Improving the Reliability of the Plagiarism Detection System

Lab Report for PAN at CLEF 2010

Jan Kasprzak and Michal Brandejs

Faculty of Informatics, Masaryk University
{kas,brandejs}@fi.muni.cz

Abstract In this paper we describe our approach at the PAN 2010 plagiarism detection competition. We refer to the system we have used in PAN'09. We then present the improvements we have tried since the PAN'09 competition, and their impact on the results on the development corpus. We describe our experiments with intrinsic plagiarism detection and evaluate them. We then discuss the computational cost of each step of our implementation, including the performance data from two different computers.

1 Introduction

Discovering plagiarism in text files by means of software is a widely-studied area. Yet it is often not clear how the proposed approaches would perform in real-world scenarios. The PAN competition[1] attempts to provide a testbed for comparing different approaches on the same data.

We discuss the performance of our plagiarism detection system on two document corpora, used in PAN 2010 competition: The development corpus PAN-PC-09 [11] is based on the PAN'09 competition data. It contains 14429 source documents, 14428 suspicious documents for the external plagiarism detection, and 6183 suspicious documents for the intrinsic plagiarism detection. The competition corpus contains both the intrinsic and external plagiarism cases in one body of suspicious documents. There are 11147 source documents in the corpus, and 15925 suspicious documents.

The next section of this paper contains a brief recapitulation of our approach used in PAN'09. We then evaluate incremental modifications to this approach and their impact on the overall score on the development corpus. The two main areas of the research are covered in separate sections: we discuss the cross-language plagiarism detection in Section 4, and the intrinsic plagiarism detection in Section 5. In Section 6 we then evaluate the computational cost of our implementation both on the same hardware which was used for PAN'09, and on the newer hardware. This should give some insight on the scalability of our system.

[1] `http://pan.webis.de/`, see [6] for the overview of the competition.

2 System Overview

We have used our system for tackling the PAN'09 external plagiarism task as a starting point, trying to improve it further. The detailed description of the system can be found in [4]. In this paper, we provide only a short summary of the main features:

- *Tokenization*: text files are split into words (sequences of at least three consecutive alphanumeric characters). We keep the offset of the first and last character of each word for further use.
- *Chunks*: we form partly overlapping word 5-grams, sort the words in them, and compute a MD5 hash[7]. The most significant 30 bits of the hash is then used as a chunk identification. For each chunk in a given document, we also keep the sequence number of this chunk.
- *Inverted index*: mapping the chunk ID to the list of structures containing the following attributes: *(document ID, chunk sequence number, offset of the first character of the chunk, offset of the last character of the chunk)*.
- *Computing the similar documents*: using the inverted index, we examine the suspicious documents (their chunks are computed the same way as for the source documents). We then find pairs of source and suspicious documents, and their common chunk IDs (including additional data we store for each chunk). We discard document pairs with less than 20 chunks in common.
- *Evaluating the similar passages*: for each document pair with 20 or more common chunks we test whether the chunks form one or more *valid intervals* (intervals where there gap between two neighbouring common chunks is not bigger than 50 chunks, and which have at least 20 common chunks) both in the source and suspicious document.
- *Generating the detections*: those valid intervals are then reported as plagiarized passages, with the only postprocessing being removal of overlapping detections.

2.1 System performance

The performance of our system in the PAN'09 was the following:[2]

$$\textit{recall} = 0.6967,\ \textit{precision} = 0.5573,\ \textit{granularity} = 1.0228,\ \textit{overall} = 0.6093$$

Unfortunately, we did not save the final version of our software for PAN'09. We had to reimplement the last phase (removal of overlapping detections) from scratch. With this change, the performance on the final version of the PAN-PC-09 data was the following:

$$\textit{recall} = 0.6442,\ \textit{precision} = 0.5725,\ \textit{granularity} = 1.0193,\ \textit{overall} = 0.5980$$

The difference here is probably caused by a different implementation, as well as possible changes between the competition corpus of PAN'09 and the final version of PAN-PC-09.

In PAN 2010, the competition corpus contained both intrinsic and external plagiarism cases. For evaluating the impact of false-positives in suspicious documents which

[2] We refer to [5] for explanation of terms *recall*, *precision*, and *granularity*. Also the formula which is used to compute the overall score from these three values is described there.

contain intrinsic plagiarism cases only, we have joined the suspicious documents from both parts of the PAN-PC-09 into one directory (renaming the intrinsic cases and editing their annotations to match the new file names), and tested the performance on this united version of the data:

$\mathit{recall} = 0.5255,\ \mathit{precision} = 0.4858,\ \mathit{granularity} = 1.0480,\ \mathit{overall} = 0.4882$

In sections 3 and 4 we present the results computed on this united version of PAN-PC-09 only.

