Approaches for Source Retrieval and Text Alignment of Plagiarism Detection

Notebook for PAN at CLEF 2013

Kong Leilei[1,2], Qi Haoliang[1], Du Cuixia[2],Wang Mingxing[2], Han Zhongyuan[1]

1Heilongjiang Institute of Technology, China
2Harbin Engineering University, China
kongleilei1979@gmail.com

Abstract. In this paper, we describe our approach at the PAN@CLEF2013 plagiarism detection competition. In sub-task of Source Retrieval, a method combined TF-IDF, PatTree and Weighted TF-IDF to extract the keywords of suspicious documents as queries to retrieve the plagiarism source document is proposed. In sub-task of Text Alignment, a method based on sentence similarity is presented. Our text alignment algorism and similar sentences merging algorism, called Bilateral Alternating Merging Algorithm, are described in detail.

1 Introduction

The great development of Internet makes it easier for people to search, copy, save, and reuse online sources. Copying another author's text and claiming its authorship is called plagiarism[1]. During the last decade, automated plagiarism detection in natural languages have attracted considerable attention from research and industry, which takes the advantage of recent developments in related fields like information retrieval, cross-language information retrieval, natural language processing, machine learning, and artificial intelligence. PAN@CLEF is dedicated to providing an environment which consists of a large scale corpus of artificial plagiarism and detection quality measures to evaluate the algorithms of plagiarism detection. There are two sub-tasks in PAN@CLEF2013: source retrieval and text alignment. The remaining sections of this paper introduce the methods we have taken in this year's competition.

2 Source Retrieval

The task of source retrieval is to retrieve all plagiarized sources while minimizing retrieval costs[2]. One document plagiarizes another document by simple cut–paste manipulations, minor or wholesale alternations and more ambiguity rewriting. One of the difficulties of efficiently detecting plagiarism source is to search the source in millions of documents from the Internet, where each document usually involves thousands of words. Another core problem of source retrieval is the keywords of suspicious document which would be used for retrieval are not specified. How to extract the keywords from the suspicious document as the queries is the significant challenge when applying the ChatNoir search engine API to retrieve the plagiarized sources. This year, we use the keywords extraction methods based on statistics, and combine with the three keyword extraction methods, which are based on TF-IDF, Pat Tree and Weighted TF-IDF. The following section describes the construction of these queries.

2.1 Keywords Based on TF-IDF

TF–IDF[3], term frequency–inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus. The term frequency is the number of occurrences of each term in a document. The inverse document frequency is a function of the number of document where a term took place. The method of keywords extraction based on TF-IDF chooses Wall Street Journal data as Corpus. After pre-treatment (removal of stop words), we calculate the TF-IDF value of each term for every suspicious document. Then, the terms which have the higher TF-IDF values are composed simply as the queries. The experiments show that we can obtain a better result on the measures of the number of queries until the first actual source is found and the number of downloads until the first actual source is downloaded when we make the only 10 terms with the highest TF-IDF value as queries. The experimental results are shown in section 2.5.

2.2 Keywords Based on PatTree

Pat tree is an efficient data structure successfully used in the area of keywords extraction in the field of information retrieval. It was developed by Gonnet[4] from Morrison's PATRICIA algorithm (Practical Algorithm to Retrieve Information Coded in Alphanumeric)[5] for indexing a continuous data stream and locating every possible position of a prefix in the stream. The PAT tree is conceptually equivalent to compressed digital search tree but smaller. Using this data structure, all possible character strings, including their frequency counts in the documents, can be got in a very efficient way. The different lengths of the character strings are used as queries in our method of source retrieval.

2.3 Keywords Based on Weighted TF-IDF

We believe that the words exist in the paragraph title and chapter headings contain more information. We give each term a weight according to its position. The final weighted TF-IDF value of each term is calculated by its TF-IDF value timing its weight. The weight is set to 3.6 if the term appears in the title of the article, while the weight is 2.5 when the term locates the first sentence or the last sentence of each paragraph.

2.4 Combination of Queries and Execution of Retrieval

All queries from a given suspicious document are extracted on the basis of the above methods and they are executed sequentially according to their priority.

Firstly, we extracte the top 20 words in the suspicious document according to their TF-IDF weights. The top 5 ranked keywords combine into the first query, and the next top 5 ranked keywords form the second query, and so on. In this manner, we get four groups of queries, and each query contains 5 terms.

Secondly, we get many groups of keywords based on PatTree which consist of different numbers of terms. We choose the keywords which composed by 2, 3, 4 and 5 terms and rank them by their sum of TF-IDF value and the frequency counting in the suspicious documents. We call them 2-gram, 3-gram, 4-gram and 5-gram PatTree keywords. One 2-gram PatTree keyword, one 3-gram PatTree keyword and two 4-gram PatTree keywords are regards as following four queries. Especially, 5-gram PatTree keywords are chosen to all the other queries. We discard the keywords whose the number of nouns and verbs under threshold 3. The queries which have no any term existing in the list of top 10 weighted TF-IDF also are filtered out by us.

