Source Retrieval for Plagiarism Detection from Large Web Corpora: Recent Approaches

Matthias Hagen, Martin Potthast, and Benno Stein

Bauhaus-Universität Weimar

pan@webis.de http://pan.webis.de

Abstract This paper overviews the five source retrieval approaches that have been submitted to the seventh international competition on plagiarism detection at PAN 2015. We compare the performances of these five approaches to the 14 methods submitted in the two previous years (eight from PAN 2013 and six from PAN 2014). For the third year in a row, we invited software submissions instead of run submissions, such that cross-year evaluations are possible. This year's stand-alone source retrieval overview can thus to some extent also be used as a reference to the different ideas presented in the last three years—the text alignment subtask will be depicted in another individual overview.

1 Introduction

The retrieval and extraction of text reuse from large document collections is central to applications such as plagiarism detection, copyright protection, and information flow analysis. Appropriate algorithms have to be able to deal with all kinds of text reuse ranging from verbatim copies and quotations to paraphrases and translations to summaries [12]. Particularly the latter kinds of text reuse still present a real challenge to both engineering and evaluation of retrieval algorithms. Until recently, one of the primary obstacles to the development of new algorithms has been a lack of evaluation resources. To rectify this lack and to enable the source retrieval subtask, we have worked on a high-quality, large-scale evaluation framework [17, 18], that has been used in the last four years.[1]

The source retrieval subtask has been running for four years in a row now and we can observe the standard multi-year life cycle of repeated shared tasks. Basically, there are three phases: an innovation phase, a consolidation phase, and a production phase. In the innovation phase, new evaluation resources are being developed and introduced for the first time, such as new corpora, new performance measures, and new technologies. The introduction of such new resources typically stirs up a lot of dust and is prone to errors and inconsistencies that may spoil evaluation results to some extent. This cannot be avoided, since only the use of new evaluation resources by many different parties will

[1] Some of the concepts found in this paper have been described earlier, so that, because of the inherently incremental nature of shared tasks, and in order for this paper to be self-contained, we reuse text from previous overview papers.

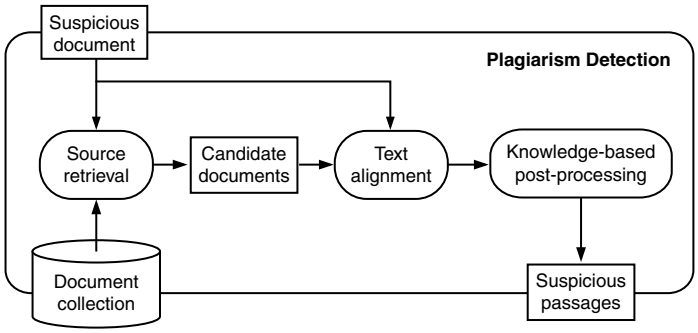

Figure 1. Generic retrieval process to detect plagiarism [25].

reveal their shortcomings. Therefore, the evaluation resources are released only sparingly so they last for the remainder of a cycle. This phase spanned the first and also to some extent the second year of the source retrieval subtask. In the consolidation phase, based on the feedback and results obtained from the first phase, the new evaluation resources are developed to maturity by making adjustments and fixing errors. This phase spanned the second and to some extent the third year of the source retrieval subtask. In the production phase, the task is repeated with little changes to allow participants to build upon and to optimize against what has been accomplished, and, to make the most of the prior investment in developing the new evaluation resources. Meanwhile, new ideas are being developed to introduce further innovation.

This third production phase in part could be observed in last year's third edition of the source retrieval subtask and was the motivation behind organizing the subtask for a fourth time. However, as will be described in more detail later, no real progress in the probably most important directions for a source retrieval method (i.e., improving the recall of reused sources and minimizing the effort until the first evidence for reuse is detected) can be observed in this year's submission. New querying strategies might be the key but no participant developed new ideas in that direction. Before further elaborating the different submitted approaches of this year's edition and the respective evaluation results, we describe the task setting and test environment in a self-contained way (note again that a lot of the respective passages have already been contained in the previous years' overview papers).

2 Setting this Overview's Scene

Terminology. Figure 1 shows a generic retrieval process to detect plagiarism in a given suspicious document d_{plg}, when also given a (very large) document collection D of potential source documents. This process is also referred to as *external* plagiarism detection since plagiarism in d_{plg} is detected by searching for text passages in D that are highly similar to text passages in d_{plg}.[2] The process is divided into three basic steps,

[2] Another approach to detect plagiarism is called *intrinsic* plagiarism detection, where detectors are given only one suspicious document and are supposed to identify text passages in it which deviate in their style from the remainder of the document.

which are typically implemented in most plagiarism detectors. First, source retrieval, which identifies a feasible set of candidate source documents $D_{\mathrm{src}} \subseteq D$ that are likely sources for plagiarism regarding d_{plg}—this is the problem tackled in the source retrieval subtask. Second, text alignment, where each candidate source document $d_{\mathrm{src}} \in D_{\mathrm{src}}$ is compared to d_{plg}, extracting all passages of text that are highly similar—this is the problem tackled in the text alignment subtask described in another overview paper. Third, knowledge-based post-processing, where the extracted passage pairs are cleaned, filtered, and possibly visualized for later inspection—this is not really reflected in the PAN plagiarism subtasks so far.

