Supporting More-Like-This Information Needs: Finding Similar Web Content in Different Scenarios

Matthias Hagen and Christiane Glimm

Bauhaus-Universität Weimar
<first name>.<last name>@uni-weimar.de

Abstract We examine more-like-this information needs in different scenarios. A more-like-this information need occurs, when the user sees one interesting document and wants to access other but similar documents. One of our foci is on comparing different strategies to identify related web content. We compare following links (i.e., crawling), automatically generating keyqueries for the seen document (i.e., queries that have the document in the top of their ranks), and search engine operators that automatically display related results. Our experimental study shows that in different scenarios different strategies yield the most promising related results.
One of our use cases is to automatically support people who monitor right-wing content on the web. In this scenario, it turns out that crawling from a given set of seed documents is the best strategy to find related pages with similar content. Querying or the related-operator yield much fewer good results. In case of news portals, however, crawling is a bad idea since hardly any news portal links to other news portals. Instead, a search engine's related operator or querying are better strategies. Finally, for identifying related scientific publications for a given paper, all three strategies yield good results.

1 Introduction

The problem considered in this paper appears whenever a user browsing or searching the web finds a document with interesting content for which she wants to identify related pages on the web. Search engines often support such information needs by providing specific operators (e.g., "`related:`" + a URL in the Google interface or the "Related articles"-link in GoogleScholar). However, in a scenario that we discussed with researchers monitoring extreme right-wing content on the web, both these possibilities failed in a pilot study. We thus examine different possibilities for finding related pages on the web with one scenario being the described monitoring. In this case, the classic crawling strategy works very well and besides automatic query formulation should form the heart of a an automatic system that supports the monitoring people.

In our study, we examine three different strategies that can be used to automatically find and suggest related documents from the web. The first idea is to simply use the available technology from search engine side (i.e., the mentioned related operators). As these are probably mainly build on click-through information in case of web pages or citation analysis and click-through information in case of scholarly articles, the search engine side techniques do not work in all scenarios. Hence, we compare the available related-operators to classic crawling-like link acquisition and to automatically generated queries. The link following strategy will prove extremely useful in case of connected networks like extreme right-wing web pages. Querying is implemented as a standard technique human users would choose to search the web. To this end, we employ the recent idea of keyqueries. A keyquery for a document is a query that returns the document in the top ranks. Other top ranked documents of that query are probably very related (as they appear for the same query) and thus good candidates to be presented to the user.

In our experimental evaluation, we compare the three strategies (crawling, querying, engine operators) for different realistic scenarios. First, we conduct a study on web pages containing extreme right-wing content. For this scenarios, the search engine operators perform not that well as probably not much click-through information is available and respective queries are probably not the main focus of commercial search engines. Instead, the link crawling works very well since typically extreme right pages are well connected on the web.

In contrast, link crawling does not yield satisfying results in the second part of our study. Namely, for news pages using the search engine related-operator performs best (potentially due to a lot of click-through and content analysis at search engine side). Also queries perform better in this case than crawling. A third scenario evaluates the three strategies on scientific publications. Here link crawling is modeled as following citations and references. Still the search engine related operator and automatic queries perform similarly well. Hence, the three scenarios contrast different use cases for which different strategies have to be applied. For the important case of providing an automatic tool that supports people who monitor extreme right-wing content on the web, traditional crawling-style techniques are the best choice.

The paper is organized as follows. In Section 2 we briefly review related work on finding similar documents. The detailed description of the examined approaches follows in Section 3. Our experimental study with the three different usage scenarios is described in Section 4. Some concluding remarks and an outlook on future work closes the paper in Section 5.

2 Related Work

We briefly describe several approaches that aim at finding similar content on the web. Note that we do not address the case of duplicate or near-duplicate detection—and thus also do not review the respective literature here. Rather, the focus of our study is on finding different documents with similar content—the “more like this”-scenario.

