World Wide Web (2016) 19:449–474
DOI 10.1007/s11280-015-0331-7

Finding seeds to bootstrap focused crawlers

**Karane Vieira · Luciano Barbosa ·
Altigran Soares da Silva ·
Juliana Freire · Edleno Moura**

Received: 23 June 2014 / Revised: 28 November 2014 /
Accepted: 9 January 2015 / Published online: 26 February 2015

Abstract Focused crawlers are effective tools for applications requiring a high number of pages belonging to a specific topic. Several strategies for implementing these crawlers have been proposed in the literature, which aim to improve crawling efficiency by increasing the number of relevant pages retrieved while avoiding non-relevant pages. However, an important aspect of these crawlers has been largely overlooked: the selection of the seed pages that serve as the starting points for a crawl. In this paper, we show that the seeds can greatly influence the performance of crawlers, and propose a new framework for automatically finding seeds. We describe a system that implements this framework and show, through a detailed experimental evaluation, that by providing crawlers a seed set that is large and varied, they not only obtain higher harvest rates but also an improved topic coverage.

Keywords Web crawling · Focused crawling · Relevance feedback

Electronic supplementary material The online version of this article (doi:10.1007/s11280-015-0331-7) contains supplementary material, which is available to authorized users.

K. Vieira · A. S. da Silva (✉) · E. Moura
Instituto de Computação, Universidade Federal do Amazonas, Manaus, Brazil
e-mail: alti@icomp.ufam.edu.br

K. Vieira
e-mail: karane@icomp.ufam.edu.br

E. Moura
e-mail: edleno@icomp.ufam.edu.br

L. Barbosa
IBM Research - Brazil, Rio de Janeiro, Brazil
e-mail: lbarbosa@research.att.com

J. Freire
Department of Computer Science and Engineering, New York University, New York, USA
e-mail: juliana.freire@nyu.edu

1 Introduction

Focused crawling has emerged as an effective strategy to locate specific information on the Web and it is essential for many important applications, such as gathering data for community information systems, and vertical search engines such as product catalogs. While generic crawlers used by search engines such as Google and Bing cover a substantial portion of the Web, they have important limitations when it comes to delivering specific information. Their keyword-based interfaces do not support queries that express complex information needs such as locating resources relevant to a topic (e.g., biology, movie) or that contain a specific object (e.g., a Web form, an artist's biography). In addition, because search engines limit the number of queries and search results returned, it may not be possible to retrieve a large collection of pages. Another limitation comes from the fact that search engines aim to obtain *breadth* but at the same time, due to resource limitations, they cannot download all the pages on the Web. As a result, pruning techniques are used and pages that might be important to a topic may be missed.

Instead of attempting to cover all pages on the Web, a focused crawler tunes its traversal strategy based on a target topic and tries to maximize the number of on-topic pages it retrieves while minimizing the number of non-relevant pages visited. This task is challenging because not only can topics be sparsely distributed over the Web graph, but often, among the billions of Web pages, there are relatively few pages for any given topic. Several focus strategies have been proposed that attempt to address these problems [1–3, 6, 11, 13, 16, 19, 20]. For example, to avoid unproductive regions of the Web, some techniques filter pages based on their contents [7, 11] while others learn patterns in URL paths that are likely to lead to pages containing a given concept [2].

But while much attention has been devoted to crawling strategies, the problem of selecting the seeds, which serve as the starting points for the crawl, has been largely overlooked. In this paper, we show that crawler efficiency and effectiveness can be improved through the selection of an appropriate seed set. The intuition for the importance of the seeds stems from the fact that even though most Web pages are highly connected [4], the topic graph induced by a focused crawler can be highly disconnected. Because focused crawlers prune the search space and avoid pages that are off-topic, they naturally create gaps in the Web

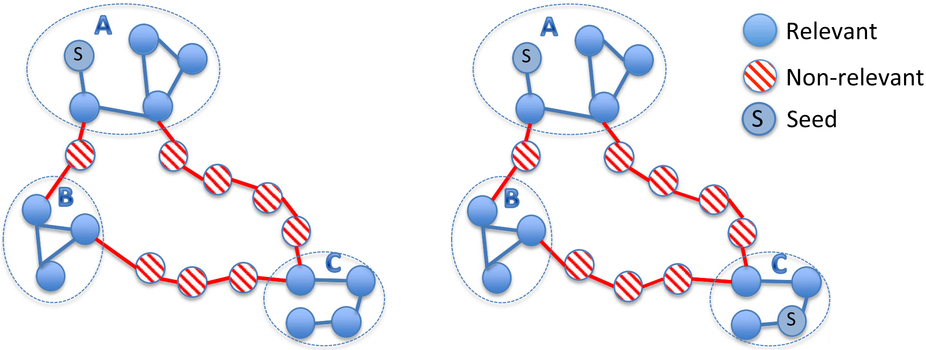

Figure 1 Connected components of a topic graph may be far apart, connected through paths of non-relevant pages

graph, leading to a series of *connected components* that contain pages belonging to the crawler's target topic. Depending on how pages within a specific topic are distributed, these components can be far apart.

As Figure 1 (left) illustrates, components can be connected through a long series of off-topic, non-relevant pages. This has two important implications: starting from component A, a crawler may not reach component C, leading to reduced coverage or it may take too long to do so, negatively affecting its harvest rate. We posit that, by selecting a suitable set of seeds, we can overcome, or at least mitigate this problem. For example, in Figure 1 (right), by selecting a seed that belongs to component C, the pages in C can be covered by the crawler, and at a lower cost—without the need to navigate through a potentially long series of non-relevant pages. As we discuss in Section 2, the importance of seed selection is also supported by studies that have examined the properties of the Web graph.

To configure a focused crawler, it is customary to provide a handful of seeds that are often manually selected by users. This can be problematic given the users inevitable limited knowledge of the Web graph. For some topics, seeds can be obtained from Web directories such as DMOZ.[1] But this strategy is ineffective for topics that are underrepresented (i.e., with few URLs), not precisely represented (i.e., not on the exact topic), or that simply do not exist in directories (e.g., regional or emerging topics).

We propose a new framework for seed selection that takes advantage of the high number of pages already crawled by general-purpose search engines, such as Google or Bing. In contrast to the common practice of feeding crawlers with a few dozen seeds [1–3, 7–9, 11, 13, 19], our framework aims to *automatically* construct an extensive seed set. Based on this framework, we have developed and implemented a system, called *BFC* (*Bootstrapping Focused Crawlers*), which constructs and issues queries to a search engine in a principled way to obtain a diverse and representative set of seeds. Because we do not know *a priori* all the terms that represent (and cover) a topic, *BFC* applies pseudo-relevance feedback [10] to iteratively compose queries and gather seeds in the search results. In addition, we have no control over or knowledge about the strategy to derive and rank the results obtained from the search engine. Thus, *BFC* makes use of a classifier to select among the returned URLs the ones that are more likely to be productive seeds. Besides simplifying the process of crawler configuration, by using an automated process, *BFC* is able to gather a high number of seed pages, doing so with a very modest overhead: (1) only a few queries are required to obtain several relevant results; (2) due to the adaptive query-generation strategy it adopts, the great majority of the results obtained are either relevant or are close to relevant pages, i.e., relevant pages can be reached from them in a few hops.

While use of search engines in focused crawling has been previously considered for constructing digital libraries [18, 24], in these studies, the search engines were used to simply provide links to scientific papers on a given subject [18] or by a given author [24]. Our framework, on the contrary, is general. It aims at improving any focused crawler by providing a high number of good seeds in any topic being sought.

We have performed an extensive experimental evaluation in which we used two well-known focused crawling strategies [7, 9] and a representative set of topics. We ran crawls that were at least five times larger than other crawls reported in the literature for these

[1] http://dmoz.org

topics. As we discuss in Section 5, the extended seed set derived by *BFC* leads to considerable improvement in performance and as well as in coverage. Contrary to previous findings [6, 7], we show that given different seed URLs, crawlers do navigate to different regions of the Web. In particular, for topics that are very sparse, there is a marked increase in coverage. For denser topics, while the gains in coverage are less striking, the crawler is able to reach relevant pages faster, leading to a noticeable gain in harvest rate. Our experiments also reinforce the usefulness of focused crawlers for finding *hard-to-find* content on the Web: a considerable number of pages obtained in our crawls were not present in the index of the search engine we used. Another interesting finding was related to the use of directories such as DMOZ. We report results that show that not only can BFC obtain a higher number of links than what is available in such directories, but it also obtains seeds that are closer to on-topic pages. Furthermore, some seeds obtained by BFC could not be reached from the seeds harvested from the directory.

Contributions To the best of our knowledge, this is the first paper that addresses the problem of seed selection for focused crawlers. Our main contributions can be summarized as follows:

- We introduce the problem of seed selection for focused crawlers and empirically verify that they can benefit from large and diverse sets of seed URLs;
- We propose a framework that harvests seeds in the page collection available in search engines. Based on this framework, we developed a fully automated system for generating seeds that requires only a modest overhead for the crawling process;
- We experimentally show, using different crawling strategies and topics, that this system is effective and leads to substantial gains in coverage and harvest rate.

