

# Unsupervised Ranking for Plagiarism Source Retrieval

## Notebook for PAN at CLEF 2013

Kyle Williams<sup>1</sup>, Hung-Hsuan Chen<sup>2</sup>, Sagnik Ray Choudhury<sup>1</sup>, and C. Lee Giles<sup>1,2</sup>

<sup>1</sup>Information Sciences and Technology

<sup>2</sup>Computer Science and Engineering

Pennsylvania State University

University Park, PA, 16802, USA

kwilliams@psu.edu, hhchen@psu.edu, szr163@ist.psu.edu, giles@ist.psu.edu

**Abstract** The source retrieval task for plagiarism detection involves the use of a search engine to retrieve candidate sources of plagiarism for a suspicious document and provides a way to efficiently identify candidate documents so that more accurate comparisons can take place. We describe a strategy for source retrieval that makes use of an unsupervised ranking method to rank the results returned by a search engine by their similarity with the query document and that only retrieves documents that are likely to be sources of plagiarism. Evaluation shows the performance of our approach, which achieved the highest  $F_1$  score (0.47) among all task participants.

## 1 Introduction

The advent of the Web has led to unprecedented levels of information access and an exponential increase in the amount of information available. These new levels of access have had a number of benefits, such as easy and immediate access to information on important topics, such as healthcare, disaster management, and research. However, one of the consequences of this access is that it has become increasingly easy to plagiarize. For instance, a study in 2010 found that 1 in 3 American high school students admitted to plagiarizing information from the Internet<sup>1</sup> and a study of college students from 2002-2005 found that 36% of undergraduates admitted to plagiarizing or copying from the Internet without citing the source [7]. Thus, the detection of plagiarism has become an important problem in academic institutions and other organizations around the world and many approaches for detecting plagiarism have been developed [6]. Generally, the most common task in plagiarism detection is:

*Problem 1.* Given a suspicious document and a potential source document for plagiarism, find all areas of overlapping text, which may have been subjected to exact copying.

This is the classic plagiarism text alignment task; however, another task for plagiarism detection involves the identification of potential sources of plagiarism before

---

<sup>1</sup> [http://charactercounts.org/programs/reportcard/2010/installment02\\_report-card\\_honesty-integrity.html](http://charactercounts.org/programs/reportcard/2010/installment02_report-card_honesty-integrity.html)

text alignment takes place. This task, known as *source retrieval*, involves querying an information retrieval system in order to retrieve documents that may be sources of plagiarism so that text alignment can be performed on the retrieved documents. This task of source retrieval can be described as:

*Problem 2.* Given a suspicious document and a search engine, use a search engine to retrieve candidate documents from the Web that may be sources of plagiarism.

There are a number of issues in solving Problem 2. For instance, it is desirable to achieve high recall by retrieving as many correct sources of plagiarism (true positives) as possible. Similarly, it is desirable to achieve high precision by, to the maximum extent possible, only retrieving sources that are true positives so as to minimize unnecessary computation and bandwidth usage. Achieving high recall is relatively simple since all that is required is to submit as many queries as possible to the search engine and download as many results as possible. Similarly, high precision can be maintained by having strict criteria for what is considered a source of plagiarism and only downloading documents that are highly likely to be true positives. However, there is a tradeoff between precision and recall since an increase in one usually comes as the expense of the other. Other goals for source retrieval involve minimizing the amount of stress put on the information retrieval system in terms of the number of queries submitted as well as minimizing bandwidth utilization and the number of documents downloaded.

In this paper, we describe a solution to Problem 2 that attempts to achieve both high precision and recall, while minimizing the amount of bandwidth utilized and documents downloaded. Core to the solution is the use of an unsupervised ranking method that re-ranks the search results from a search engine before they are downloaded in order to increase the probability that only true positives are retrieved (high precision) while increasing the likelihood that the true positives that exist are retrieved (high recall). In describing this solution, the rest of this paper is structured as follows. Section 2 describes related work, while Section 3 describes the approach we used to solve the problem. Section 4 describes the evaluation of our approach and, lastly, conclusions are drawn in Section 5.

## 2 Related Work

The source retrieval task was first introduced as a sub-task at PAN 2012 and thus the approaches used in that year serve as a good starting point for a discussion of related work. As mentioned in the overview of the plagiarism detection task [10], most approaches to source retrieval in 2012 included the separation of text into chunks, followed by the extraction of keyphrases and query formulation, and then the control and downloading of results. The majority of the approaches taken seemed to be very similar, with the most significant differences occurring in the steps for query formulation.