Shared Tasks on Plagiarism Detection. We have organized shared tasks on plagiarism detection annually since 2009. In the innovation phase of our shared task at PAN 2009 [21], we developed the first standardized evaluation framework for plagiarism detection [20]. This framework was consolidated in the second and third task at PAN 2010 and 2011 [13, 14], and it has since entered the production phase while being adopted by the community. Our initial goal with this framework was to evaluate the process of plagiarism detection depicted in Figure 1 as a whole. We expected that participants would implement source retrieval algorithms as well as text alignment algorithms and use them as modules in their plagiarism detectors. However, the results of the innovation phase proved otherwise, since participants implemented only text alignment algorithms, whereas they resorted to exhaustively comparing all pairs of documents within our evaluation corpora, even when the corpora were tens of thousands of documents large. To establish source retrieval as a shared task of its own, we introduced it at PAN 2012 next to the text alignment task [15], thus entering a new task life cycle for this task. We developed a new, large-scale evaluation corpus of essay-length plagiarism cases that have been written manually, and whose sources have been retrieved manually from the ClueWeb09 corpus [18]. Given our above observation from the text alignment task, the ClueWeb09 was deemed too large to be exhaustively compared to a given suspicious document in a reasonable time. Furthermore, we developed a new search engine for the ClueWeb09 called ChatNoir [17], which serves participants who do not wish to develop their own ClueWeb09 search engine as a means of participation. We then offered source retrieval as an individual task based on the new evaluation resources [15, 16], whereas this year marks the fourth time we do so, and the continuation of the source retrieval task's production phase.

Contributions. Since the source retrieval subtask probably now is in the production phase of its life cycle, we refrain from changing the existing evaluation resources too much, whereas we continue to maintain them. Therefore, our contributions this year consist of (1) a survey of this year's submitted approaches, which reveals that there are hardly any new trends among participants in the source retrieval subtask, and (2) an analysis of this years participants' retrieval performances in direct comparison to participants from previous years, which reveals that no real progress in the important direction of increased recall can be observed.

In this connection, our goal with both shared tasks is to further automate them. Hence, we continue to develop the TIRA evaluation platform [5, 6], which gives rise to software submissions with minimal organizational overhead and secures the execution of untrusted software while making the release of the test corpora unnecessary [4]. Like

last year, the fully-fledged web service as a user interface enables the participants to remote control their evaluations on the test corpus under our supervision. Within this framework, we will probably enable further evaluations of new approaches but probably we will for now refrain from organizing a fifth edition of a source retrieval subtask at PAN 2016.

3 Source Retrieval Evaluation Framework

In source retrieval, given a suspicious document and a web search engine, the task is to retrieve all source documents from which text has been reused whilst minimizing retrieval costs. The cost-effectiveness of plagiarism detectors in this task is important since using existing search engines is perhaps the only feasible way for researchers as well as small and medium-sized businesses to implement plagiarism detection against the web, whereas search companies charge considerable fees for automatic querying their APIs.

In what follows, we briefly describe the building blocks of our evaluation setup, provide brief details about the evaluation corpus, and discuss the performance measures (see the task overview from 2013 for more details on these three points [16]). We then survey the submitted softwares in Section 4, and finally in Section 5, report on their achieved results in this year's setup.

3.1 Evaluation Setup

For the evaluation of source retrieval from the web, we consider the real-world scenario of an author who uses a web search engine to retrieve documents in order to reuse text from them in their to-be-written text. A plagiarism detector typically uses a search engine, too, to find reused sources of a given document. Over the past years, we assembled the necessary building blocks to allow for a meaningful evaluation of source retrieval algorithms; Figure 2 shows how they are connected. The setup was described in much more detail in the task overview of 2013 [16].

Two main components are the TIRA experimentation platform and the ClueWeb09 with two associated search engines. TIRA [5] itself consists of a number of building blocks; one of them, depicted in Figure 2 bottom left, facilitates both platform independent software development and software submissions at the same time by its capability to create and remote control virtual machines on which our lab's participants deploy their source retrieval systems.

The ClueWeb corpus 2009 (ClueWeb09)[3] is one of the most widely adopted web crawls which is regularly used for large-scale web search-related evaluations. It consists of about one billion web pages, half of which are English ones. Although an updated version of the corpus has been released,[4] our evaluation is still based on the 2009 version since our corpus of suspicious documents was built on top of ClueWeb09. Indri[5] and

[3] http://lemurproject.org/clueweb09
[4] http://lemurproject.org/clueweb12
[5] http://lemurproject.org/clueweb09/index.php#Services

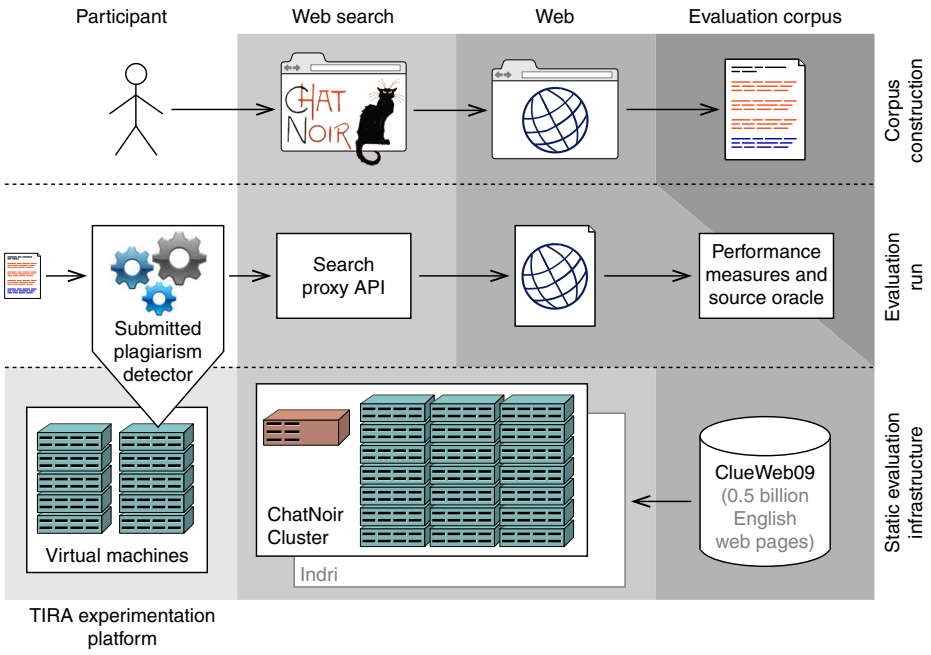

Figure 2. Overview of the building blocks used in the evaluation of the source retrieval subtask. The components are organized by the two activities corpus construction and evaluation runs (top two rows). Both activities are based on a static evaluation infrastructure (bottom row) consisting of an experimentation platform, web search engines, and a web corpus.

