PaSE: Locating Online Copy of Scientific Documents Effectively

Byung-Won On[1] and Dongwon Lee[2]

[1] Department of Computer Science and Engineering &
[2] School of Information and Sciences and Technology,
The Pennsylvania State University, PA 16802, USA
on@cse.psu.edu, dongwon@psu.edu

Abstract. The need for fast and vast dissemination of research results has led a new trend such that more number of authors post their documents to personal or group Web spaces so that others can easily access and download them. Similarly, more and more researchers use online search for accessing documents of interest in Web, instead of paying a visit to libraries. Currently, to locate and download an online copy of a particular document *D*, one typically (1) uses Search Engines with the citation information and browses through returned web pages (e.g., author's homepage) to see if any contains *D*, or (2) uses searching facilities of an individual Digital Library (e.g., CiteSeer, e-Print) looking for *D*, and if not found, repeats the search in another Digital Library. However, the scheme (1) involves human browsing to get to the final online copy, while the scheme (2) suffers from incomplete coverage. To remedy these shortcomings, in this paper, we present a system, named as *PaSE*, which can effectively locate online copies (e.g., PDF or PS) of scientific documents using citation information. We consider a myriad of alternatives in crawling and parsing the Web to arrive at the right document quickly, and present a preliminary experimental study. Using some of the best alternatives that we have identified, we show that PaSE can locate online copy of documents more accurately and conveniently than human users would do at the cost of elongated search time.

1 Introduction

With the arrival of the World-Wide Web, authors often post their documents onto personal web space for others' easy access and fast dissemination of ideas. Recent study [19] also shows that online scientific documents are more likely to be cited than offline ones, boosting this phenomenon. As such a trend continues, the way researchers look for interesting documents changes as well; instead of searching through catalogues in the traditional library, researchers now search for online copies of documents via (1) Search Engines such as Google [11], or (2) Digital Libraries such as DBLP [13], CiteSeer [14], e-Print arXiv [15], research repositories [16]. Let us call the former as *SE-scheme* and the latter as *DL-Scheme*. For instance, to download the latest paper, one often enters citation data to Google to find author's home page, where a downloadable PDF version of the document may be found.

Z. Chen et al. (Eds.): ICADL 2004, LNCS 3334, pp. 408-418, 2004.

Sometime, to find the scholar's home page, people even use a specialized Search Engine such as MOPS [1]. As another venue to look for documents, one may also search documents in some Digital Libraries, hoping to find an archived copy of the document.

Despite the excellent coverage of modern Search Engines or huge amount of archived documents of Digital Libraries, however, these schemes are not without problems. For instance, SE-Scheme assumes human users. That is, when Google returns a list of candidate pages (mostly HTML web pages) that are likely to contain the target document, a human must sift through the links and determine which one to follow further. Such a task can be trivial for human users, but no so trivial for software agents. Search Engines like Google can be advised to search for only specific document formats (e.g., PDF, PS, DOC) in advanced interfaces, but only so at the cost of decreased precision and recall. Similarly, in DL-Scheme, since the coverage of Digital Libraries is limited, when a document is not found in one Digital Library, one has to continue the search in next Digital Libraries. The limited access to Digital Libraries (e.g., subscription is required to access ACM or IEEE Digital Libraries) only exacerbates the problem.

To demonstrate our motivation, we ran a simple experimentation as follows. We first randomly gathered 200 real citations published from 1986 to 2004 and their corresponding PDF files (i.e., this is our solution set). Then, for each citation, we submit its "title" to Google using two interfaces – normal and advanced ones. Normal interface [11] would search any web pages (HTML, XML, XHTML, PDF, PS, PPT, DOC, etc.) that contain keywords of the title, while advanced interface [12] would search only PDF documents, excluding HTML web pages (the advanced search in Google can also be achieved by appending additional construct "`filetype:pdf`" to the normal search). For the normal search, if a PDF document identical to one of the solution set is found at web pages within at most 2 hops, starting from any of the top-10 links returned, we considered it as "Match." This is based on the recent study [9] that majority of people only look at the first returned page (i.e., 10 links) of Google. Therefore, if the first link returned from Google points to an author's home page that in turn points to the author's publication page that finally contains the PDF document with a matching citation, then it is considered to be a match.

