A plagiarism detector for intrinsic plagiarism
Lab Report for PAN at CLEF 2010

Pablo Suárez[1], José Carlos González[1,2], Julio Villena-Román[1,3]

[1] DAEDALUS – Data, Decisions and Language, S.A. Avda. De la Albufera, 321
28031 Madrid, Spain
{psuarez, jgonzalez, jvillena}@daedalus.es

[2] ETSI Telecomunicación, Universidad Politécnica de Madrid,
28040 Madrid, Spain
josecarlos.gonzalez@upm.es

[3] Telematic Engineering Department, Universidad Carlos III de Madrid,
28911 Leganés, Spain
jvillena@it.uc3m.es

Abstract. In this paper, we describe the algorithm that has been used to carry out our plagiarism detection within the context of PAN10 competition. Our system is based on the LempelZiv distance, which is applied to extract structural information from texts. Then the algorithm tries to find outliers in the vector of distances between each fragment of the text and the whole document itself.

1 Introduction

This paper is structured as follows: Section 2 is devoted to the description of the intrinsic plagiarism algorithm. Section 3 is devoted to the system evaluation. Finally, Section 4 includes some conclusions and future work.

2 Intrinsic plagiarism

The first algorithm in which we worked was the intrinsic plagiarism one, and it was the only type of analysis that we carried out for PAN10 competition.

2.1 Global architecture

Next figure shows the global architecture for our intrinsic plagiarism algorithm.

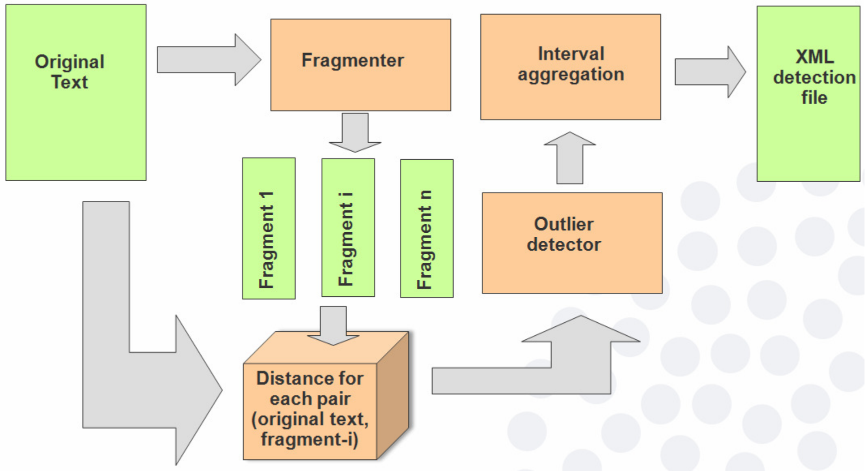

Figure 1. Intrinsic plagiarism global architecture.

2.2 Fragmenter

This module fragments the original text in blocks. Our software offers two different possibilities: 1) fragmentation by sentences, and 2) fragmentation by paragraphs. The minimum size allowed for the fragments or text blocks is a configurable parameter in our system. It is necessary, since over a small fragment is not valid to detect the presence of plagiarism.

2.3 Detection distances

The current version of our algorithms includes, among others, the implementation of the next definitions for distances:

Basile distance: proposed by Basile and others, that define a distance between two texts *x* and *y* from its n-grams ([1], [2]):

LempelZiv distance: it is a Kolmogorov distance implemented by means of the LempeZiv compression algorithm, as described in [3].

RHonore distance: as described in [4].

Our algorithms can use one or a subset of the available distances by means of a configurable parameter. In our detection of intrinsic plagiarism for PAN10 we have only taken into account the LempelZiv distance, since it has been shown that measures based on Kolmogorov complexity (using a lossless compression algorithm) are a good way to extract structural information from texts for the intrinsic plagiarism detection [6].