2 Focused crawling: background

The main components of a typical focused crawler and their operation are illustrated in Figure 2. Before the crawling process starts, a model that encodes the notion of relevance of a page must be generated. This model, the *Page Model*, is used by the *Relevance Classifier*. Typically, the features learned by this classifier are related to the contents of the pages, e.g., the terms that are representative of the topic [1–3, 7, 9, 11, 13, 19]. However, features related to the structure of the page can also be considered [20]. When content-based features are used, training examples are usually obtained from pre-existing topic taxonomies such as *Yahoo!* or *DMOZ* and from the users themselves. In [7], for instance, the user is required to select from a taxonomy those classes that best represent the topic of interest. Users can also adapt the original taxonomy by rearranging classes or manually providing instances (i.e., pages) to populate classes.

Also prior to starting the crawling process, a set of *Seed Pages* must be selected. These pages are used by the *Crawler* as the starting points for the crawl. These are usually on-topic pages from which, according to the user's judgment, many other on-topic pages can be reached. The choice of representative seeds is crucial for focused crawlers, as we discuss below.

During the crawling process, for each page retrieved by the crawler, the *Relevance Classifier* assigns a relevance value. Based on the determination made by this classifier, pages

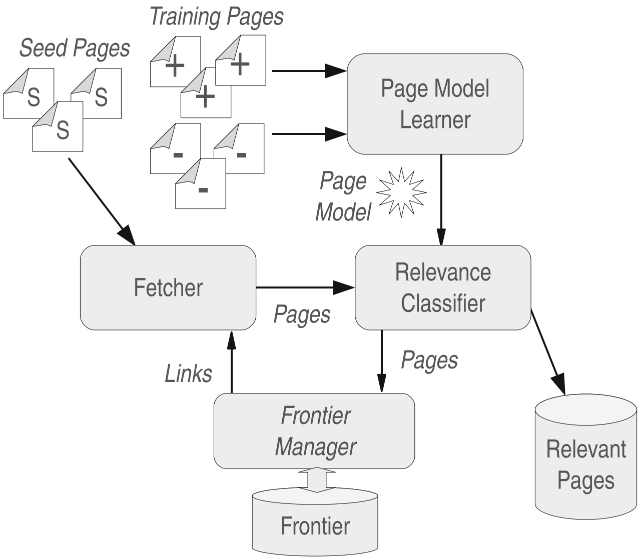

Figure 2 Main components of a typical Focused Crawler

considered as relevant to the topic are stored. In addition, the links from these pages are extracted and sent to the *Frontier Manager*, where the unvisited links are added to the *frontier*. In some focused crawlers, the links in the frontier are visited based on the estimated relevance of the pages that contains the link [7] (e.g., according to Relevance Classifier). More sophisticated strategies [1, 2, 9, 13, 19] consider link-specific features to estimate the importance of a link.

2.1 The importance of seeds

Although previous approaches [17, 22, 23] have proposed seed selection methods for general Web crawlers, this problem has been largely overlooked in the focused crawling literature. Here, we argue that the effectiveness of a crawler, regardless of the search strategy it uses, can be improved if a large enough set of (good) seeds is provided. We base our discussion on principles that have been implicitly observed in previous studies on focused crawling, but that can be better justified by analyzing certain properties on the Web graph. A number of results on such properties were presented in [12] and [8]. Here, we are particularly interested in the results related to subgraphs of the Web composed only of pages on a given topic, which we summarize below.

Let G be a directed subgraph of the Web whose nodes are pages on a given topic and the edges are the links between these pages. We call them *topic graphs* or *t-graphs*. According to [12] and [8], the number of out-links of the pages in t-graphs follows a Zipfian distribution. This means that most of the pages, i.e v., those in the heavy tail of the distribution curve, have only a few out-links to other pages on the same topic. Another interesting property of t-graphs is that the size of their connected components also follows a Zipfian distribution [8]. Consequently, t-graphs have a few large and many small connected components. This implies that, if a crawler reaches one component, it may not be able to reach another component by traversing only through on-topic pages.

Chakrabarti et al. [8] noted that isolated components in a t-graph are connected by a *navigational backbone*, which is common to the whole Web. This means that, to go from one component to another, a crawler must navigate through a potentially high number of off-topic pages. In Figure 1, we illustrate some of these properties by means of a very simple example of t-graph G. This t-graph has 3 connected components, represented by ellipses. Within each component, gray circles represent relevant (on-topic) pages and circles labeled with “S” represent seeds. Each connected component is linked to off-topic pages, represented as hashed circles. Now, consider a focused crawler that receives as input the seed within component A. By using a naive strategy where only links that lead to other pages on the topic are selected, a focused crawler would only collect pages within A. More sophisticated strategies, such as the ones proposed in [2, 9, 11], would take the crawler to other components by estimating the benefit of fetching off-topic pages that can lead to relevant pages. However, if the path of non-relevant pages needed to reach a relevant component were too long, even crawlers based on such strategies would miss important pages. This is likely to occur in topics whose pages are more sparsely distributed over the Web. This is illustrated in Figure 1: pages in component C, which is “far away” from component A, are less likely to be visited than pages on a “closer” component such as B.

Given more seeds as input, a crawler is likely to cover a higher number of connected components in G. For crawlers that attempt to reach beyond the on-topic pages, providing a single additional seed within a “far way” component, say C, might be enough to cover the entire t-graph. A large seed set that covers many of the components can be especially beneficial for very sparse topics.

In all cases illustrated above, providing more seeds improves the coverage of the t-graph obtained by some crawler. However, even if we consider that a focused crawler is able to traverse all components A, B and C, it would need to traverse several off-topic pages. For this case, providing seeds within the components would help to improve *crawler efficiency*, which is often evaluated in terms of its *harvest rate* [7], i.e., the rate at which relevant pages are acquired.

Based on the observations above, the ideal seed set should be diverse, covering different regions of the Web graph where pages related to the topic being sought are found.

Seed selection and taxonomies Previous approaches to focused crawling used Web taxonomies as sources for obtaining seeds [7, 9]. However, taxonomies have two important shortcomings. First, if the topic of interest is underrepresented and there are very few or no pages at all which belong to the topic, there will be insufficient support for guiding the crawler. This is most likely to occur for new, emergent topics (e.g., information about “H1N1”). Second, pages on a given topic may match only partially one or more classes on the taxonomy. For instance, pages on “Ornamental Fish From Amazon” are spread among many classes in DMOZ. For such cases, as discussed in [7], users have to modify the original taxonomy to accommodate this requirement.

3 A framework for finding seeds

Our approach for obtaining seeds on a given topic is based on the observation that a large portion of the Web has already been crawled by general-purpose search engines. By issuing queries related to the topic, it is possible to retrieve relevant pages that can serve as seeds. However, applying such an approach involves a number of challenges. First, we do not know in advance all the terms that describe the topic of interest and that could be used to compose the queries. To address this problem, we apply *Pseudo-Relevance Feedback* [10]: we start with an initial, simple query composed of a couple of terms clearly related to the topic, and, based on an analysis of the retrieved documents, select new terms that can be used to compose a new query.

Another challenge stems from the fact that we do not have access to a whole collection of documents in the search engine index. Thus, the judgment of the relevance of the documents is made by the search engine's own ranking algorithm. This means that, even if a high number of documents are retrieved by a query, not all of them are necessarily suitable to be used as seeds. Indeed, in virtually any ranking method there is a precision decay with growth in recall. In our framework, we use a classifier that filters from the documents retrieved by the query, those that are more likely to be related to the topic and be good seeds. We call this classifier *topic filter*. This classifier is also useful to prevent query drift that sometimes occurs with a pseudo-relevance feedback process, which can then use the results of the classifier to select terms for the new query. Indeed, Cao et al. [5] showed that the retrieval effectiveness of pseudo-relevance feedback methods can be improved when a classifier is used for selecting relevant documents, instead of simply selecting the top-K documents retrieved, as is done in traditional pseudo-relevance feedback approaches [10].

In practice, the number of results a search engine returns is only a fraction of the entire set of the documents in the answer. Thus, if a significant number of seeds are to be obtained, it is better to submit several small queries than a single long one. This strategy is also justified by the precision decay mentioned above. For instance, submitting 10 short queries with relevant terms and retrieving 100 answers from each of these queries is likely to lead to more relevant pages than submitting a long query and retrieving 1000 answers from it. Based on this observation, we adopt an iterative pseudo-relevance feedback process, in which new queries are continuously generated using terms from documents retrieved in the previous query. These documents are selected by the classifier from the results of each query. The number of queries issued, though, is limited by the target search engine. Thus, it is important to correctly select terms, so that only a small number of queries are submitted to retrieve a high number of relevant documents.

We use an iterative classification-based pseudo-relevance feedback approach. The fact that the queries are continually refined with the help of the classifier allows the initial seed query to be small and simple, as long as it is intuitively highly related to the topic. For instance, in our experiments we use topic names such as "cycling" and 'call for papers" as seed queries.

As our goal is to obtain the highest coverage of relevant pages by issuing fewest queries as possible, the query-issuing policy must combine two factors: exploitation, by choosing the best actions based on already known information; and exploration, by exploring actions that might be sub-optimal at the moment, but that can improve results in further steps. In our case, considering the exploration factor is important for ensuring diversity on the set of seeds generated.

4 The BFC system

In this section, we present *BFC* (*Bootstrapping Focused Crawlers*), a novel system we developed based on the approach described above. We begin by describing the models we use in the pseudo-relevance feedback process, which were designed to balance the exploration and exploitation factors required in our approach. Then, we detail our method by means of an algorithmic description.