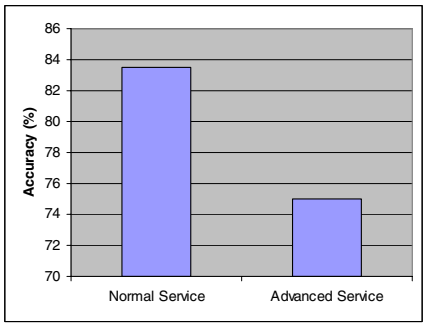

Fig. 1. Google accuracy

Fig. 1 shows the result of our experimentation. Note that normal service of Google can locate online copies of documents using its citation information with about 83% accuracy, where accuracy is `#_of_match/200`. Since the test data included some of the latest citations that may not be indexed by Google yet, the overall accuracy was not near 100%. Interestingly, the accuracy drops to 75% for the advanced service. That is, using Search Engines like Google, human users can locate online copies of documents fairly well (up to 83% accuracy) since one of the top 10 links returned from Google is likely to lead to the right URL (although the link itself does not point to the PDF directly). Therefore, just a little effort of sifting through and clicking a few returned links should be sufficient to get to the PDF document. However, since advanced service excludes all such possibilities and only focus on PDF documents, its accuracy degrades significantly. Note that it is this facility similar to the *advanced service* when a software agent needs to directly locate online copy of a document since it requires less human intervention. Therefore, the goal of this research is to build a function similar to the advanced search, only with a better accuracy:

$$[PDF_1, PDF_2, \ldots] \leftarrow PaSE(citation)$$

That is, given a citation information, we want to find the online copies of the documents (e.g., PDF or PS documents) more directly and effectively. Toward this goal, in this paper, we present a software system, named as *PaSE (Paper Search Engine)*, which can locate the publically-available online copies of documents, given proper citation information. More specifically, we use the normal search of Google to implement PaSE with the following challenges to cope with:

1. Given candidate pools (i.e., top-10 links) from Google's normal search, one needs good "crawling methods" to quickly get to the right web page that is likely to contain the online copy of documents. For this, we examine the *heuristic-based Random*, *BFS*, and *DFS* crawling algorithms.
2. Once arriving at the right web page, we need to identify the right *(citation, PDF)* pair among many candidates. This is important since typical scholar's web page contains a long list of publications, where often different publications share similar titles (e.g., conference and journal versions). To make the problem simple, instead of considering all the fields of citations (i.e., title, author, venue, year, etc), we only consider the "title" field since we believe that title has much less probability of being written in different formats (compared to author name or publication venue field).

The rest of this paper is organized as follows. In Section 2, we discuss the background and related work. In Section 3, we introduce our main ideas. In Section 4, we report preliminary experimental results. Finally, some discussion and conclusion follow in Section 5.

2 Related Work

System: There are only a few known systems that bear similarities to PaSE. The MOPS [1] is an approach to seek scientific papers relevant to a pre-defined research area. It searches for web pages which are created by some active scientists of the domain, but does not search for web pages which contain matching keywords. The

name of these scientists is obtained from the DBLP server. Using HPSearch [20], MOPS first finds homepages, and research papers close to the homepages.

BibFinder [2] is an integrated bibliographic digital library on computer science domain, with links to online copies. However, it mainly focuses on citation data itself, not the online copies. Also, many times, links lead to web page near the online copies so that users have to sift through again. PaperFinder [3] is a tool that maintains user's personal profile, queries several digital libraries for new articles, and filters the results according to the profile. It is mainly designed as an add-on service to Digital Libraries.