Table 1 shows the combination of queries.

Rank	Query type	Extracted Query Number
1 to 2	Top 10 TF-IDF	2
3	2-Gram PatTree Keyword	1
4	3-Gram PatTree Keyword	1
5 to 6	4-Gram PatTree Keyword	2
7 to 8	Top 10 to 20 TF-IDF	2
other	5-Gram PatTree Keyword	about 40

Table 1. Combination of queries.

After committing all queries we extract from a suspicious document, we decide whether to actually download the web pages or not according to the cosine similarity between the snippet and the suspicious document. We extract about 40 queries for each suspicious document.

2.5 Result

According to the five performance measures that PAN@CLEF2013 employs, the more web pages are download, the better recall could be got. For example, if the downloading number of web page for each document returned is set to 1000, the recall can reach 0.5719. By setting this value 600 we could lower the total number of web pages downloaded by 400 with only 2.05% of recall lost. To ensure the evaluation measures of *Number of queries submitted*, *Number of queries until the first actual source is found* and *Number of downloads until the first actual source is downloaded* better in this evaluation, we try to improve the measure of precision and recall of web pages downloaded regarding actual sources of a suspicious document, and reduce the measure of *Number of web pages downloaded*.

In the experiment, we have found that the queries constructed by top 10 TF-IDF words got a better result. The experiment result without using snippet filter on the training data of Source Retrieval is shown in Table 2.

Total workload	Queries	80
	Downloads	1289
Time to 1st Detection	Queries	1.225
	Downloads	473.475
Retrieved Sources	Precision	0.004
	Recall	0.2091

Table 2. TF-IDF Result with only top 10 TF-IDF words

Table 3 shows results of PAN@CLEF2013 Source Retrieval subtask.

Workload	Queries	48.50
	Downloads	5691.47
Time to 1st Detection	Queries	2.46
	Downloads	285.66
Retrieved Performance	Precision	0.01
	Recall	0.65
No Detection		3

Table 3. Results of PAN@CLEF2013 Source Retrieval subtask

3 Text Alignment

Given a pair of documents, the task of text alignment is to identify all contiguous maximal-length passages of reused text between them[2].The subtask of Text Alignment has been called Detailed Comparison or Detailed Analysis. The Detailed Analysis method of plagiarism detection is summarized in [6]. It can be divided into 3 steps: (1) seeding, (2) match merging and (3) extraction filtering. Given a suspicious document and a source document, seeding refers to use seeds for matches. These matches are called "seed". The purpose of match merging is to combine the seeds obtained in step (1) which having a maximum length of text fragments. The extraction filtering combined fragments to get the final plagiarism fragments according to certain standards. Our method also uses the above-described process. The following sections will describe our approach.

Algorithm 1 describes the Text Alignment algorithm of plagiarism detection.

Algorithm 1. Text Alignment algorithm of the plagiarism detection

Input: Suspicious document dplg，the plagiarism alternative documentation set Dsrc

Output: The collection of the pairs(splg , ssrc) from the plagiarism fragment of the suspicious document splg and the plagiarism source document fragment ssrc

```
Pre-Processing dplg and Dsrc
Feature represent dplg and  Dsrc
for each dsrc ∈ Dsrc {
    Measure sentences similarity of dplg and  dsrc
}
Merge similar sentences to similar passages
(splg , ssrc)← Post-Processing the pairs of similar passages
return the set of passage pairs (splg , ssrc)
```

We described the process of seeding in [7].

The result of seeding is that we get a number of scattered one-to-many sentence pairs. We designed an algorithm called Bilateral Alternating Merging Algorithm to combine these fragments into a larger fragment as plagiarism fragment. The description of the algorithm is shown as follows.

Algorithm2. Bilateral Alternating Merging Algorithm

Input: Similar sentence list casesList

Output: The collection of the pairs(splg , ssrc) from the plagiarism fragment of the suspicious document splg and the plagiarism source document fragment ssrc

```
public void merger(List<String> casesList, int sign) {
    if (sign = -1) {
        leftSort(casesList);
        if (end >= leftEndge && end <= rightEndge) {
        resultList.add(dto);
          } else {
        for (i = 1..casesList.size()) {
            if (|now – last| > dist) {
                checkAdjacent();
              merger();
            }
        }
     } else {
          rightSort(casesList);
           if (endgeFirstInfs=endgeEndInfs && end >= leftEndge&& end <=
              rightEndge) {
                resultList.add(dto);
          } else {
              for (i = 1..casesList.size()) {
                  if (|now – last| > dist) {
                          checkAdjacent();
                          merger();
                  }
        }
    }
return resultList;
}
```

In paper [7], we discard passages whose word overlap under a threshold. This year, we improve the method of post-processing. We apply a sliding window surrounding the original answer to refine the final plagiarism fragment which has a larger Jaccard coefficient value than the initial result we got. Due to time constraints, the parameters are not fully adjusted, which makes our PlagDet scores increased only by 0.5%.