Classic approaches to identify content from the web are crawling strategies. Given some seed set of URLs, a crawler tries to identify links in the seeds and fetch the respective documents, then links in the new documents are identified, etc. [2]. We follow a very similar approach but only follow links from one given page (the source of the more-like-this need) and also do not crawl the entire part of the web reachable from that document (cf. Section 3 for more details). Note that for papers, similar ideas are to follow citations to and from a given paper to identify related publications. The SOFIA search tool [7] extracts references from a paper and by weighting author groups and topic words in paper titles, extracts a set of publication that are suggested as related to the source document. Since the prototype of SOFIA search is not available, we implemented a basic similar strategy that "crawls" the references and citations (cf. Section 3 for more details).

For another source of related documents that we exploit, no literature exists. In particular, we are using the Google operator `related` that for a given URL returns up to about 200 related web pages. It is probably based on click-through information from a search engine and page content analysis. A probably similar approach is described by Lee et al. [9] who identify related news stories by observing queries and clicks in a new search engine. However, concrete details about Google's `related`-operators are not available online such that we use the system as a "black box." In case of scientific papers, we use the "related articles" and "cited by" functionality offered by GoogleScholar as a replacement of the `related` operator (cf. Section 3 for more details).

Different studies have proposed to derive queries for a given document and use the queries to retrieve similar documents. Fuhr et al. [5] build a theoretical framework for optimum clustering (OCF) based on not comparing document-term-vectors but vectors of document-query similarities. Based on a set of predefined queries, documents with similar similarities for these queries would be grouped in the same cluster. One way of storing the important queries for a document is the reverted index presented by Pickens at al. [13]. Different to the traditional inverted indexes used in most IR systems that basically store for a given term, which documents contain the term, the reverted index stores for each documents for which queries it is returned (weights would correspond to the document's rank in the result set). Initially planned as a means for query expansion, the reverted index could also be applicable to store the queries used in the OCF. Our idea of assuming relatedness of documents returned for the same query builds up on the OCF proposal. Also a couple of previous query formulation strategies to identify documents with similar content on the web are very related to the ideas in Fuhr et al.'s paper.

For instance, Bendersky and Croft deal with the scenario of text reuse detection on the web [3]. Different to previous approaches that deal with text reuse on small-scale corpora, their focus is on reuse of single sentences on the web (but not on complete documents as input). As web-scale prohibits several previous reuse detection strategies, Bendersky and Croft suggest a querying strategy to identify other documents with occurrences of very similar sentences. They also try to identify which of the found documents was the earliest and to analyze information flow which is not our topic. Even though reused text is one form of similarity, our scenario is much different. Still the developed basic query formulation strategy inspired later work that we will employ.

In our setting it would be desirable to use the given document as a query itself ("query by document"). Yang et al. [15] focus on such a scenario in the context of analyzing blog posts. They also try to derive a keyword query that reflects the document's (blog post's) content. Their approach extracts keyphrases from the document, but formulates only a single query from them—backed up by knowledge from Wikipedia and different sources. In contrast, our query formulation will be based on keyphrases instead of words—which was shown beneficial in later studies. Furthermore, Yang et al.'s approach requires to manually select the number of "good" keywords for each document which is not applicable in a fully automatic system.

A more applicable setting which is also related to ours is Dasdan et al.'s work on finding similar documents by using only a search engine interface [4]. Although Dasdan et al. focus on a search engine coverage problem (resolve whether a search engine's index contains a given document or some variant of it), their approach of finding similar documents using keyword interfaces is very related to our setting. Dasdan et al. propose two querying strategies and experimentally show that their approaches indeed find similar documents. However, a later study by Hagen and Stein [8] showed that other query formulation strategies yield even better results. Similar to the text reuse scenario of Bendersky and Croft, Hagen and Stein try to identify potential source documents from the web for text reuse in a given suspicious document. They show that keyphrases are better components for good automatic queries than single keywords. Their proposed strategy also does not formulate just a single query but a whole collection whose combined results are used in the end. Hagen and Stein show their strategy to be much more effective than previous strategies while also being comparably efficient.