Crawling Algorithms: Among many outgoing links in a given web page, choosing the right order of visit is an important issue for overall performance and accuracy of Search Engines. Toward this issue, [4] considers four approaches: (1) Similarity to a driving query Q, which is similar to TFIDF approach, (2) *Backlink Count*, where the priority of a visit is favored toward the link that is contained by more pages, (3) *PageRank*, that recursively defines the importance of a page to be the weighted sum of backlinks to it, and (4) *Location Metric*, in which importance of a page is a function of its location, not its contents (e.g., URLs with fewer "/" are more useful than otherwise). Fish-Search algorithm [5] is based on the assumption that relevant documents have relevant neighbors, and determines to pursue the exploration in that direction based on the relevancy. Shark-Search algorithm [6] uses a similarity engine which returns a fuzzy score between 0 and 1. Finally, an incremental crawling algorithm [7] continuously updates and refreshes the local collection of documents retrieved to have better results.

Citation Matching Algorithms: Citation matching problem is a specialization of a more general problem known as Record Linkage problem; i.e., given two lists of strings, find all pairs *a* and *b* whose distance is within some threshold. In our setting, the problem can be summarized to: given an input citation *a* and a list of citations b_1, … b_n, found in a web page, determine b_i with the smallest *distance(a,* b_i*)*. As the citation matching algorithm, [8] examined (1) word matching – token based matching, (2) word and phrase matching – variation of *n*-gram, (3) Edit distance, and (4) subfield algorithms – citation is broken into each field (author, title, etc) and compared separately.

3 Paper Search Engine (PaSE)

3.1 The Architecture

Fig. 2 illustrates the overall architecture of PaSE. As shown at the top of the figure, *a Web services client* to Google [10] uses the keyword search in the Google's normal service [11]. Since the Google Web service supports Simple Object Access Protocol (SOAP), which is a technology to allow for Remote Procedure Call (RPC) over the Web, the client program creates a SOAP request message that contains citation information entered by a user, and then sends it to Google's Web services server. After the client receives a SOAP response message from the server, it parses the SOAP response, and then extracts the top-10 links. The links would be the URLs of web pages that are likely to contain the target document.

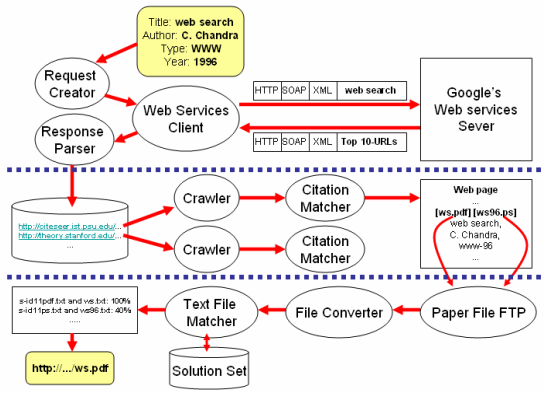

Fig. 2. Overview of PaSE

The *Crawlers* and *Citation Matcher (CM)* are shown in the middle. After ten Crawlers are created simultaneously, they start at their initial pages of top-10 links and stop if one of Citation Matchers finds the right web page that is likely to contain the online copy of the target document.

At the bottom, the *File Matcher (FM)* is illustrated. The FM is not part of PaSE system, but added for experimental validation purpose. That is, once the CM finds the candidate online copies, the FM downloads PDF or PS documents from the web page, converts them into text files using the Sherlock program [18], and compares them with solution set. Since the solution set contains the correct PDF document for each citation, we can estimate how good/bad the PaSE is.

3.2 The Crawler

Given a link to a web page, there are various orders to visit the link and its descendents: for instance, Breadth First Search (BFS), Depth First Search (DFS), Backlink Count (BC), Page Rank (PR), and Random schemes. Among these, we do not consider BC and PR since they were shown to be ineffective in a small domain [4]. Since our candidate pool contains only top-10 links and some of their descendents, our context is also a small domain, where most web pages have only a small number of backlinks.

