Information Processing and Management 45 (2009) 380–391

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Finding what is missing from a digital library: A case study in the Computer Science field

Allan J.C. Silva [a], Marcos André Gonçalves [a,*], Alberto H.F. Laender [a], Marco A.B. Modesto [a], Marco Cristo [b], Nivio Ziviani [a]

[a] Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte, Brazil
[b] FUCAPI – Technological and Research Foundation, Manaus, Brazil

ARTICLE INFO

Article history:
Received 28 May 2008
Received in revised form 4 December 2008
Accepted 14 December 2008
Available online 18 February 2009

Keywords:
Full text
Digital libraries
Search engines
DBLP
Scholar
Google

ABSTRACT

This article proposes a process to retrieve the URL of a document for which metadata records exist in a digital library catalog but a pointer to the full text of the document is not available. The process uses results from queries submitted to Web search engines for finding the URL of the corresponding full text or any related material. We present a comprehensive study of this process in different situations by investigating different query strategies applied to three general purpose search engines (Google, Yahoo!, MSN) and two specialized ones (Scholar and CiteSeer), considering five user scenarios. Specifically, we have conducted experiments with metadata records taken from the Brazilian Digital Library of Computing (BDBComp) and The DBLP Computer Science Bibliography (DBLP). We found that Scholar was the most effective search engine for this task in all considered scenarios and that simple strategies for combining and re-ranking results from Scholar and Google significantly improve the retrieval quality. Moreover, we study the influence of the number of query results on the effectiveness of finding missing information as well as the coverage of the proposed scenarios.

1. Introduction

On-line access to the full text of cataloged documents is an important requirement for satisfying the needs and expectations of the users of a digital library (DL) of scientific articles (Laender et al., 2008). However, in many of such DLs, mainly those built by aggregating metadata from heterogeneous sources, not all metadata records have a direct pointer (e.g., a URL) to the corresponding full text. This situation is also common when the DL makes available information about citations and references but there is no direct access to the referenced items.

Even the presence of a direct pointer may not be useful for the user, for instance, in the cases in which the access to the full text requires the payment of a fee or the pointer became invalid due to the Web dynamics. In these cases, a service that retrieves the respective missing full texts from other Web sources would be of great value to the DL's users.

In this article, we propose a process to provide such a service. It explores general purpose and specialized Web search engines to retrieve the URLs of full-text documents for which metadata records exist in a DL catalog but a full-text pointer is not available. The idea is to explore the potentiality of the existing search engines and to study how they behave in this specific task. For this, we experimented with records of documents from DLs in the Computer Science field. We also study the potential of the proposed process in finding other related documents and information that may be useful for the user, such as

* Corresponding author.
E-mail addresses: allan@dcc.ufmg.br (A.J.C. Silva), mgoncalv@dcc.ufmg.br (M.A. Gonçalves), laender@dcc.ufmg.br (A.H.F. Laender), mabm@dcc.ufmg.br (M.A.B. Modesto), marco.cristo@fucapi.br (M. Cristo), nivio@dcc.ufmg.br (N. Ziviani).

0306-4573/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2008.12.006

other publications (e.g., a thesis or dissertation) authored by one of the authors of the searched article and additional metadata that complements the information already in the DL (e.g., references). In our experiments, we consider content freely available on the Web (e.g., in one of the authors' homepage) as well as that provided by restricted sources (e.g., a publisher's Web site).

We investigate how to specify, for each considered search engine, the most effective queries for the task at hand and which search engine is the best given distinct user requirement levels. We also explore search engine combination strategies to improve the overall effectiveness of our process. Finally, we analyze the capability of the proposed process to find the desired content in the considered scenarios.

In our evaluation, we use metadata records from the BDBComp – *Brazilian Digital Library of Computing*[1] metadata catalog (Laender, Gonçalves, & Roberto, 2004) complemented with a set of records extracted from DBLP – *The DBLP Computer Science Bibliography*[2] corresponding to conference papers authored by Brazilian researchers but not present in the first collection. We randomly sampled a set of records from these two collections and used them to build queries which we submitted to three general purpose search engines (Google, Yahoo!, and MSN) and two specialized ones (Scholar and CiteSeer) aiming at retrieving the corresponding full texts or other relevant but missing information. Our experimental results demonstrate that our proposed process is effective and provides a very simple strategy for finding the full text of documents cataloged in a DL for which a corresponding URL is missing. They have also shown that, among the five tested search engines, Scholar is the most effective one for this task and that, when combined with Google, significant gains are achieved for all considered scenarios.

It is important to note that, despite the study presented in this article has been carried out with metadata records of documents from Computer Science, a field very well represented on the Web, recent studies show that search engines, such as Scholar, also cover in a reasonable manner the content of other fields (Walters, 2007). Since the proposed process does not rely on intrinsic characteristics of any academic field, such as publication patterns, standards or preferences, but depends only on the metadata cataloged in DLs, we believe that it can be applied to other fields with similar results.

In summary, the main contributions of this article are:

(1) The proposal of a process for finding the URL of the corresponding full text, or of any relevant related material, for those documents cataloged in a DL but for which this information is missing.
(2) A comprehensive study of this process using different query strategies applied to different search engines and considering different user needs and profiles.
(3) The proposal of a strategy for combining the results coming from specific search engines and re-ranking them, which improves the overall quality of the retrieved URLs.
(4) An analysis of the existent trade-off between the efficiency and the effectiveness of the proposed process.

The remainder of this article is organized as follows. Section 2 addresses related work. Section 3 describes the process proposed for retrieving the missing URLs. Section 4 discusses our experimental environment. Section 5 describes the methodology and the metrics used to compile the experimental results. Section 6 presents and discusses these results. Section 7 analyzes the impact of several factors in the likelihood of finding the missing URLs. Section 8 presents the conclusions and future work.

2. Related work

Current approaches to find documents missing from DLs rely mostly on focused crawlers (Chakrabarti, van den Berg, & Dom, 1999). For instance, in Zhuang, Wagle, and Giles (2005), the authors investigate the feasibility of using publication metadata to guide the crawler towards author's homepages to harvest documents that are missing from a DL collection. However, relying on focused crawlers to maintain collections of scientific documents requires the construction of a complex software infrastructure. Therefore, in this work we advocate taking advantage of the current content already indexed by existing search engines but having just a small effort of formulating appropriate queries to such search engines.