ChatNoir [17] are currently the only publicly available search engines that index the ClueWeb09 corpus; their retrieval models are based on language modeling and BM25F, respectively. For developer convenience, we also provide a proxy server which unifies the APIs of the search engines. At the same time, the proxy server logs all accesses to the search engines for later performance analyses.

3.2 Evaluation Corpus

The evaluation corpus employed for source retrieval is based on the Webis text reuse corpus 2012 (Webis-TRC-2012) [19, 18]. The corpus consists of 297 documents that have been written by 27 writers who worked with our setup as shown in the first row of Figure 2: given a topic, a writer used ChatNoir to search for source material on that topic while preparing a document of 5700 words length on average, reusing text from the found sources.

As in the last year, we use the same 98 documents from the Webis-TRC-2012 as training documents and another 99 documents are sampled as test documents—also the same as in 2014. The remainder of the corpus will be used within future source retrieval evaluations.

3.3 Performance Measures

Given a suspicious document d_{plg} that contains passages of text that have been reused from a set of source documents D_{src}, we measure the retrieval performance of a source retrieval algorithm in terms of precision and recall of the retrieved documents D_{ret} taking into account the effect of near-duplicate web documents as follows (cf. the 2013 task overview [16] for more details).

For any $d_{\text{ret}} \in D_{\text{ret}}$, we employ a near-duplicate detector to judge whether it is a true positive detection; i.e., whether there is a $d_{\text{src}} \in D_{\text{src}}$ of d_{plg} that is a near-duplicate of d_{ret}. We say that d_{ret} is a true positive detection for a given pair of d_{src} and d_{plg} iff (1) $d_{\text{ret}} = d_{\text{src}}$ (equality), or (2) the Jaccard similarity of the word n-grams in d_{ret} and d_{src} is above 0.8 for $n = 3$, above 0.5 for $n = 5$, and above 0 for $n = 8$ (similarity), or (3) the passages in d_{plg} known to be reused from d_{src} are contained in d_{ret} (containment). Here, containment is measured as asymmetrical set overlap of the passages' set of word n-grams regarding that of d_{ret}, so that the overlap is above 0.8 for $n = 3$, above 0.5 for $n = 5$, and above 0 for $n = 8$. This three-way approach of determining true positive detections inherently entails inaccuracies. While there is no straightforward way to solve this problem, this error source affects all detectors, still allowing for relative comparisons.

Let d_{dup} denote a near-duplicate of a given d_{src} that would be considered a true positive detection according to the above conditions. Note that every d_{src} may have more than one such near-duplicate and every d_{dup} may be a near-duplicate of more than one source document. Further, let D'_{src} denote the set of all near-duplicates of a given set of source documents D_{src} of d_{plg} and let D'_{ret} denote the subset of D_{src} that have at least one corresponding true positive detection in D_{ret}:

$$D'_{\text{src}} = \{d_{\text{dup}} \mid d_{\text{dup}} \in D \text{ and } \exists d_{\text{src}} \in D_{\text{src}} : d_{\text{dup}} \text{ is a true positive detection of } d_{\text{src}}\},$$
$$D'_{\text{ret}} = \{d_{\text{src}} \mid d_{\text{src}} \in D_{\text{src}} \text{ and } \exists d_{\text{ret}} \in D_{\text{ret}} : d_{\text{ret}} \text{ is a true positive detection of } d_{\text{src}}\}.$$

Based on these sets, we define precision and recall of D_{ret} regarding D_{src} and d_{plg} as follows:

$$prec = \frac{|D_{\text{ret}} \cap D'_{\text{src}}|}{|D_{\text{ret}}|}, \qquad rec = \frac{|D'_{\text{ret}} \cap D_{\text{src}}|}{|D_{\text{src}}|}.$$

Rationale for this definition is that retrieving more than one near-duplicate of a source document does not decrease precision, but it does not increase recall, either, since no additional source is obtained. A further graphical explanation of how we take near-duplicates into account for precision and recall is given in Figure 3. Note that D_{ret} as defined above does not actually contain all duplicates of the retrieved documents, but only those that are already part of D_{src}.

Finally, to measure the cost-effectiveness of a source retrieval algorithm in retrieving D_{ret}, we count the numbers of queries and downloads made and compute the workload in terms of queries and downloads until the first true positive detection is made (i.e., until the first real evidence for text reuse is found). The last measure highlights the probable end user needs that some evidence should be found fast in order to quickly flag such a suspicious document for a further detailed analysis.

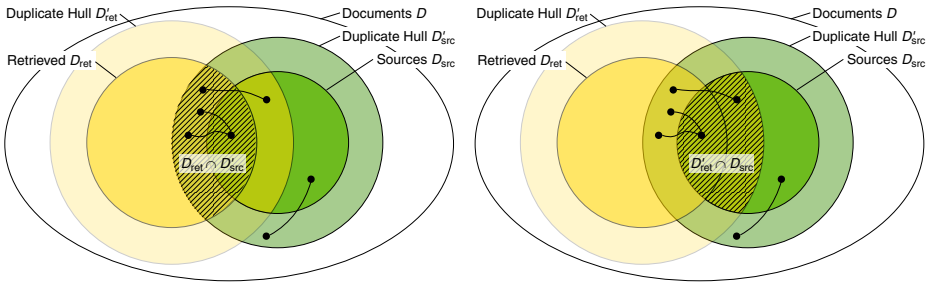

Figure 3. Effect of near-duplicates on computing precision (left) and recall (right) of retrieved source documents. Without taking near-duplicates into account, a lot of potentially correct sources might be missed.

