A Cluster-Based Plagiarism Detection Method

Lab Report for PAN at CLEF 2010

Du Zou
South China University of Technology
GuangZhou, China
duzou@scut.edu.cn

Wei-jiang Long
South China University of Technology
GuangZhou, China
wjlong@scut.edu.cn

Zhang Ling
South China University of Technology
GuangZhou, China
ling@scut.edu.cn

Abstract. In this paper we describe a cluster-based plagiarism detection method, which we have used in the learning management system of SCUT to detect plagiarism in the network engineering related courses. And we also used this method to detect external plagiarism in the PAN-10 competition. The method is divided into three steps: the first step, called pre-selecting, is to narrow the scope of detection using the successive same fingerprint; the second step, called locating, is to find and merge all fragments between two documents using cluster method; the third step, called post-processing, is to deal with some merging errors. Our method ran 19 hours in the PAN-10 competition, and the result ranked the second place, which met our expectation.

Keywords. Plagiarism detection, Similar text, Locating, Cluster

1 Introduction

Plagiarism detection, also known as text copy detection, is designed to determine whether a document is copied from other documents in whole or in part without any reference indicated. Besides copying text without any change, changing the order of the original text and replacing synonym are also regarded as plagiarism [1]. Text copy detection technology is widely used in intellectual property protection, search engine, e-library and student paper checks.

2 Related work

Text copy detection originated from program code similarity detection in the 1970's. Natural language text copy detection technique appeared in the 1990s, and has produced three detection approaches [2]: 1) Grammar-based method. This method focuses on the grammatical structure of documents, and uses a string-based matching approach to measure similarity between the documents. Huang[3] proposed a similar web pages detection method based on the LCS(Largest Common Subsequence) algorithm by finding the largest common string between two pages to calculate the similarity of the two pages. Winnowing algorithm [4] uses overlapping k-gram method to get hashes of the documents, and it uses moving window to select the minimum hash value from each window to obtain the fingerprints of the document, and then it calculates the rate of the matching fingerprint to get the similarity between the two documents. Hashbreaking[5], DCT[6] are also the grammar-based methods,

the only difference between them is how to get the fingerprints of the document. Using grammar -based method to detect verbatim copying can get better results than using it to detect the copied text including synonym replacement or rewriting. 2) Semantics-based method. This method uses the vector space model of the Information Retrieval Technology, and statistics word frequency in a document to obtain feature vector of the document, then uses dot product, cosine, etc. to measure the feature vector of the two documents. This feature vector is the key of the document similarity. This method is not always effective to detect partial plagiarism, because it is difficult to determine the location of copied text. 3) Grammar semantics hybrid method [1]. This method is used to solve the problems of the two methods mentioned above, and improve the detection results.

Locating is an important step of text copy detection technology. It is required to give the position of the plagiarized content in the document in addition to calculating the document similarity. By-word comparison is a common locating method. Sediyono et al proposed LCCW (Longest Commonly Consecutive Word) algorithm [7]. It regards a paragraph as a comparison unit, and splits it into a collection of consecutive words. The position of the words in the paragraph is recorded. Then using by-word comparison, the longest commonly successive word can be identified. Eventually, the content and position of the similar text in the document can be obtained. Zaslavsky et al proposed the MDR (Match detect Reveal) system [8]. In this system, preprocessed document is split into words or fixed-length strings. And using Matching Statistics Algorithm, a suffix tree is constructed for each fixed-length string, and then the longest common substring can be found among the suffix trees. According to the longest common substring, the similar text and its position in the document can be obtained. By-word comparison method builds the index by words, which is an exact matching method. Its locating performance is not good enough for the text which includes obfuscation. The top three of PAN-09 plagiarism detection contest [10] [11] [12] process the document by word or by sentence and combine with heuristic matching methods, to achieve approximate text matching. This method is suitable in English documents, however, it can not solve the Chinese word segmentation problem in Chinese documents.

The Cluster-based plagiarism detection method we propose is the grammar-based method. This method is divided into three steps: pre-selecting, locating and post-processing. We use the pre-selecting step to find out the documents which may be copied, so as to shorten the locating time and improve the detecting efficiency. And we use clustering method to locate plagiarism fragments, which can reduce the impact of obfuscation to some extent. In the last step, we deal with some errors which are found in locating step.

3 The datasets

We participated in the PAN-10 external plagiarism detection competition. The main objective of the external plagiarism detection is: given a collection of suspicious documents and source documents, to identify all possible copy fragments from the

suspicious documents and their locations in the suspicious documents and the corresponding source documents.

