promoting access to White Rose research papers

White Rose Research Online

Universities of Leeds, Sheffield and York
http://eprints.whiterose.ac.uk/

This is an author produced version of a paper published in **CLEF 2010 LABs and Workshops, Notebook Papers.**

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/78590

Published paper

Nawab, R., Stevenson, M. and Clough, P. (2010) *University of Sheffield: Lab Report for PAN at CLEF 2010.* In: CLEF 2010 LABs and Workshops, Notebook Papers. 4th International Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, 22nd - 23rd September 2010, Padua, Italy.

White Rose Research Online
eprints@whiterose.ac.uk

University of Sheffield
Lab Report for PAN at CLEF 2010

Rao Muhammad Adeel Nawab, Mark Stevenson and Paul Clough

University of Sheffield, UK.
{r.nawab|m.stevenson}@dcs.shef.ac.uk, p.d.clough@sheffield.ac.uk

Abstract This paper describes the University of Sheffield entry for the 2nd international plagiarism detection competition (PAN 2010). Our system attempts to identify extrinsic plagiarism. A three-stage approach is used: pre-processing, candidate document selection (using word n-grams) and detailed analysis (using the Running Karp-Rabin Greedy String Tiling string matching algorithm). This approach achieved an overall performance of 0.20 in the official evaluation with a precision of 0.40, recall of 0.16 and granularity of 1.21.

Keywords: extrinsic plagiarism dteection, greedy string tiling, GST, n-grams

1 Introduction

This paper describes the University of Sheffield entry for the 2nd international plagiarism detection competition (PAN 2010). The system we submitted only attempted to identify *extrinsic plagiarism*: in this scenario the assumption is that the source (i.e. original) document(s) for a plagiarized document is hidden within a large collection of documents that can be accessed, i.e. within a closed–set of documents. The task is to identify the portions of the documents that have been plagiarized by aligning them with the corresponding text within the relevant source documents.

1.1 Related Work

A simple approach to extrinsic plagiarism detection is to compare each suspicious document with a collection of source documents to find the potential source(s) of a plagiarized text. However, the problem is that if the reference collection is very large, for instance the World Wide Web, then the computational resources required can be prohibitive for many approaches. The vector space model is considered as a baseline for the external plagiarism detection task. This model has been used to detect duplicate and near duplicate documents [7]. A variant of this model was used for copy detection at word level (the SCAM system) [13] and sentence level (the COPS system) [2]. *Fingerprinting* is another popular approach. Fingerprints, a means of uniquely identifying texts, are used to represent the contents of the source documents. To query a collection of documents, fingerprints are also generated for the query document. The number of

common fingerprints between the source and query document determines the similarity between them. This approach has also been used by other researchers [12,8]. Another approach that has also proved effective is based upon n-gram overlap. The source and suspicious documents are converted into fixed length n-grams (either characters or words) and the proportion of common n-grams used to determine the similarity between documents. This approach has been applied using different similarity measures, such as the Jaccard, Dice and Overlap coefficients [9,10,1,6]. Similar approaches were used by several of the systems that participated in the 1st international PAN-PC-09 plagiarism detection competition [15].

2 External Plagiarism Detection

We divided the task of external plagiarism detection into three phases: preprocessing, candidate document selection and detailed analysis using Running Karp-Rabin Greedy String Tiling (RKR-GST).

2.1 Preprocessing

In the first phase we used TextCat[1] [3], an existing language recognizer, to identify the language of each source document. Documents that were identified as being in German or Spanish were translated into English using the Google Translate [2] tools (see [4] for an overview of the Google translation tools). Each document was preprocessed by converting the text to lowercase and removing any non-alphanumeric characters. Each document was then converted into a set of word n-grams ($n = 5$). Word n-grams of length 5 were chosen for the text representation since they were found to generate high recall in the 2009 PAN competition (see [15]).

2.2 Candidate Document Selection

Detailed comparison of each suspicious document with all the source documents could be computationally prohibitive. Therefore, for each suspicious document we attempt to identify likely source documents. The similarity between two documents can be computed by counting the number of n-grams they have in common. This technique has been used in copy detection and plagiarism detection [2,13,14,10]. We used one such technique, the overlap coefficient [11], to compare each suspicious document with the source documents. The overlap coefficient, $Sim_{overlap}(A, B)$, is computed using equation 1, where $S(A, n)$ and $S(B, n)$ are the set of distinct n-grams in documents A and B respectively. Values for the overlap coefficient range from 0 to 1 where 0 means that two documents are entirely different and 1 means that two documents are exactly the same.

$$Sim_{overlap}(A, B) = \frac{|S(A, n) \bigcap S(B, n)|}{min(|S(A, n)|, |S(B, n)|)} \quad (1)$$

[1] http://www.let.rug.nl/vannoord/TextCat/

[2] http://translate.google.com/

The potential source documents were sorted in descending order according to their overlap score and the top 10 selected. Any documents with overlap score below a certain threshold, $\alpha_{overlap}$, are discarded and the remaining source-suspicious pairs passed to the next stage of processing.