2.4 Outlier detection

Next step consists of detecting which distance can be considered as an outlier in the vector of distances between each fragment of the text and the whole document itself. Our software implements three classical ways of detecting an outlier in a list of data [5]. They are: standard deviation (Chebyshev), percentiles and MAD (Median Absolute Deviation). In particular, the selected threshold for each case is: $t=\alpha*\sigma+\overline{x}$ (for standard deviation), $t=Q_3 + \beta*(Q_3-Q_1)$ (for percentiles) and $t=\overline{x}+\gamma*\text{MAD}$ (for MAD). Where α, β and γ are configurable weights that we used with values $\alpha=0.9$, $\beta=1.5$ and $\gamma=3.0$. It can be used only one or a subset of outlier thresholds by means of a configurable parameter. We only used MAD for PAN10.

2.5 Interval aggregation

Interval aggregation is an optional module that can be used in the output of our system. It aggregates a group of separated detected plagiarism intervals into one interval when interval separation is smaller than a configurable threshold. It permits detecting as a unique plagiarized block some close blocks that were separated by the fragmenter. For PAN10 we did not use this interval aggregation module.

3 Evaluation

With respect to PAN10 competition, as stated above, we have only participated in the intrinsic plagiarism detection task, because of (software or hardware) bad performance of our system for external plagiarism. In this case, the configurable parameters of our plagiarism detector are: fragmentation level (sentence, paragraph), minimum length of interval (minimum length for being considered a valid sentence or paragraph), use of interval aggregation (true, false), aggregation interval (minimum distance between intervals for aggregation), minimum fragment length (minimum fragment length for plagiarism detection), active comparison distances (Basile, LempelZiv, RHonore), outlier detection method (standard deviation, percentiles, MAD), α, β and γ weights for outlier detection. Our settings, after from different tests on the training corpus PAN-PC-09, were: fragmentation level = paragraph, minimum length of interval = 200, use of interval aggregation = false, aggregation interval = 50, minimum fragment length = 200, active comparison distances = only LempelZiv, outlier detection method = standard deviation, weights for outlier detection $\gamma = 3.0$.

The detection performance that our system achieves on the training corpus PAN-PC-09, using the PAN evaluation measures, was: recall=0.185225576213, precision=0.075230788299, overall=**0.0743645119788**, granularity=1.71111111111.

Whereas our final results in the PAN10 were: recall=0.0615, precision=0.1349, overall=**0.0498**, granularity=2.2376. These results rank 16th in the participant list.

4 Conclusion

As we noted earlier, we have only participated in the intrinsic plagiarism detection task. Since the results of the competition cover the detection of both intrinsic and external plagiarism globally, and not separately, the overall results had to be necessarily worse. In that sense, we are sure that we can greatly improve our current system with our future work. In any case, the results have not been too good at the moment. Our future work will include, in fact, the following tasks: 1) Improve intrinsic and external plagiarism performance; 2) Combine intrinsic and external plagiarism; 3) Develop the Internet module; 4) Implement new detection distances; 5) Implement new outlier detection methods; 6) Implement 'obfuscation' detection algorithms; 7) Implement a report generator module.

Acknowledgements

This work has been partially supported by the Spanish Center for Industry Technological Development (CDTI, Ministry of Industry, Tourism and Trade), through the CONTENIDOS A LA CARTA project, INGENIO 2010 Programme, AVANZA I+D 2008.

References

1. BASILE, C. et al. 2008: “An example of mathematical authorship attribution”. In: *Journal of Mathematical Physics*, 49:125211–125230.
2. BASILE, C. et al. 2009: “A plagiarism detection procedure in three steps: selection, matches and ‘squares’”. In: PAN-09 Competition.
3. BELABBES, Sigem et al. 2008: “On Using SVM and Kolmogorov Complexity for Spam Filtering”. In: *Proceedings of the Twenty-First International FLAIRS Conference*.
4. BARRÓN, Luis Alberto 2008: “Detección automática de plagio en texto”. In: <*http://mavir2006.mavir.net/docs/Barron-DeteccionPlagioTexto.pdf*>.
5. IRANZO PÉREZ, David 2007: Análisis de Outliers: un caso a estudio. PhD Thesis. Universitat de València. Servei de publicacions. In: <*http://www.tesisenxarxa.net/TESIS_UV/AVAILABLE/TDX-1007108-124618//iranzo.pdf*>.
6. SEAWARD, Leane and MATWIN, Stan 2009: “Intrinsic Plagiarism Detection using Complexity Analysis”. In: Stein, Rosso, Stamatatos, Koppel, Agirre (Eds.). PAN'09, pp. 56-61.