In a later paper, the idea of Hagen and Stein is refined to so-called keyqueries [6]. A keyquery for a given document returns the given document in the top ranks while also retrieving other documents. The query is then viewed as very descriptive for the given document's content (since the document is in the top ranks) and since also the other top ranked documents are retrieved, the query probably also is very descriptive for their content. Following Fuhr et al.'s OCF framework and previous query formulation papers, the keyquery's results in some sense then are the most related documents for the input. We will employ the keyquery technique in our query formulation strategy (cf. Section 3 for more details).

3 Approach

In this section, we describe the employed strategies for finding similar content web pages. The classical approach of following links (i.e., crawling) is contrasted by search engine provided related-operators, that we employ as a "black box" due to the lack of publically available information on their inner methods, and an automatic query formulation based on keyqueries (i.e., queries that return a given document in the top of their ranks).

3.1 Link crawling

Following links to crawl documents from the web is a classic building block of modern web search engines [2]. The typical implementation extracts hyperlinks from the main content of found web pages —for main content detection we use the boilerpipe[1] library—and then fetches the respective documents. In case of web pages, we simply employ this basic strategy, but only collect links that point to pages on other domains. We thus differentiate between internal links (same domain) and external links (different domain). The underlying assumption is that probably same-domain pages are rather similar and that more interesting pages (especially in the right-wing monitoring scenario) are pages from different domains. The found external links are added to a standard crawler frontier (i.e., a queue) that returns unseen links in a FIFO manner. Crawling

[1] https://code.google.com/p/boilerpipe/, last accessed may 13, 2014

was stopped when 200 external links where retrieved—due to a limitation of 200 results in case of Google's related-operator (see below).

In case of scientific articles, just extracting web links from the documents is not the best choice. Instead, in this case, links are formed by citations and references. For reference extraction from a given paper, we employ the ParsCit[2] tool and the GoogleScholar search for finding citing papers. Differentiating between internal and external links for papers could be modeled by author overlap. However, as a searcher would probably also be interested in related papers from the same author group we simply crawl all papers.

Note that some implementation issues arise for non-available links, password-protected pages, or for differentiating between internal and external links in case of usage of virtual hosts. However, as these issues are not the focus of this paper, we usually just ignored non-available or password protected links and simply checked the URL-strings in case of doubts about internal or external nature.

3.2 Search engine related-operator

As a representive of commercial search engines, we use the Google search. Google provides a related-operator as part of its query language.[3] A query `related:`+URL returns pages that are similar to the given URL. There is no information about the inner method of the operator but it probably is based on clickthrough information (i.e., people with similar queries clicking on differnt URLs) and a bit of page content analysis. In a pilot study with people monitoring extreme right-wing content on the web, we observed that for such content the related-operator often did not bring up any results. This is probably in part also due to the fact that in Germany pages promoting hate speech have to be removed from the index—still not all right-wing content pages actually contain hate speech. The lack of support from Google's related operator for monitoring right-wing content was one of the driving inspirations of the presented study. In contrast to right-wing content, for prominent domains like news portals, the related-operator works very well. This also underpins the assumption that clickthrough is an important signal since big news portals probably are much more prominent web pages that right-wing content; resulting in more available clickthrough. Typically, when the related-operator does provide results, the returned list has a length of about 200 entries. Most of the top entries then also are results for related-queries on each others domains.

As for scientific articles we employ the "Related articles" link from the search engine result page that basically provides the same functionality as the related-operator from the main Google page. In this case, the operator might also be based on clickthrough and content analysis but citations (i.e., linking) probably play the biggest role. In this sense, the GoogleScholar related-operator should produce similar results as link crawling for papers (which basically is following references and citations, see above).

3.3 Keyqueries

The keyqueries concept was introduced by Gollub et al. [6]. Basically, a keyquery for a given document d is a query that returns d in the top ranks but also returns other

[2] http://aye.comp.nus.edu.sg/parsCit/, last accessed may 13, 2014

[3] http://www.googleguide.com/advanced_operators_reference.html, last accessed May 13, 2014

documents besides d and thus is not too specific. The original idea is to represent documents by their keyqueries. In our scenario, we will employ keyqueries to identify related content—namely the other results from the top ranks besides d. The underlying assumption is that documents returned in the top ranks for the same queries cover very similar content, similar to the OCF assumption [5].