To the rest of three BFS, DFS, and Random schemes, we add a simple but very effective heuristics – if words like "*research*", "*publication*", "*paper*", "*group*", "*laboratory*", "*citation*", or "*proceeding*" appear in anchors or URLs, then those links are favored.

In the BFS scheme, to give such a priority to web pages including the words, each crawler keeps two queues of URLs to visit. The first queue stores URLs with the words in anchors or URLs while the second queue keeps the rest of URLs to visit. Crawlers always prefer to take URLs to visit from the first queue. Algorithm 1 is *our heuristic-based BFS* crawling algorithm.

Algorithm 1. The heuristic-based BFS crawling algorithm

```
Procedure:
  enqueue(SecondQueue, startingURL)
  while (not empty(FirstQueue))
    if (not empty(FirstQueue))
      then URL = dequeue(FirstQueue)
      else URL = dequeue(SecondQueue)
    Page = crawlPage(URL)
    URLlist = extractURLs(Page)
    for each u in URLlist
      if (u is not in FirstQueue and SecondQueue)
        if (u contains topic words in anchor or url)
          then enqueue(FirstQueue, u)
```

Other schemes are similar and omitted due to space constraint.

3.3 The Citation Matcher (CM)

Next, when the Crawler visits a web page that has many citations in it, one needs to find out (1) which citation in the page matches the most with what a user specified; and (2) which PDF or PS is the corresponding online document of the matched citation? The CM does this job of finding the right (citation, PDF) pair that matches the given citation. Often, different users use different citation format to refer to the same document (e.g., "ICADL" vs. "Int'l Conf. on Asian Digital Libraries", or "J. Ullman" vs. "Jeffrey D. Ullman"). Therefore, it is not trivial to match what user specified with what is found on Web. In our setting, to make the problem simple, we assume that the user specified "title" of the citation, which is less likely to have different formats. That is, the CM uses the given title of the citation, finds the most similar citation of the page, and identifies the "start" and "end" of the citation. (called a citation block).

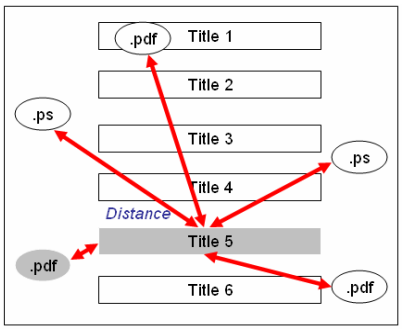

Fig. 3. The shortest distance title matching

Once the right citation block is identified, we may still have a problem, as illustrated in Fig 3. In the given web page, the "Title 5" is found to be the closest one to the input citation. However, there are various PDF or PS links near the "Title 5" block in the web page, and it is not always easy to find the right one. This often

occurs since not all citations have matching links to online copies in a web page. To make matters worse, links to online copies may be found in front of, in the middle of, or after the matching citation. For instance, in Fig 3, the shaded PDF is the corresponding document of "Title 5".

Since the way to link a citation to PDF or PS document in HTML varies by persons and by pages, to remedy this problem, we use the notion of *distance*. That is, once the right citation block (e.g., "Title 5") is found in a web page, the CM measures the distance (i.e., word count, byte, etc) from the citation block to each neighboring PDF or PS document, and pick the one with the shortest distance with some threshold. In Fig 3, the shaded PDF will be chosen. It is also important to set proper threshold to this distance to avoid the chase of matching far-away citation and online copy (e.g., "Title 2" and shaded PDF).

