Automatic External Plagiarism Detection Using Passage Similarities

Clara Vania and Mirna Adriani

Fakultas Ilmu Komputer
Universitas Indonesia
Kampus Depok
Depok 16424, Indonesia
clara.vania@ui.edu, mirna@cs.ui.ac.id

Abstract. In this paper, we report our approach in detecting external plagiarism. For the pre-processing stage, we identify non-English documents and translate them into English using an online translator tool. Then we index and retrieve the top documents that are similar to the suspicious documents. We divide the retrieved documents into passages where each passage contains twenty sentences. The plagiarism is detected by identifying the number of overlapped words between suspicious and source passages.

Keywords: plagiarism detection, overlapping n-grams, passage retrieval

1 Introduction

Nowadays, plagiarism happen easily and more difficult to detect. With the advances of technology, especially the Internet, plagiarism can happen across languages and has different level of obfuscation. People can easily *copy* and *paste*, paraphrase, or translate websites, papers, or other sources from the Internet without mentioning its source and acknowledge it as their own work. This situation motivates in constructing an accurate automatic plagiarism detector. A plagiarism detector is a tool to detect if a suspicious document contains plagiarized work.

In recent years, some research in the text plagiarism detection have been published and developed. Mozgoyov et.al. (Mozgoyov, Kakkonen, and Sutinen, 2007) develop natural language parser to find swapped words and phrases to detect intentional plagiarism. Chen et.al. (Chen, Yeh, and Ke, 2010) use n-gram co-occurrence statistic to detect verbatim copy while LCS (Longest Common Subsequence) is used to handle text modification.

According to Potthas et al. (Potthast, et al., 2009), it is still difficult to determine the best system or algorithm to detect plagiarism because there is no controlled evaluation environment to compare the results. So, the PAN track on Plagiarism Detection was held last year to overcome this plagiarism problem. The plagiarism track offers two topics to detect text plagiarism automatically: external plagiarism and intrinsic plagiarism. The external plagiarism is intended to detect plagiarism section in a suspected document and its corresponding source document. While the intrinsic plagiarism detects a plagiarized section without comparing the suspect documents to the source documents.

Grozea et.al. (Grozea, Gehl, and Popescu, 2009) use character-16 gram VSM (Vector Space Model) for their retrieval model and get most similar documents to each suspicious document using cosine similarity score. To extract the pair sections, they join the matches based on a Monte Carlo Optimization. Basile et.al. (Basile et al., 2009) use word 8-grams VSM to retrieve similar documents and use their "joining algorithm" to extract the plagiarized passage. Kasprzak et.al. (Kasprzak et al., 2009) apply word-5-gram VSM to retrieve documents which share at least 20 n-grams with each suspicious document. Then they extract pairs of section which share at least 20 matching n-grams and at most 49 not-matching n-grams.

In this paper we report our approach in detecting plagiarism (external plagiarism). The remaining of this paper is organized as follows: section 2 discusses our methods in plagiarism detection, section 3 describes the evaluation and section 4 is the conclusion.

2 External Plagiarism Detection

In this section, we describe the method that we use in our plagiarism detection. There are four main steps in our detection method such as preprocessing stage, finding candidate documents, extract similar passages, and post-processing stage.

2.1 Preprocessing Phase

The pre-processing phase is mainly analyzing the corpus. The PAN '10 corpus[1] consists of 11.148 source documents and 15.925 suspicious documents. The corpus not only contains English documents but also several other languages. The external plagiarism cases also include the cross-lingual plagiarism cases. So, at the beginning we identify the language used in the documents using an automatic language identifier. The result shows that the non-English documents only occur in the source document set. The language identifier recognizes 10.480 English documents, 474 German documents, and 194 Spanish documents. Then we translate all non-English documents into English using an online language translator. We substitute the non-English documents in the corpus with their translated documents.

[1] http://pan.webis.de/

2.2 Finding Candidate Documents

The procedure in finding candidate documents is the same as document retrieval using suspicious document as queries. In this phase, we index the overall source documents and use suspicious documents as queries. We use Lucene[2] to index and retrieve the corpus. Lucene is an open source information retrieval system based on combination of Boolean Model and Vector Space Model. During the indexing process, we remove the stopwords, however we do not apply any stemming algorithm. In this work, for each suspicious document (as query), we retrieve the 10 most similar source documents.