More formally, given the vocabulary $W_d = \{w_1, w_2, \ldots, w_n\}$ of a document d, let $\mathcal{Q}_d$ denote the family of search queries that can be formulated from W_d without word repetitions; i.e., $\mathcal{Q}_d$ is the power set of W_d, $\mathcal{Q}_d = 2^{W_d}$. Note that no distinction is made with respect to the ordering of the words in a query. If it is clear from the context, we omit the subscripts and just use W and $\mathcal{Q}$ to denote the vocabulary and the potential queries from d.

A query $q \in \mathcal{Q}$ is a *keyquery* for d with respect to a reference search engine S iff: (1) d is among the top-k results returned by S on q, and (2) no subset $q' \subset q$ returns d in its top-k results when submitted to S. The parameter k controls the level of keyquery generality and is usually set to some small integer, such as 10 in our case. Let $\mathcal{Q}^*$ denote the set of keyqueries for d.

As in the original paper, we form keyqueries from keyphrases extracted from a document's text via the TextRank algorithm [10]. TextRank basically forms a graph with the words in a text as its vertices and edges between vertices when the words are neighbors in the text (after stopword removal). On the graph, in a PageRank style computation [12], weights for the vertices are computed and after convergence phrases are formed from neighboring heavy weight vertices.

Contrary to Gollub et al.'s original approach [6] that uses the Apriori algorithm [1] to find the family $\mathcal{Q}^*$ of all keyqueries for a given document, we employ a simpler gready search to find a handful of keyqueries from the top 12 keyphrases extracted by TextRank. We first try the first phrase, then add the next phrases as long as the desired document is not in the top k ranks. Whenever the document is in the top ranks, we try to find a keyquery starting with the second phrase etc. From the found keyqueries, we use the top-k documents such that 200 documents are fetched—similar to the Google related operator that always presents about 200 documents when successful. For instance, in case of four found keyqueries, the top-50 documents from each form the final result set. Compared to the exhaustive Aprior search, our gready approach significantly reduces the number of queries submitted.

4 Evaluation

Having presented the applied approaches for finding related documents on the web, we develop an empirical evaluation based on the following hypothesis. The first hypothesis was formed in a pilot study with people monitoring extreme right-wing content on the web and also is in line with related research on the web structure of right-wing communities [11].

Hypothesis 1: The link crawling strategy is a good choice for highly connected communities like web pages containing extreme right-wing content. For less connected related pages like different news portals, link crawling is not the best choice.

Hypothesis 2: The related-operator is a good choice for frequently visited web pages like news portals while documents with less traffic and much more specific content are not well-covered.

Hypothesis 3: Keyqueries as an automatic query formulation strategy are a good backup strategy whenever some other technique does not perform well enough.

To test our hypothesis and to again emphasize the use case of monitoring extreme right-wing content, our evaluation corpus consists of four different parts (two for extreme right-wing content, two other). Each part contains documents available on the web for which the three strategies described in the previous sections each are run to identify related content. In a last step, human annotators evaluate the quality of the returned documents with respect to their relatedness to the input corpus document.

The first two parts of the corpus are formed by German weblogs (part 1) and web pages (part 2) with extreme right-wing background. These pages form the use case of people monitoring the web for extreme right-wing content to study for instance information spread or to protect young people from seeing the content. The pages in part 1 of our corpus are mined from public German sources collecting extreme right-wing weblogs from less organized right-wing structures. Part 2 pages are formed by web pages of the German extreme right-wing party NPD (a more organized and publically viewable player). With these two different right-wing page types (weblogs and NPD pages), we want to evaluate two different standard use cases that people monitoring such content on the web have. Typically, they manually find such pages, follow links and submit queries. Our study on the first two parts of our corpus should give a first idea of whether such a behavior can be semi- or fully automated.

To contrast the rather “niche”-style pages in the first two parts of our evaluation corpus, we also include public German and English news portals as a third part. Finally, the fourth part is more research oriented as it aims to examine to what extend the search for related work or similar scientific articles can be automated or at least semi-automatically supported. We thus include scientific articles from the field of information retrieval as the fourth part in our corpus.