The use of the infrastructure provided by search engines has been beneficial in many situations. In Qin, Zhou, and Chau (2004), for example, the authors discuss limitations of traditional focused crawling algorithms and argue that the use of meta-search can help overcome such deficiencies. They also propose that answers of queries submitted to search engines can be used to make more diverse the search space of such algorithms, which are normally limited to the content located close to the seeds selected as initial points for the crawling process. In Harrison and Nelson (2006), the authors describe strategies for finding information related to pages missing from Web sites. Cached versions of the missing pages retrieved from search engines are used for generating a lexical signature of a set of terms that captures the essential information presented in the page which is then used to find similar documents or alternative copies of the original document. This strategy is the basis of a framework that aims at preserving the information available on the Web.

Some systems that provide searching or crawling services for scientific articles have been reported in the literature, such as HPSearch and Mops (Hoff & Mundhenk, 2001), and Paper Search Engine (PaSE) (On & Dongwon, 2004). However, these

[1] http://www.lbd.dcc.ufmg.br/bdbcomp/.
[2] http://dblp.uni-trier.de/.

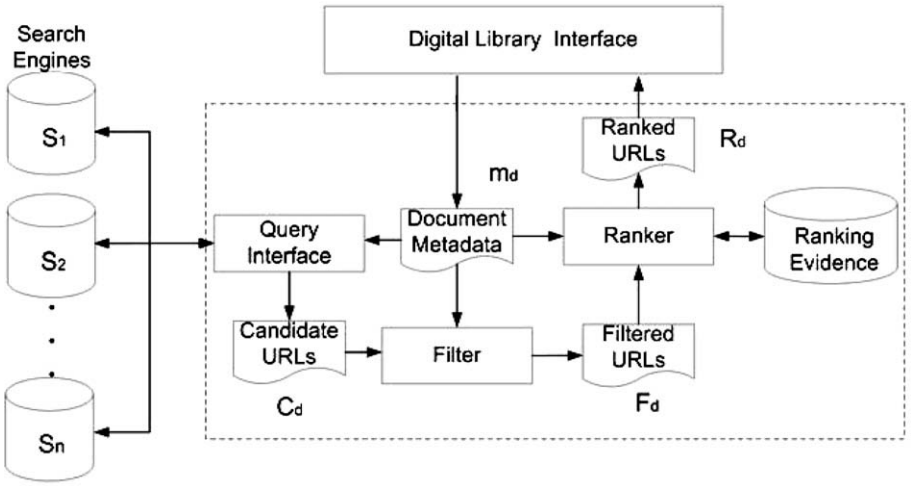

Fig. 1. Service architecture.

works focus on searching for scientific articles in general. In our work, we restrict our investigation to articles for which metadata records exist in a DL but a full-text pointer is not available. Thus, we evaluate the effectiveness of existing search engines to accomplish this task.

Finally, comparative studies evaluating the effectiveness of generic search engines to satisfy general information needs are very common (Bharat & Broder, 1998; Chu & Rosenthal, 1996; Gordon & Pathak, 1999; Lawrence & Giles, 1998, 1999). However, we have been unable to find any work comparing the use of generic and specialized search engines for the specific task described here.

3. Proposed process

We envision a service aimed at helping users to find DL missing content on the Web. The proposed process is depicted in Fig. 1. Interacting with the *Digital Library Interface*, if the user notices that a document *d* has no full text in the DL, she may request the service to search for the missing information. By using the *Article Metadata* record m_d, the *Query Interface* automatically generates and submits queries to one or more search engines requesting the missing information. *Candidate URLs* are extracted from the resulting pages as an ordered list C_d. The results in C_d follow a ranking that prioritizes answers coming from vertical search engines[3] since they index collections of scientific articles, potentially minimizing noise. Results from a same search engine have their relative positions preserved.

Then, the *Filter* removes from C_d those URLs with no or little interest to the user, contributing to reduce the processing costs of the next steps. For this, we adopt a simple procedure that mimics the behavior of typical users who usually examine only results whose titles are minimally similar to the title of the requested document. In particular, we consider that the title t_c of a candidate document is similar to the title t_d of the requested document if the Jaccard similarity coefficient (Tan, Steinbach, & Kumar, 2005), computed over their terms, according to Eq. (1), is greater than a certain threshold *j*. We used the Jaccard coefficient because it is a very simple and fast alternative to other measures such as the cosine similarity (Salton, Wong, & Yang, 1975). As a result, the *Filter* generates a list F_d of *Filtered URLs*.

$$J(t_c, t_d) = \frac{|t_c \cap t_d|}{|t_c \cup t_d|}. \quad (1)$$

Next, the *Ranker* ranks the F_d list and generates a new ranked list R_d possibly using additional evidence such as the title of the returned document and the source of the result. The *Ranker* works by trying to put, on top of the ranked list, those documents with higher chance of satisfying the user's needs, as we shall see later. The R_d list is then returned to the *Digital Library Interface*, which shows it to the user.

4. Experimental environment

In order to find the best configuration for the proposed service when searching for the full text of documents whose metadata records have been taken from the two Computer Science collections described next, we first investigated the

[3] Vertical search engines are specialized search engines focused on a specific field (for instance, a search engine specialized in the biomedical and life sciences fields) or document genre (for instance, a search engine specialized in academic articles, course syllabi, or videos).

effectiveness of individual search engines for this specific task. We tested five popular search engines available on the Web: Google,[4] Yahoo!,[5] MSN Search,[6] Google Scholar,[7] and CiteSeer.[8] The first three are general purpose search engines and are among the ones with the largest audience on the Web. The last two are specialized search engines that index scientific publications, being CiteSeer focused on the Computer Science field.

Our experimental environment is a simplified version of the service architecture described in Fig. 1, where the *Sample Catalog* contains metadata records of Computer Science conference papers for which a URL is missing from the DL and each query q_d, generated for a metadata record m_d, is submitted to a single search engine in order to find a relevant URL for m_d. For each search engine, except CiteSeer, we developed a specific *Query Interface* that submits the queries directly to the respective query processor and extracts from P_d, the set of returned pages, the title and the URL of the retrieved documents in order to create the list C_d of candidate URLs. For CiteSeer, the queries were submitted to Google, using its filter option, restricted to the CiteSeer domain. This was due to the fact that CiteSeer was constantly unavailable at the time of our experiments. We have adopted such an alternative based on the results of a previous experiment in which we randomly selected 1060 records from the CiteSeer metadata catalog[9] and then submitted queries to Google requesting for related content. For $98 \pm 1\%$ of these records, using a 95% confidence interval, it was possible to retrieve at least one document with a title similar to the one in the metadata record.

