UFRGS@PAN2010: Detecting External Plagiarism

Lab Report for Pan at CLEF 2010

Rafael Corezola Pereira, Viviane P. Moreira, Renata Galante

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{rcpereira, viviane, galante}@inf.ufrgs.br

Abstract. This paper presents our approach to detect plagiarism in the PAN'10 competition. To accomplish this task we applied a method which aims at detecting external plagiarism cases. The method is specially designed to detect cross-language plagiarism and is composed by five phases: language normalization, retrieval of candidate documents, classifier training, plagiarism analysis, and post-processing. Our group got the seventh place in the competition with an overall score of 0.5175. It is important to notice that the final score was affected by our low recall (0.4036) which arose as a result of not detecting intrinsic plagiarism cases, which were also present in the competition corpus.

1 Introduction

This paper describes our participation on the plagiarism detection task during the PAN competition at CLEF 2010. In order to detect the plagiarism cases present in the competition corpus we used the method described in [7], which focuses on detecting plagiarism based on a reference collection. In particular, the method is specially designed to detected cross-language plagiarism, which is also present in the competition corpus. Thus, our task is to detect the plagiarized passages in the suspicious documents and their corresponding text passages in the source documents.

The method is composed by five phases: language normalization, retrieval of candidate documents, classifier training, plagiarism analysis, and post-processing. Since the method is also designed to detect cross-language plagiarism, an automatic translation tool is used to translate the documents into a common language. A classification algorithm is used to build a model that is able to differentiate a plagiarized text passage from a non-plagiarized one. Note that the use of classification algorithms is common in the area of intrinsic plagiarism analysis [2, 4], but not in the area of external plagiarism analysis.

Based on the text passages extracted from the suspicious documents, an information retrieval (IR) system is used to retrieve the passages that are more likely to be the source of plagiarism cases. This is an important phase since the time necessary to perform a complete analysis of each suspicious document against all the documents in the reference collection would not be feasible. Only after the candidate passages of the source documents are retrieved, the plagiarism analysis is performed. Finally, a post-processing technique is applied in the results in order to join the contiguous plagiarized passages.

The remainder of this paper is organized as follows: Section 2 presents the employed method. Section 3 describes how training was done and shows the results achieved in the competition. Finally, Section 4 presents our conclusions.

2 The Method

We present here a brief description of how the method we used in the experiments works. A detailed description can be seen in [7]. The applied method is divided into five main phases, which are all briefly described below:

- *Language Normalization*: at this phase, the documents in the collection are translated into a default language so they can be analyzed in a uniform way. The English language was chosen as the default language. A language guesser is used to identify the documents that must be translated and an automatic translation tool is used to translate the documents.
- *Retrieval of Candidate Documents*: at this phase, an information retrieval system is used to retrieve, based on each suspicious document, the documents in the source collection that are candidates of being used as source of plagiarism. Before indexing the source documents, they are divided into several subdocuments, each one containing a single paragraph of the original document. Thus, when submitting a query to the system it will only return the relevant subdocuments, not the entire source document. For each passage in the suspicious document, the index is queried and the most relevant subdocuments are returned. These candidate subdocuments are the ones selected to be analyzed in the next phases of the method. It is important to notice that both the terms *passage* and *subdocument* represent a paragraph of the suspicious or source document.
- *Feature Selection and Classifier Training*: at this phase, a classification model is built to enable the method differentiates between a plagiarized and a non-plagiarized text passage. Thus, for each pair [*suspicious passage, candidate subdocument*] the following features are considered during the classifier training: *(i)* the cosine similarity between the suspicious passage and the candidate subdocument; *(ii)* the score assigned by the IR system to the candidate subdocument; *(iii)* the position of the candidate subdocument in the rank returned by the IR system; *(iv)* the length (in characters) of both the suspicious passage and the candidate subdocument. Note that a training collection (with the plagiarism cases annotated) must be supplied in order to create the training instances to train the classifier.
- *Plagiarism Analysis*: at this phase, for each pair [*suspicious passage, candidate subdocument*] we extract the necessary information to create the test instance and pass it to the classifier. Thus, the classifier is able to decide whether the suspicious passage is plagiarized from the candidate subdocument.
- *Post-Processing*: at this phase, the detection results of each suspicious document are post-processed to join the contiguous plagiarized passages. The goal is to report a plagiarism case as a whole instead of several small plagiarism cases. The following heuristic is applied: *(i)* separate the detections in groups, each group containing the detections referring to a single source document; *(ii)* for each group, sort the detections in order of appearance in the suspicious document; *(iii)* join adjacent detections that are close to each other (less than a pre-defined num-

ber of characters); *(iv)* for each plagiarized passage, keep only the detection with the largest length in the source document, i.e., do not report more than one possible source of plagiarism for the same passage in the suspicious document.