2.3 Detailed Analysis

In the final stage of processing detailed analysis is carried out using the Greedy String Tiling (GST) algorithm. This performs an alignment between the contents of the suspicious document and each source document to identify potential plagiarized sections from the set of source-suspicious pairs generated by the previous stage. The advantages of using GST compared with alternative string similarity algorithms, such as *longest common subsequence* or *edit distance*, is its ability to detect *block moves*: treating the *transposition* of a substring of contiguous words as a single move rather than considering each word separately. GST has previously been used in program code plagiarism detection [18], biological sequence alignment [17,16] and measuring text re–use in journalism [5]. The algorithm has a run time of $O(n^3)$, but has been optimised to run in linear time using a string matching algorithm Running Karp-Rabin Greedy String Tiling (RKR-GST) [16].

An example of the output of GST for a fragment of source text and a rewritten (or plagiarized) version is shown below. ($[1]$ and $[2]$ indicate aligned matches between the two texts.)

Source A dog$_{[1]}$ bit the postman$_{[2]}$.
Rewrite The postman$_{[2]}$ was bitten by a dog$_{[1]}$.

Match merging heuristics were applied to the output of GST to identify contiguous portions of plagiarized text from suspicious documents. GST computes the *length* of the longest tile[3] in common between the suspicious and source documents. If this is greater than a pre–defined length, α_{length}, the tiles are combined to identify plagiarized sections. If the distance between adjacent tiles is less than or equal to a given number of characters, α_{merge}, then they are combined into a single section.

3 Evaluation

3.1 System Development

Candidate Document Selection (Section 2.2). We initially experimented with using a copy detection system (an implementation of SCAM [13]) to extract candidate documents. However, due to the processing speed and memory problems this approach had to be re-designed. The $\alpha_{overlap}$ threshold was set using a small corpus of 895 source and 88 suspicious documents from the first PAN competition (PAN-PC-09 corpus). We calculated the precision, recall and F1 measures for overlap scores with thresholds from 0.001 to 0.100. The best performing threshold, 0.005, gave a precision, recall and F1 measure of 0.74, 0.93 and 0.78 respectively.

[3] A tile is a consecutive subsequence of maximal length that occurs as one-to-one pairing between two input strings.

Detailed Analysis (Section 2.3). To avoid accidental matches a parameter, α_{mml}, was used to set the minimum length of tiles that could participate in merges. We set α_{mml} to 5. A value of 1 produced the best performance but the processing time was prohibitive. The α_{length} parameter was set using a small set of 22 suspicious and 125 source documents from the PAN-PC-09 corpus. The length of the longest matching substring was computed for each source document and suspicious document in which it appeared. This value was always greater than 10. A small corpus of 30 suspicious documents from the PAN-PC-09 corpus was used to find a suitable value for the α_{merge} parameter for combining the adjacent tiles to create a section. For α_{merge}=250 characters, our heuristic achieved a performance of 0.58 overall, precision of 0.87, recall of 0.49, and granularity of 1.12.

3.2 System Performance

Our system achieved an overall performance of 0.20, precision of 0.40, recall of 0.16 and granularity of 1.21 in the formal evaluation.

Table 1 shows the performance of the candidate document selection (Section 2.2) and detailed analysis (Section 2.3) stages. Various performance metrics are shown for each stage and results broken down for each of the different types of obfuscation used in the corpus.

The focus of candidate selection step was to maximise recall, thereby ensuring that the source document is included in the set that are returned so that it can be identified in the detailed analysis stage. However, the recall values are quite low, indicating that the source document is often missed. Recall also varies according to the type of obfuscation. The best recall (0.7105) is obtained for low obfuscation and the worst for the simulated plagiarism (0.4154).

The results are somewhat different for the detailed analysis stage and the best performance is obtained for simulated plagiarism. We analyzed the documents and found that simulated obfuscated documents contain cases that have not been obfuscated and the number of source documents for this type of obfuscation is very small compared to other types.

	Candidate Document Selection			Detailed Analysis			
Obfuscation	Precision	Recall	F1	Precision	Recall	PlagDet Score	Granularity
None	0.8446	0.6693	0.7468	0.6366	0.2162	0.3205	1.0100
Low	0.7963	0.7105	0.7510	0.5960	0.1934	0.2619	1.1660
High	0.6729	0.6907	0.6817	0.6918	0.1775	0.2041	1.6110
Simulated	0.7643	0.4154	0.5383	0.6659	0.3980	0.4864	1.0330
Translated	0.9622	0.5941	0.7346	0.3994	0.2219	0.2610	1.3290

Table 1. Performance of candidate selection and detailed analysis stages for various types of obfuscation.

3.3 Sources of Error

Preprocessing (Section 2.1). Documents in German and Spanish were translated using Google Translate. These automatic translations contained errors that may have affected the accuracy of later processing stages.