3 External Plagiarism Detection Improvements

In order to improve the performance for the external plagiarism detection many ad-hoc modifications have been tested. The results from PAN'09 suggested that our method has a good recall[3], so the focus has been mostly on improving the precision and granularity.

As for the recall, there is probably not a big room for improvement: the chunking method cannot possibly detect heavily obfuscated passages.

3.1 Adjacent Detections

One modification to the post-processing stage was removing *both* overlapping detections (instead of e.g. keeping a longer one only), provided they were short enough—the threshold we have used was 600 characters. Interestingly enough, this this has improved the precision score, but did not hurt recall much:

$\mathit{recall} = 0.5252,\ \mathit{precision} = 0.4941,\ \mathit{granularity} = 1.0465,\ \mathit{overall} = 0.4929$

A possible explanation is that those short overlapping detections are passages of relatively common phrases.

As for the granularity measure, we have attempted to merge the adjacent detections pointing to the same source document. The criteria we have used were the following:

- When the gap is under 600 characters, merge.
- When the gap is under 4000 characters and is smaller than half of the average length of both adjacent detections, merge.
- Otherwise, keep the both detections separated.

The resulting score confirmed the improvement in both precision and granularity:

$\mathit{recall} = 0.5256,\ \mathit{precision} = 0.5302,\ \mathit{granularity} = 1.0233,\ \mathit{overall} = 0.5192$

3.2 False positives

One of the drawback of the system in PAN'09 has been that some of the document similarities detected were not annotated in the gold standard as plagiarism passages. The examination of those false positives suggested that they are often in structured passages (like table of contents, etc.). The new system attempts to exclude such passages from our detection: it computes the percentage of letter characters in the detection, and the

% of letters	recall	precision	overall
baseline	0.5256	0.5302	0.5192
> 0.550	0.5256	0.5323	0.5202
> 0.575	0.5256	0.5327	0.5204
> 0.600	0.5255	0.5335	0.5208
> 0.625	0.5253	0.5348	0.5213
> 0.650	0.5251	0.5373	0.5224
> 0.675	0.5244	0.5420	0.5243
> 0.700	0.5230	0.5501	0.5275
> 0.725	0.5193	0.5639	0.5321
> 0.750	0.5160	0.5810	0.5386
> 0.775	0.3657	0.5946	0.4475
> 0.800	0.0736	0.4846	0.1264
> 0.825	0.0023	0.1747	0.0045

Table 1. Excluding passages with low percentage of letter characters

detections with low percentage of letter characters were excluded. The measurements are in Table 1.

Because of a bug in our software, we had the above values miscomputed during the competition, and the best overall score was with threshold 0.675, which is what we have used for final results computation:

$$\textit{recall} = 0.5244,\ \textit{precision} = 0.5420,\ \textit{granularity} = 1.0233,\ \textit{overall} = 0.5243$$

This modification has added the access to the original suspicious documents again to the post-processing stage, which brings a small performance penalty. In principle, the similar effect could be obtained by excluding non-letters (e.g. digits) in the tokenization phase, and then look at the interval length with respect to number of chunks in the said interval.

4 Cross-language Plagiarism Detection

Both the development and competition corpora contained also translated documents. All the suspicious documents were in English, and the source documents were in English, German, and Spanish.

One of the cheapest ways for improving the performance would be to exclude the non-English documents from processing altogether in order to avoid false positives. Another approach would be to detect cross-language plagiarism using e.g. keyword extraction (for example, by finding words which are much more frequent in the text itself than in the general language).

With the relatively small document corpus size it was feasible to translate the non-English source documents to English, and to use it as alternative source documents.

[3] In fact, we had the highest recall value from all teams in PAN'09.

4.1 Language Detection

The automatic language detection is a widely studied area. One possible approach is to compare character n-gram based language models [2]. We have used the `Text::Language::Guess` Perl module [8] from CPAN, primarily because it was ready-to-use and worked well enough. Its classification method is based on detecting stop-words in the document. While not being extremely accurate, it has provided a preselection of documents, which have been then manually checked whether they are indeed in Spanish or German. The most misdetections were on structured data[4].