3 Experiments

3.1 Setting up the detector

In order to tune our plagiarism detector to the PAN'10 competition, we used the PAN-PC-09 [1] training corpus, which is a large-scale corpus containing artificial plagiarism offenses. It is important to mention that all the steps presented here are the same ones described in [7], the only difference is that we analyzed a different corpus.

As in [7] , we used the Terrier Information Retrieval Platform [6] as our IR system. We also employed the same IR techniques: the TF-IDF weighting scheme, stop-word removal (a list of 733 words included in the Terrier Platform), and stemming (Porter Stemmer [8]). To train our classifier, we used the Weka Data Mining Software [9]. In particular, we applied the J48 classification algorithm to build the classifier.

We divided the source documents into several subdocuments before translation in order to keep the original offset and length of each passage in the original document. As mentioned before, during the language normalization phase, we translate all the non-English documents in the corpus to English. We used LEC Power Translator 12 [5] as our translation tool and the Google Translator [3] as our language guesser.

After all documents in the reference collection are divided into subdocuments and translated into English, the collection is indexed. To reduce index size and speed up retrieval, only the subdocuments longer then 250 characters were indexed.

Before analyzing each document, we first have to train the classifier. To accomplish this, we randomly selected 50 suspicious documents. For each suspicious passage the top ten candidate subdocuments were retrieved. Based on each pair [*suspicious passage*, *candidate subdocument*], we can extract the information necessary to create the 500 training instances. The annotations provided with the corpus allowed us to check if the suspicious passage was actually plagiarized from the candidate subdocument. After the training instances were created, we generated the ARRF (Attribute-Relation File Format) file containing the training instances according to the Weka file format. Once we have the ARRF file with examples of plagiarized and non-plagiarized passages, we applied the J48 classification algorithm to build the classification model. After the classifier is trained, we can proceed to the analysis of the suspicious documents of the training corpus.

To analyze the suspicious documents, we divided them into passages. For each passage, we queried the index to get the top ten candidate subdocuments. Thus, for each pair [*suspicious passage*, *candidate subdocument*] we extracted the information needed by the classifier, and let it decide whether the suspicious passage was, in fact, plagiarized from the candidate subdocument. After we analyzed all the suspicious documents, we post-processed the results to join the contiguous plagiarized passages according to the heuristic described previously.

The parameters shown on Table 1 were defined based on tests with the training corpus. These same parameters were used for analyzing the competition corpus.

Table 1. Method Parameters.

Retrieval Parameters	
Subdocument length (in characters)	250
Subdocuments retrieved per suspicious passage	10
IDF threshold	8
IR score threshold	11
Post-Processing Parameters	
Merge threshold (in characters)	3000

As shown in Table 1, to reduce the index size and speed up retrieval, we only indexed the subdocuments with length greater than 250 characters. The IR system returned at most 10 candidate subdocuments for each suspicious passage. Also, to speed retrieval, instead of using all the terms of the suspicious passage to query the index, we discarded the terms which had an IDF (inverse document frequency) value lower than 8. We also discarded the subdocuments that received (by the IR system) a score lower than 11. Finally, in the post-processing phase, we joined the contiguous plagiarized passages that were at most 3000 characters distant from each other.

3.2 Evaluation

In order to analyze the competition corpus, we proceeded the same way described in the previous section. Note that we used the same classifier built during the analysis of the training corpus. Table 2 shows our overall result in the competition as well as the result of the analysis when considering only the external plagiarism cases. Note that since the competition corpus had both external and intrinsic plagiarism cases mixed up, the recall value ended up getting affected since the applied method was designed to detect only external plagiarism cases.

With the final score of 0.5175 our group got the seventh place in the competition. Table 3 shows an in-depth analysis of the results. We provide an overall analysis considering the results of the competition and we also analyze our results in detecting only the external plagiarism cases (which is the focus of the applied method). To analyze in which situations the method performs better, we investigated how well it handles text obfuscation and in what level the length of the plagiarized passage affects its overall performance. We divided the plagiarized passages according to their textual lengths: *short* (less than 1500 characters), *medium* (from 1501 to 5000 characters), and *large* (greater than 5000 characters).

According to Table 3, during the competition the method detected 29,486 out of 68,558 plagiarized passages (i.e., 43%). When ignoring the intrinsic plagiarism cases, the method detected 29,486 out of 55,723 plagiarized passages (i.e., 53%). It is possible to see that the method performed poorly while detecting short plagiarized passages. This is partially explained by our decision of indexing only the subdocuments with length greater than 250 characters (to speed up retrieval). Table 3 also shows that, other than translation, the intrinsic plagiarism cases did not suffered any kind of obfuscation. While detecting medium plagiarized passages, the performance of the method decreased as the level of obfuscation increased (none to high). It is worth noticing that the translated and the simulated plagiarized passages did not seem to have a negative impact in the performance of the method, since the percentage of the passages detected is not lower than for the other types of obfuscation. Finally, when detecting large plagiarized passages the method detected almost all of them, regardless of the type of obfuscation (note that that were no large simulated plagiarized passage).