To create the *Sample Catalog* we carried out a stratified random sampling of two collections: (a) a set of 3969 metadata records obtained from the complete catalog of BDBComp, a collection of papers published in proceedings of major Brazilian Computer Science conferences, and (b) a set of 3181 records extracted from DBLP, which we call DBLP-Br, corresponding to papers published by Brazilian researchers in proceedings of international conferences. Note that no records from the second collection belong to the first one. The percentage of articles without a full-text URL is about 66% in BDBComp and 36% in DBLP-Br. The resulting *Sample Catalog* comprises 200 metadata records with missing URLs.

Each metadata record m_d was then used to generate the queries to be used for requesting the full text of the respective papers. We have tested seven types of query, as illustrated by the following examples, derived from the article *Data Extraction By Example* whose authors, as cataloged at DBLP, are Laender, Ribeiro-Neto and da Silva:

- *AS*: surnames of all cataloged authors (e.g., *Laender Ribeiro-Neto Silva*).
- *UT*: unquoted title (e.g., *DEByE – Data Extraction By Example*).
- *UT + FS*: unquoted title followed by the surname of the first cataloged author (e.g., *DEByE – Data Extraction By Example Laender*).
- *UT + AS*: unquoted title followed by the surnames of all cataloged authors (e.g., *DEByE – Data Extraction By Example Laender Ribeiro-Neto Silva*).
- *QT*: quoted title (e.g., "*DEByE – Data Extraction By Example*").
- *QT + FS*: quoted title followed by the surname of the first cataloged author (e.g., "*DEByE – Data Extraction* By Example" Laender).
- *QT + AS*: quoted title followed by the surnames of all cataloged authors (e.g., "*DEByE – Data Extraction By Example*" *Laender Ribeiro-Neto Silva*).

For each query q_d, the *Query Interface* generated a list C_d of candidate URLs with the 40 first results returned. From this list, the *Filter* removed elements whose titles lead to Jaccard coefficients lower than the threshold value[10] of 0.22, giving rise to a list F_d of filtered URLs.

5. Evaluation

After submitting the queries to the five search engines, we combined the obtained result lists into a single one. This procedure resulted in 3676 pairs (m_d, u), where u is a plausible URL for accessing the full text (or related information) of a specific paper a. To classify these pairs according to their usefulness, we first defined the different user scenarios we are interested.

Different users of a DL may have different interests and needs in different circumstances. For instance, a user looking for material on a certain topic may be interested in having access only to the abstracts of the related articles. Having found these abstracts, the interest of this user may be shifted towards their full-text content. However, if this user is not prepared to pay for this content, she might prefer to have access only to those articles whose full texts are freely accessible. Later on, when complementing some bibliographic references, this same user may become interested in missing metadata such as page

[4] http://www.google.com/.
[5] http://search.yahoo.com/.
[6] http://search.msn.com/.
[7] http://scholar.google.com/.
[8] http://citeseer.ist.psu.edu/.
[9] Obtained from http://citeseer.ist.psu.edu/oai.html.
[10] This threshold was chosen to keep the number of selected results manageable, since they would be manually classified for evaluation.

Table 1
User scenarios (note that, in the accessibility row, an 'f' stands for *free* and an 'r' stands for *restrict*).

Content	Full text		Similar full text		Useful metadata		Similar metadata		Redundant metadata		Other	
Accessibility	f	r	f	r	f	r	f	r	f	r	f	r
Scenario												
Strict	X	X										
Strict & *Free*	X											
Flexible	X	X	X	X								
Flexible & *Free*	X		X									
Highly Flexible	X	X	X	X	X	X						
Strict & *Restricted*		X										
At Least Metadata	X	X	X	X	X	X	X	X	X	X		
No Requirements	X	X	X	X	X	X	X	X	X	X	X	X

numbers. Therefore, different sort of material may be useful in different situations. Thus, in this context, bibliographic material can be classified according to its content and accessibility, as described below.

With respect to content, we have considered the items in our result list as belonging to the following six categories:

(1) *Full text*: the URL u points to the full text (or to a document containing a pointer to it) of the article described by m_d.
(2) *Similar full text*: the URL u points to the full text (or to a document containing a pointer to it) of a document d similar to a, such as another related article or a thesis or dissertation authored by one of the authors of a.
(3) *Useful metadata*: the URL u does not belong to any of the above categories and points to a document containing metadata information about a not present in m_d.
(4) *Similar metadata*: the URL u does not belong to any of the above categories and points to a document containing metadata that describe a document d similar to a, such as a related article or a thesis or dissertation authored by one of the authors of a.
(5) *Redundant metadata*: the URL u does not belong to any of the above categories and points to a document describing only metadata information already present in m_d.
(6) *Others*: the URL u does not belong to any of the above categories.

Regarding accessibility, we have considered the items in our result list as belonging to the following two categories:

(1) *Restricted*: the URL u provides access to the full text (or related document) by means of some sort of payment or subscription.
(2) *Free*: the URL u provides free access to the full text (or related document).

Based on the previously described categories, we derived eight scenarios that model users with different interests, as shown in Table 1. In this table, each scenario is derived based on the type of content a user is interested in and the required accessibility, i.e., free (f) or restricted (r). In the *Strict* scenario, users are interested only in the full text of an article, no matter how it can be accessed, while in the *Strict* & *Free* scenario, the full text is required to be freely accessible. In the flexible scenarios, besides the full text, related documents and unknown useful metadata may also satisfy the users. The scenario *Strict* & *Restricted* covers, for example, users interested only in "official" documents coming from trusted sources. In the *At Least Metadata* scenario, additionally to all relevant documents considered in the *Highly Flexible* scenario, users are also interested in documents that contain either redundant metadata describing a searched article or metadata describing another document similar to the desired one, no matter how they can be accessed. Finally, in the *No Requirements* scenario, users have no requirements. Any kind of content related to a searched article is considered relevant, no matter also how it can be accessed. The last three scenarios are defined only for the purpose of analyzing coverage, as discussed in Section 7.

We then asked 27 subjects, members of our research group, to classify the resulting 3676 pairs as useful or not according to the previously described usage scenarios. For instance, a page containing additional metadata related about a paper but not its full text could be considered useful in the *At Least Metadata* scenario, but not useful in the *Strict* scenario.

To evaluate our results, we use three metrics: average precision at seen relevant documents (P_q), mean average precision (*MAP*), and mean reciprocal rank (*MRR*).