The Source Oracle To allow for participation in the source retrieval task without the need of having a text alignment component at hand, we provide a source oracle that automatically enriches a downloaded document with information about whether or not it is considered a true positive source for the given suspicious document. Note that the oracle employs the aforementioned conditions to determine whether a document is a true positive detection. However, the oracle does not, yet, tell for which part of a suspicious document a downloaded document is a true positive detection. Hence, applying a custom text alignment strategy can still be beneficial to derive such a mapping and potentially to adjust the query strategy accordingly.

4 Survey of Retrieval Approaches Submitted for PAN 2015

Five teams submitted softwares for the source retrieval task, all of whom also submitted a notebook describing their approach—the approaches of Han [9] and Kong et al. [10] are described in the same notebook. An analysis of the individual descriptions reveals the same building blocks that were commonly used in last years' source retrieval algorithms: (1) chunking, (2) keyphrase extraction, (3) query formulation, (4) search control, and (5) download filtering. Some participants only slightly changed their approach from the previous year or adopted ideas from other approaches; in what follows, we describe the employed ideas in a little more detail.

Chunking Given a suspicious document, it is divided into (possibly overlapping) passages of text. Each chunk of text is then processed individually. Rationale for chunking the suspicious document is to evenly distribute "attention" over a suspicious document so that algorithms employed in subsequent steps are less susceptible to unexpected characteristics of the suspicious document.

The chunking strategies employed by this year's participants are no chunking (i.e., the whole document as one chunk) [26], 500-word chunks [22] (overlap size not detailed), paragraphs as chunks [23] (not detailed how paragraphs are split), paragraphs split at empty lines as chunks [26], or individual sentences and headings as chunks [10, 9].

Note that chunks typically seem to be implemented as non-overlapping by the participating approaches. The potentially interesting question of whether overlapping chunks might help was not really tackled by any approach so far—except that Suchomel and Brandejs [26] use different types of chunks in combination. A problem with non-overlapping longer chunks might be that typical plagiarism cases have no fixed length and overlapping chunks might reduce the risk of, for instance, having more than one source in one chunk of 500 words. Furthermore, relying on the given document structure (e.g., chunking by lines or paragraphs) bears the risk of failing for some unseen documents that are not as well-formatted as the ones in our evaluation corpus (e.g., all the text in a single line). Maybe mixed chunking strategies as seen in Suchomel and Brandejs [26]' approach is the most promising direction. Notably, their document level queries seem to also guarantee an early recall measured in queries (cf. Section 5) and mixed chunking probably at least will not decrease total recall.

Keyphrase Extraction Given a chunk, "keyphrases" are extracted from it in order to formulate queries with them. Rationale for keyphrase extraction is to select only those phrases (or words) which maximize the chance of retrieving source documents matching the suspicious document. Keyphrase extraction may also serve as a means to limit the amount of queries formulated, thus reducing the overall costs of using a search engine. This step is perhaps the most important one of a source retrieval algorithm since the decisions made here directly affect the overall performance: the fewer keywords are extracted, the better the choice must be or recall is irrevocably lost.

Some participants use single words while others extract whole phrases. Most of the participants preprocessed the suspicious document by removing stop words before the actual keyphrase extraction. Phrasal search was provided by the Indri search engine. All participants did use Indri when submitting phrasal queries; some of which also combine phrases with non-phrasal ChatNoir queries, the search engine that the original essay authors had used. In particular, Han [9] and Kong et al. [10] extract nouns and verbs according to the Stanford POS-tagger as their keywords. Rafiei et al. [22] use the words with highest $tf \cdot idf$ scores from a chunk. The idf is derived from the PAN 2011 corpus although it is not really clear why this would be a good choice since it contains rather different documents than a web corpus and most of the documents are artificially created with some more or less "random" word distributions. Ravi N and Gupta [23] also use single words as keywords (verbs, nouns, and adjectives according to the NLTK Python package) and also score by $tf \cdot idf$ (without further details on how idf is computed). Suchomel and Brandejs [26] apply a similar strategy as in their previous years' approaches: they also use the highest scoring $tf \cdot idf$ terms where idf is computed from a 4 billion word collection they also used in the last year.

Altogether, the participants' approaches to "keyphrase extraction" are more or less simplistic and do not really rely on established keyphrase extraction techniques from the NLP community. Giving several such established methods a try at least on document level might be an interesting direction for some more general keyphrases. Queries from such phrases in combination with queries from "extracted keyphrases" or just single words from shorter chunks might be a promising direction. Some first steps in this direction are shown in the method of Suchomel and Brandejs [26] but might still be enhanced. Not relying on just one strategy alone but combining different keyphrase

extraction ideas might ensure a higher recall. This way, just as with chunking, the risk of algorithm error is further diminished and it becomes possible to exploit potentially different sources of information that complement each other.

Query Formulation Interestingly, most of the participants hardly put effort in finding good keyphrase combinations as queries. Instead, typically the top-k $tf \cdot idf$-ranked terms form the first query, then the next k terms, etc. This way, mostly non-overlapping queries are generated for the individual chunks. This non-overlap-approach is in line with many query-by-document strategies [1, 3] but in contrast to previous source retrieval strategies that were shown to better identify highly related documents using overlapping queries from several keyphrase combinations [8]. Also note that hardly any of the participants made use of advanced search operators offered by Indri or ChatNoir, such as the facet to search for web pages of at least 300 words of text, and the facet to filter search results by readability.