The experiments are carried out in the PAN@CLEF2012 training data set. Table 4 shows the results of our Text Alignment sub-task which applied different parameters.

Sub-Corpus	Plagdet Score	Recall	Precision	Granularity
02_no_obfuscation	0.928215	0.989573	0.874021	1.00000
03_artificial_low	0.843241	0.807480	0.951852	1.05077
04_artificial_high	0.583010	0.435398	0.946562	1.03220
06_simulated_paraphrase	0.777298	0.704972	0.869630	1.00248
Overall	0.767636	0.686730	0.914581	1.03058

Table 4. Performance on the training corpus

Table 5 shows results of PAN@CLEF2013 Text Alignment subtask.

Sub-Corpus	Plagdet Score	Recall	Precision	Granularity
02_no_obfuscation	0.82740	0.90682	0.76077	1.00000
03-random-obfuscation	0.82281	0.78682	0.86224	1.00000
04-translation-obfuscation	0.85181	0.84626	0.85744	1.00000
05-summary-obfuscation	0.43399	0.30017	0.96384	1.07742
Overall	0.81896	0.81344	0.82859	1.00336

Table 5. Performance on the test corpus

4 Conclusions

In the sub-task of Source Retrieval, we apply a method based on TF-IDF, Weighted TF-IDF and PatTree to extract the keywords to compose the queries of suspicious documents. The number of queries submitted is reduced by two filters. The performance measures of *Number of queries until the first actual source is found* and *Number of downloads until the first actual source is downloaded* get the promotion by combining and ranking the queries. To achieve better performance measure *recall* and *precision*, we try to keep a balance between *Number of web pages downloaded* and *Precision and recall of web pages downloaded regarding actual sources of a suspicious document.*

In the sub-task of Text Alignment, our method is more adaptable to obfuscation plagiarism. We achieve the better performance in sub-corpus 03-random-obfuscation, 04-translation-obfuscation and 05-summary-obfuscation. But the performance for sub-corpus 02-no-obfuscation still remains to rise.

There are many high-performance methods in dealing with no-obfuscation plagiarism, therefore the current results show that it is possible to build up a classifier based on different plagiarism types. Furthermore, we can use different methods to deal with different plagiarism problems to obtain a better performance. Moreover, more efforts are required to figure out more plagiarism features to be used in the design of the plagiarism type classifier to improve the performance.

Acknowledgements This work is supported by NSF of China(61272384).
Remark This work was done in Heilongjiang Institute of Technology.

Reference

[1] Potthast M, Eiselt A, Barrón-Cedeno A, et al. Overview of the 3rd international competition on plagiarism detection[C].Notebook Papers of CLEF 2011 LABs and Workshops. 2011: 19-22.

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

[3] SALTON G, CLEMENTT Y. On the construction of effective vocabularies for information retrieval[C]. Proceedings of the 1973 Meeting on Programming Languages and Information Retrieval New York: ACM, 1973: 11.

[4] Gaston H. Gonnet, Ricardo A. Baeza-yates, Tim Snider. New Indices for Text: Pat Trees and Pat Arrays. Information Retrieval Data Structures & Algorithms, Prentice Hall, pp. 66-82, 1992.

[5] Morrison, D.. PATRICIA: Practical Algorithm to Retrieve Information Coded in Alphanumeric. JACM, pp. 514-534, 1968.

[6] Martin Potthast, Tim Gollub, Matthias Hagen, Jan Graßegger, Johannes Kiesel,Maximilian Michel, Arnd Oberländer1 Martin Tippmann, Alberto Barrón-Cedeño, Parth Gupta, Paolo Rosso, Benno Stein. Overview of the 4th International Competition on Plagiarism Detection. CLEF 2012 Evaluation Labs and Workshop–Working Notes Papers.17-20 September, Rome, Italy. ISBN 978-88-904810-3-1. ISSN 2038-4963. 2012.

[7] Leilei Kong, Haoliang Qi, Shuai Wang, Cuixia Du, Suhong Wang, and Yong Han. Approaches for Candidate Document Retrieval and Detailed Comparison of Plagiarism Detection—Notebook for PAN at CLEF 2012. In Forner et al. [6]. ISBN 978-88-904810-3-1. URL.