4.1 Relevance models

Language models provide a probabilistic framework to represent topics in documents and queries [15]. A basic assumption underlying these models is that words that tend to occur often when discussing a topic T have high probabilities in the corresponding language model θ_T. Thus, one can model the probability of generating a query or a document given the language model of a topic. When used to evaluate the relevance of an item (i.e., term, query or document) to a topic, a language model is called a *relevance model*. Formally, relevant items are samples from this model: the probability $P(x|\theta_T)$ of an item x given a relevance model θ_T indicates the likelihood that x has been generated by the model.

Relevance models are particularly useful in pseudo-relevance feedback settings such as ours. Besides providing an estimate of the probability of a term w, the relevance model θ_T can also be refined using the documents retrieved in different iterations. As we describe below, we use different relevance models for modeling the exploration and exploitation factors.

4.1.1 Exploitation models

Let θ_R be a relevance model for the retrieved documents that are relevant to the topic—the positive relevance model, and θ_N be the relevance model for the retrieved documents that are not relevant to the topic—the negative relevance model. Thus, for a given term w, we can estimate $P(w|\theta_R)$ and $P(w|\theta_N)$. Notice that these probabilities are independent, since w can be generated by each model independently. To take these two models into account, we compute a score to rank terms w as follows:

$$s(w) = P(w|\theta_R) - P(w|\theta_N) \tag{1}$$

While $P(w|\theta_R)$ and $P(w|\theta_N)$ cannot be computed directly without having access to all of the documents related to the topic, they can be approximated using the output of a classifier over the results of a search query. This can be accomplished as follows:

$$\begin{aligned} P(w|\theta_R) &\approx \sum_{D \in \mathcal{D}^+} P(w|D)P(D|\theta_R) \\ P(w|\theta_N) &\approx \sum_{D \in \mathcal{D}^-} P(w|D)P(D|\theta_N) \end{aligned} \tag{2}$$

where $\mathcal{D}^+$ and $\mathcal{D}^-$ are respectively the set of documents classified as being on-topic and off-topic.

Notice that, in a traditional pseudo-relevance feedback scenario [10, 15], $\mathcal{D}^+$ is the set of top-K retrieved documents, and only these documents are considered for evaluating terms. Here, similar to [5], we rely on a classifier for this task. However, we consider *both* the positive and negative classes. Since many of the documents retrieved by the search engine can be in the positive class, tracking the occurrence of terms in the negative class improves the discriminative power of our method with respect to term importance.

If we assume that the classifier separates the positive and the negative subsets of the retrieved documents correctly, we can say that $P(D|\theta_R)$ and $P(D|\theta_N)$ are both equal to 1 in (2). To compute $P(w|D)$, we use the *Maximum Likelihood Estimation (MLE)* and obtain

$P(w|D) = f(w, D)/|D|$, where $f(w, D)$ is the frequency of w within D and $|D|$ is the length of the document:[2]

$$P(w|\theta_R) \approx \sum_{D \in \mathcal{D}^+} \frac{f(w, D)}{|D|} \text{ and } P(w|\theta_N) \approx \sum_{D \in \mathcal{D}^-} \frac{f(w, D)}{|D|} \tag{3}$$

As observed by Cao et al. [5], the classification process can provide valuable information on the importance of terms for a given relevance model. Similar to their *soft filtering* approach, our method uses this information to re-weight the terms. However, instead of directly using the classifier score to compute the new weights, we use the *precision* achieved by a query having the output of the classifier as a reference. By doing so, we avoid the need for normalization and numeric smoothing operations. Note that in our scenario, the re-weighting is used to adjust the importance of query terms according to the results generated by the query.

Let $Q = q_1, \ldots, q_n$ be a query submitted to the target search engine and let $\mathcal{D}_k$ be set of k documents in the answers that were actually retrieved (e.g., due to limits imposed by the search engine). As before, let $\mathcal{D}^+$ be the set of positive documents selected by the classifier from $\mathcal{D}_k$. The precision achieved by the search engine for Q having the classifier as a reference is $prec_Q = |\mathcal{D}^+|/|\mathcal{D}_k|$. Then, the new weights are obtained as follows:

$$\begin{aligned} P^{new}(q_i|\theta_R) = P^{old}(q_i|\theta_R) \times prec_Q \\ P^{new}(q_i|\theta_N) = P^{old}(q_i|\theta_N) \times prec_Q \end{aligned} \tag{4}$$

After the re-weighting, these probabilities are used to compute a score for each term according to (1).

4.1.2 Exploration models

The exploration relevance model aims to include diversity in the set of terms used to construct queries. Here, we face a situation similar to that described in [14] for user relevance feedback: as an extreme case, if all documents selected by the classifier used in the exploitation model have identical contents, the topic becomes ill represented.

For the discussion below, assume that there exists a sub-topic T_E of the topic T being sought, so that at a certain point in the iterative pseudo-relevance feedback process, only terms related to T_E have been selected. This means that there should be another *unexplored* sub-topic T_U of T whose terms are not related to T_E. We look for terms w with a high probability in the language model θ_{T_U}.

$$P(w|\theta_{T_U}) = P(w|\theta_T) \times (1 - P(w|\theta_{T_E})). \tag{5}$$

In this equation, if we consider that the topic T is correctly characterized by the relevance models θ_R and θ_N, we can also say that there are models θ_{R_U} and θ_{N_U} that characterize T_U, which are similar to the exploration models defined in (2). Thus, for θ_{R_U}, we have:

$$P(w|\theta_{R_U}) = P(w|\theta_R) \times (1 - P(w|\theta_{T_E}))$$

[2] No smoothing is required here since we only use terms that occur in some document.

Note that $P(w|\theta_{T_E}) = 1$ if w appears in some query $Q_1, \ldots, Q_m$ generated on the iterative pseudo-relevance feedback process so far. Otherwise, $P(w|\theta_{T_E}) = 0$. Thus

$$P(w|\theta_{R_U}) = \begin{cases} 0, & \text{if } w \in \mathcal{Q} \\ P(w|\theta_R), & \text{otherwise} \end{cases}$$

where $\mathcal{Q} = Q_1 \cup Q_2 \ldots \cup Q_m$ is the set of all terms used in the queries so far. A similar equation can be used for estimating $P(w|\theta_{N_U})$.

Finally, to select terms to be included in the query according to the exploration models θ_{R_U} and θ_{N_U}, we can define a score similar to the one in (1) as follows:

$$s'(w) = \begin{cases} 0, & \text{if } w \in \mathcal{Q} \\ s(w), & \text{otherwise} \end{cases} \tag{6}$$

Equation (6) provides a simple and convenient criterion for selecting novel terms to be included in queries: we select terms that are relevant to the topic and that were not selected in previous queries.

4.2 The SeedFinder algorithm

The SeedFinder algorithm for BFC is shown in Algorithm 1. Given an initial seed query Q_0 as input, it produces a set of URLs (*seeds*) that can be used as seeds for a focused crawler. SeedFinder iteratively constructs new queries using terms selected from the answer pages according to our relevance models (Lines 7–12). *ProcessQuery* (Line 8) obtains two sets of documents returned by a query Q_i: documents classified as being on-topic (relevant) and off-topic (non-relevant). The URLs for the relevant documents $\mathcal{D}^+$ are added to the set of selected seeds (*seeds*) (Line 9). Next, SeedFinder devises a new query by calling the procedure *BuildNextQuery*. These steps are repeated until the stop criterion is reached, for example, a pre-defined number of seeds are obtained.

Algorithm 1 SeedFinder

```
1: SeedFinder($Q_0$)
2: Input: $Q_0$ {initial query}
3: Output: seeds {seeds pages obtained}
4: seeds ← $\emptyset$
5: $i \leftarrow 1$
6: $Q_i \leftarrow Q_0$
7: repeat
8:     $\langle \mathcal{D}^+, \mathcal{D}^- \rangle \leftarrow$ ProcessQuery($Q_i$)
9:     seeds ← seeds $\cup\, \mathcal{D}^+$ {adds positive pages as seeds}
10:    $Q_{i+1} \leftarrow$ BuildNextQuery($Q_i$, $\mathcal{D}^+$, $\mathcal{D}^-$)
11:    $i++$
12: until stop criterion
13: return seeds
```

In Algorithm 2 we describe how *ProcessQuery* works. This Algorithm assumes that no more than N answer pages can be supplied by the search engine, i.e., the search engine allows no more than N consecutive interactions to obtain the results of a query, and that each page has at most K answers. Thus, a maximum of $K \times N$ total answers would be processed. The counter j (Line 8) simply tracks which is the current answer page whose results are being processed. It also controls the end of the loop, when all possible results have already been retrieved (Line 21). In Line 10 the algorithm requests the j-th answer page returned for query Q_i and compiles a list of URLs U_j returned in this page.