We merged PAN-09 training set and test set into a big set to debug our algorithm. This big set contains 14,429 source documents (training set 7214, and test set 7215), 14428 suspicious documents (training set 7214, and test set 7214). For each suspicious document, there is an xml document which notes all copy fragments in the suspicious document and their locations in the suspicious document and source documents.

Based on language, plagiarism fragments can be divided into monolingual (translation=false) and cross-lingual (translation=true). Based on the degree of obfuscation[9], plagiarism fragments can be divided into none obfuscation (obfuscation=none), low obfuscation (obfuscation=low) and high obfuscation (obfuscation=high). The length of plagiarism fragments is distributed between several hundred characters to ten thousand characters. The data set contains 73522 plagiarism fragments, among which English fragments are 67141 and multi-language fragments are 6381. In all of the English fragments, the number of none obfuscation is 26855 and the number of low obfuscation is 26628. The remainders are high obfuscation fragments.

4 Method

The plagiarism detection method we propose uses Winnowing's fingerprint extraction algorithm. The method consists of three steps: The first step is pre-selecting. For each suspicious document, the task is to find out a small list of candidate documents in which the plagiarized content may exist from the source document set quickly. The second step is locating, which compares the suspicious document with each candidate document to get the copy fragments out of the suspicious document. The last step is post-processing, which discards some fragments without plagiarism from the end result.

4.1 Pre-selecting

Supposed it takes an average of 100ms to process a pair of documents, and then the total computation time will be more than 200 days. Even if parallelizing the tasks with multi-core processors, the time needed is still unacceptable to the competition.

We found out that the most important step is locating. When we analyzed the algorithm, the locating step cost the most time. So we use the pre-selecting method to reduce the number of candidate documents, thereby the time of locating is shortened. Using the pre-selecting method we can save 90% of the running time.

C.Basile[11] computed the distance between each suspicious and source document, then selected the top 10 source documents with minimum distance for further processing. At the beginning, we used this approach to calculate the similarity of each suspicious and source document, and we selected the top 50 source documents according to the similarity.

However, during the testing, we found that there was possibility of false collision when using Winnowing method to obtain fingerprint. For example, the values of two fingerprints are 1024 and 2024, if they are divided by 1000, their remainders are both 24, but in fact the two fingerprints are not the same. We found that the larger the file is, the greater the probability of collision and the higher the similarity will be. So using this method affects the efficiency and accuracy of locating.

To improve the accuracy of pre-selecting, we use successive same fingerprint thresholds to get candidate documents. We define two parameters, valid interval and successive same fingerprint. There are two fingerprint vectors D1 and D2 representing two different documents, if the number of different fingerprint among a given pairs of the fingerprint is not greater than a given number, we consider the pairs of the fingerprint are the same. And this given number is a valid interval. The pair of the same fingerprint is called a successive same fingerprint if it meets the condition of valid interval. By comparing each source document with the suspicious document, the source document will be regarded as candidate document of the suspicious document if the value of the successive same fingerprint is greater than a given threshold.

By setting the successive same fingerprint threshold to pre-select, the results are as follows: we can find 99.9% of the documents which contain none obfuscation (obfuscation=none) fragments, 89.3% of the documents which contain only low obfuscation (obfuscation=low), and there are not a source document is selected which contains only high obfuscation fragments.

4.2 Locating

Locating is done between a suspicious document and a source document. In this step, we compare the suspicious document with each source document in the candidate document set, and then get a plagiarism fragment list of the suspicious document. The steps of locating are as follows:

1) Preprocessing. The purpose of this step is to remove all symbols which do not affect document semantic information, such as punctuation, whitespace, etc., converting all letters to lower case. And then we record the position of each word before and after preprocessing.

2) Sampling. The overlapping word-5-grams approach is used to obtain the initial fingerprint of the document. Referring to Winnowing's fingerprint sampling algorithm, we use 6 fingerprints as a window, and select the minimum fingerprint as a sample fingerprint of the window. We move the window by a fingerprint until the end of the initial fingerprint, the fingerprint that is repeatedly selected in the same position will be discarded. Finally the sample fingerprint vector of the document can be obtained. The beginning and the end position of the original text before preprocessing can be recorded by each fingerprint of the vector. Then the inverted index of the document can also be generated. The position of the original text is computed by the following formula:

$$SP_i = EP_{i-1} - 1; \quad EP_i = P_{cur} + w + k - 2$$

Where SP_i, EP_i are the beginning and the end position of the original text which is represented by the i-th fingerprint, P_{cur} is the beginning position of the current window, w is the size of the sample window, and k is the length of the original text.