Table 2. Overall results.

---	Competition	Only External Cases
Recall	0.4036	0.4966
Precision	0.7242	0.7242
F-Measure	0.5183	0.5892
Granularity	1.0024	1.0017
Final Score	0.5175	0.5881

Table 3. In-depth analysis of the results.

Short Plagiarized Passages						
-	***Competition***			***Only External***		
Obfuscation	**Detected**	**Total**	**%**	**Detected**	**Total**	**%**
None	78	9395	0.83	78	4088	1.90
Low	63	3798	1.65	63	3798	1.65
High	37	3729	0.99	37	3729	0.99
Translated	194	2417	8.02	194	1754	11.06
Simulated	211	2362	8.93	211	2362	8.93
Medium Plagiarized Passages						
-	***Competition***			***Only External***		
Obfuscation	**Detected**	**Total**	**%**	**Detected**	**Total**	**%**
None	2509	9907	25.32	2509	5911	42.44
Low	1832	4722	38.79	1832	4722	38.79
High	1415	4752	29.77	1415	4752	29.77
Translated	980	2358	41.56	980	1851	52.94
Simulated	268	624	42.94	268	624	42.94
Large Plagiarized Passages						
-	***Competition***			***Only External***		
Obfuscation	**Detected**	**Total**	**%**	**Detected**	**Total**	**%**
None	6755	8733	77.35	6755	6785	99.55
Low	6343	6363	99.68	6343	6363	99.68
High	6171	6275	98.34	6171	6275	98.34
Translated	2630	3123	84.21	2630	2709	97.08
Simulated	0	0	100.00	0	0	100.00

4 Conclusions

This paper described our approach to the plagiarism detection task during the PAN competition at CLEF 2010. We applied the method presented in [7], which focuses on detecting external plagiarism. In particular, the method is specially designed to detect cross-language plagiarism, which is also present in the competition corpus.

We used the training corpus PAN-PC-09 to set up the detector. The training corpus was also used to build the classification model used during the analysis of the competition corpus. With an overall score of 0.5175 we ended up in the seventh place in the competition. Our overall score was affected by our low recall (0.4036) since the applied method was designed to detect only the external plagiarism cases, leading the detector to ignore the intrinsic plagiarism cases present in the competition corpus.

An in-depth analysis was conducted to check in what situations the method performs better. Regarding the textual length of the plagiarized passage, the larger is the passage the easier is the detection. In fact, when analyzing large plagiarized passages the method detected almost all of them, regardless of the type of obfuscation. However, the method performed poorly while detecting short passages. We attribute this low performance to the fact that we only indexed subdocuments with length greater than 250 characters. Finally, the translated and the simulated plagiarized passages did not seem to have a negative impact in the performance of the method, since the percentage of the passages detected are not lower than the other types of obfuscation.

Acknowledgements

This project was partially supported by CNPq-Brazil and INCT-Web.

References

1. Webis at Bauhaus-Universität Weimar & NLEL at Universidad Politécnica de Valencia PAN Plagiarism Corpus 2009 (PAN-PC-09). http://www.webis.de/research/corpora M. Potthast, A. Eiselt, B. Stein, A. Barrón-Cedeño, and P. Rosso (editors).
2. Argamon, S. and S. Levitan, *Measuring the Usefulness of Function Words for Authorship Attribution*, in *Association for Literary and Linguistic Computing/ Association Computer Humanities*. 2005: University Of Victoria, Canada.
3. Google Translator http://www.google.com/translate_t.
4. Koppel, M. and J. Schler. *Authorship Verification as a One-Class Classification Problem*. in *Proceedings of the 21st International Conference on Machine Learning*. 2004. Banff, Canada: ACM.
5. LEC Power Translator http://www.lec.com/power-translator-software.asp.
6. Ounis, I., G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson, *Terrier Information Retrieval Platform*, in *Proceedings of the 27th European Conference on Information Retrieval (ECIR 05)*. 2005, Springer. p. 517-519.
7. Pereira, R.C., V.P. Moreira, and R. Galante, *A New Approach for Cross-Language Plagiarism Analysis*, in *Proceedings of the CLEF 2010 Conference on Multilingual and Multimodal Information Access Evaluation*, M. Agosti, et al., Editors. 2010, Springer: Padua, Italy.
8. Porter, M.F., *An algorithm for suffix stripping*, in *Readings in information retrieval*. 1997, Morgan Kaufmann. p. 313-316.
9. Weka http://www.cs.waikato.ac.nz/ml/weka/.