Each part of the corpus is formed by 25 documents (100 in total). For each document, each of the described three strategies identifies 200 related documents when possible. Two human annotators subjectively classified a sample of 20 pages for each source document following some rough guidelines as “related” or as “not related.” Thus a sample of 20 out of at most 200 results for each of the 100 corpus documents was classified. Note that the sampling favored top retrieved documents from the 200 potential ones (the top-10 were always included); however, lower ranked results did have a small probability of also being sampled for classification.

In case of disagreement among the two annotators, a short discussion was arranged. Whenever the two annotators did not agree even after discussion, the result was labeled as “no consensus.” Note that in general this case did not occur too often such that most of the cases have a consensus—the exception being right-wing weblogs that often probably somewhat try to hide their real “orientation” such that our annotators had a tough task for these cases.

Table 1. Classification of the link crawling results.

Corpus part	**Classification**			**Total classified**
	related	not related	no consensus	
Right-wing Blogs	173	70	248	491
NPD web pages	289	24	14	327
News portals	18	443	39	500
Scientific publications	216	140	109	465

4.1 Individual classification results

We first show the individual performances of the different strategies before we compare them on the whole corpus and check the validity of our initial hypotheses.

Link crawling Table 1 contains the classification results for the link crawling strategy. Each line corresponds to a specific part of our evaluation corpus. The classification columns show how many of the retrieved documents were classified as related or not by our assessors. The last column shows the total number of classified results. Two interesting observations are striking.

First, not for all the 100 source documents even 20 related results could be identified by crawling. The lowest number is achieved for NPD web pages. On average, only 13 pages were found by link crawling (remember that we are only interested in external links such that other NPD pages do not count).

Second, for extreme right-wing blogs, our annotators faced a tough task depicted by the many results for which no classification consensus could be reached. Still the ratio of related to not-related pages is very good for right-wing documents. As expected, for news portals, link crawling does not yield many related pages. Again, this is not too surprising as typically different news portals do not link to each other—probably in order not to lose their readers.

Google related Table 2 contains the classification results for the strategy employing Google's related operator. Interestingly, the related operator does not work at all for the extreme right-wing blogs. One reason could be a policy of removing hate speech content from display. Another explanation based on the assumption that the related-operator is based on query click-through information, is that there is not much available

Table 2. Classification of the Google related results.

Corpus part	**Classification**			**Total classified**
	related	not related	no consensus	
Right-wing Blogs	0	0	0	0
NPD web pages	237	136	13	386
News portals	406	91	3	500
Scientific publications	262	138	25	425

Table 3. Classification of the keyqueries results.

Corpus part	**Classification**			**Total**
		not	no	**classified**
	related	related	consensus	
Right-wing Blogs	52	219	37	308
NPD web pages	0	0	0	0
News portals	0	0	0	0
Scientific publications	82	107	71	260

in the logs. This might show that not much traffic is lead to such pages via Google—which would be a very good sign in our opinion. One further reason probably also is the volatility of the respective blogs that often change their URLs etc. For the other right-wing corpus documents (the NPD pages) the related-operator does produce acceptable results, however, returning rather many not-related documents—but also here not for all corpus documents at least 20 related ones could be identified.

As for the news portals, the assumed underlying click-through information really shows its power. More than 80% (406 out of 500) of the returned results are relevant.

Keyqueries Table 3 contains the classification results for the keyquery strategy. The most striking observations are the failure to produce keyqueries for NPD pages and news portals. For all the corpus documents in these groups no keyqueries could be computed. The reason was not that Google did not return any results due to some treatment of hate speech removal. Instead, typically, the query containing all the 12 extracted keyphrases still did not return the corpus document in its top 10 results—a sign that the phrases are very generic—or even short combinations of only few keyphrases did only return the single corpus document—a sign that the phrases in combination are too specific. In case of news portals, even short queries typically are very specific as they contain non-related phrases from different news stories shown on the news portals' starting pages. Such queries are very specific and often did not yield any other result. In case of the NPD pages, often also the full query containing all the keyphrases was to generic not showing the particular corpus page in the top 10 results such that no keyquery could be computed from the 12 extracted keyphrases. Adding more phrases in this case might help but would harm the comparability with the results on other classes. As for the right-wing blogs, the results are not really satisfying with a lot of results no related to their respective source document.