The source retrieval task can be formulated within the framework of the search methodology known as *Query by Document (QBD)*. In QBD, whole documents are submitted to a search engine that supports QBD, usually with the goal of retrieving similar documents [14]. In previous works investigating QBD, the workflow usually includes the submission of a whole document as a query, the automatic extraction of

queries from the document, the submission of the queries, and the ranking of results by their similarity with the query document [2]. QBD has been used for retrieving similar documents on the Web [9] and research paper recommendation [8], among other applications. Applying QBD to the PAN source retrieval task, a suspicious document is received as input and the goal is to use the document to construct queries that can be used to retrieve candidate plagiarism sources and rank the returned results by the level of plagiarism that occurs. Since this task is concerned only with the retrieval of sources of plagiarism, the actual overlap among the texts is not calculated but rather an Oracle is consulted to determine whether or not a downloaded text is a source of plagiarism.

A key aspect of source retrieval and QBD is the construction of the queries that are submitted to a search engine in order to retrieve plagiarized sources from the Web. The construction of queries could be viewed as a keyword extraction problem since the goal is to identify words in a document that can be used to construct good queries. There have been a number of studies investigating different methods for keyword extraction. For instance, two studies compared unsupervised methods for keyword extraction and, in both cases, found that TF-IDF ranking of keywords in a document performed well for different corpora [5][3]. TF-IDF-based ranking was used in the PAN 2012 source retrieval task by several participants [10], though one of the best performing approaches at PAN 2012 constructed queries from sequences of POS-tagged words [4].

Lastly, commercial tools like Turnitin<sup>2</sup> and The Essay Verification Engine (EVE)<sup>3</sup> also exist for plagiarism detection. For Turnitin, users upload documents via a Web interface and plagiarism detection takes place remotely and EVE takes a document as input, constructs queries from the document and conducts a Web-based search in order to identify similar documents.

### 3 Approach

Our approach involves four main steps: 1) query generation, 2) query submission, 3) result ranking; and 4) candidate document downloading. The general approach methodology of our approach is based on the idea that it is easy to achieve either high precision or recall for source retrieval, though achieving both is relatively more difficult. In order to do this, it is essential to submit queries that are likely to return true positives as well as only download documents for evaluation that are likely to be true positives. Our approach to accomplishing these goals involves the use of unsupervised ranking method to re-rank the results returned by the search engine for a set of queries by their similarity to the suspicious document before downloading them. Algorithm 1 shows our source retrieval algorithm, which we describe in more detail in the sections below.

For most of the steps in Algorithm 1, there are parameters for which the values need to be set. In the majority of cases, the best values for the parameters were empirically found by varying each parameter while keeping the others constant and then measuring precision and estimating recall.

---

<sup>2</sup> <http://www.turnitin.com/>

<sup>3</sup> <http://www.canexus.com/eve/abouteve.shtml>

---

**Algorithm 1** General overview of source retrieval strategy

---

```
1: procedure SOURCERETRIEVAL(doc)
2:   paragraphs  $\leftarrow$  SPLITINTOPARAGRAPHS(doc)
3:   for all p  $\in$  paragraphs do
4:     p  $\leftarrow$  PREPROCESS(p)
5:     queries  $\leftarrow$  EXTRACTQUERIES(p)
6:     for i = 0  $\rightarrow$  n do                                      $\triangleright$  n is the top n queries
7:       results  $\leftarrow$  SUBMITQUERIES(queries[i])
8:     end for
9:     results  $\leftarrow$  RANK(results)
10:    for all result  $\in$  results do
11:      if SIMILARITY(result)  $\geq$  t then                        $\triangleright$  t is the similarity threshold
12:        if PREVIOUSSOURCE(result) = false then
13:          source  $\leftarrow$  DOWNLOAD(result)
14:        end if
15:        if ISSOURCE(source) then
16:          print source
17:          continue
18:        end if
19:      end if
20:    end for
21:  end for
22: end procedure
```

---

### 3.1 Query Generation

We automatically generate queries that are submitted to a search engine where the goal is to maximize the probability that the results they return contain true positives. Our query generation strategy is based on the idea that different parts of a document are plagiarized from different sources and similar to that of Jayapal [4]. The text is first partitioned into *paragraphs* that are made up of 5 sentences as tagged by the Stanford Tagger [13] and stop words are removed. It was found that the number of sentences included in a paragraph had a large effect on the performance; for instance, increasing the number of sentences to 10 led to an increase in precision by about 2%, but a decrease in estimated recall of over 20%.

After paragraphs are extracted, each word in each paragraph is tagged using the Stanford POS Tagger and, following Liu et al. [5], only verbs, nouns, and adjectives are considered as keywords. Queries are constructed by combining each non-overlapping sequence of  $k$  keywords, where  $k = 10$ , in order to create a set of queries for each paragraph. Furthermore, each keyword was only included in a query once per paragraph.