2.3 Extract Similar Passages

We divide the top 10 source documents and suspicious documents into small passages. Each passage contains 20 sentences. Then we index and retrieve passages that are similar to the sections found in the source documents. We only use the top-5 similar source passages for each suspicious passage.

2.4 Post-processing Phase

In the post-processing phase, we analyze both of the pair passages. We filter the top-5 most similar source passages by removing pair passages that have low similarity score. After that, we compute the overlapping n-grams (Broder, 1997; Lyon et.al., 2001) between two passages. For the final result, we take pair passages that have at least three overlapping 6-grams. Small n-grams parameter is used because the size of the passages is also small (twenty sentences).

3 Evaluation

We don't have time to try our method using the training corpus, so the evaluation is only done using the testing corpus. Based on the evaluation measure given by the organizer (Potthast, 2010), the detail score of our algorithm can be seen in Table 2.

Table 2. The Evaluation Result

Measures	Score
Precision	0.9114
Recall	0.2620
Granularity	6.7764
Overall	**0.1375**

[2] http://lucene.apache.org

Our result show that our method performs quite good precision score (we were 4th for this parameter), but it has very low recall score. In other words, for the precision score, 91.14% of our detections are correct while 8.86% are incorrect. On the other hand, the recall means that our detector can only detect 26.2% of the overall plagiarism cases.

Based on our result, we need to explore further in terms of plagiarism with different level of obfuscation. The translation process at early stage is quite effective to overcome cross-language plagiarism, but in the detailed step, passage retrieval and n-grams overlapping technique just can handle exact match plagiarism. Plagiarism using word modification such as the use of synonym, word reordering, and paraphrasing still can't be identified using our method.

4 Conclusion

We report our participation in identifying external plagiarism in CLEF 2010. We apply N-grams overlapping words to measure the plagiarism between pair passages found in the documents. Our result achieves high precision (0.9114), but still low in terms of recall (0.2620). This method can identify the cross-language plagiarism, however it fails to detect plagiarism with various word modifications. In the future we will include words variations and develop method to detect plagiarism with different level of obfuscation.

References

Basile et al. 2009. A Plagiarism Detection Procedure in Three Steps: Selection, Matches and "Squares". In Stein et al. (Stein et al., 2009).

Broder, A Z. 1997. On the resemblance and containment of documents. In *Compression and Complexity of Sequences.* IEEE Computer Society.

Chen, Chien-Ying, Jen-Yuan Yeh, and Hao-Ren Ke. Plagiarism Detection using ROUGE and WordNet. *Journal of Computing*, *2(3)*, pages 34-44, March 2010. https://sites.google.com/site/journalofcomputing/. ISSN 2151-9617.

Grozea, Cristian, Christian Gehl, and Marius Popescu. 2009. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In Stein et al. (Stein et al., 2009).

Kasprzak, Jan, Michal Brandejs, and Miroslav Křipač. 2009. Finding Plagiarism by Evaluating Document Similarities. In Stein et al. (Stein et al., 2009).

Lyon et al. 2001. Detecting short passages of similar text in large document collections. *In Conference on Empirical Methods in Natural Language (EMNLP2001)*. pp. 118-125.

M. Mozgovoy, T. Kakkonen, and E. Sutinen. Using Natural Language Parsers in Plagiarism Detection. *In Proceeding of SLaTE'07 Workshop, Pennsylvania, USA,* October 2007.

Potthast, Martin et al. 2010. An Evaluation Framework for Plagiarism Detection. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China,* August 2010. Association for Computational Linguistics.

Potthast, Martin et al. 2009. Overview of the 1st International Competition on Plagiarism Detection. In Benno Stein, Paolo Rosso, Efstathios Stamatatos, Moshe Koppel, and Eneko Agirre, editors, *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09),* pages 1-9, September 2009. CEUR-WS.org. ISSN 1613-0073.