The case of scientific publications is the strongest for keyqueries among the four different parts of our evaluation corpus. Still, only about 10 documents were found on average and a little more not-related results were classified. One frequent reason (50 of the 71 cases) for the no-consensus decision in this case were access-restricted portals from which our assessors could not acquire a pdf of the proposed document.

4.2 Comparison and Hypotheses' Validity

As can be seen from the classification results each of the three techniques has its individual strengths and weaknesses. The link crawling strategy is the best among the tested techniques for extreme right-wing content with a high ratio of related pages found. This confirms our first hypothesis formed with people monitoring extreme right-wing web content on a daily basis. In such scenarios of tightly connected networks, simply following links that also often dynamically point to moved content is the best choice.

Our second hypothesis that Google related has its strengths on frequently visited pages also is clearly confirmed by comparing the results on the news portals. Here, about 80% of related documents found is way ahead of the other techniques.

Our third hypothesis stating that for cases where the others fail is not really confirmed by our experiments. Still, for scientific publications, the keyquery strategy shows some promising results but for news portals or NPD pages completely fails. Thus, the third hypothesis can only partly be confirmed but for pages with diverse content (as news portals or NPD pages are) the hypothesis is falsified.

In total, our results clearly show that the choice of a strategy for finding related content often heavily depends on the input document. In case of our focus use case of finding related right-wing content and building a semi-automatic tool to assist people who monitor such pages, the classic idea of following links still clearly beats advanced search engine features like the related-operator or automatic query formulation strategies based on keyqueries.

4.3 Further Observations: Overlap and Efficiency

We could observe an interesting effect when we compared the overlap of the retrieved related documents for the different techniques in the different parts of our corpus. For each two techniques and each corpus category, the overlap of the found related results was at most 10% (often much lower). This means that the different techniques are somewhat complementary to each other and find different related results. Whenever the results of one technique do not yield enough similar documents another technique can be used as a backup; of course, probably only for corpus categories where it retrieves something related. For instance, for our use case of finding related right-wing content, crawling is the best standalone technique but can be backed up by Google-related for NPD pages or keyqueries for blogs since the retrieved related results of that techniques complement the crawling results very well.

As for runtime, using the search engine built-in operators by far is the fastest approach. Crawling links comes with the timing issues that crawling usually exhibits. This includes politeness—not fetching to often from the same server—and also latency—waiting for server responses. Thus crawling usually is slower than a search engine operator. Automatic query formulation was the slowest approach since even our simple greedy strategy submits about 50–80 queries on average to identify the final keyqueries for a document. With the available interfaces of commercial search engines submitting these queries costs a significant amount of time—submitting too many queries simultaneously or in short time frames may even result in blocking from search engine side.

5 Conclusion and Outlook

In this paper, we have examined different strategies of finding related content for a given document on the web. Our primary use case emerged from discussions with people monitoring extreme right-wing content on the web. They would like to have an (semi-)automatic tool that retrieves related pages from the web that they then can examine without the burden of retrieval.

We compared three different approaches. Namely, classic link crawling, using Google's related-operator, and automatic query formulation with the recent keyqueries approach. Our evaluation corpus consists of four different parts aiming at examining the three retrieval strategies on different scenarios. In the two first corpus parts, we focus on extreme right-wing content in the form of weblogs and pages from the German NPD party. The third part consists of popular news portals while the fourth part is formed by scientific publications—a use case of particular interest to ourselves as researchers.

Our experimental study is guided by three main hypotheses. The first hypothesis states that link crawling is a particularly good choice for finding related content in scenarios of tightly connected networks. This hypothesis was formed form observations of the people monitoring extreme right-wing content on the web—an example of a volatile but very connected network.