4.2 Translating the Documents

Several web sites provide automatic translation services. We have tried to use Yahoo! Babelfish system [1] and Google Translate [3]. With Babelfish we had a problem with reliability: for some documents we did not get any reply at all. Google Translate was better, and it even kept the line breaks in the translated text. This allowed us to match the translated plagiarism back to the original position with a good accuracy.

The only problem with Google Translate is that it silently truncates longer documents. We suspect they impose a limit to the CPU time their hardware spends on a single document. We have therefore split documents to blocks with the size betweeen 15 and 22 KB, splitting preferably at the paragraph boundary or a line boundary. We have got 2562 document parts for PAN-PC-09 for Spanish, 4887 parts for the competition corpus for German, and 2562 parts for the competition corpus for Spanish. Because of time limitations we have omitted German in the development corpus.

4.3 Problematic Sentences

Even with small document parts, there were paragraphs or sentences on which Google Translate stopped processing altogether. One such example was the line 6256 of source document 6696 from the competition corpus:

> *unser Los. Und ich bin ja auch glücklich, wenn ich nur weiß, daß Moina sich vergnügt.< Sie*

Google Translate has always stopped translating after the word *Moina*, no mater how many lines of text we have put before or after the above line.

When splitting the data further was not possible, we have translated the rest of the document, and put blank lines instead of the problematic ones.

4.4 Character Offset Calculation

We have modified our post-processing stage to recompute the offsets in detections of the translated source documents using line-to-line mapping to the original source documents. Offsets within a given line were linearly interpolated using the offsets of the first and last characters of that line.

[4] as an example, the source document number 112 from PAN-PC-09 was detected as French.

For example, when the detection in the translated source document started at character 1000, which was 20th character out of 40 on line 125, we have mapped it to the offset of the character which was in the middle of line 125 in the original source document.

4.5 Performance on the Training Corpus

After translating all the appropriate source documents from Spanish to English and removing the source documents in German we have re-ran the computation. The final score on the training corpus was the following:[5]

$$\textit{recall} = 0.5386,\ \textit{precision} = 0.5476,\ \textit{granularity} = 1.0236,\ \textit{overall} = 0.5340$$

The performance on the competition corpus is expected to be a bit higher, because of translating also the source documents from German.

5 Intrinsic Plagiarism Detection

For the intrinsic detection part, the most important aspect was to not lower the overall score by using the generally low-performing intrinsic plagiarism detection. As a proof-of-concept, we have reimplemented the approach of the PAN'09 winner of the intrinsic task, described in [9]:

5.1 System Outline

The intrinsic plagiarism detection system splits the document into a set of partly overlapping *windows*, and computes the *style change* function as a distance between the character trigram profile of the whole ocument and of the given window. Those areas of the suspicious document which have higher style change function values are then supposed to be plagiarized[6].

The score of the PAN'09 winner was the following:

$$\textit{recall} = 0.4607,\ \textit{precision} = 0.2321,\ \textit{granularity} = 1.3839,\ \textit{overall} = 0.2462$$

Our implementation has not been able to match this score. The best overall score our implementation has reached was around 0.172.

[5] After the competition, we have tried to modify the threshold value as described in Section 3.2. The best score we have got was with threshold of 0.725:

$$\textit{recall} = 0.5334,\ \textit{precision} = 0.5693,\ \textit{granularity} = 1.0230,\ \textit{overall} = 0.5418$$

[6] This is an oversimplification. Refer to the original paper [9] for details on how the plagiarized passages are determined.

5.2 Our Modifications

We have further improved the system by modifying how the plagiarized passages are detected from the style change function. Essentially we should only consider those areas, where the style change function rises from “generally low” values to “generally high” values and stays there for some interval.

In order to reach a better granularity and to detect steep changes in the style change function, we have computed the Gaussian-smoothed variants of this function:

$$\mathrm{sc}_\sigma(x) = \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\mathrm{sc}(x_i)^2}{2\sigma^2}}$$

For each document, we have computed two functions: for $\sigma = 1$, and for $\sigma = 10$. For each window, we have then computed the nearest local minimum and local maximum of the $\mathrm{sc}_1(x)$ function.

As a beginning of the plagiarised passage we have considered the point where $\mathrm{sc}_1(x)$ becomes higher than $\mathrm{sc}_{10}(x)$, and the corresponding local minimum is low enough (lower than median of $\mathrm{sc}(x)$ + stddev of $\mathrm{sc}(x)$), the corresponding local maximum is high enough (greater than median of $\mathrm{sc}(x) + 2.2 \cdot$ stddev of $\mathrm{sc}(x)$), and the $\mathrm{sc}(x) >$ median of $\mathrm{sc}(x) + 1.7 \cdot$ stddev of $\mathrm{sc}(x)$.