3) Clustering and merging. Take the source-document10383.txt and suspicious-document07957.txt as example. The sample fingerprint vectors of the two documents have been obtained in previous steps. By comparing the two fingerprint vectors, a list of matches between the suspicious document and the source document can be obtained. We represent the pair of vectors in a bi-dimensional plane [11] with the vector of the suspicious document as x axis and the source document as y axis, and then we get the coordinates of all matches in the plane. A point in the figure 1 represents a match in the coordinates(x,y), namely, the x-th fingerprint in the vector of the suspicious document is equal to the y-th fingerprint in the vector of the source document.

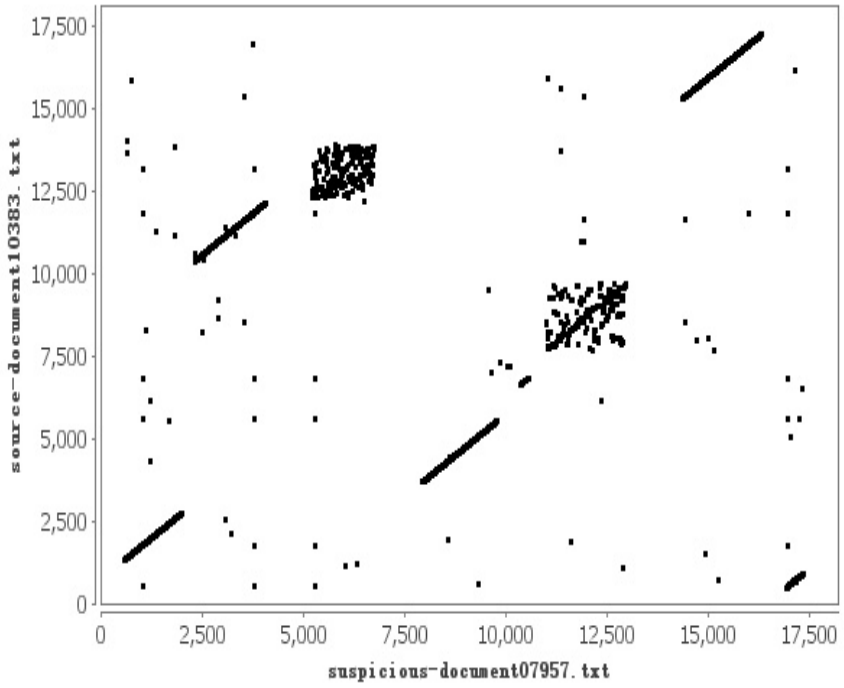

Figure 1. the same fingerprint between documents

By analyzing the sample fingerprint vectors of the documents, we can know, if a successive text in the suspicious document is copied from the source document, there is a set of points distributed along the direction of 45 degrees in the figure. None-obfuscated copy text corresponds to a line, and obfuscated copy text corresponds to a shadow square that contains a lot of lines and points. The task of our plagiarism detection method is to determine the location of copy text in respective document by the sample fingerprint vector.

We propose a two-stage approach. The first stage is merging, which uses the improved LCS (Longest Common Substring) algorithm to merge common substrings

of the two vectors. The algorithm finds all common substrings from the two vectors, and sets a threshold. If the distance between two substrings is less than the threshold, then merge them. Finally a set of approximate successive fingerprint sections can be found. There is an example of an approximate successive fingerprint section in the figure 2:

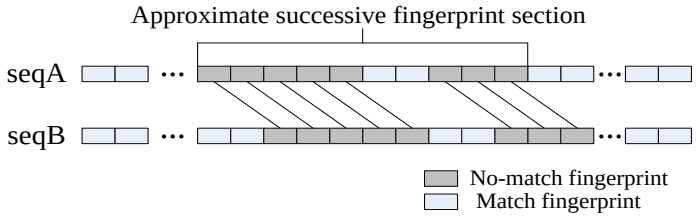

Figure 2. Approximate successive Fingerprint Section

The second stage uses the clustering method to reduce the impact of the obfuscated text on the locating. As shown in figure 1, a text with obfuscation corresponds to a shadow square that contains a lot of lines and points with a distribution along the direction of 45 degrees. Given (x_i, y_i, l_i) represents line a_i, x_i is the beginning position of line a_i in the fingerprint vector of the suspicious document, y_i is the beginning position of line a_i in the fingerprint vector of the source document, l_i is the length of the line a_i. There is a class $\overline{a}_i$, and line a_i belongs to class $\overline{a}_i$. The idea of the clustering is that, given a passage with a_i as axis and width is $2\delta_y$, if line a_j falls within this passage, and the distance between a_i and a_j along the passage's direction is less than δ_x, then we merge a_j into class $\overline{a}_i$. Clustering formula is as follows:

Given line a_i, class $\overline{a}_i$ and $a_i \in \overline{a}_i$, $\forall j > i$, if the following two conditions are fitted:

$$\left|x_i + y_j - (x_j + y_i)\right| \le \delta_y$$

$$\left|(x_i + y_i + l_i) - (x_j + y_j + l_j)\right| - 1.4 \times (l_i + l_j) \le \delta_x$$

then $a_j \in \overline{a}_i$.