We also tried constructing queries by ranking keywords by TF-IDF and BM25 and then combining the top  $k$  keywords. We empirically found that: TF-IDF keyword ranking performed the poorest; BM25 performed better with similar precision to TF-IDF but with estimated recall that was approximately 15% higher; and sequential keywords had a precision that was approximately 8% lower but that had recall that was about 13% higher than BM25. Furthermore, we found that sequential keywords required that fewer queries be submitted to the search engine in order for it to return true positives.

### 3.2 Query Submission

The first 3 queries extracted from each paragraph are submitted to the ChatNoir search engine [11]. This number was chosen since it was empirically found that, in the majority of cases, the first true positive would be returned by one of the first 3 queries. The queries are submitted as a batch based on the intuition that the results returned by all three queries combined are more likely to contain true positives than the results returned by any query individually. For each query, only the top 3 results are returned since this value was found to perform well. Returning more results increased recall and decreased precision, while returning fewer had the opposite effect. The top 3 results are then combined for the 3 queries leaving a final maximum of 9 results; however, there may be fewer if a query returned fewer than 3 results.

### 3.3 Result Ranking

High precision is achieved by assuming that the order in which the results were ranked by the search engine does not truly reflect the probability that a result is a true positive. Thus, all results for the three queries are combined and re-ranked before downloading.

For any document in the ChatNoir search engine, it is possible to submit the document ID and a selection of keywords and a snippet of the document is returned that shows the matched keywords and the context in which they occur. Thus, for each of the results returned, a snippet of the document is downloaded using the original query that returned that result as the keywords and then all results are ranked by the similarity of their snippet (after stripping HTML markup) with the suspicious document.

The similarity between the snippet of each document and the suspicious document is calculated based on an unsupervised method for calculating document similarity [1]. For each snippet and the suspicious document, the  $w$ -shingles are extracted, where the  $w$ -shingles are the overlapping sequences of  $w$  tokens in the snippet or the document. The similarity between a snippet  $s$  and a suspicious document  $d$  is then calculated as:

$$Sim(s, d) = S(s) \cap S(d), \quad (1)$$

where  $S(\cdot)$  is a set of shingles. The pairs of snippets and suspicious documents are then ranked by their similarity. We set  $w = 5$  and found that decreasing  $w$  led to higher recall and lower precision, while increasing it had the opposite effect.

### 3.4 Candidate Document Downloading

Each document for which the similarity was above a threshold  $t$  was downloaded in the order in which they were ranked. We set  $t = 5$  and found that decreasing  $t$  led to higher recall and lower precision, while increasing it had the opposite effect. For each downloaded document, the ChatNoir Oracle is used to determine if the candidate document is a true positive for the suspicious document. If it is, no more candidate results are downloaded for the current paragraph and the whole process is repeated for the next paragraph. Furthermore, a list is maintained that contains a record of each document downloaded since, if a candidate document has already been downloaded, downloading it again does not improve performance for the current suspicious document.

## 4 Evaluation

Table 1 shows that performance of our approach on the test data and as evaluated by the PAN organizers using the PAN corpus<sup>4</sup> [12].

**Table 1.** Performance of approach on test data

Retrieval Performance			Workload		Time to 1st Detection		No	Runtime
$F_1$	Precision	Recall	Queries	Downloads	Queries	Downloads	Detection	
0.47	0.55	0.50	116.40	14.05	17.59	2.45	5	69781436

### 4.1 Retrieval Performance

As can be seen from Table 1, our approach achieved precision and recall of 0.55 and 0.50 respectively. Overall, both the precision and recall were the second highest achieved by all participants; however, the harmonic mean  $F_1$  score, which captures the tradeoff between precision and recall, was the highest achieved by all participants in the task. Thus, in terms of the overall retrieval performance, our approach was very competitive.

### 4.2 Workload

An average of 116.4 queries were submitted per document and 14.05 results downloaded. The number of queries is directly related to the size of the paragraphs extracted from the text because increasing the number of sentences per paragraph increases the number of queries since a fixed number of queries are submitted per paragraph. Thus, it is possible to vary the number of queries submitted by changing the number of sentences and queries per paragraph. However, as already mentioned, the values for these variables were chosen based on their performance.

The average number of downloads per document was 14.05 and thus on average at least 14 of the results returned per document had at least 5 shingles in common with the query document. It is possible that there were more; however, they may not have been downloaded due to previously being downloaded or due to a true positive already being found. Of these 14.05 documents downloaded, on average just over half of them were true positives as is evident from the precision of 0.55.

### 4.3 Detection

On average, 2.45 results were downloaded until the first true positive. This suggests that the unsupervised results ranking method employed worked relatively well with relatively few false positive documents being downloaded. The average number of queries until first detection suggest that, on average, a true positive was identified based on

<sup>4</sup> The  $F_1$  score is computed by averaging the  $F_1$  score of each run rather than from the average precision and recall.

queries generated from one of the first 6 paragraphs that occurred in the text (6 paragraphs \* 3 queries). Furthermore, the fact that the first true positive was within the first 3 queries confirms that submitting 3 queries per paragraph was a good approach.