In particular, Han [9] and Kong et al. [10] formulate one query from the keywords extracted per sentence (at most 10 words in a query as a threshold). Rafiei et al. [22] employ a rather involved scheme of first identifying the 10 words with highest $tf \cdot idf$ scores for each chunk, then selecting three sentences with these keywords of the respective chunk and then formulating queries from the keywords in these sentences until some not-detailed maximum is reached. Then they also seem to formulate individual queries for every sentence. Ravi N and Gupta [23] formulate two queries per paragraph from the respective top-n $tf \cdot idf$ terms (without further details on how n is chosen). Suchomel and Brandejs [26] apply a similar strategy as in their previous years' approaches: queries with 6 terms on the document level, phrase queries from their collocations, and then individual queries with the 10 highest scoring $tf \cdot idf$ terms per chunk.

Search Control Given sets of keywords or keyphrases extracted from chunks, queries are formulated which are tailored to the API of the search engine used. Rationale for this is to adhere to restrictions imposed by the search engine and to exploit search features that go beyond basic keyword search (e.g., Indri's phrasal search). The maximum number of search terms enforced by ChatNoir is 10 keywords per query while Indri allows for longer queries.

Given a set of queries, the search controller schedules their submission to the search engine and directs the download of search results. Rationale for this is to dynamically adjust the search based on the results of each query, which may include dropping queries, reformulating existing ones, or formulating new ones based on the relevance feedback obtained from the search results. Han [9] and Kong et al. [10] download 100 results per query but the downloads seem to start only after all queries were submitted. Rafiei et al. [22] drop a query when more than 60% of its terms are contained in an already downloaded document. Ravi N and Gupta [23] omit duplicate queries without giving further details on what constitutes a duplicate. Suchomel and Brandejs [26] apply a similar strategy as in their previous years' approaches: they schedule queries dependent on the keyphrase extractor which extracted the words. The order of precedence corresponds to the order in which they have been explained above. Whenever later queries were formulated for chunks of the suspicious document that were already mapped to a source, these queries are not submitted and discarded from the list of open

queries. Depending on how many plagiarism sources are contained in a paragraph-long chunk, this might potentially miss some further sources when for instance two sources were used and one was already found.

Note that still (just as in the last years) none of the teams did try to reformulate existing queries or formulating new ones based on the available number of search results, the search snippets, or the downloaded documents, which probably leaves room for substantial improvement. Another interesting aspect might be the scheduling of the queries themselves. The experimental results (cf. Section 5) seem to suggest that some document-level queries in the first submission positions guarantee an early recall with respect to the number of submitted queries (e.g., Suchomel and Brandejs [26]). Simply scheduling queries in the order of chunks in the documents instead, might run into problems with early recall as maybe there is not that much reused text at the beginning of a document. This might also be an interesting point for future research that none of the approaches has investigated so far.

Download Filtering Given a set of search engine results, a download filter removes all documents that are probably not worthwhile being compared in detail with the suspicious document. Rationale for this is to further reduce the set of candidates and to save invocations of the subsequent detailed comparison step. Many of the participants in the last years and also this year seem to focus a little too much on that part of a source retrieval system. This can be seen in the not improved overall recall and also in the number of submitted queries till the first evidence of text reuse is found compared to the approaches from 2013. Another evidence is that some approaches hardly download ten documents per suspicious document. In this case, the download filtering needs to be almost optimal which puts a higher burden on the search strategy. As downloads probably are not the most important bottleneck and probably queries are more costly in a general environment, reducing downloads too much does not seem to be the most promising strategy. A text alignment system is probably easily able to compare a suspicious document against even several thousands of potential sources in a couple of minutes. Maybe downloading more documents and putting more effort on the formulation of good queries is a promising future area.

In this year, Han [9] and Kong et al. [10] focus on the top-100 results of a query and download them depending on the number of queries for which they appear. This basically means that all queries are submitted before any download and that downloads have no influence on potential query scheduling etc. This strategy obviously has a high number of queries submitted until the first evidence is found since basically all queries are submitted before any evidence can be found. Han [9] additionally seems to use some learning approach similar to Williams et al. [27] employing snippet features but do not provide many details. Note that this learning approach seems to be the only difference between the two approaches of Kong et al. [10] and Han [9]. Rafiei et al. [22] download at most the top-14 results of a query when the individual snippets (no length information) contain at least 50% of the query terms. Ravi N and Gupta [23] download a document when the snippet (no length information) has a high cosine similarity to the document chunk for which the query was generated. However, there are no details on the similarity threshold or on how many documents are checked per query. Suchomel and Brandejs [26] apply a similar strategy as in their previous years' approaches: they

Table 1. Source retrieval results with respect to retrieval performance and cost-effectiveness.

Software Submission Team (alphabetical order)	Year	Downloaded Sources			Total Workload		Workload to 1st Detection		No Detect.	Runtime
		F_1	Prec.	Rec.	Queries	Dwlds	Queries	Dwlds		
Elizalde	2013	0.16	0.12	0.37	41.6	83.9	18.0	18.2	4	11:18:50
Elizalde	2014	0.34	0.40	0.39	54.5	33.2	16.4	3.9	7	04:02:00
Foltynek	2013	0.11	0.08	0.26	166.8	72.7	180.4	4.3	32	152:26:23
Gillam	2013	0.06	0.04	0.15	**15.7**	86.8	16.1	28.6	34	**02:24:59**
Haggag	2013	0.38	**0.67**	0.31	41.7	**5.2**	13.9	**1.4**	12	46:09:21
Han	2015	0.36	0.55	0.32	194.5	11.8	202.0	1.7	12	20:43:02
Kong	2013	0.01	0.01	**0.59**	47.9	5185.3	**2.5**	210.2	**0**	106:13:46
Kong	2014	0.12	0.08	0.48	83.5	207.1	85.7	24.9	6	24:03:31
Kong	2015	0.38	0.45	0.42	195.1	38.3	197.5	3.5	3	17:56:55
Lee	2013	0.40	0.58	0.37	48.4	10.9	6.5	2.0	9	09:17:10
Prakash	2014	0.39	0.38	0.51	60.0	38.8	8.1	3.8	7	19:47:45
Rafiei	2015	0.12	0.08	0.41	43.5	183.3	5.6	24.9	1	08:32:37
Ravi N	2015	0.43	0.61	0.39	90.3	8.5	17.5	1.6	8	09:17:20
Suchomel	2013	0.05	0.04	0.23	17.8	283.0	3.4	64.9	18	75:12:56
Suchomel	2014	0.11	0.08	0.40	19.5	237.3	3.1	38.6	2	45:42:06
Suchomel	2015	0.09	0.06	0.43	42.4	359.3	3.3	39.8	4	161:51:26
Williams	2013	**0.47**	0.60	0.47	117.1	12.4	23.3	2.2	7	76:58:22
Williams	2014	**0.47**	0.57	0.48	117.1	14.4	18.8	2.3	4	39:44:11
Zubarev	2014	0.45	0.54	0.45	37.0	18.6	5.4	2.3	3	40:42:18