Algorithm 2 ProcessQuery

```
 1: **ProcessQuery**($Q_i$)
 2: **Input:** $Q_i$ {current query}
 3: **Output:** $\mathcal{D}^+, \mathcal{D}^-$ {positive and negative documents returned}
 4: **let** $N$ be the maximum number of iterations per query
 5: **let** $K$ be the maximum number of answers allowed per iteration
 6: **let** $prec_{min}$ be a threshold for the minimum acceptable precision
 7: $\mathcal{D}^+ \leftarrow \emptyset, \mathcal{D}^- \leftarrow \emptyset$
 8: $j \leftarrow 0$;
 9: **repeat**
10:    $U_j \leftarrow \text{submit}(Q_i, j)$
11:    **for all** url $u \in U_j$ **do**
12:       $D \leftarrow$ fetch $u$
13:       **if OnTopic**($D$) **then**
14:          $\mathcal{D}^+ \leftarrow \mathcal{D}^+ \cup \{D\}$
15:       **else**
16:          $\mathcal{D}^- \leftarrow \mathcal{D}^- \cup \{D\}$
17:       **end if**
18:    **end for**
19:    $prec \leftarrow |\mathcal{D}^+|/|\mathcal{D}^+ \cup \mathcal{D}^-|$
20:    $j++$
21: **until** $prec < prec_{min}$ **or** $j = N$ **or** $U_j = \emptyset$
22: **return** $\langle \mathcal{D}^+, \mathcal{D}^- \rangle$
```

Then, it fetches each URL in U_j and separates the documents into positive and negative sets $\mathcal{D}^+$ and $\mathcal{D}^-$ (Lines 11–18). This separation is derived from the outcome of the *topic filter* (Section 3) in Line 13. The steps in the main loop (Lines 9-21) are repeated until the precision over the documents retrieved for the query Q_i is smaller than the $prec_{min}$ threshold, or the number of iterations j is equal to N, which is the maximum number of iterations allowed, or there is an empty answer page ($U_j = \emptyset$).

New queries are iteratively constructed by Algorithm 3 *BuildNextQuery*. It takes as input the current query Q_i and the positive and negative sets, $\mathcal{D}^+$ and $\mathcal{D}^-$ respectively, and outputs a new query Q_{new}. It starts by calculating the scores of all terms in $\mathcal{D}^+$ and $\mathcal{D}^-$ (Lines 5–14) according to (1) and (3). Notice that, in practice, not all terms need to be considered since some, for instance, stop words, are non-representative. Such terms may be simply filtered out, as we in fact do in our implementation. Next, the scores of the terms in Q_i are re-weighted using (4) (Lines 17–20). In case the precision for Q_i is smaller than the minimum precision allowed $prec_{min}$, the size of the query is increased by one (Lines 22–24). Finally, the new query $Q_{new} = \langle q_1, q_2, ..., q_n \rangle$ is constructed taking the $n-1$ highest ranked terms according to (2) and the highest rank unused term according to (6) (Lines 25–31).

It is worth noticing that Algorithm 3 uses two different strategies to re-weight terms: one for all terms (Lines 5–14) and another for query terms only (Line 17–19). The rationale for this is trying to prevent queries from changing too much from one iteration to the next. If only the strategy of Lines 5–14 was used, many new terms that are frequent in the documents retrieved could lead the whole query to change, as it in fact occurred in initial experiments we carried out. By doing so, we effectively enforce a bias towards exploitation in the algorithm, that is, we try to keep terms from previous queries if these terms led to a good precision.

Algorithm 3 BuildNextQuery

```
BuildNextQuery($Q_i, \mathcal{D}^+, \mathcal{D}^-$)
Input: $Q_i$ {current query}
Input: $\mathcal{D}^+$, $\mathcal{D}^-$ {positive/negative pages retrieved by $Q_i$}
Output: $Q_{new}$ {next query}
for all $D \in \mathcal{D}^+ \cup \mathcal{D}^-$ do
    for all $w \in D$ do
        {updates scores of terms (Eqs. 1 and 3)}
        if $D \in \mathcal{D}^+$ then
            $score[w] \leftarrow score[w] + f(w, D)/|D|$
        else
            $score[w] \leftarrow score[w] - f(w, D)/|D|$
        end if
    end for
end for
$prec \leftarrow |\mathcal{D}^+|/|\mathcal{D}^+ \cup \mathcal{D}^-|$
let $Q_i = \langle q_1, \ldots, q_n \rangle$
for all $q \in Q$ do
    {re-weights scores of the terms in the query (Eq.4)}
    $score[q] \leftarrow score[q] \times prec$
end for
let $prec_{min}$ be the minimum acceptable precision
if $prec < prec_{min}$ then
    $n{+}{+}$
end if
let $Q_{new} = \langle q_1, \ldots, q_n \rangle$
for $k = 1$ to $n - 1$ do
    $q_k \leftarrow$ the term $w$ with the $k-$th highest value for $score[w]$
end for
{selects the "best" unused term for the new query (Eq.6)}
$u \leftarrow$ the term not used in $Q_1, \ldots, Q_i$ with the highest $score[u]$
$q_n \leftarrow u$;
return $Q_{new}$
```

A possible concern regarding the process we propose to generate seeds is the overhead it introduces in the overall crawling process. However, we argue that this overhead is fairly beareable. First of all, we consider that the costly operation to be aware of in this case is the fetching of pages. To estimate the number of pages fetched, we observe that Algorithm 2 fetches at most N pages for each query Q_i. So, the overhead is N times the number of queries issued. Considering that N is typically 100 and that issuing a few tens of queries is enough to produce good results, we have that some thousands of page fetches are spent to generate queries. We argue that this cost is bearable, in particular because finding seeds lead to save page fetches in the later crawling process.

5 Experimental evaluation

In this section, we report the results of an extensive experimental evaluation we carried out to assess the effectiveness of our approach. Our goals in this evaluation are to study the effect of BFC-derived seeds on different crawling strategies, how they affect the harvest rate and the diversity of the pages retrieved, and how the BFC seeds compare to seeds obtained through other widely accepted strategies.

5.1 Experimental setup

5.1.1 Topics

We selected four distinct topics: *Call*, *Cycling*, *HIV* and *Bossa*. *Call* consists of pages which contain Call for Papers announcements for scientific conferences or journals; (2) *Cycling* consists of pages related to cycling (e.g., sports, cycling trips, bike parts, etc.); (3) *HIV* denotes pages whose content is related to the AIDS disease and HIV; and (4) *Bossa* refers to pages on the Brazilian music style Bossa Nova. As previous results [2, 7, 9, 11] and our own experiments have shown, these topics present distinct degrees of *sparsity*: Call and Bossa are sparse, while HIV and Cycling are denser. Note that the sparsity degree of a topic influences the maximum harvest rate that can be achieved. We discuss this issue in more detail in Section 5.4.

5.1.2 BFC configuration

We executed BFC with all topics using Bing as the target search engine. In all cases, we set SeedFinder (Algorithm 1) to stop after a total of 10,000 URLs were fetched from the answers of the search engine considering all queries. Therefore, this is the stop criterion adopted for $SeedFinder$ in our experiments. For *ProcessQuery* (Algorithm 2), the value of K is 50, which corresponds to the maximum number of answers returned per request. This limit is imposed by the Bing API. After tuning experiments, the $prec_{min}$ threshold (see Algorithms 2 and 3) was set to 0.5. As an illustration of the output derived by BFC, Table 1 shows, for each topic, the number of queries and a sample of 10 words in queries generated in the experiments.

For the topic filters, we implemented SVM classifiers from a set of 30 positive and 30 negative examples for each topic. Despite the small number of sample pages, BFC is capable of generating queries composed of words that are highly related to each topic. Although we could have re-used the relevance classifier used by the crawler, which is stricter, to make BFC general and independent of the crawler, we opted to use a simpler classifier. Nonetheless, for the sake of comparison, we have run experiments that use both classifiers as topic filters.

5.1.3 Crawler configurations

To evaluate the effect of BFC seeds on crawling, we used multiple crawler configurations that vary not only the crawling strategy but also the source of the seed set. Regarding crawling strategies, we used the ones described in [7], which we call *Basic*, and in [9], which uses a link classifier and to which we refer to as *Apprentice*. Notice that *Basic* and *Apprentice* do not play the role of baseline systems here, since our contribution is not a new focused crawling method. Indeed, we could have used any focused crawler method instead, but we choose to use them to demonstrate that the seeds generated by BFC can improve the performance of crawlers, even if they do not use any additional resources such as graph contexts [11] and meta-search [18]. Regarding sources of the seed set, besides BFC we also experimented with seeds manually provided (MAN), seeds provided by DMOZ, which has been used as a source of seeds in previous works on focused crawling [7, 9], and also with

Table 1 Examples of terms used in queries generated by BFC

Topic	#queries	Sample of words composing queries
Call for Papers	29	Conference submission paper call information program author abstract
HIV	20	hiv aids health testing research contact care program center who
Cycling	20	cycling bike ride club bicycle home mountain here page contact
Bossa Nova	58	music brazilian gilberto artist latin guitar choro popular video dance

URLs resulting from the submission of the initial query Q_0 to the search engine (Bing). We name this set as BFC$_0$. This last source corresponds to a simpler form of the BFC's approach, in which, instead of continuously generating new queries using pseudo-relevance feedback, the system would simply take the URLs returned from the first query as seeds. We noticed that, due to the limits imposed by Bing's API, not all of the returned URLs could be used.