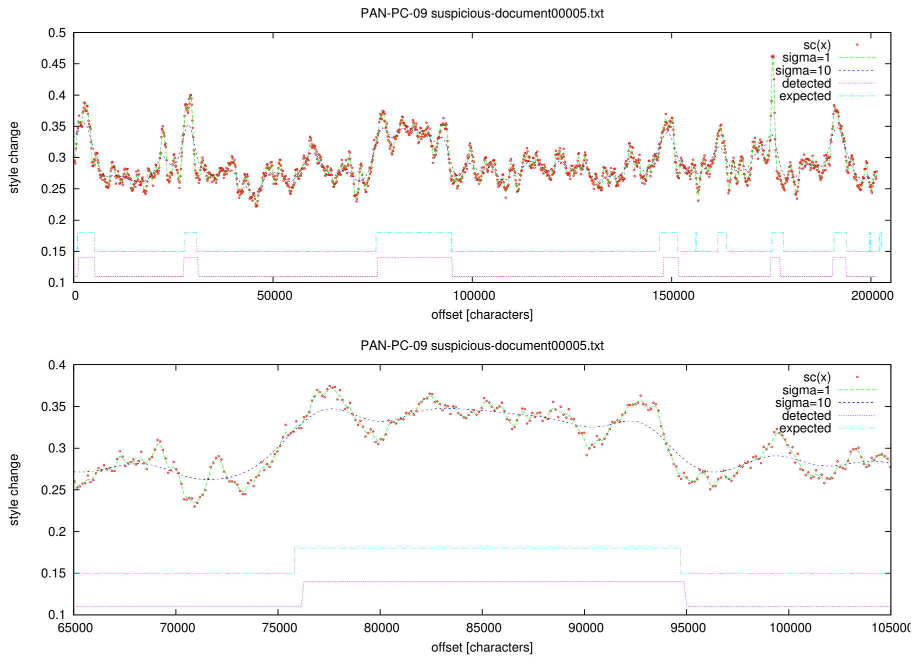

Figure 1. Smoothed style change functions

The smoothed versions of the style change function of the document 00005 from PAN-PC-09 can be seen in Figures 5.2 and 5.2, together with the detailed view of a single plagiarism case.

After the competition we did further experiments: we have found that the style change function is not as stable with varying document structure as the original paper states.

We have obtained a significant improvement when considering windows not with the fixed length in terms of characters, but in terms of trigrams instead (skipping trigrams with all non-letter characters). We suspect that using a different dissimilarity function to compute the style change function would be even better. One possibility is to compute the "out-of-place" n-gram profile distance as described in [2]. Another approach to explore would be to use the cosine similarity.

With the above modifications, we have been able to obtain a better overall score than the PAN'09 winner:

$\mathit{recall} = 0.2627,\ \mathit{precision} = 0.2969,\ \mathit{granularity} = 1.072,\ \mathit{overall} = 0.2562$

However, we were not able to reach any significant improvement by using the intrinsic detector as a hint to the external plagiarism detection, partly because of time constraints. Thus in our final submission, the intrinsic detection was not used at all.

6 Performance of Our Approach

The system for external and cross-language plagiarism (without the intrinsic plagiarism detection) has been used to process the competition corpus. The resulting score earned us the first place in the competition:

$\mathit{recall} = 0.6915,\ \mathit{precision} = 0.9405,\ \mathit{granularity} = 1.0004,\ \mathit{overall} = 0.7968$

We had the highest recall measure from all teams (the second best recall was 0.6907, by the Know-Center Graz team, which was classified at the third place). Also our granularity measure was the best (the second best was 1.0018 by the Universidad de Huelva team, which got the 6th place). Our precision value was the second best of all teams (the best precision of 0.9559 had the Anna University Chennai team on the 9th place).

The results are somewhat surprising, as they differ significantly from the performance on the development corpus. We have expected to have higher recall score on the competition corpus because of additional translations from German, but the difference is still too big.

Also the precision score is higher than expected. Nevertheless, it is higher for many teams in the competition than it was in PAN'09. This could mean that they have improved their systems, or that the competition corpus is significantly different from PAN-PC-09.