obtain snippets for at most 100 results per query and download documents when more than 20% of the word 2-grams in the concatenated snippets also appear in the suspicious document.

Some participants download very few documents based on their filtering while others download more documents per query than most participants did in the last years. As described above, the second option seems to be more promising with respect to recall since downloads typically are not that costly. An further interesting option for future research might be based on the User-over-Ranking hypothesis [24, 7] taking into account some goal number of candidate documents against which a detailed text alignment can be performed after the source retrieval phase of plagiarism detection.

5 Evaluation Results

Table 1 shows the performances of the five plagiarism detectors that took part in this year's source retrieval subtask as well as those of the last years' participants whose approaches were also evaluated on this year's test corpus using the TIRA experimentation platform. Since there is currently no single formula to organize retrieval performance and cost-effectiveness into just one absolute score or order, the detectors are ordered alphabetically, whereas the best performance value for each metric is highlighted—note that these individual "best" performances should be compared to the other metrics as for instance the fastest approach also has the highest number of no detections etc. As can be seen, there is no single detector that performs best on all accounts. Rather, different detectors have different characteristics.

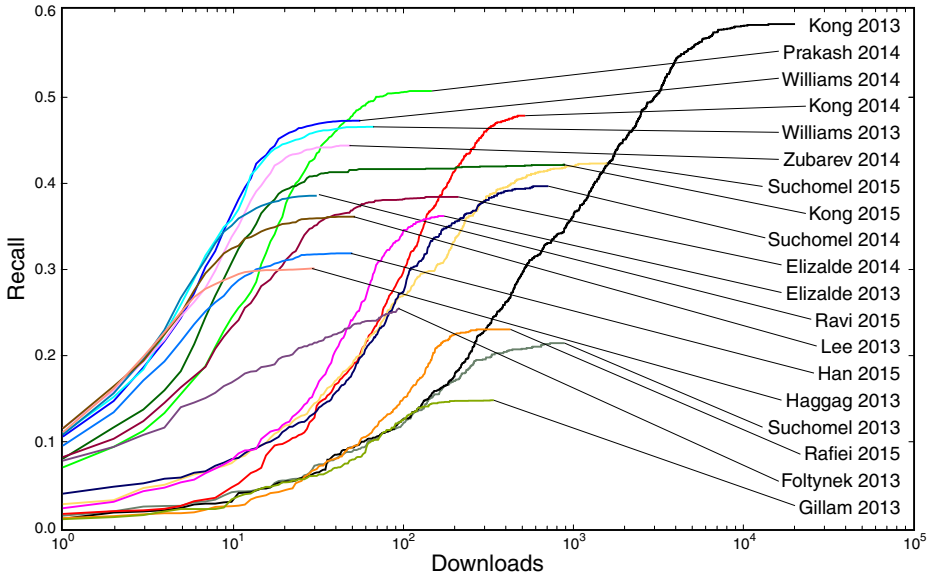

Figure 4. Recall at a specific number of downloads per participant averaged over all topics.

Arguably, highest possible recall at a reasonable workload (queries and downloads) is the goal of source retrieval while, at the same time, it would be nice to quickly detect some first evidence for text reuse if there is one. Since downloads in most environments would be much cheaper than query submissions (that would accompanied by costs at most major web search engine's APIs) the most interesting metrics probably are: recall and number of no detections, number of submitted queries, and number of queries and downloads until the first detection.

Focusing on recall first, interestingly the top-6 approaches are from previous years with the best one by far still being the approach of Kong et al. [11] (at some high costs in number of queries and downloads and a poor precision, though). Thus, after last year's progress in the overall recall of most systems, this year the participating systems seem not to be able to actually increase their recall substantially. To further shed some light on the recall of the different approaches, Figure 4 shows the recall against the number of downloaded documents. It can be seen that recall is typically gained over the whole process of downloading documents and not with the very first downloads (the plateau effect at the upper right end of each plot is due to the averaging). Unsurprisingly, some of the low-workload approaches achieve higher recall levels with fewer downloads while approaches with more downloads typically achieve their better final recall levels only at a much higher number of downloads—which still can be good depending on probably rather low practical costs for downloads.

The ensemble of all submitted approaches of the last three years would achieve an average recall of 0.85 retrieving all sources for 48 topics. Only for 14 topics the recall is below 0.6 (which is the best individual average recall). These numbers did not change

from the ensemble of the 2013 and 2014 approaches also indicating that this year's methods do not contain real innovations with respect to recall-oriented source retrieval.

A per-participant analysis also reveals some interesting observations when comparing the approaches from different years. For instance, after doubling the recall in the last year, Suchomel and Brandejs [26] were only able to slightly increase their recall this year with way more query and downloading effort. Rafiei et al. [22] and Ravi N and Gupta [23] manage to enter the competition with medium recalls while Kong et al. [10] and Han [9] could not reach their still state-of-the-art recall from the 2013 edition.