The crawler configurations adopted in our experiments are listed below:

- $\boldsymbol{Basic + MAN}$ The Basic crawling strategy using manually selected seeds.
- $\boldsymbol{Basic + BFC}$ The Basic strategy using seeds found by BFC.
- $\boldsymbol{Basic + DMZ}$ The Basic strategy using seeds provided by DMOZ.
- $\boldsymbol{Basic + BFC_0}$ The Basic strategy using seeds from the BFC$_0$ set.
- $\boldsymbol{Basic + BFCPartitioned}$ The Basic strategy using one out of four randomly-selected partitions of similar sizes from the seed set generated by BFC.
- $\boldsymbol{Apprentice}$ The Apprentice crawling strategy using manually selected seeds.
- $\boldsymbol{Apprentice + BFC}$ The Apprentice strategy using seeds found by BFC.
- $\boldsymbol{Apprentice + DMZ}$ The Apprentice strategy using seeds provided by DMOZ.
- $\boldsymbol{Apprentice + BFC_0}$ The Apprentice strategy using seeds from the BFC$_0$ set.
- $\boldsymbol{Apprentice + BFCPartitioned}$ The Apprentice strategy using one out of four randomly-selected partitions of similar sizes from the seed set generated by BFC.

In the experiments, each crawler ran until downloading a total of 100,000 pages, both relevant and non-relevant pages included. For this, we used to 2 servers with Xeon quad-core/64GB/8TB and Xeon quad-core/16GB/2TB of processor/memory/disk space. These servers were connected to the Internet through a 1Gbps link using the infrastructure of the Brazilian National Education and Research Network (RNP). In all cases, the entire crawling process took a few hours to complete. It is worth noting that our experimental crawls were at least five times larger than the ones reported in the focused crawling literature for the same topics [7, 9]. In fact, our crawls are much larger than those carried out in most previous studies on focused crawling [1, 3, 16, 19].

Crawler configurations using manually selected seeds all used 30 seed pages. This is roughly the same number used in the experiments reported in [7, 9]. The seed pages were selected from the top-ranked results returned by Bing using a handful of handcrafted queries related to each topic.

For the configurations using DMOZ seeds, pages from categories "Call for Papers", "Bossa Nova" and "Cycling" were used as seeds for crawling on the corresponding topics. For the topic HIV, the seeds were obtained from the union of the pages from categories "HIV and AIDS", "HIV" and "HIV-AIDS". These are the categories that best match our topics.

For each topic, the initial query used in BFC was formulated using the topic name, i.e., "Call for Papers", "Bossa Nova", "Cycling" "HIV AIDS", "Cycling". As described earlier, these were the queries used to generate the BFC$_0$ sets.

The configurations that use randomly-selected partitions of the seeds generated by BFC are used to better evaluate the way crawlers exploit BFC seeds. We decided to partition the seed set after observing that the crawlers did not use all the seeds provided by BFC. By running the crawlers with smaller seed sets, more of the seeds are actually used by the crawler.

For the topics Call, HIV and Cycling, Chakrabarti et al. [7, 9] constructed Naive Bayes classifiers to determine the relevance of pages. In our experiments, we decided to use SVM classifiers because, in general, they are known to perform better than Naive Bayes classifiers for text classification. The accurate classification of pages in crawling is critical for a reliable graph analysis, presented in Section 5.5. For each of the topics, we generated an SVM classifier from a set of 1000 positive and 1000 negative example pages. The generated classifiers were evaluated by manually inspecting a sample of 100 random pages from the set of all pages crawled. For all classifiers, we obtained a ratio of true-positives and true-negatives between 70-98 % with a 95 % confidence level.

5.1.4 Crawl pool

In the experiments, we often need to estimate the coverage of the crawls over the set of pages comprising a topic. Unfortunately, this can only be precisely evaluated if the full Web graph or the graph from an exhaustive crawl on the topics is available. Since it is not feasible to obtain either, we approximate the exhaustive crawl by constructing a *topic crawl pool*. The crawl pool for a topic T is a graph generated from the union of the graphs obtained with all crawling configurations described above for T. Both Basic and Apprentice strategies were used. Thus, each crawl pool is composed by the union of 16 different crawls for each topic.

Notice that, even though the coverage estimated using the crawl pool may not precisely represent the real coverage, the pool is useful for the purpose of comparing the crawl configurations in our experiments [21, 25]. This technique has been used in comparative evaluations of Information Retrieval methods in many other tasks and contexts.

5.2 Overall effectiveness

We present in Table 2 results obtained with all crawler configurations for the four topics. Most of these results will be discussed in detail later. We present them here to provide a broader perspective of the experiments we have performed to evaluate our approach.

In this table, column “Seeds/Set” describes the seed set used as input to the crawler and “Seeds/#” refers to the size of this set. The row “MAN” refers to the seed set manually selected; “DMZ” refers to the seed set obtained from the DMOZ directory; “BFC_0” refers to the set of seeds obtained with BFC_0; “BFC” refers to the complete set of seeds obtained with BFC; and “P1” to “P4” refer to the randomly-selected partitions of “BFC”. Columns “Basic/# R” and “Appr./# R” present the number of relevant pages obtained using the corresponding seed set with crawling strategies Basic and Apprentice, respectively. Columns “Basic/ % C” and “Appr./ % C” show the coverage of relevant pages in the crawl pool, also obtained using the corresponding set of seeds with crawling strategies Basic and Apprentice, respectively. Notice that the size of each crawl pool is given in the header of the table.

An important point to note is that BFC generated a high number of seeds. In all cases, this number is at least one order of magnitude higher than the number of manually provided seeds. Manually obtaining such a high number of seeds would be very time consuming to say the least. This number is also much higher than the number of seeds that can be obtained from DMOZ. In fact, the number of seeds from DMOZ reported in Table 2 accounts only for valid URLS, since we have found that between 10 % and 20 % of the links were stale.

The crawls that use BFC and BFC_0 seed sets obtained far more pages than those using manual and DMOZ seeds for Call and Bossa topics. In the case of Call, the higher number of pages obtained was observed not only for the whole set of BFC seeds, but also for crawls that use the smaller partitions (i.e., P1 to P4). With respect to coverage, the highest value was obtained with the crawl that used BFC seeds, both in Call and Bossa, followed by the

Table 2 General results obtained with distinct sets of seeds and different crawler configurations for our four topics

	Seeds		**Basic**		**Appr.**	
	Set	**#**	**#R**	**%C**	**#R**	**%C**
Call (Pool = 37,511 pages)						
	MAN	30	2383	6.35	2221	5.92
	DMZ	83	1975	5.27	3738	9.97
	BFC$_0$	605	6355	16.94	6533	17.42
	BFC	3884	6673	17.79	7845	20.91
	P1	900	4627	12.34	6441	17.17
	P2	900	4607	12.28	7096	18.92
	P3	900	5420	14.45	5182	13.81
	P4	900	4839	12.90	7009	18.69
Bossa (Pool = 53,477 pages)						
	MAN	30	2199	4.11	2743	5.13
	DMZ	2	2216	4.14	2408	4.5
	BFC$_0$	486	12889	24.10	8116	15.18
	BFC	2119	15638	29.24	9990	18.68
	P1	489	12755	23.85	5843	10.93
	P2	489	13945	26.08	7168	13.4
	P3	489	10695	20.00	5998	11.22
	P4	489	11824	22.11	6588	12.32
HIV (Pool = 344,604 pages)						
	MAN	30	63283	18.36	66157	19.2
	DMZ	797	63460	18.41	63553	18.44
	BFC$_0$	421	64235	18.64	56056	16.27
	BFC	6593	64518	18.72	64868	18.82
	P1	1536	65171	18.91	54882	15.93
	P2	1536	63302	18.37	55332	16.06
	P3	1536	63361	18.39	54487	15.81
	P4	1536	65333	18.96	54087	15.7
Cycling (Pool = 916,402 pages)						
	MAN	30	70956	7.58	81538	8.71
	DMZ	3755	72626	7.76	80998	8.65
	BFC$_0$	770	75144	8.03	77483	8.28
	BFC	8767	78576	8.39	86991	9.20
	P1	1803	75424	8.06	77533	8.28
	P2	1803	75183	8.03	77739	8.30
	P3	1803	76414	8.16	79009	8.44
	P4	1803	75891	8.10	78341	8.37

crawls that use partitions of BFC seeds. In these cases, the higher coverage is a consequence of the greatest number of pages obtained with these crawls.

For HIV and Cycling, the gains obtained by BFC were smaller. Interestingly, there were cases in which a small number of seeds led to a slightly higher number of pages. For instance, in HIV the larger BFC seed set resulted in fewer pages than P1 and P4. This is explained by the fact that current crawling strategies benefit less from a high number of seeds in dense topics, such as HIV and Cycling, than in sparse topics, such as Call and Bossa. This issue is examined in detail in Section 5.5.

The analysis of the coverage obtained for HIV and Cycling reveals an interesting aspect regarding the seeds generated by BFC. As in all crawls, if the coverage is low and the number of pages obtained is about the same, it is possible to conclude that the relative intersection between the crawls is also low. This follows from the fact that a total intersection would lead to a coverage of 100 % in all crawls, but, both in HIV and Cycling, the coverage is far below that. While this could be expected if we compare crawls that use different sets of seeds, e.g., BFC, MAN , DMZ and BFC_0, such a small intersection is also observed between crawls using the full set of BFC seeds and its partitions. This is explained by the fact that Basic and Apprentice crawl the same seed set in different ways, but obtain a high number of pages in all cases. We discuss this further in Section 5.5.

