Exploring Fingerprinting as External Plagiarism Detection Method

Lab Report for PAN at CLEF 2010

Yurii Palkovskii, Alexei Belov, Irina Muzika

Zhytomyr State University, SkyLine, Inc.
palkovskiy@yandex.ru

Abstract. This paper outlines the main approach and the general design of the plagiarism detection prototype application we have developed to take part in the 2nd International Plagiarism Detection Competition. The developed system is a part of the larger application used at Zhytomyr State University as CMS Thesis Storage and comes under the title "Plagiarism Detector Accumulator". This application prototype uses fingerprinting and hash search methods to locate similarities between different documents and thus detect plagiarism.

1 Introduction

Facing the new challenge of the 2nd International Competition on Plagiarism Detection we continued our previous research in external plagiarism detection using fingerprinting methods. The main task of our research was to get most effective results of the existing framework we have developed last year. Irrespective of the fact that vector based plagiarism detection performs much better than fingerprinting in relation to quality of plagiarism detection, we decided to go on with the fingerprinting as we find scalable enough to be used in products that require the processing of excessively large amounts of documents and we strongly believe that computational powers required to process the same amount of data using the VSM approach are too large to be used in solving practical tasks whereas fingerprinting can be easily scaled up to linear data processing. One of the main tasks we tried to achieve this year was to build a fingerprinting based plagiarism detection prototype that can compete with the VSM based systems.

2 External Plagiarism Detection

The exact algorithm of plagiarism detection is the following:

1. Indexing of the suspicious documents.
2. Searching the suspected document against the created index.
3. Accumulating search results, defining Plagiarism Sections.
4. Refining the newly found Plagiarism Sections, finding the offsets, writing the result xml file.

All the above sequence is applied using windowing technique to be able to use RAM for storing the index for best performance. This approach has several benefits and drawbacks - extremely high speed as for both indexing and searching while requiring to save the intermediate results for each document that was marked as the one that contains plagiarism.

The index structure used:

1. Record ID.
2. Fingerprint Hash.
3. Fingerprint offset.
4. Source Document ID.

No preprocessing was done to the source documents.

The indexing metaparameters are the following:

1. "The number of Words" in a Fingerprint.
2. "Fingerprint Overlapping Step" - the distance from the 1st word of the 1st Fingerprint to the 1st word of the next Fingerprint.

The searching metaparameters are the following:

1. "The number of Words" in a Fingerprint. Must be equal to "The number of Words" in a Fingerprint used during the Indexing step.
2. "Fingerprint Overlapping Step" - variation of this parameter results in a tradeoff between speed and the exactness of the results.

We used a big number of automated tests to define the best suitable values for the above parameters. We compiled a number of corpora made of the test corpus with different criteria used to check the metaparameters influence Precision, Recall and Granularity.

We created the following corpora:

1. 500 documents with 0 obfuscation , no translated parts.
2. 100 documents with maximum obfuscation, no translated parts.
3. 200 documents with low obfuscation, no translated parts.

Then, we moved to the mixed types of corpora and finally got the best metaparameters values.

Our main prototype Indexing system uses hashes formed out of the sequences of 5 words with the step of 3 words starting with the initial word in the hash. The search is done with all possible hashes: 5 words in hash, 1 word step. MySQL Database heap storage engine was used as a hash index storage with the windowing applied to mitigate the memory requirements. A single window could have contained up to 50.000.000 records before the system memory was depleted. We ran a number of practical tests to define the memory limits to be applied as base values for our research. We have applied special configurations both to the MySQL engine and to the operating system to best improve the performance and reach system stability while in processing stage. As for MySQL we have significantly increased the amounts for key sorting buffers and fine-tuned the configuration files, as for the operating system - we disabled the swap file, "prefetch" and "superfecth" features as we have discovered their great negative influence onto the total performance. Intel Q9550 with a memory of 6GB was used as a computational platform for the prototype application written in VB.NET running on MS Windows7 64bit. Total processing time was about 6 hrs under the IDE. The switch to the 64bit operating system was dictated by Windows OS memory limitations. The whole development process was done on Windows XP with 2 GB of RAM with the MySQL MyISAM on-disk engine. The total performance of the disk based system resulted in the excessive time required to process this large amount of hashes, so we decided to switch to the memory based MySQL Heap model. This boosted the total search speed dramatically with the indexing speed being a constant value. We used a straightforward approach to the definition of the suspected documents - in case 3 or more hashes occurred within a specified distance, so-called "Plagiarism Section Minimal Detection Distance", we considered this document a suspected one and recorded the case of plagiarism. The database main index contained two base values - the Hash value and the Fingerprint Offset of the proper fingerprint - making it easy to define the values required for saving the result set. During the search stage all the results have been accumulated into a single file before moving the next window thus allowing to use the windowing index and search.

3. Evaluation

Due to the ability to measure the exact effectiveness of the new improvements applied to our prototype this year we successfully automated the process of any critical parameter variations testing - so that we were able to monitor their exact trends and most optimal values. Thus we did a large number of tests trying to figure out the general trends of our most critical parameters. We used a special test corpora that represent selections made of the available training corpus. We selected these sets to evaluate the plagiarism detection effectiveness on different obfuscation levels, document sizes, translation types etc. Unfortunately we did not have an opportunity to run exhaustive tests on the entire training corpus for the lack of time. Our overall score is 0.50 against 0.30 the last year, this shows that the applied modifications and the automated result evaluation boosted the total effectiveness nearly twice.

4. Conclusions

We hope that even a further development of the fingerprinting plagiarism detection combined with the VSM analysis will eventually produce the most effective plagiarism detection system in relation to both quality and speed. We would like to thank the organizers of the Competition for their devoted work, assistance and prompt replies in the Google Group as the time span for the competition was rather close.

5. References

1. Barron-Cedeno, Alberto and Paolo Rosso. 2009. On Automatic Plagiarism Detection based on n-grams Comparison. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy, editors, ECIR 2009, volume 5478 of LNCS, pages 696–700, Toulouse, France. Springer.
2. Grozea, C. 2004. 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, August.
3. International Competition on Plagiarism Detection. 2009. http://www.webis.de/pan-09/competition.php.

4. Potthast, Martin et.al. (editors). 2009. PAN Plagiarism Corpus PAN-PC-09. http://www.webis.de/research/corpora.
5. V. Keselj, F. Peng, N.Cercone and C. Thomas. 2003. n-gram-based author profiles for authorship attribution.