As some further not-just-recall-oriented observations, for no metric, any of this year's participants achieved the best performance. This is not too surprising given the fact that no real "innovations" are contained in this year's submissions. They are rather very similar to methods that participated in the last year such that no "radical" change could be expected. Still, some notable achievements can be observed for the five participants of this year's competition. Rafiei et al. [22] in their first year manage to find sources for all but one of the suspicious documents. Also Ravi N and Gupta [23] enter the competition with a very good result: their number of downloads till the first detection is almost the top-performing one and is a little better than the also very good one of Han [9]; however, both probably download too few documents overall to achieve a good recall. The number of submitted queries of Suchomel and Brandejs [26] is still very good taking into account the achieved recall. Still, a recall of "only" 0.42 suggests that they might still not make the most of their interesting idea of combining more general with more focused queries.

Altogether, the current strategies might still be a little too focused on saving downloads (and queries) compared to for instance increase the recall. Also runtime should probably not be the key metric to optimize (e.g., using threads instead of sequential processing does not decrease the actual costs for using the search engines). A reasonable assumption probably is that recall is most important to the end user of a source retrieval system. Investing a couple of queries and a couple of downloads (maybe even hundreds to thousands) to achieve a recall above 0.8 might be a very important research direction since still none of the participating approaches can reach such levels. In the end, whatever source the source retrieval step misses, cannot be found by a later text alignment step. This probably is a key argument for a recall-oriented source retrieval strategy that also takes into account basic considerations on total workload of query submissions and downloads. It would be interesting to see efforts in that direction of substantially improved recall at a moderate cost increase in future approaches.

Another interesting direction from a practical standpoint is to find at least one source for a document that contains text reuse and to report a first detection as early as possible. This way the real end user of a detection system could focus on the really important suspicious documents very quickly and could even deepen the search depth or increase the allowed number of queries without the fear of missing text reuse in too many suspicious documents. However, this probably should be future work when better overall recall levels are reached.

6 Conclusion and Outlook

Altogether, even though new participants entered the source retrieval subtask this year, the ideas underlying the approaches did not contain "real" innovations compared to the approaches from the previous years since mostly the same ideas are just reused. Even though the source retrieval subtask now is expected to be in the production phase of its shared task life cycle—it is well-defined and all evaluation resources are set up and provide for a challenging testbed—most of the approaches still struggle to retrieve at least 50% of the text reuse sources with no real progress on the recall side this year. The combined ensemble of all approaches would result in a really good recall but at a rather high cost in the number of queries and downloads. None of this year's approaches really tried to work towards that recall at reduced costs resulting in the described "lack" of innovation.

Without really new ideas, the task does not seem to be interesting enough for another edition. Instead, we are planning to continue to pursue automating source retrieval evaluations without the requirement of an organized shared task. The key is the development of the TIRA experimentation platform [6] that facilitates software submissions, where participants submit their plagiarism detection software to be evaluated at our site [4]. The web front end of TIRA allows any researcher to conduct self-service evaluations on the test data of the source retrieval task under our supervision and guidance, whereas the test data remains hidden from direct access from participants.[6] This has enabled us to put the participants back in charge of executing their software while the software itself remains in a running state within virtual machines managed by TIRA. Based on this technology, we conduct cross-year evaluations of all source retrieval systems that have been submitted since 2013. This platform will be further available for comparison against the state of the art in source retrieval but a new edition of the source retrieval subtask at PAN might probably not happen next year.

Acknowledgements

We thank the participating teams of the editions of the source retrieval subtask for their shared ideas and for their devoted work towards making their softwares run on TIRA.

Bibliography

1. Bendersky, M., Croft, W.: Finding Text Reuse on the Web. In: Baeza-Yates, R.A., Boldi, P., Ribeiro-Neto, B.A., Cambazoglu, B.B. (eds.) Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009. pp. 262–271. ACM (2009)
2. Cappellato, L., Ferro, N., Jones, G., San Juan, E. (eds.): CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France. CEUR Workshop Proceedings, CEUR-WS.org (2015), http://www.clef-initiative.eu/publication/working-notes