A good ranked list should maximize the placement of relevant content near to the top positions since these are the most likely positions to be inspected by the users. To accomplish this, the evaluation metric should take into consideration the number of relevant results and the order in which they appear. An example of such metric is P_q, which is based on the *non-interpolated average precision*, a measure commonly used in TREC evaluations (Harman, 1996). It is defined as

$$P_q = \frac{1}{n}\left(\sum_{i=1}^{m} r_{q_i} \times \frac{1}{i}\sum_{j=1}^{i} r_{q_j}\right), \tag{2}$$

Table 2
MAP intervals for each type of query in the *Strict* scenario.

Type of query	Google		Yahoo!		MSN		Scholar		CiteSeer	
	j	*MAP* (%)	*j*	*MAP* (%)	*j*	*MAP* (%)	*j*	*MAP* (%)	*j*	*MAP* (%)
AS	0.28	23.5 ± 6.5	0.22	28.0 ± 8.8	0.22	27.3 ± 19.0	0.32	30.6 ± 8.4	0.31	59.4 ± 15.9
UT	0.26	37.9 ± 7.1	0.34	58.8 ± 9.7	0.27	25.6 ± 18.9	0.37	54.9 ± 9.2	0.24	64.3 ± 14.2
UT + FS	0.23	$\mathbf{39.0 \pm 7.2}$	0.22	$\mathbf{63.8 \pm 9.7}$	0.22	53.4 ± 20.6	0.36	$\mathbf{71.6 \pm 7.4}$	0.23	$\mathbf{66.4 \pm 13.7}$
UT + AS	0.26	36.4 ± 7.6	0.30	54.5 ± 10.2	0.22	62.5 ± 19.9	0.36	66.3 ± 8.1	0.29	57.0 ± 15.6
QT	0.22	32.2 ± 7.4	0.22	57.4 ± 10.0	0.22	$\mathbf{75.0 \pm 17.8}$	0.30	66.2 ± 8.3	0.22	55.3 ± 16.0
QT + FS	0.22	31.8 ± 7.4	0.22	54.7 ± 10.0	0.22	$\mathbf{75.0 \pm 17.8}$	0.30	65.9 ± 8.2	0.22	45.3 ± 16.5
QT + AS	0.26	28.8 ± 6.6	0.30	49.7 ± 10.2	0.22	64.8 ± 20.1	0.30	63.1 ± 8.5	0.22	51.6 ± 16.8

where m is the number of documents returned as result for a query q, n is the total number of relevant documents for q, and r_{q_i} is 1 if the ith returned document is relevant or 0 otherwise. This metric intuitively weights higher relevant documents that appear on top positions of the ranked list.

Since P_q provides only one evaluation per query, and we need a metric to evaluate a whole set of queries submitted to a search engine, we use *MAP* to provide a global estimate to a set of queries. *MAP* is the mean of the average precisions (P_q) calculated over a set of queries and defined as

$$MAP = \frac{1}{k} \sum_{i=1}^{k} P_i, \tag{3}$$

where k is the total of queries. This is exactly the mean of the average precisions obtained for the metadata records in the *Sample Catalog*. Finally, *MRR* (Voorhees, 1999) estimates how close a document is from the top of the ranking. It is defined as

$$R_q = 1/i, \tag{4}$$

where i is the position of the first relevant document observed as result for query q. This metric assumes that users are usually interested only in one item and that this item should be ranked at the first position. Thus, it is very useful in this work because users interested in the full text of an article usually expect that article to appear as the first result for the query submitted to find it.

For all *MAP* and *MRR* results, we report value intervals with a 95% confidence level. We assume c, the central value of an interval $c \pm e$ as being its representative value. For all reported comparisons, we tested the statistical significance using the pairwise t-test (Jain, 1991). We consider statistically significant the results with, at least, 95% confidence level.

6. Results and discussion

In this section, we present the results obtained, considering the five first scenarios described in Table 1. In Section 6.1, we analyze seven types of query for the task of retrieving the full text of an article. In Section 6.2, we compare the five search engines considering, for each one, the best query type in each scenario. In Section 6.3, we present strategies to improve the overall performance of our proposed process.

6.1. Query type analysis

Tables 2 and 3 present the MAP and MRR value intervals achieved by the seven types of query for each tested search engine considering the scenario *Strict*. We focus our analysis on this scenario because users in scenario *Strict* are only interested in finding the full text of a specific article cataloged in a DL, our main goal in this study. The Jaccard coefficient threshold j values shown are the ones that, for each query type, achieved the highest MAP value for records of the *Sample Catalog*.

Note that the values for each query type are not comparable among different search engines, since these values have been computed considering relevance information in a pool of results obtained by the union of the results of all seven query types for the same search engine. These values can only be used to compare the relative performance of each query type for a same search engine. Global comparisons among the tested search engines are presented in Section 6.2.

Table 2 presents the MAP results for the seven types of query submitted to the five tested search engine. As we can see, in general, the best results are those of unquoted title queries (UT, UT + FS, and UT + AS). In particular, UT + FS queries are significantly better than the other ones for Google, Scholar, and CiteSeer. For Yahoo!, the apparently best performance of unquoted queries over the other types was not statistically significant. In general, AS queries presented a very poor performance, except for CiteSeer. For MSN, the number of papers for which we have been able to retrieve relevant results was only 10% of the total, which makes it hard to draw any significant conclusion. Despite this, we note in MSN a very poor performance for UT queries and a general advantage of quoted queries. As we can see from Table 3, much of the observation drawn for MAP results holds when we analyze the MRR values.

Table 3
MRR intervals for each type of query in the *Strict* scenario.