By clustering, we can merge these small lines in the shadow square in figure 1 into a long line. The beginning and the end positions of the long line are the lower left and upper right of the shadow square.

4.3 Post-processing

The locating step uses the distance between approximate successive fingerprint sections to decide whether needs to merge or not. If using the same clustering parameter to merge those copy texts, which have different lengths and different obfuscated degrees, may cause merging errors.

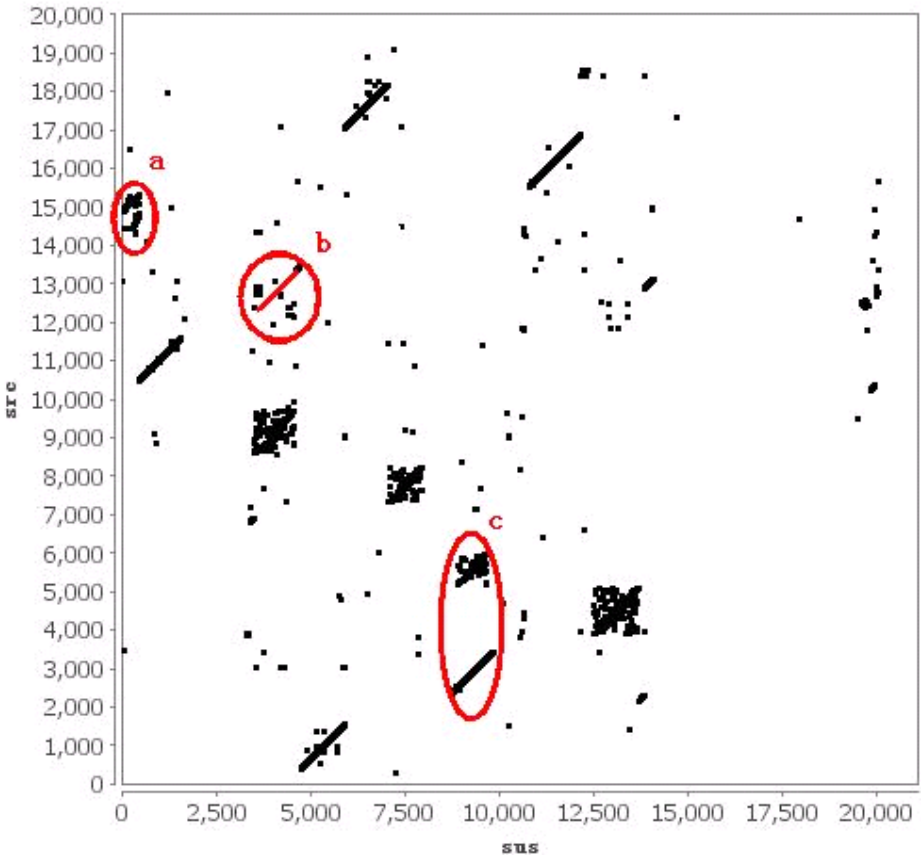

Figure 3. Three merging errors: a is the first error, b is the second error, c is the third error

Analysis to the experimental results shows, there are three major merging errors. The first error is the angle of the merged line deviates from 45 degree too much, because the length of the source fragment and the suspicious fragment that we found are so different. The second error is to merge sparse points. That means only a few fingerprints are the same in merged text. The third error is the same copy text in the suspicious document is found repeatedly in the source document.

There are two approaches to deal with these copy texts whose angles deviate from 45 degree too much. One approach is to reduce the cluster threshold to re-merge. Another approach is to directly discard these copy texts. Known by experiment, the re-merging will partly improve the precision, but the granularity will be worse, and has negative effects on the overall score, meanwhile, the re-merging impacts performance seriously. Therefore, in actual operation, we directly discard these copy texts.