Candidate Document Selection (Section 2.2) Analysis of the system performance (Section 3.2) showed that the source documents are often not identified.

Detailed Analysis (Section 2.3). Several parameters ($\alpha_{overlap}$, α_{length} and α_{merge}) were set using small training corpora. It is possible that larger training corpora would lead to better values for these thresholds. Processing limitations led the α_{mml} to be set to 5. This is a relatively high value which caused plagiarized sections with different types of obfuscation to be missed.

4 Conclusion

This paper described the University of Sheffield's entry to the 2nd international PAN plagiarism detection competition which attempted to identify extrinsic plagiarism. Our system did not attempt to identify intrinsic plagiarism. A three stage approach was used: preprocessing, candidate selection and detailed analysis. The approach achieved an overall performance of 0.20 in the evaluation with precision of 0.40, recall of 0.16 and granularity of 1.21 in the formal evaluation in which extrinsic and intrinsic plagiarism cases were evaluation together.

In the future we plan to improve our approaches for both the candidate document selection and detailed analysis stages. For the candidate document selection stage we plan to experiment with using approaches from information retrieval and copy detection (e.g. SCAM) to identify potential source documents since this may be faster than the approach that is currently used and may also improve recall. For the detailed analysis stage we plan to experiment with alignment techniques that can cope better with high obfuscation and also plan to make more use of approaches from natural language processing to improve matching. We also want to improve our approach at computing byte offsets in the source documents.

Acknowledgements

Rao Muhammad Adeel Nawab thanks the COMSATS Institute of Information Technology, Lahore, Pakistan for funding this work under the Faculty Development Program (FDP).

References

1. Bao, J., Shen, J., Liu, X., Liu, H., Zhang, X.: Finding Plagiarism Based on Common Semantic Sequence Model. In: Lecture notes in computer science. vol. 3129/2004, pp. 640–645. Springer (2004)

2. Brin, S., Davis, J., Garcia-Molina, H.: Copy detection mechanisms for digital documents. In: Proceedings of the 1995 ACM SIGMOD international conference on Management of data. pp. 398–409. ACM New York, NY, USA (1995)
3. Cavnar, W.B., Trenkle, J.M.: N-Gram-Based Text Categorization. In: Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval. pp. 161–175. UNLV Publications/Reprographics (1994)
4. Chen, J., Bao, Y.: CrossâĂŞlanguage search: The case of Google Language Tools. First Monday 14(3) (2009)
5. Clough, P., Gaizauskas, R., Piao, S., Wilks, Y.: Measuring text reuse. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania. pp. 152–159 (2002)
6. Clough, P., Stevenson, M.: Developing A Corpus of Plagiarised Short Answers. In: Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis. Springer (2010)
7. Hoad, T., Zobel, J.: Methods for identifying versioned and plagiarized documents. In: Journal of the American Society for Information Science and Technology. vol. 54, pp. 203–215 (2003)
8. Kent, C., Salim, N., Johor, M.: Features Based Text Similarity Detection. In: Journal of Computing. vol. 2 (2010)
9. Lane, P., Lyon, C., Malcolm, J.: Demonstration of the Ferret plagiarism detector. In: Proceedings of the 2nd International Plagiarism Conference (2006)
10. Lyon, C., Malcolm, J., Dickerson, B.: Detecting short passages of similar text in large document collections. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing. pp. 118–125 (2001)
11. Manning, C., Schutze, H.: Foundations of statistical natural language processing. MIT Press,Massacchusetts Institute of Technology, Cambridge MA (1999)
12. Schleimer, S., Wilkerson, D., Aiken, A.: Winnowing: local algorithms for document fingerprinting. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data. pp. 76–85. ACM (2003)
13. Shivakumar, N., Garcia-Molina, H.: SCAM: A copy detection mechanism for digital documents. In: Proceedings of the Second Annual Conference on the Theory and Practice of Digital Libraries,Austin,Texas (1995)
14. Shivakumar, N., Garcia-Molina, H.: Building a scalable and accurate copy detection mechanism. In: Proceedings of the first ACM international conference on Digital Libraries. pp. 160–168 (1996)
15. Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E.: 3rd PAN Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse. In: 25th Annual Conference of the Spanish Society for Natural Language Processing, SEPLN 2009. pp. 1–77 (2009)
16. Wise, M.: Running karp-rabin matching and greedy string tiling. Tech. Rep. 463, Basser Department of Computer Science Technical Report, Sydney University (1993)
17. Wise, M.: Neweyes: A system for comparing biological sequences using the running karp-rabin greedy string-tiling algorithm. In: Third International Conference on Intelligent Systems for Molecular Biology, Cambridge, England. pp. 393–401 (1995)
18. Wise, M.: YAP3: improved detection of similarities in computer program and other texts. In: Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education, Philadelphia, USA. pp. 130–134 (1996)