Type of query	Google		Yahoo!		MSN		Scholar		CiteSeer	
	j	*MRR* (%)	*j*	*MRR* (%)	*j*	*RR* (%)	*j*	*MRR* (%)	*j*	*MRR* (%)
AS	0.28	50.8 ± 10.0	0.22	38.9 ± 11.3	0.22	29.6 ± 20.1	0.32	34.5 ± 9.4	0.31	72.5 ± 16.6
UT	0.26	$\mathbf{76.6 \pm 8.7}$	0.34	74.0 ± 10.4	0.27	28.2 ± 20.1	0.37	59.6 ± 9.7	0.24	84.3 ± 13.7
UT + FS	0.23	76.4 ± 8.5	0.22	$\mathbf{74.8 \pm 10.2}$	0.22	61.4 ± 21.6	0.36	$\mathbf{78.3 \pm 7.6}$	0.23	$\mathbf{86.8 \pm 13.2}$
UT + AS	0.26	70.5 ± 9.4	0.30	66.9 ± 11.0	0.22	70.5 ± 20.1	0.36	74.3 ± 8.5	0.29	76.8 ± 16.4
QT	0.22	65.4 ± 10.0	0.22	69.4 ± 10.8	0.22	$\mathbf{79.6 \pm 17.7}$	0.30	73.8 ± 8.5	0.22	80.8 ± 16.3
QT + FS	0.22	64.9 ± 9.9	0.22	66.8 ± 11.0	0.22	$\mathbf{79.6 \pm 17.7}$	0.30	73.8 ± 8.5	0.22	68.8 ± 19.2
QT + AS	0.26	66.9 ± 10.2	0.30	62.4 ± 11.3	0.22	70.5 ± 20.1	0.30	72.0 ± 8.8	0.22	77.3 ± 17.2

Table 4
Comparison of the search engines according to MAP intervals. Gains are given as percent values over the search engine immediately below.

Search engines	*Strict*		*Strict* & *Free*		*Flexible*		*Flexible* & *Free*		*Highly Flexible*	
	MAP (%)	Gain	*MAP* (%)	Gain	*MAP* (%)	Gain	*MAP* (%)	Gain	*MAP* (%)	Gain
Scholar	32.6 ± 6.2	**117.9**	31.7 ± 7.3	**129.7**	29.4 ± 5.1	**63.4**	23.7 ± 5.3	**48.7**	29.9 ± 5.1	**47.9**
Google	15.0 ± 3.6	17.4	13.8 ± 4.2	−13.9	18.0 ± 4.2	**61.7**	15.9 ± 4.5	13.7	20.2 ± 4.2	**93.0**
Yahoo!	12.7 ± 4.2	**241.3**	15.7 ± 5.3	**161.6**	11.1 ± 3.5	**108.1**	14.0 ± 4.4	60.8	10.5 ± 3.3	38.1
CiteSeer	3.7 ± 1.8	14.5	6.0 ± 2.7	55.3	5.3 ± 2.6	58.6	8.7 ± 3.7	111.9	7.6 ± 3.0	**120.3**
MSN	3.3 ± 2.2	–	3.8 ± 2.7	–	3.4 ± 2.4	–	4.1 ± 2.8	–	3.4 ± 2.3	–

Although intuitively quoted title queries were expected to return better results, it seems that quoted terms exclude many relevant results mainly due differences between the document title cataloged in the DL and the title of results returned by the search engines. These differences may happen for a number of reasons such as typographical errors at cataloguing time and conversion errors of PDF documents, particularly when the document titles include formulas and subscripts/superscripts.

6.2. Comparison among search engines

The comparison among search engines has been carried out by using the most effective query type for each tested search engine in each scenario. For example, in scenario *Strict*, we use query types QT + FS for MSN and UT + FS for the other search engines. Relevance assessments have been made with regard to a global pool, i.e., the union of the results of all types of query for all search engines in a specific scenario. The analysis of the results is shown next. Note that MAP and MRR values are comparable only within a specific scenario.

Table 4 shows the effectiveness of the search engines according to the MAP metric. For each search engine, we also show gains over the search engine immediately below. Bold values correspond to statistically significant gains. Scholar clearly outperforms the other search engines for our particular task. Its gains, however, are increasingly smaller as the user is more flexible regarding which she considers a relevant material. Google and Yahoo! present similar performances, being Google significantly better than Yahoo! only when restricted content is considered relevant (scenarios *Flexible* and *Highly Flexible*). Yahoo! slightly outperforms Google in scenario *Strict* & *Free* probably due to the great amount of restricted content indexed by Google. MSN and CiteSeer have not shown to be good alternatives in general.

As shown in Table 5, results for MRR are similar to those obtained with MAP but with gains of smaller magnitude. Additionally, in flexible scenarios, Scholar and Google present similar performance. Thus, we can say that users do not notice much difference between Scholar and Google when they are only interested in one relevant result that is close to the top of the ranking.

Table 5
Comparison of the search engines according to MRR intervals. Gains are given as percent values over the search engine immediately below.

Search engines	*Strict*		*Strict* & *Free*		*Flexible*		*Flexible* & *Free*		*Highly Flexible*	
	MRR (%)	Gain	*MRR* (%)	Gain	*MRR* (%)	Gain	*MRR* (%)	Gain	*MRR* (%)	Gain
Scholar	65.8 ± 8.4	**20.0**	55.9 ± 9.5	**10.0**	69.8 ± 7.9	4.8	50.7 ± 8.7	−10.6	70.2 ± 7.8	−0.4
Google	54.8 ± 9.0	17.7	50.8 ± 10.3	14.6	66.6 ± 7.8	**33.2**	56.7 ± 9.0	21.9	70.5 ± 7.5	**42.7**
Yahoo!	46.5 ± 9.4	**127.3**	44.3 ± 10.2	**77.9**	50.0 ± 8.7	**93.7**	46.5 ± 9.1	**59.9**	49.4 ± 8.5	**46.4**
CiteSeer	20.5 ± 7.7	24.0	24.9 ± 9.2	49.7	25.8 ± 7.5	52.4	29.1 ± 8.2	**56.1**	33.7 ± 8.1	**81.6**
MSN	16.5 ± 7.1	–	16.7 ± 7.9	–	16.9 ± 6.6	–	18.6 ± 7.3	–	18.6 ± 6.7	–

Table 6
Comparison of Scholar with the combinations SGC′ and SGC″ in each scenario.

Scenario	Scholar	SGC′		SGC″	
	MAP (%)	*MAP* (%)	*G* (%)	*MAP* (%)	*G* (%)
Strict	32.6 ± 6.2	43.2 ± 6.1	**32.6**	35.0 ± 6.2	7.2
Strict & *Free*	31.7 ± 7.3	39.7 ± 7.2	**25.2**	34.2 ± 7.3	7.8
Flexible	29.2 ± 5.1	43.2 ± 5.4	**47.9**	35.0 ± 5.5	**19.8**
Flexible & *Free*	23.6 ± 5.3	35.5 ± 6.0	**50.2**	30.1 ± 6.0	**27.6**
Highly Flexible	29.7 ± 5.1	46.5 ± 5.4	**56.5**	37.1 ± 5.5	**25.0**

6.3. Improvements

By analyzing our results, we observed that the tested search engines have different coverages of the Web. This is consistent with the literature (Bharat & Broder, 1998). Further, the ranking yielded by Scholar is oriented towards more reliable sources that, in general, provide no free access to their content. This ranking is also heavily influenced by citation count. Based on these observations, in this section, we present strategies to improve our best results. In particular, in Section 6.3.1, we describe how to combine rankings to improve content coverage. In Section 6.3.2, we study re-ranking strategies aimed at improving the quality of the retrieval, including strategies to promote free content. As seen in previous sections, conclusions drawn from MRR and MAP results are very similar, therefore, in the next sections we only analyze MAP results.