4 Experimental Results

We first made a solution set with 1,000 pairs of "(citation, PDF)", randomly collected from CiteSeer [14]. An example of our input file is as follows:

```
NUM: 7
AUTHOR 1: jun yang
AUTHOR 2: Jennifer widom
TITLE: incremental computation and maintenance of temporal aggregates
TYPE: icde
YEAR: 2001
```

Fig. 4 shows the accuracies of Google Advanced Search and PaSE's *heuristic-based* BFS, Random, and DFS schemes.

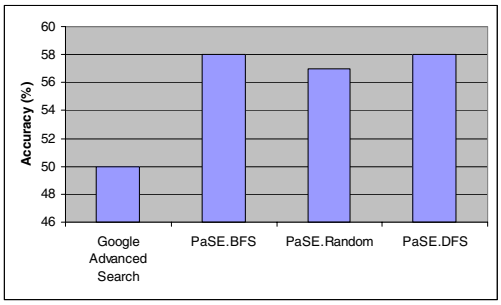

Fig. 4. Accuracy in schemes

In the graph, all the schemes of PaSE show higher accuracy than Google Advanced Search. It is mainly because PaSE can find online copies of documents hidden to Google during its crawling process. The relatively low accuracy compared to one of Fig. 1 is due to the fact that test set has many of the latest citations in 2004 that might not have been indexed by Google yet. Since the test data are drawn from CiteSeer, all of the corresponding online copies can be found in the CiteSeer. Interestingly, however, in our experimentations, Google Advanced Search did not

include any links to CiteSeer. This is different from our previous experimentation done on 2004/April, where most of top ranked links to online copies are toward CiteSeer (see Table 1).

Table 1. An example of top-10 links returned from Google's Web services

Citation	NUM: 4 AUTHOR 1: george karypis AUTHOR 2: eui-hong (sam) han TITLE: concept indexing a fast dimensionality reduction algorithm with applications to document retrieval & categorization TYPE: university of minnesota YEAR: 2000
2004/4	1. http://citeseer.ist.psu.edu/karypis00concept.html 2. http://citeseer.ist.psu.edu/article/yang99reexamination.html 3. http://www-users.cs.umn.edu/~karypis/publications/Papers/Abstracts/CI.html 4. http://www-users.cs.umn.edu/~karypis/publications/ir.html 5. http://portal.acm.org/citation.cfm?id=354772&dl=ACM&coll=GUIDE&CFID=11111111&CFTOKEN=2222222 6. http://www.cs.rutgers.edu/~mlittman/courses/lightai03/keller.pdf 7. http://www710.univ-lyon1.fr/~hassas/gjan/Divers/liens_classif.html 8. http://davis.wpi.edu/~xmdv/docs/tr0314_mds_som.pdf 9 http://www.isse.gmu.edu/~carlotta/teaching/INFS-795-s04/info.html 10. http://www-a2k.is.tokushima-u.ac.jp/~kita/eprint/ICCPOL01.ps
2004/6	1. http://www.cs.rutgers.edu/~mlittman/ courses/lightai03/keller.pdf 2. http://www-users.cs.umn.edu/~karypis/publications/Papers/Abstracts/CI.html 3. http://www-users.cs.umn.edu/~karypis/publications/ir.html 4. http://portal.acm.org/citation.cfm?id=354772&dl=ACM&coll=GUIDE&CFID=11111111&CFTOKEN=2222222 5. http://portal.acm.org/citation.cfm?id=963661&dl=ACM&coll=portal&CFID=11111111&CFTOKEN=2222222 6. http://dx.doi.org/10.1145/354756.354772 7. https://wwws.cs.umn.edu/tech_reports/index.cgi?selectedyear=2000&mode=printreport&report_id=00-016 8. http://sie.mimuw.edu.pl/literature.php 9. http://www.iturls.com/English/TechHotspot/TH_DocCluster.asp 10. http://www.di.uniovi.es/~dani/ publications/presentaciones/icwe.ppt

Fig. 5 illustrates the individual accuracies of top-10 ranks. Google Advanced Search returns the online copies of the target document as #1 rank more than 80%. Also, when it cannot find the right match up to rank 3, it is very unlikely that a matching document can be found in the remaining ranks of top-10. We believe this is due to the effectiveness of PageRank algorithm. Interestingly, however, all of the PaSE schemes find target documents evenly across all top-10 ranks. We believe this illustrates the reason why PaSE schemes were able to achieve higher accuracy that Google Advanced Search. That is, when target documents cannot be found by Google, PaSE follows some of the middle-ranked links such as rank #4-#6, and was able to recover the hidden matching documents, at the cost of elongated search time.