In sum, these numbers show that BFC and BFC_0 seeds led to very good results in all topics. And unlike manual seeds or seeds obtained by DMOZ, BFC and BFC_0 seeds are gathered in a completely unsupervised fashion. Obviously, for sparse topics, such as Call and Bossa, BFC seeds had a greater impact, precisely due to the difficulty in manually finding good seeds for them.

5.3 Crawl coverage

Table 3 shows the number of connected components resulting from crawls using configurations Basic+DMZ, Basic+BFC_0, Basic+BFC, Apprentice+DMZ, Apprentice+BFC_0, Apprentice+BFC over the four topics. Here, we consider weakly connected components, i.e., sets of nodes/pages such that for any pair $< u, v >$ there is an undirected path from u to v. Notice that, for our purposes, to consider weakly connected components it is appropriate to characterize a group of linked on-topic pages, even if they are not all reachable from each other, as in strong connected components.

As can be observed, BFC configurations led to more components than DMOZ and BFC_0 configurations in all topics. Greater differences are mostly found for Basic crawls. This is explained by the fact the Apprentice, as explained in Section 2, attempts to expand the traversal beyond on-topic pages, and thus, can reach more "far-away" components. Interestingly, comparing the results from Table 3 with those from Table 2 reveals that, although the number of relevant pages found with BFC seeds is about the same as those found with DMOZ seeds for less sparse topics such as HIV and Cycling, the number of components

Table 3 Number of connected components found

	Call			**Bossa**			**HIV**			**Cycling**		
	DMZ	**BFC_0**	**BFC**	**DMZ**	**BFC_0**	**BFC**	**DMZ**	**BFC_0**	**BFC**	**DMOZ**	**BFC_0**	**BFC**
Basic	830	2048	2755	281	2696	2747	231	148	1093	2079	454	4634
Appr	1738	2436	3717	529	777	1651	1879	2324	2603	1473	863	4191

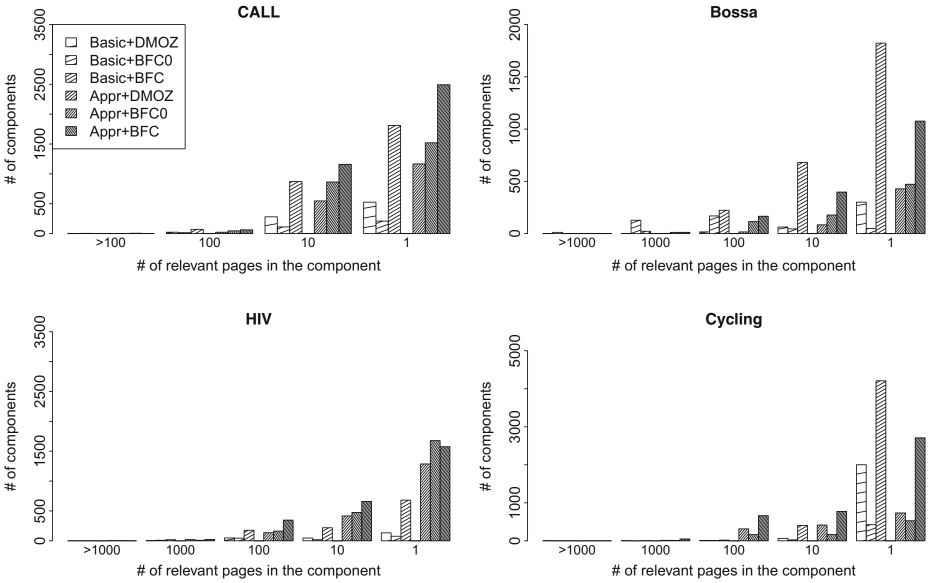

Figure 3 Distribution of component sizes

reached is higher. Notice that this it is also true when comparing BFC and BFC$_0$ configurations. BFC$_0$ configurations reach approximately the same number of relevant pages than BFC in all topics, but BFC lead the crawlers to a higher number of components. This is an indication that BFC indeed provides better coverage of the Web for a given topic.

In Figure 3, we further examine the connected components from Table 3. This figure shows that the size of the components (i.e., the number of relevant pages in each component) follows a Zipfian distribution in all cases. As discussed in Section 2, this is expected, but notice that BFC obtains more components of almost all sizes. Also in Figure 3, the majority of components are of size 1. We have found that most of these components are on-topic seeds that were not exploited by the crawlers in this particular experiment. Despite this, we could verify that these seeds can indeed lead to relevant pages in other experiments we have performed, as described in the next section.

5.4 Quality of BFC seeds

Table 4 shows the percentage of on-topic seed pages, i.e., those considered as relevant for each topic. This relevance judgment was made by the classifier used in the crawler configurations described in Section 5.1. While a high percentage of on-topic pages were found among the seeds for topics HIV and Cycling, this percentage is low in Call and Bossa.

Table 4 On-topic seed pages per topic

Topic	# BFC seeds	% On-Topic
Call	5742	40.5%
Bossa	701	31.52%
HIV	6529	79.33%
Cycling	8767	84.64%

Although at a first glance, this may seem problemematic regarding the quality of the seeds generated by BFC, being strictly relevant to the topic is not the only aspect to be considered. Indeed, considering the discussion in Section 2, a good seed must *lead* to relevant pages on the topic, even if the seed itself is not relevant.

5.4.1 Seed yield

To verify the quality of the seeds generated by our method, we measured the number of relevant pages that can be reached from them. For this, we used a metric we call *Seed Yield* or simply *Yield*. Let s be a seed page on the Web graph. For a given topic, the *yield* of s at distance k, $yield(s, k)$, is the number of relevant pages reachable from s at distance of k or fewer hops. Although the actual yield of a seed can only be accurately computed over the whole Web graph, here we approximate this measure using the crawl pool described in Section 5.1. The seed yield numbers are presented in Figure 4. For each curve in this figure, each point corresponds to the sum of the yield of all seeds provided by users (i.e., manually), or taken from DMOZ, or using BFC/BFC$_0$ at k hops. The curves labeled BFC+ considers only seeds judged as being on-topic from Table 4.

The curves indicate that BFC seeds are closer to relevant pages than those obtained manually, using DMOZ or BFC$_0$. Thus, any crawler using BFC seeds finds relevant pages with less effort in terms of page downloads. Note in particular the plot for Call. Because this topic is very sparse, several hops are necessary to get from the manual seeds or DMOZ seeds to 40 % of relevant pages, while with BFC seeds only 2 hops are necessary. In the case of Bossa, 4 hops with BFC seeds were enough to reach the same number of relevant pages as manual and DMOZ seeds with 10 hops. Note also that, despite the fact that the coverage in crawling configurations using seeds in the BFC$_0$ set are close to BFC (see Table 2), the

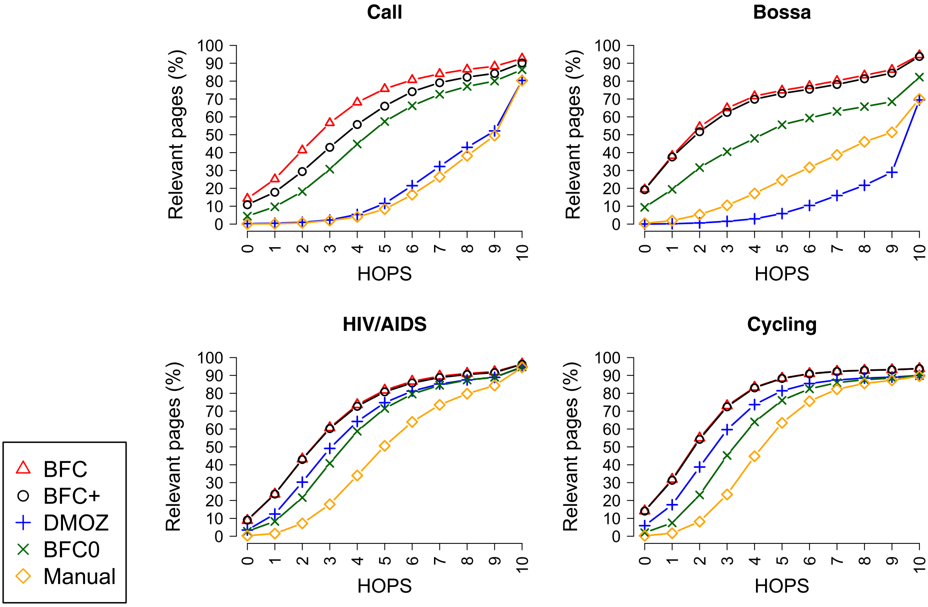

Figure 4 Number of relevant pages in the crawl pool at different distances (number of hops) from the seeds

coverage in early hops are not as high as those using BFC seeds. This clearly indicates that BFC seeds are also closer to relevant pages than BFC_0 seeds.

By comparing curves of BFC, which considers all seeds generated and BFC+, which considers only seeds judged as relevant by the (stricter) classifier used by the crawlers, it is possible to see that using a less strict classifier as we did in BFC does not compromise the quality of the seeds obtained. Indeed, the seed yield is slightly better for topics Call and Bossa. In the case of HIV and Cycling, almost all seeds are considered as being on-topic, thus the seed yield is the same.