No plagiarism sources were detected for 5 suspicious documents, which was 8.6% of the suspicious documents in the test data. This was the second fewest among all participants, thereby demonstrating that the method performs relatively well at retrieving sources of plagiarism for the majority of documents.

#### 4.4 Bandwidth Utilization

The number of queries was relatively high compared to the other methods; however, this was a feature of our approach rather than a shortcoming. We estimated the amount of bandwidth used by our approach by storing the data returned by any HTML request in a text file and checking the size of the text file. We found that the amount of bandwidth used to submit the top 3 queries with each returning 3 results for a single test document (suspicious-document001) was < 4 KiB and the size of each of the snippets downloaded was around 1 KiB. Contrasting this with amount of bandwidth used to download the first result for the first query, which was 90 KiB, and it can be seen that queries are relatively cheap from a bandwidth perspective compared to downloading documents. Thus, there is evidence that our approach is quite efficient when it comes to bandwidth utilization; however, since a relatively large number of queries are submitted, the one downside is the additional query processing that needs to take place on the server.

## 5 Conclusions

The goal in source retrieval is, ultimately, to retrieve as many sources of true plagiarism as possible while minimizing unnecessary computation. The first and perhaps most important step is query formulation where the question is: *how best can queries be formulated that capture the information content of suspicious documents?* This is an open research question that has importance not only in plagiarism detection, but also in key-word extraction, text summarization, and other query by document applications.

In addition to query formulation, the management of the potentially large number of results returned by search engines is important since it is desirable to determine which of those results are likely to be true positives so as to minimize unnecessary downloads and document comparisons in order to improve processing speed.

Our approach to the source retrieval task addressed both issues. Query construction was based on a method that has previously been shown to work well [4], while the result management was based on an unsupervised re-ranking method that, to the best of our knowledge, is novel in its application. Using this approach, competitive recall and precision were achieved as well as the best performing tradeoff between the two.

The method employed involved a number of parameters for which values that performed well were found empirically. However, there may be some theoretical foundations for estimating some of these parameter values so as to maximize performance and investigating this could form the basis for future work. Future work could also investigate new methods for query formulation as well additional methods for result ranking so as to improve both precision and recall.

## Acknowledgments

We gratefully acknowledge partial support by the National Science Foundation under Grant No. 1143921 and the PAN Lab organizers for their effort in organizing PAN 2013.

## References

1. Broder, A., Glassman, S., Manasse, M., Zweig, G.: Syntactic clustering of the Web. *Computer Networks and ISDN Systems* 29(8-13), 1157–1166 (Sep 1997)
2. Dasdan, A., D'Albeto, P., Kolay, S., Drome, C.: Automatic retrieval of similar content using search engine query interface. In: *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*. pp. 701–710 (Nov 2009)
3. Hasan, K., Ng, V.: Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. *proceedings of the 23rd International Conference on Computational Linguistics (1999)*, 365–373 (2010)
4. Jayapal, A.: Similarity Overlap Metric and Greedy String Tiling at PAN 2012: Plagiarism Detection. *CLEF (Online Working Notes/Labs/Workshop)* (2012)
5. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. pp. 620–628. No. June (2009)
6. Maurer, H., Media, C., Kappe, F., Zaka, B.: Plagiarism - A Survey. *Journal of Universal Computer Science* 12(8), 1050–1084 (2006)
7. McCabe, D.L.: Cheating among college and university students : A North American perspective. *International Journal for Educational Integrity* 1(1), 1–11 (2004)
8. Nascimento, C., Laender, A.H., da Silva, A.S., Gonçalves, M.A.: A source independent framework for research paper recommendation. In: *Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries - JCDL '11*. pp. 297–306 (Jun 2011)
9. Pereira, A., Ziviani, N.: Retrieving similar documents from the web. *Journal of Web Engineering* 2(4), 247–261 (2004)
10. Potthast, M., Gollub, T., Hagen, M., Graß egger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-cede no, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th International Competition on Plagiarism Detection pp. 17–20 (2012)
11. Potthast, M., Hagen, M., Stein, B., Graß egger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A Search Engine for the ClueWeb09 Corpus. In: *Proceedings of the 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12)*. p. 1004 (Aug 2012)
12. Potthast, M., Hagen, M., Völske, M., Stein, B.: Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In: *51st Annual Meeting of the Association of Computational Linguistics (ACL 13)*
13. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*. vol. 1, pp. 173–180 (May 2003)
14. Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N., Papadias, D.: Query by document. In: *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. pp. 34–43 (2009)