6.3.1. Combination strategies

Our main aim here is to analyze some possible strategies for combining the results returned by Scholar and Google, the two search engines with the best performance.[11] Note that it is not our intent to produce the "best" combination strategy but only to show that even simple combinations can improve results for this specific task. In particular, we test two strategies, which we call SGC′ and SGC″. In SGC′, a query is sent to both search engines and the final answer is composed of Scholar documents followed by those returned by Google, after removing duplicates. SGC″ is similar to SGC′, but in this case a query is sent to Google only if Scholar returns no answer.

As we can see in Table 6, our combination strategies yielded significant gains in all scenarios, except SGC″ in the case of scenarios *Strict* and *Strict* & *Free*. These gains can be credited to the existence of relevant URLs covered by Google but not by Scholar. Gains are higher in scenarios in which there is more flexibility. We also note that SGC′ gains are the highest, which can be explained by the larger coverage of this strategy.

6.3.2. Re-ranking strategies

Having considered the above combination strategies, in order to better assess their retrieval effectiveness we then compare Scholar and Google rankings with an ideal one, i.e., a ranking where all relevant entries appear on top of the irrelevant ones. From this comparison, we observed that there is a large room for improving Scholar results in the scenarios *Strict*, *Strict* & *Free*, and *Flexible* & *Free*. With respect to Google, the room for gains is more limited in all scenarios.

The larger room for gains with Scholar is due to the fact that its ranking is strongly influenced by citation count, which in some cases becomes more important than the actual similarity between the query and the document. Additionally, Scholar "prefers" URLs coming from DLs of well-known publishers such as ACM, IEEE, Springer and Elsevier in detriment of free ones. This makes sense since the "official" URL is probably the most reliable one. However, such a preference obviously does not work in scenarios in which restricted content is considered irrelevant.

In order to improve Scholar's ranking, we reorder document positions of the result list as follows. Given two documents in the result list, f_i and f_j, we change their positions if the title of f_j is more similar to the document title than the title of f_i since the higher the similarity between the titles the higher the likelihood of they being related. In case of similar titles, we change positions if the URL frequency of the domain[12] of f_j is greater than that of f_i since we believe that the higher the domain frequency of an document full-text source the smaller the likelihood that it will supply free content.

To calculate the title similarity, we used the cosine distance weighting terms according to the traditional TF-IDF weighting scheme (Baeza-Yates & Ribeiro-Neto, 1999). To calculate the domain frequencies, we sampled 200 metadata records of missing URL papers from our test collection. From these records, we built queries and submitted them to Scholar. We then calculated the domain frequencies from the resulting lists in a pre-processing batch mode based on the number of times a particular domain appears in the results for those queries. In sum, our hypothesis is that, when two titles in the list of results are similar, the one coming from the most infrequent domain (e.g., personal or research group lab pages) have a higher probability of providing free access to the full text of documents than those coming from a highly frequent domain (e.g., large publishers or commercial DLs).

[11] We have also tested other search engine combinations following the same procedures, but no significant improvements have been achieved.

[12] In this work, we consider that a domain is a string typically in a three-level "server.organization.type" format, used to identify an entity on the Internet, such as an organization (e.g., http://www.acm.org).

Table 7
Comparison of the combinations SGC′ and SGC″ with their re-ranked versions RSGC′ and RSGC″.

Scenario	SGC′	RSGC′		SGC″	RSGC″	
	MAP (%)	*MAP* (%)	*G* (%)	*MAP* (%)	*MAP* (%)	*G* (%)
Strict	43.2 ± 6.1	46.5 ± 6.4	**7.5**	35.0 ± 6.2	38.2 ± 6.6	**9.2**
Strict & *Free*	39.7 ± 7.2	44.6 ± 7.6	**12.5**	34.2 ± 7.3	39.1 ± 7.8	**14.5**
Flexible	43.2 ± 5.4	43.6 ± 5.5	1.0	35.0 ± 5.5	35.4 ± 5.5	1.3
Flexible & *Free*	35.5 ± 6.0	38.0 ± 6.2	**7.3**	30.1 ± 6.0	32.7 ± 6.2	**8.6**
Highly Flexible	46.5 ± 5.4	47.0 ± 5.5	1.0	37.1 ± 5.5	37.6 ± 5.5	1.2

We applied the described re-ranking strategy to Scholar obtaining gains of about 10% in scenario *Strict*, 16% in scenario *Strict* & *Free*, and 11% in scenario *Flexible*. Gains in other scenarios were not significant. When applied to Google, the reordering resulted in a worse performance, which means that Google's ranking is already a good one. We also applied the described re-ranking strategy to SGC′. In this case, only the answers from Scholar are reordered. We refer to this new strategy as RSGC′. Likewise, we applied the same re-ranking strategy to SGC″ yielding a new strategy called RSGC″. Results are shown in Table 7.

As expected, gains for the re-ranking strategies are higher in scenarios that require free accessibility. Note, however, that there is no meaning on re-ranking results for the cases in which URLs are only provided by either a free source or by a restricted one. In Table 8, we show results of experiments that apply the proposed combination and re-ranking method only to

Table 8
Impact of re-ranking over the results of the combinations SGC′ and SGC″ for the restricted and free access cases.

Scenario	SGC′	RSGC′		SGC″	RSGC″	
	MAP (%)	*MAP* (%)	*G* (%)	*MAP* (%)	*MAP* (%)	*G* (%)
Strict & *Free*	36.6 ± 9.5	46.4 ± 11.8	**26.9**	34.5 ± 10.1	45.6 ± 12.9	**32.2**
Flexible & *Free*	34.8 ± 7.3	40.8 ± 7.9	**17.5**	34.4 ± 7.5	41.6 ± 8.1	**20.7**

Table 9
Comparison between the combinations RSGC″ and RSGC′ with respect to compensatory gains.