For the second merging error, we use the similarity to decide whether discard copy texts. After merging, we compute the similarity of all copy texts, and discard those copy texts whose similarities are less than the threshold. This approach will discard some high-obfuscated copy texts, because the similarity of these copy texts is

relatively low. Through repeated experiments, copy texts whose similarities are less than 0.05 have mostly merging error, so we discard them.

From the analysis of the third merging error, we can find that there are two reasons causing the merging error. One is that we merge some sparse points into a fragment that in fact is not plagiarism. Another is that some fragments in the source document are indeed similar to the same fragment in the suspicious document. In the real competition, we do not know the right answer, so for this situation, we rank the product of the copy text's length and the respective similarity, and then select the copy text whose product is the largest, and discard other copy texts. Through testing in the training dataset, this approach can get a good result.

5 Experiment

We tested our method in two datasets. One is the union of training and test corpus in PAN-09, including 14429 source documents and 14428 suspicious documents. The other is only test corpus for PAN-09, which is composed of 7215 source documents and 7214 suspicious document.

Our experiments ran on 8 entries HPC (High Performance Computing). Each entry has a Dual-route Intel Xeon 4 cores 5500 series processor and 4G memory. In each experiment, we started 24 threads, 3 threads of each entry. The total running time was 25 hours in the union corpus, and 7 hours in the test corpus.

There are 15925 suspicious documents and 11148 source documents in the PAN-10 corpus, and the source documents contain 665 non-English documents. We used google translation api to translate about two thirds of the non-English documents into English. We used the cluster-based plagiarism detection method mentioned above to run this corpus in the same experimental environment, and the overall score is 0.7087, and the rank is: 2.

6 Conclusion

The method we proposed is cluster-based plagiarism detection method, which has been used in South China University of Technology to check plagiarism in network engineering related courses. And we also used it to detect external plagiarism in PAN-10 competition. The method uses Winnowing's fingerprint extraction algorithm, consists of three steps: The first step is pre-selecting. For each suspicious document, the task is to find out a small list of candidate documents in which the plagiarized content may exist from the source documents set quickly. The second step is locating, which compares the suspicious document with each document in the candidate documents to get the copy fragments out of the suspicious document. The last step is post-processing, which discards some fragments without plagiarism from the end result

References

[1] Bao Jun-Peng,Shen Jun-Yi,Liu Xiao-Dong,Song Qin-Bao, “A Survey on Natural Language Text Copy Detection[J]”,*Journal of Software*, 2003, vol.14, No.10, pp.1753-1760(Ch).

[2] Wang Tao, Fan Xiao-Zhong, Liu Jie, “Plagiarism Detection in Chinese Based on Chunk and Paragraph Weight”,in *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Kunming, pp.2574-2579, July 2008.

[3] Huang Lian’en, “On the Technologies for Building and Accessing a Web Archive”, PhD thesis, Peking University, 2008(Ch)

[4] Schleimer.S, D.S.Wilkerson, and A. Aiken, “Winnowing: Local Algorithms for Document Fingerprinting”, in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, ACM New York, NY, USA , pp.76–85, 2003

[5] O. Abdel-Hamid, B.Behzadi, Stefan Christoph, Monika Henzinger, “Detecting the Origin of Text Segments Efficiently”, *WWW 2009*, Madrid, Spain,pp.61-70,2009

[6] J. Seo and W. B. Croft, “Local Text Reuse Detection”, *SIGIR’08*, pp.571–578. ACM, 2008

[7] A Sediyono and KRK Mahamud, “Algorithm of the Longest Commonly Consecutive Word for Plagiarism Detection in Text Based Document”, *Digital Information Management*,pp. 253-259,2008

[8] Zaslavsky, Arkady et al,“Using Copy-Detection and Text Comparison Algorithms for Cross-Referencing Multiple Editions of Literature works”, *the 5th European Conference on Research and Advanced Technology for Digital Libraries*,pp.103-114, 2004.3−21.

[9] M.Potthast,B.Stein,A.Eiselt,A.Barón-Cedeño,and P.Rosso, “Overview of the1st International Competition on Plagiarism Detection”. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*,pp.1-9,2009.

[10] C.Grozea,C.Gehl,and M.Popescu. “ENCOPLOT:Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection”. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*,pp.10-18,2009.

[11] C.Basile,G.Cristadoro,D.Benedetto,E.Caglioti,and M.Degli Esposti. “A Plagiarism Detection Procedure in Three Steps:Selection, Matches and Squares”. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, pp.19-23,2009.

[12] Jan Kasprzak,Michal Brandejs,Miroslav Křipă. “Finding Plagiarism by Evaluating Document Similarities”. In *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*,pp.24-28,2009.