6.1 Computational Cost

The performance of the system has been tested on two different computers: the development has been done on a main student's server of Faculty of Informatics, Masaryk University: it is HP DL-585 G6 with four six-core AMD Opteron 8439 SE processors at 2.8 GHz (24 cores total), 128 GB RAM. For purpose of comparison, we have also evaluated the system on the same hardware as in PAN'09 (SGI Altix XE), which has two quad-core Intel Xeon E5472 CPUs at 3.0 GHz (8 cores total), 64 GB RAM.

Our implementation is written in Perl (tokenization, generating chunks and their hashes, postprocessing), and in plain C (generating the inverted index, searching the inverted index).

In Table 2, we present the times spent by processing the development and competition corpora. The time is divided into three parts: the first one is tokenizing the source documents and generating the inverted index, the second part is tokenizing the suspicious documents and looking up common chunks, and the last part is postprocessing (removing overlaps, joining adjacent detections, generating the XML files).

	Development corpus		Competition corpus	
Task	8-core SGI	24-core HP	8-core SGI	24-core HP
Inverted index	1:06:02	0:12:41	0:49:44	0:10:28
Chunk pairs	2:07:25	0:20:44	1:39:20	0:15:55
Postprocessing	0:09:22	0:03:17	0:04:40	0:01:16
Total	3:22:55	0:36:42	2:33:44	0:27:39

Table 2. Computational cost of the system (times as hours:minutes:seconds)

7 Future work

There are several areas where possible improvements of the approach presented in this paper can be present:

- *N-gram profile distance function in the intrinsic plagiarism detector.* Different means of computing the distance of the window profile to the profile of the whole document should be further evaluated with the expectations of reaching better stability on variable-sized documents and windows.
- *Using the intrinsic detector* as a hint for the external detector, possibly allowing to detect also intrinsic plagiarism cases if the lower precision will not impact negatively the overall performance.
- *Density of common chunks.* Instead of computing the valid intervals and then joining the detections in the post-processing phase, it could be feasible to allow wider gaps in the detection for either large detections (which we partly use), or for detections where the density of common chunks is higher.
- *Differences between the corpora.* Much better score in the competition when compared to the training corpus is definitely a phenomenon which should be further explored. Because of late release of annotations for the competition corpus we are unable to do it in this paper.

8 Conclusions

We have presented several improvements to the system for detecting plagiarism, which is described in [4]. The result was the best-performing system in the PAN 2010 plagiarism detection competition. The core of this system has also been in production use in the Czech National Archive of Graduate Theses [10] and several other production systems since 2008, handling milions of text documents.

Some modifications we have described would be usable also in large-scale production systems, while others are made purely for the sake of the PAN competition. One such modification with limited impact outside the competition is our approach to detecting cross-language plagiarism. Using public translation services like Google Translate or Yahoo! Babelfish is not feasible for large sets of documents. On the other hand, we have verified that even machine translation can detect at least some cross-language plagiarism cases using the general-purpose external plagiarism detector.

Acknowledgements

We would like to thank Pavel Rychlý for discussing the properties of n-gram profiling and intrinsic plagiarism detections with us. Also we thank the organizing comitee of the PAN 2010 competition for an interesting event.

References

1. Yahoo! Babelfish. http://babelfish.yahoo.com/ (2010)
2. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. pp. 161–175 (1994)
3. Google Translate. http://translate.google.com/ (2010)
4. Kasprzak, J., Brandejs, M., Křipač, M.: Finding plagiarism by evaluating document similarities. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 24–28 (2009)
5. 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 (COLING 2010) (to appear). Association for Computational Linguistics, Beijing, China (Aug 2010)
6. Potthast, M., Stein, B., Eiselt, A., Cedeño, A.B., Rosso, P.: Overview of the 1st international competition on plagiarism detection. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 1–9 (2009)
7. Rivest, R.: RFC1321: The MD5 Message-Digest Algorithm (1992), http://www.rfc-editor.org/rfc/rfc1321.txt
8. Schilli, M.: `Text::Language::Guess` Module. http://search.cpan.org/~mschilli/Text-Language-Guess-0.02/lib/Text/Language/Guess.pm (2005)
9. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles. In: Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 38–46 (2009)
10. Czech National Archive of Graduate Theses. http://theses.cz/ (2008–2009)
11. Webis at Bauhaus-Universität Weimar, NLEL at Universidad Polytécnica de Valencia: PAN Plagiarism Corpus 2009 (PAN-PC-09). http://www.webis.de/research/corpora (2009), Martin Potthast, Andreas Eiselt, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso (editors)