Another perspective of seed yield is presented in Table 5. In this table, for each topic, we show, in column " %S", the portion of BFC seeds whose yield is in the range indicated in column "Range", considering any distance k. For all topics, the great majority of BFC seeds were able to provide a high number of relevant pages. These results show that BFC seeds indeed lead the crawlers to relevant pages.

Also, we verified, in all cases, that all on-topic seeds reached some relevant pages, and, as we expected, many off-topic seeds also led to relevant pages. For instance, in Call, 59.95 % of the seeds were classified as off-topic (see Table 4), while only 20.82 % of the seeds were non-productive. Thus, in this topic, 39.13 % of the seeds which are classified as being off-topic are productive.

Notice that, for all topics, most of the seeds reach approximately the same high number of relevant pages in the crawl pool, while a few seeds lead to a much smaller number of pages. This can be explained by the fact that most of the seeds are connected to the same large component on the Web. This explanation is consistent with the findings in [8], where the authors predicted that the size of components in a topic graph follow a Zipfian distribution. On the other hand, this could also indicate that BFC generated seeds that are limited to only a few hosts on the Web. But as we discuss next, this is not the case.

Table 5 Alternative perspective for seed yield

	Range	%S
Call		
	[30080, 30455]	75.90
	[1, 24]	3.28
	0	20.82
Bossa		
	[37140, 39599]	84.61
	[1, 215]	3.91
	0	8.09
HIV		
	[325617, 325785]	65.63
	[1, 208]	2.58
	0	31.80
Cycling		
	[820793, 821312]	55.49
	[1, 681]	37.92
	0	6.59

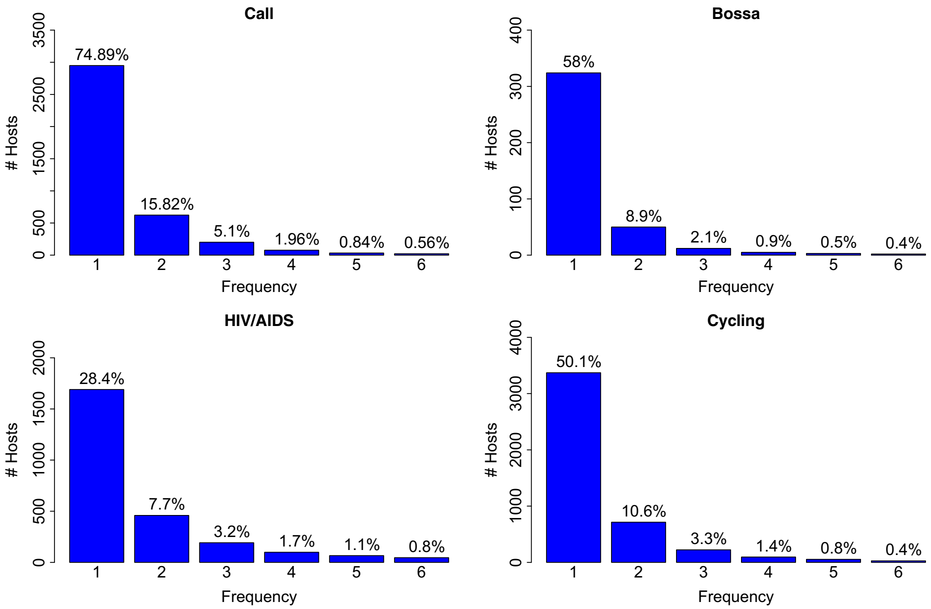

Figure 5 Distribution of hosts among BFC seeds, with frequency up to 6

5.4.2 Seed diversity

In order to verify the diversity of the seeds generated by BFC, as shown in Figure 5, we plotted for each topic, the distribution of hosts to which BFC seeds belong, with frequency up to 6. Over each bar in the plot, we also show the corresponding portion of URLs accounted for. We only include productive seeds, i.e., those that lead to at least one relevant page. We have applied well-known URL deduplication techniques to avoid counting the same host more than once. Observe that, in all topics, the great majority of hosts appears just once in the set of BFC seeds. This shows that BFC seeds are diverse and do not restrict the crawlers to a small set of hosts.

5.5 Impact of BFC seeds on crawls

5.5.1 Harvest rate

A common measure used to evaluate focused crawlers is the *harvest rate*—the rate at which relevant pages are acquired [7]. The plots in Figure 6 show for Basic and Apprentice, the number of relevant pages found as a function of the number of downloaded pages. Each plot has a curve corresponding to each set of seeds considered in Table 2, i.e., MAN, DMZ, BFC, BFC_0 and P1 to P4.

For the sparse topics, Call and Bossa, BFC seeds retrieved a higher number of relevant pages compared to crawls using MAN, DMZ and BFC_0 seeds. For HIV and Cycling, which are dense topics, all configurations behave similarly. Again, we stress that unlike DMOZ and MAN, BFC obtains seeds in an unsupervised way.

5.5.2 *Seed novelty*

Another question we investigated was whether BFC is able to reach *novel* seeds, i.e., seeds that would be hard to find otherwise, or if these seeds could be found, for instance, by just crawling from the manually provided seeds.

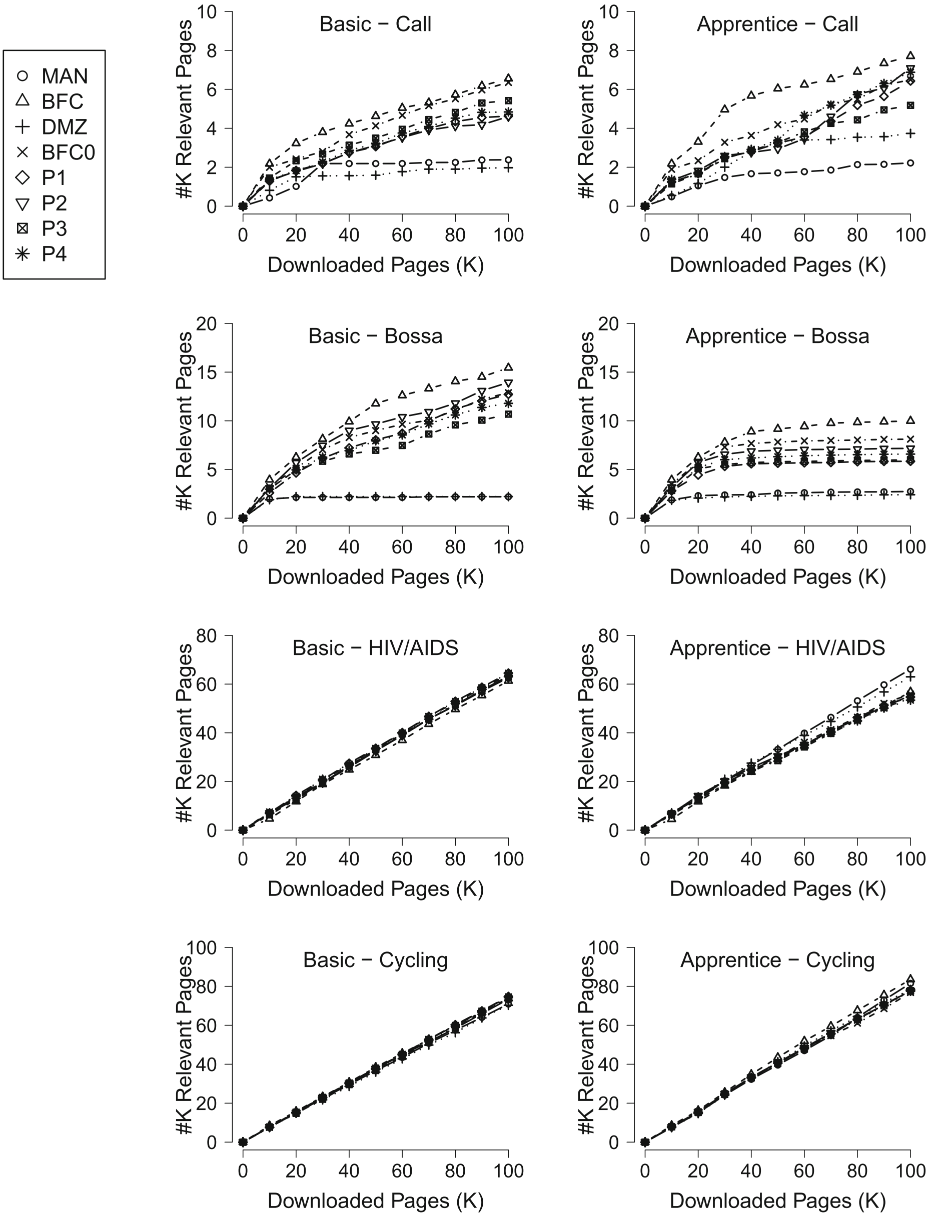

Figure 6 Comparison of the number of relevant pages found as a function of the number of downloaded pages

Table 6 Seed novelty

Topic	DMOZ		BFC_0	
	Basic	**Appr.**	**Basic**	**Appr.**
Call	0.57 %	0.71 %	0.87 %	3.58 %
Bossa	0.42 %	0.32 %	0.32 %	3.98 %
HIV	0.22 %	0.15 %	0.30 %	1.01 %
Cycling	0.38 %	0.35 %	0.29 %	1.11 %

To determine this, we examined the intersection of BFC seeds and all relevant pages obtained by the crawler configurations with DMOZ and BFC_0 seeds. As Table 6 shows, crawls starting from DMOZ and BFC_0 seeds reach very few of the seeds obtained with BFC. This means that BFC seeds are novel and are not reachable by these crawls.