Scenario	RSGC″	RSGC′		
	MAP (%)	*MAP* (%)	*G* (%)	CG (%)
Strict	38.2 ± 6.6	46.5 ± 6.4	**21.7**	9.2
Strict & *Free*	39.1 ± 7.8	44.6 ± 7.6	**14.1**	14.1
Flexible	35.4 ± 5.5	43.6 ± 5.5	**23.1**	2.2
Flexible & *Free*	32.7 ± 6.2	38.0 ± 6.2	**16.3**	7.8
Highly Flexible	37.6 ± 5.5	47.0 ± 5.5	**25.0**	1.6

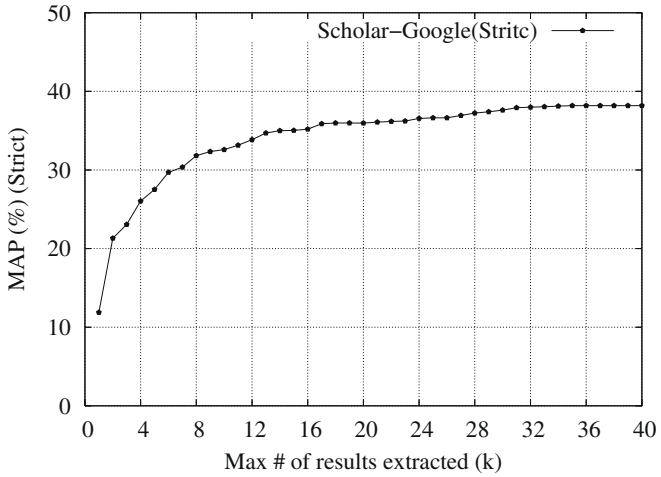

Fig. 2. Effectiveness *versus* maximum number of extracted results.

Table 10
Coverage comparison in each scenario.

Scenario	Min (%)	Mean (%)	Max (%)
No Requirements	60.1	66.5	72.9
At Least Metadata	56.5	63.0	69.5
Highly Flexible	48.8	55.5	62.2
Flexible	44.8	51.5	58.3
Strict	35.8	42.5	49.2
Flexible & Free	33.4	40.0	46.6
Strict & Free	26.7	33.0	39.4
Strict & Restricted	13.3	18.5	23.7

documents for which we had both free and restricted sources, considering scenarios *Strict* & *Free* and *Flexible* & *Free*. As we can see, our gains in these situations are even higher. Similarly, by applying the re-ranking strategy without considering the frequency of the URLs, we obtained gains smaller and not significant.

Unlike strategy RSGC″, RSGC′ always requires the inspection of two rankings. We now investigate whether the effectiveness gains of RSGC′ compensate the additional costs. For this, we define a compensatory gain as the one in which there is an increase in the effectiveness of RSGC′ when no relevant URL was found by using RSGC″. Other types of gain, such as new relevant URLs returned by RSGC′ when it is already possible to find relevant content with RSGC″ are considered non-compensatory.

Table 9 shows the gains (G) of RSGC′ over RSGC″ for all considered scenarios and the percentage of these gains coming from compensatory gains (CG). We can see that the compensatory gains were inferior to 15% of the overall gain in all scenarios. Thus, due to the extra effort to extract and process two lists of results in strategy RSGC′, we consider RSGC″ as the more appropriate combination in terms of both effectiveness and processing cost (from now on, we refer to RSGC″ as Scholar–Google combination). Further, if processing time is a critical problem, we can always reduce the maximum number of candidates to be extracted. For instance, Fig. 2 shows for the scenario *Strict* the impact of the number of extracted ranking results on effectiveness of the Scholar–Google combination. As we can see, reducing the number of extracted results to half (from 40 to 20) led to just a slightly decreasing in effectiveness.

7. Coverage analysis

In this section, we analyze some factors that may affect the likelihood of finding the URL of a specific full-text document. For this, we use the notion of coverage, that is, the proportion of documents for which a search engine finds relevant information. More formally, we define coverage as

$$C_{qcs} = \frac{m}{n}, \tag{5}$$

where n is the total number of documents for which a query q is submitted and m is the number of documents for which at least one relevant document is returned by search engine s as result of q in scenario c.

Since the cost of computing C_{qcs} can be prohibitive due to the need of a manual inspection of the results, we estimate the coverage as a variation interval using a random sampling. Thus, we now compare the effectiveness of the search engines according to their coverage over the resulting sample. Variation intervals are estimated with a 95% confidence level. Intervals without intersection correspond to significantly different coverages.

The results reported in this section are based on the Scholar–Google combination considering the best query types for these two search engines in the scenario *Strict*. We start by analyzing the influence of the users' requirement on the results. Table 10 shows the coverage intervals for the scenarios that we have studied so far considering the sample of 200 metadata records we have used in our experiments. For completeness, we include all the scenarios described in Table 1.

In Table 10, we show minimum, mean, and maximum values for the collection coverage interval in each scenario ordered from the highest (least restrictive) to the lowest (most restrictive) mean values. Coverage for all scenarios below *Highly Flexible* are significantly lower than for the scenario *No Requirements*. This difference shows that the coverage of less restrictive scenarios can be significantly improved with respect to the more restrictive ones. We also note that the percentage of documents for which is possible to obtain free access to its full text is significantly higher than for those with restricted access.

8. Conclusions

We have proposed in this article a process that uses results from queries submitted to search engines for finding the URL of the corresponding full text (or of any relevant related material) for those documents cataloged in a DL but for which this information is missing. This process can be used in the implementation of a service for users in the case a DL does not offer pointers to the full text of certain items catalogued or cited in its collections. The service would be also useful if the pointers that a DL contain are broken or point only to restricted items, by means of payment, and the user is not willing to complete

the transaction. Finally, such a service would also be useful to obtain additional metadata about an item or to discover related material.

We have presented a comprehensive study of this process for conference papers in the Computer Science field by investigating different query strategies applied to several search engines and user scenarios. According to our experimental results, we have concluded that Scholar is the best alternative for this task. Google can be considered as the second alternative, but achieving a performance equivalent to Yahoo! in scenarios where users consider only documents that can be freely accessed. For these three search engines, queries that include the unquoted title along with the surname of the first author (UT + FS) are more effective than those that include quoted titles (QT + FS). CiteSeer and MSN have not shown to be good alternatives for this particular task.