Our second hypothesis is based on the assumption that Google heavily uses click-through information for its related-operator; it states the Google's related-operator should particularly perform well for frequently visited pages but has lower performance for rather unpopular pages like extreme right-wing content. Also this hypothesis could clearly be confirmed.

Our third hypothesis that automatic keyqueries are a good backup when the other techniques might fail, can only be confirmed for scientific publications. Interestingly, for news portals or NPD pages, the keyqueries technique did not retrieve any results since no keyqueries could be computed. The reason often being too specific or too general queries.

Altogether, our results clearly show that the input document's characteristic is an important signal for choosing the "best" strategy of retrieving related content from the web. In the cases represented in our corpus, different strategies have clear strengths and weaknesses for different document characteristics. Thus, an automatic classification and choice of a good strategy for a given input document is an interesting task in the direction of building an automatic related content finder. The work by Qi and Davison [14] might be a good starting point for classifying the input document. Still, in some cases, like our focus topic of monitoring right-wing content, the keyqueries and Google-related complement the crawled results very well as they find different related results (when they find any).

Interesting directions for future work would be a large-scale study of the observed effects. Our corpus consists of only 100 documents (25 for each of the four scenarios) and only 20 potential results were judged by two assessors whether they are related. A large-scale study should contain hundreds of documents for each scenario probably also including different use cases. We are currently evaluating to what extend such relatedness judgments can be crowdsourced—one ethical issue being the extreme right-wing content for two important parts of our corpus that might not be appropriate for potential assessors.

In order to build a semi-automatic system that supports people monitoring extreme right-wing content, also the recall of the strategies is an important but difficult to estimate issue. So far, the link crawling strategy has the lowest rate of false positives in these cases (while Google-related has the lowest false positive rate for news portals). In order to further reduce the number of false positives presented to the user, machine learning classifiers could be trained for different scenarios that are able to detect the retrieve not-related documents. Research in that direction would probably further smooth the user experience of using the semi-automatic crawling strategy.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of VLDB 1994. pp. 487–499.
2. Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the web. ACM Trans. Internet Technol. 1(1), 2–43 (2001)
3. Bendersky, M., Croft, W.B.: Finding text reuse on the web. In: Proceedings of WSDM 2009. pp. 262–271.
4. Dasdan, A., D'Alberto, P., Kolay, S., Drome, C.: Automatic retrieval of similar content using search engine query interface. In: Proceedings of CIKM 2009. pp. 701–710.
5. Fuhr, N., Lechtenfeld, M., Stein, B., Gollub, T.: The optimum clustering framework: Implementing the cluster hypothesis. Information Retrieval 15(2), 93–115 (2011)
6. Gollub, T., Hagen, M., Michel, M., Stein, B.: From keywords to keyqueries: Content descriptors for the web. In: Proceedings of SIGIR 2013. pp. 981–984.
7. Golshan, B., Lappas, T., Terzi, E.: SOFIA search: A tool for automating related-work search. In: Proceedings of SIGMOD 2012. pp. 621–624.
8. Hagen, M., Stein, B.: Candidate document retrieval for web-scale text reuse detection. In: Proceedings of SPIRE 2011. pp. 356–367.
9. Lee, Y., Jung, H.y., Song, W., Lee, J.H.: Mining the blogosphere for top news stories identification. In: Proceedings of SIGIR 2010. pp. 395–402.
10. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of EMNLP 2004. pp. 404–411.
11. O'Callaghan, D., Greene, D., Conway, M., Carthy, J., Cunningham, P.: Uncovering the wider structure of extreme right communities spanning popular online networks. In: Proceedings of WebSci 2013. pp. 276–285.
12. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999).
13. Pickens, J., Cooper, M., Golovchinsky, G.: Reverted indexing for feedback and expansion. In: Proceedings of CIKM 2010. pp. 1049–1058.
14. Qi, X., Davison, B.D.: Web page classification: Features and algorithms. ACM Comput. Surv. 41(2), 12:1–12:31 (2009)
15. Yang, Y., Bansal, N., Dakka, W., Ipeirotis, P., Koudas, N., Papadias, D.: Query by document. In: Proceedings of WSDM 2009. pp. 34–43. (2009)