09 corpus it was able to indicate the source for each plagiarism instance, with 75% accuracy[4].

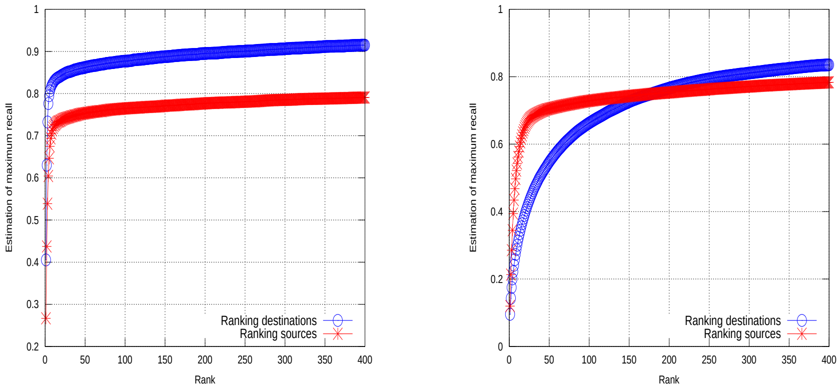
2 External Plagiarism Detection

The method consists in two stages. In a first stage the values from a string kernel matrix are computed, that give a rough approximation of the similarity between each two documents (a source and a suspicious one). The string kernel used is the normalization of the kernel that counts for each two strings how many character-based N-grams types of a fixed length N they share.

In the second stage the most promising pairs are examined in detail by creating the encoplot data for them and the passages in correspondence are extracted based on some simple heuristics. The encoplot data is a subset of the dotplot. It consists of a subset of the set of indexes on which the two documents have the same N-gram, thus: the first occurrence of an N-gram in one document is paired with the first occurrence of the same N-gram in the other document, the second occurrence with the second one and so on. Computationally, it is worth mentioning that computing the encoplot data is done in linear time, with the algorithm we have published in [3]. On the other hand, since our method is based on pairwise comparisons of documents, its runtime is of quadratic order in the number of documents.

* corresponding author – cristian.grozea@first.fraunhofer.de

A critical choice is how to rank the similarity measures, how to prune the huge list of document pairs without loosing too much from performance. We had noticed in the previous competition that ranking all possible target documents for each fixed source leads to higher recall than ranking for each suspicious document (target) the sources. This can be seen in Figure 1(a), ranking the targets offers a consistent about 10% higher recall, therefore this is what we have used. Based on those graphs, we have decided for looking this time at the first 400 most likely targets for each source, up from the 50 we considered in the previous competition.



(a) Pan'09

(b) Pan'10

Figure 1. Pruning: Is it better to rank the possible sources for each suspicious document or the other way around?

We have kept the same method without any extra tuning - all hyperparameters but this one were kept to the values given in the Encoplot paper [3]. As a result of the increased size of the dataset – $12000 * 16000$ document pairs instead of $7000 * 7000$ – computing the kernel matrix took this time 40 hours on a similar 8-core machine. Leveraging the speed of Encoplot, we have been able to extend the detailed analysis to the first (closest) 400 suspicious documents to each source document (4.8 million document pairs, about 2.5% of the total number before ranking and pruning). This took about a week to process on the available machine.

It turns out that for this dataset, for low number of neighbors, it is much better to rank the possible sources for each target than to rank the possible targets for each source, as in Figure 1(b). The difference is as extreme as between 38% and 64% when the first 20 neighbors are used. For the number of neighbors we have chosen, 400, our choice looks on the first glance close to ideal – more about that in the Evaluation and Discussion section below.

3 Internal Plagiarism Detection

Our position is that given the weaknesses of the intrinsic plagiarism detection and the higher probability that the source is anyway available to the investigator, it is better to attempt external plagiarism detection with an enlarged set of sources instead of risking with the low-performing intrinsic plagiarism detection methods. It was unclear whether this would be allowed by the rules – as a side note, in other data competitions such as VOC[1] there are separate tracks for learning only from the data explicitly provided and for the methods/submissions using supplementary data. Therefore we didn't test the suspicious documents against the Gutenberg archive – the most probable source. Instead we have performed a natural test that is typical done in a way or another by any professor suspecting the homeworks received of potential plagiarism [2]: we compared the suspicious documents (restricted on purpose only to the ones for which we had not found any external plagiarism) to each other. This comparison was done again using the method Encoplot with standard hyper-parameters. We have filtered the resulting passages in order to retain only the ones with very high probability of being copies to their found corresponding passage. The criteria used were: the length of each of the two passages in correspondence being over 1500 and having the matching score (roughly the percentage of N-grams that are matched) at least 0.95.

Those passages have been reported as intrinsic plagiarism, as they have been identified as plagiarism by this document-set method but the source is not known. It may well be that in fact the two passages are copied from a third source.

4 Evaluation and Discussion

Since our method was already proven in the last year competition (on which PAN-PC-09 is based), we did not evaluate it again on this dataset.

Our results in the competition are given in the Table 1.

As far as the current competition is concerned, we have the confirmation that our decision to add cross-suspicious-documents detected passages led to an increase in recall that eventually gave a similar increase in the overall score. The precision is also slightly higher, as a result of the strong filtering of the cross-suspicious-documents detections. Hadn't we filter those, we would have obtained an higher recall (2.5% higher) at the expense of a lower precision (2% lower). Overall this would have improved further (but only slightly – 1%) the overall score.

Table 1. Results on various datasets

Dataset	Overall Score	Recall	Precision	Granularity
PAN'09	0.6957	0.6585	0.7418	1.0038
PAN'10	0.6154	0.4742	0.9073	1.0168
PAN'10 external detections only	0.6048	0.4602	0.9016	1.0106
PAN'10 with unfiltered internal detections	0.6252	0.5042	0.8852	1.0387

The most interesting question is why is our score lower this year? Could it be that we didn't detect the non-artificial plagiarism as good as the artificial one? The score went down mostly as a result of the decrease in recall. We suggest as a possible explanation that the Figures 1(a) and 1(b) are over-optimistic for high number of neighbors. The actual recall we obtained in these two competitions corresponds to the ideal values for 10 to 20 neighbors. A likely explanation is that, as the kernel and the encoplot agree on their conclusions, encoplot will fail to notice common passages in the documents where the kernel failed to notice similarity. If this is true, then, according to Figure 1(b) we have used the lower performance ranking and better should have been to rank the sources for each target, that would have brought us a substantial performance boost without any change in the algorithms.

5 Conclusion

Although three groups have succeeded to outperform our method in the challenge, we have argued here that our results could have been improved through pruning by ranking the source documents for each suspicious one, instead of ranking the suspicious documents for each source document. Given the consistent performance and the versatility of Encoplot, we plan to apply it to new fields in the future.

References

1. Visual Object Classes Challenge, http://pascallin.ecs.soton.ac.uk/challenges/VOC/
2. Grozea, C.: Plagiarism detection with state of the art compression programs. Report CDMTCS-247, Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, Auckland, New Zealand (Aug 2004), http://www.cs.auckland.ac.nz/CDMTCS/researchreports/247Grozea.pdf
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
4. Grozea, C., Popescu, M.: Who's the thief? automatic detection of the direction of plagiarism. In: CICLing. pp. 700–710 (2010)
5. Webis at Bauhaus-Universität Weimar, NLEL at Universidad Politécnica de Valencia: PAN Plagiarism Corpus PAN-PC-09. http://www.webis.de/research/corpora (2009), Martin Potthast, Andreas Eiselt, Benno Stein, Alberto Barrón Cedeño, and Paolo Rosso (editors)