Further, we have shown that, by using a combination of Google and Scholar along with a re-ranking strategy, the overall quality of the process is significantly improved. We have also shown that some reduction in the processing costs can be achieved with low impact on the overall effectiveness. In addition, we have noticed that the percentage of papers in the test collections for which is possible to obtain free access to the full text is significantly higher than for those with restricted access.

Finally, given the fact that search engines such as Scholar also have a good coverage of other fields (Walters, 2007), an interesting future work would be to experiment with fields such as Health and Medical Sciences or Physics, which have a large body of knowledge published on the Web. We also intend to study how digital libraries can be enriched by the process discussed here. For this, an experimental service based on the process proposed in this article has been deployed to BDBComp (Santos, Silva, Santos, Laender, & Gonçalves, 2007) and is under evaluation.

Acknowledgements

This research is partially supported by the MCT/CNPq/CT-INFO projects 5S-VQ (Grant Number 551013/2005-2) and InfoWeb (Grant Number 550874/2007-0), and by the authors' scholarships and individual research grants from CAPES and CNPq.

References

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. New York, NY, USA: Addison Wesley.

Bharat, K., & Broder, A. (1998). A technique for measuring the relative size and overlap of public web search engines. *Computer Networks and ISDN Systems, 30*(1–7), 379–388.

Chakrabarti, S., van den Berg, M., & Dom, B. (1999). Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks, 31*(11–16), 1623–1640.

Chu, H., & Rosenthal, M. (1996). Search engines for the World Wide Web: A comparative study and evaluation methodology. In *Proceedings of the annual conference of the American society for information science, Baltimore, MD, USA* (pp. 127–135).

Gordon, M., & Pathak, P. (1999). Finding information on the World Wide Web: The retrieval effectiveness of search engines. *Information Processing and Management, 35*(2), 141–180.

Harman, D. K. (1996). Overview of the fourth Text REtrieval Conference TREC-4. In *Proceedings of the fourth Text REtrieval Conference (TREC-4), Gaithersburg, Maryland, USA* (pp. 1–24).

Harrison, T. L., & Nelson, M. L. (2006). Just-in-time recovery of missing web pages. In *Proceedings of the 17th ACM conference on hypertext and hypermedia, Odense, Denmark* (pp. 145–156).

Hoff, G., & Mundhenk, M. (2001). Finding scientific papers with HomepageSearch and MOPS. In *Proceedings of the 19th annual international conference on computer documentation, Santa Fe, New Mexico, USA* (pp. 201–207).

Jain, R. (1991). *The art of computer systems performance analysis*. New York, NY, USA: John Wiley and Sons, Inc.

Laender, A. H. F., Gonçalves, M. A., & Roberto, P. A. (2004). BDBComp: Building a digital library for the Brazilian computer science community. In *Proceedings of the fourth ACM/IEEE joint conference on digital libraries, Tucson, Arizona, USA* (pp. 23–24).

Laender, A. H. F., Gonçalves, M. A., Cota, R. G., Ferreira, A. A., Santos, R. L. T., & Silva, A. J. C. (2008). Keeping a digital library clean: New solutions to old problems. In *ACM symposium on document engineering, São Paulo, Brazil* (pp. 257–262).

Lawrence, S., & Giles, C. L. (1998). Searching the World Wide Web. *Science, 280*(5360), 98.

Lawrence, S., & Giles, C. L. (1999). Accessibility of information on the web. *Nature, 400*, 107–109.

On, B.-W, & Dongwon (2004). PaSE: Locating online copy of scientific documents effectively. In *Proceedings of the seventh annual international conference of Asian digital libraries, Shanghai, China* (pp. 408–418).

Qin, J., Zhou, Y., & Chau, M. (2004). Building domain-specific web collections for scientific digital libraries: A meta-search enhanced focused crawling method. In *Proceedings of the fourth ACM/IEEE-CS joint conference on digital libraries, Tucson, AZ, USA* (pp. 135–141).

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Santos, R. L. T., Silva, A. J. C., Santos, H. S., Laender, A. H. F., & Gonçalves, M. A. (2007). A component-based digital library service for finding missing documents. In *Proceedings of the 22nd Brazilian symposium on databases, João Pessoa, PB, Brazil* (pp. 5–19).

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Voorhees, E. M. (1999). The TREC-8 question answering track report. In *Proceedings of TREC-8* (pp. 77–82).

Walters, W. H. (2007). Google Scholar coverage of a multidisciplinary field. *Information Processing and Management, 43*(4), 1121–1132.

Zhuang, Z., Wagle, R., & Giles, C. L. (2005). What's there and what's not? Focused crawling for missing documents in digital libraries. In *Proceedings of the fifth ACM/IEEE-CS joint conference on digital libraries, Denver, CO, USA* (pp. 301–310).

Allan J.C. Silva received a Computer Science M.Sc. degree from the Federal University of Minas Gerais, Brazil, in 2007. His research is focused on digital libraries and information retrieval topics. Currently, he is a database analyst at SYDLE, a company specialized in the development of mission critical systems.

Marcos André Gonçalves is an Assistant Professor at the Federal University of Minas Gerais, Brazil. He concluded his doctoral degree in Computer Science at Virginia Tech, USA, in 2004. His research interests include digital libraries, information retrieval, and machine learning.

Alberto H.F. Laender is a Computer Science Professor at Federal University of Minas Gerais, Brazil. He received his Ph.D. degree in Computing from the University of East Anglia, UK, in 1984. He is a member of the ACM SIGMOD Advisory Board and of the ACM SIGMOD Ph.D. Dissertation Award Committee. His research interests include database modeling and design methods, cooperative database user interfaces, web data management, digital libraries, and web information systems.

Marco A.B. Modesto is an M.Sc. student at the Information Retrieval Group in the Computer Science Department of Federal University of Minas Gerais and a Business Intelligence Analyst at Gerdau Group, Brazil. His research interests include data extraction, web crawling, digital libraries, and data warehousing.

Marco Cristo is an Assistant Professor at FUCAPI, Brazil. He concluded his doctoral degree in Computer Science at the Federal University of Minas Gerais in 2007. His research interests include information retrieval, machine learning, digital libraries, and web advertising.

Nivio Ziviani is a Professor Emeritus in Computer Science at the Federal University of Minas Gerais, Brazil. He holds a Ph.D. in Computer Science from the University of Waterloo, Canada, 1982. He is a member of the Brazilian Academy of Sciences and the National Order of the Scientific Merit in the class Commendador. His research interests include algorithms, information retrieval, text indexing, text searching, and web information systems.