5.5.3 Search engine exploitation

BFC is able to generate a high number of seeds. In fact, for the dense topics, most of these seeds are relevant pages themselves (see Section 5.4). Thus, an interesting question that arises is whether BFC itself could be used as a focused crawler. Answering this question involves two other related questions: (i) is it possible to obtain a high number of pages on a given topic from the search engine index?, and (ii) can a focused crawler harvest pages that are not in the search index?

To answer the first question, we noticed that, although it could be theoretically possible to group pages on a certain topic if one has access to a search engine's document collection, in practice this is not possible since BFC uses an external search interface. This is due not only to the limits on the number of requests and results obtained imposed by search engine interfaces, but also due to the well-known precision decay with the recall growth, which is found in virtually any practical ranking method.

To answer the second question, we have carried out an experiment to estimate the fraction of relevant pages found with a crawler using BFC seeds that were already indexed by BING, our target search engine. As it was unfeasible to execute this experiment with the entire set of relevant pages obtained with both Basic and Apprentice strategies for all topics, we used two types of samples from this set. The first sample, which we call Random, corresponds to a random sample of 1500 pages from the whole set of relevant pages. This was enough to allow a confidence level of 98 %. The second sample, which we call Deepest, consists

Table 7 Crawled relevant pages found in the search engine

Topic	Random		Deepest	
	Basic	**Appr.**	**Basic**	**Appr.**
Call	33 % ± 2.48	33 % ± 2.49	59 %	59 %
Bossa	20 % ± 2.33	21 % ± 2.30	64 %	64 %
HIV	67 % ± 2.35	66 % ± 2.37	69 %	68 %
Cycling	75 % ± 2.17	75 % ± 2.17	59 %	61 %
Averages	48.75 %	49.00 %	62.75 %	63.00 %

of relevant pages with the deepest URL in each host found in the crawls. This is motivated by the fact the search engine crawling strategies are usually based on a breadth-first search, thus prioritizing shallow URLS.

The results are presented in Table 7. Notice that at least 25 % of the pages found by the focused crawlers were not found on the search engine index. This percentage grows to at least 31 % if we consider only the deepest relevant pages. As expected, sparse topics had the lowest coverage in all cases.

6 Conclusions and discussion

We proposed a framework for automatically discovering seeds that takes advantage of the high number of pages already crawled by general-purpose search engines, such as Google or Bing. Although our original motivation to develop BFC was to simplify crawler configuration and relieve the users from the task of manually selecting the seeds, as it turned out, the ability to automatically collect a high number of seeds had another notable benefit: it improved crawler efficiency and coverage.

Indeed, when used to feed focused crawlers, the experiments we performed show that BFC seeds led to gains in the number of relevant pages obtained and in harvest rate achieved. Although BFC had positive impact on all topics tested, it was specially good for the case of topics whose pages are sparsely distributed over the Web.

By examining the obtained results in our experiments, we could verify a number of properties previously identified in the literature regarding the organization of topics on the Web [12]. For instance, for all topics, we observed a Zipfian distribution on the size of the connected components of the graphs formed by relevant pages. On the other hand, our findings contradict the prior belief that starting from different seeds, focused crawlers would produce sets of relevant pages which broadly overlap [6, 7]. In fact, our results showed minor ovelapping. This indicates that having an extensive seed set is important for obtaining a good topic coverage.

Besides providing a scalable means to obtain seeds, BFC could potentially serve as a surrogate that uses search engines to perform focused crawling. Of course, such an approach would only be applicable for tasks that do not exceed the limits imposed by the search engines on the number of queries and result size. But even if these limits were not present, to obtain greater coverage, the use of a focused crawler can be beneficial. As we show in the experimental results, by combining BFC with a focused crawler, it is possible to retrieve a significant number of pages that are not present in the search engine index.

The framework we proposed can be implemented in different ways. BFC is one possible implementation. We should note that our goal with BFC was to provide a proof-of-concept rather than develop the best solution for all steps comprising the framework. For instance, Thus, there are several questions that we intend to investigate in future work, including: how to best construct seed queries—BFC uses topic names such as "cycling" and "call for paper" as seed queries, but other approaches such as the use of a sample document are possible; how to select terms from answer pages—currently, BFC extracts all terms, but it is also possible to be more selective, for example, extract only terms in the title of the page, or include terms taken from anchor texts referring to the page; how to better balance the exploration and exploitation factors when composing queries—while BFC favors exploitation, other balancing schemes could be considered.

Acknowledgments This research was partially sponsored by projects TTDSW (PRONEM/FAPEAM/CNPq), e-vox pesquisa (FAPEAM), e-spot (CNPq Universal), and by individual CNPq fellowship grants to Edleno Moura (308130/2014-6) and Altigran da Silva 311433/2014-6). This material is based on research sponsored by DARPA under agreement number FA8750-14- 2-0236. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

1. Almpanidis, G., Kotropoulos, C., Pitas, I.: Combining text and link analysis for focused crawling-an application for vertical search engines. Inf. Syst. **32**, 886–908 (2007)
2. Barbosa, L., Freire, J.: An adaptive crawler for locating hidden-web entry points. In: Proceedings of the 16th International Conference on World Wide Web, pp. 441–450 (2007)
3. Batsakis, S., Petrakis, E.G.M., Milios, E.: Improving the performance of focused web crawlers. Data Knowledge & Engineering **68**, 1001–1013 (2009)
4. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the Web. Comput. Netw. **33**(1–6), 309–320 (2000)
5. Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval, pp. 243–250 (2008)
6. Chakrabarti, S.: Focused web crawling. In: Encyclopedia of Database Systems, pp. 1147–1155 (2009)
7. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: A new approach to topic-specific web resource discovery. Comput. Netw. **31**(11–16), 1623–1640 (1999)
8. Chakrabarti, S., Joshi, M., Punera, K., Pennock, D.: The structure of broad topics on the web. In: Proceedings of the 11th International Conference on World Wide Web, pp. 251–262 (2002)
9. Chakrabarti, S., Punera, K., Subramanyam, M.: Accelerated focused crawling through online relevance feedback. In: Proceedings of the 11th International Conference on World Wide Web, pp. 148–159 (2002)
10. Croft, W.B., Metzler, D., Strohman, T.: Search Engines - Information Retrieval in Practice. Pearson Education (2009)
11. Diligenti, M., Coetzee, F., Lawrence, S., Giles, C.L., Gori, M.: Focused crawling using context graphs. In: Proceedings of 26th International Conference on Very Large Databases, pp. 527–534 (2000)
12. Dill, S., Kumar, R., Mccurley, K.S., Rajagopalan, S., Sivakumar, D., Tomkins, A.: Self-similarity in the Web. ACM Trans. Internet Technol. **2**(3), 205–223 (2002)
13. Johnson, J., Tsioutsiouliklis, K., Giles, C.L.: Evolving strategies for focused web crawling. In: ICML, pp. 298–305 (2003)
14. Karimzadehgan, M., Zhai, C.: Exploration-exploitation tradeoff in interactive relevance feedback. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1397–1400 (2010)
15. Lavrenko, V., Croft, B.: Relevance-based language models. In: Proceedings of the 23st Annual International ACM SIGIR Conference on Research and Development on Information Retrieval, pp. 120–128 (2001)
16. Menczer, P.G., Srinivasan, P.: Topical web crawlers: Evaluating adaptive algorithms. ACM Trans. Internet Technol. **4**(4), 378–419 (2004)
17. Prasath, R., Öztürk, P.: Finding potential seeds through rank aggregation of web searches. Pattern Recog. Mach. Intell., 227–234 (2011)
18. Qin, J., Zhou, Y., Chau, M.: Building domain-specific web collections for scientific digital libraries: a meta-search enhanced focused crawling method. In: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL '04, pp. 135–141. ACM, New York (2004)
19. Sizov, S., Theobald, M., Siersdorfer, S., Weikum, G., Graupmann, J., Biwer, M., Zimmer, P.: The bingo! system for information portal generation and expert web search. In: CIDR (2003)
20. Vidal, M.L., da Silva, A.S., de Moura, E.S., Cavalcanti, J.M.B.: Structure-driven crawler generation by example. In: Proceedings 29th of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 292–299 (2006)
21. Voorhees, E.M.: The philosophy of information retrieval evaluation. In: Evaluation of Cross-Language Information Retrieval Systems, pp. 355–370. Springer (2002)

22. Wu, J., Teregowda, P., Ramírez, J.P.F., Mitra, P., Zheng, S., Giles, C.L.: The evolution of a crawling strategy for an academic document search engine: Whitelists and blacklists. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 340–343 (2012)
23. Zheng, S., Dmitriev, P., Giles, C.: Graph-based seed selection for web-scale crawlers. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 1967–1970 (2009)
24. Zhuang, Z., Wagle, R., Giles, C.L.: What's there and what's not?: focused crawling for missing documents in digital libraries. In: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL '05, pp. 301–310. ACM, New York (2005)
25. Zobel, J.: How reliable are the results of large-scale information retrieval experiments? In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 307–314 (1998)