[6] www.tira.io

3. Dasdan, A., D'Alberto, P., Kolay, S., Drome, C.: Automatic retrieval of similar content using search engine query interface. In: Cheung, D.W.L., Song, I.Y., Chu, W.W., Hu, X., Lin, J.J. (eds.) Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009. pp. 701–710. ACM (2009)
4. Gollub, T., Potthast, M., Beyer, A., Busse, M., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Recent Trends in Digital Text Forensics and its Evaluation. In: Forner, P., Müller, H., Paredes, R., Rosso, P., Stein, B. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 4th International Conference of the CLEF Initiative (CLEF 13). pp. 282–302. Springer, Berlin Heidelberg New York (Sep 2013)
5. Gollub, T., Stein, B., Burrows, S.: Ousting Ivory Tower Research: Towards a Web Framework for Providing Experiments as a Service. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). pp. 1125–1126. ACM (Aug 2012)
6. Gollub, T., Stein, B., Burrows, S., Hoppe, D.: TIRA: Configuring, Executing, and Disseminating Information Retrieval Experiments. In: Tjoa, A.M., Liddle, S., Schewe, K.D., Zhou, X. (eds.) 9th International Workshop on Text-based Information Retrieval (TIR 12) at DEXA. pp. 151–155. IEEE, Los Alamitos, California (Sep 2012)
7. Hagen, M., Stein, B.: Applying the User-over-Ranking Hypothesis to Query Formulation. In: Advances in Information Retrieval Theory. 3rd International Conference on the Theory of Information Retrieval (ICTIR 11). Lecture Notes in Computer Science, vol. 6931, pp. 225–237. Springer, Berlin Heidelberg New York (2011)
8. Hagen, M., Stein, B.: Candidate Document Retrieval for Web-Scale Text Reuse Detection. In: 18th International Symposium on String Processing and Information Retrieval (SPIRE 11). Lecture Notes in Computer Science, vol. 7024, pp. 356–367. Springer, Berlin Heidelberg New York (2011)
9. Han, Y.: Submission to the 7th International Competition on Plagiarism Detection. http://www.uni-weimar.de/medien/webis/events/pan-15 (2015), http://www.clef-initiative.eu/publication/working-notes, From the Heilongjiang Institute of Technology
10. Kong, L., Lu, Z., Han, Y., Qi, H., Han, Z., Wang, Q., Hao, Z., Zhang, J.: Source Retrieval and Text Alignment Corpus Construction for Plagiarism Detection—Notebook for PAN at CLEF 2015. In: [2]
11. Kong, L., Qi, H., Du, C., Wang, M., Han, Z.: Approaches for Source Retrieval and Text Alignment of Plagiarism Detection—Notebook for PAN at CLEF 2013. In: Forner, P., Navigli, R., Tufis, D. (eds.) CLEF 2013 Evaluation Labs and Workshop – Working Notes Papers, 23-26 September, Valencia, Spain (Sep 2013)
12. Potthast, M.: Technologies for Reusing Text from the Web. Dissertation, Bauhaus-Universität Weimar (Dec 2011), http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:gbv:wim2-20120217-15663
13. Potthast, M., Barrón-Cedeño, A., Eiselt, A., Stein, B., Rosso, P.: Overview of the 2nd International Competition on Plagiarism Detection. In: Braschler, M., Harman, D., Pianta, E. (eds.) Working Notes Papers of the CLEF 2010 Evaluation Labs (Sep 2010), http://www.clef-initiative.eu/publication/working-notes
14. Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B., Rosso, P.: Overview of the 3rd International Competition on Plagiarism Detection. In: Petras, V., Forner, P., Clough, P. (eds.) Working Notes Papers of the CLEF 2011 Evaluation Labs (Sep 2011), http://www.clef-initiative.eu/publication/working-notes
15. Potthast, M., Gollub, T., Hagen, M., Graßegger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th

International Competition on Plagiarism Detection. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) Working Notes Papers of the CLEF 2012 Evaluation Labs (Sep 2012), http://www.clef-initiative.eu/publication/working-notes
16. Potthast, M., Gollub, T., Hagen, M., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., Stein, B.: Overview of the 5th International Competition on Plagiarism Detection. In: Forner, P., Navigli, R., Tufis, D. (eds.) Working Notes Papers of the CLEF 2013 Evaluation Labs (Sep 2013), http://www.clef-initiative.eu/publication/working-notes
17. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A Search Engine for the ClueWeb09 Corpus. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). p. 1004. ACM (Aug 2012)
18. Potthast, M., Hagen, M., Völske, M., Stein, B.: Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In: Fung, P., Poesio, M. (eds.) Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 13). pp. 1212–1221. Association for Computational Linguistics (Aug 2013), http://www.aclweb.org/anthology/P13-1119
19. Potthast, M., Hagen, M., Völske, M., Stein, B.: Exploratory Search Missions for TREC Topics. In: Wilson, M.L., Russell-Rose, T., Larsen, B., Hansen, P., Norling, K. (eds.) 3rd European Workshop on Human-Computer Interaction and Information Retrieval (EuroHCIR 2013). pp. 11–14. CEUR-WS.org (Aug 2013), http://www.cs.nott.ac.uk/ mlw/euroHCIR2013/proceedings/paper3.pdf
20. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An Evaluation Framework for Plagiarism Detection. In: Huang, C.R., Jurafsky, D. (eds.) 23rd International Conference on Computational Linguistics (COLING 10). pp. 997–1005. Association for Computational Linguistics, Stroudsburg, Pennsylvania (Aug 2010)
21. Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., Rosso, P.: Overview of the 1st International Competition on Plagiarism Detection. In: Stein, B., Rosso, P., Stamatatos, E., Koppel, M., Agirre, E. (eds.) SEPLN 09 Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse (PAN 09). pp. 1–9. CEUR-WS.org (Sep 2009), http://ceur-ws.org/Vol-502
22. Rafiei, J., Mohtaj, S., Zarrabi, V., Asghari, H.: Source Retrieval Plagiarism Detection based on Noun Phrase and Keyword Phrase Extraction—Notebook for PAN at CLEF 2015. In: [2]
23. Ravi N, R., Gupta, D.: Efficient Paragraph based Chunking and Download Filtering for Plagiarism Source Retrieval—Notebook for PAN at CLEF 2015. In: [2]
24. Stein, B., Hagen, M.: Introducing the User-over-Ranking Hypothesis. In: Advances in Information Retrieval. 33rd European Conference on IR Resarch (ECIR 11). Lecture Notes in Computer Science, vol. 6611, pp. 503–509. Springer, Berlin Heidelberg New York (Apr 2011)
25. Stein, B., Meyer zu Eißen, S., Potthast, M.: Strategies for Retrieving Plagiarized Documents. In: Clarke, C., Fuhr, N., Kando, N., Kraaij, W., de Vries, A. (eds.) 30th International ACM Conference on Research and Development in Information Retrieval (SIGIR 07). pp. 825–826. ACM, New York (Jul 2007)
26. Suchomel, Šimon., Brandejs, M.: Improving Synoptic Quering for Source Retrieval—Notebook for PAN at CLEF 2015. In: [2]
27. Williams, K., Chen, H.H., Giles, C.: Supervised Ranking for Plagiarism Source Retrieval—Notebook for PAN at CLEF 2014. In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (eds.) CLEF 2014 Evaluation Labs and Workshop – Working Notes Papers, 15-18 September, Sheffield, UK. CEUR Workshop Proceedings, CEUR-WS.org (Sep 2014)