Fig. 6 shows overall result of citation matching after file comparisons. Most matching results are 90% or more. However, small amount of matching results are 90% or less because of the possible errors during file download and/or conversion.

We ran several cycles of experimentations, and found that the matching threshold of 20% gave acceptable results.

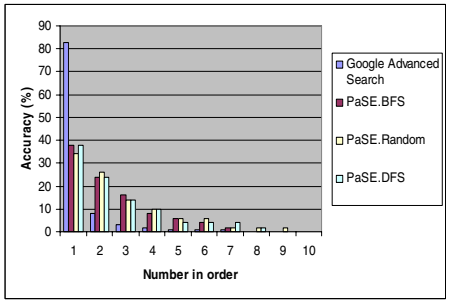

Fig. 5. Accuracy in rank

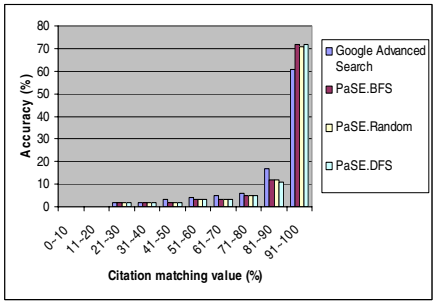

Fig. 6. Accuracy in citation matching

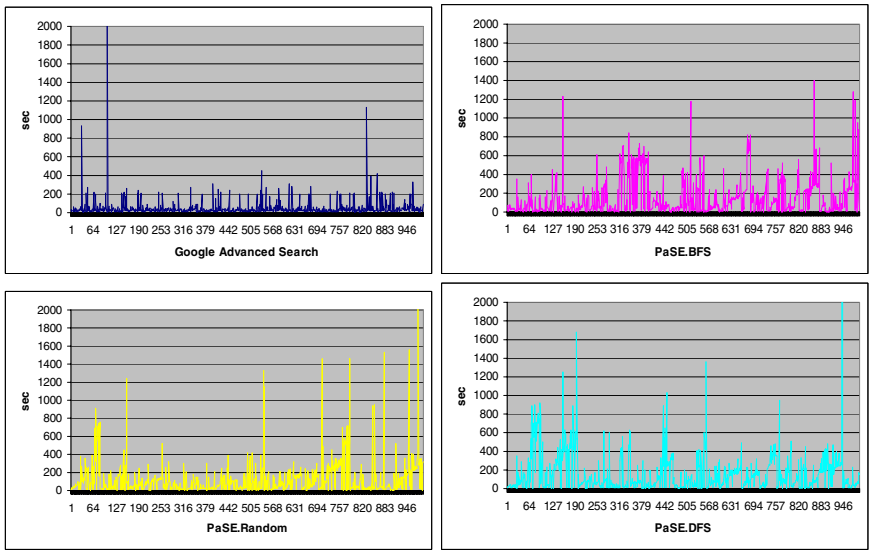

Fig. 7. The elapsed time per citation

Although PaSE can find target documents when Google cannot, it does so at the cost of time. To see how much additional time is needed for PaSE, we also check the time. Fig. 7 shows the elapsed time per citation. Clearly, Google Advanced Search takes a shorter time (on average) than the other three schemes of PaSE, which need to spend time on additional crawling and citation matching. The average crawling time of three PaSE's schemes is about 22 sec. per citation. However, in many cases, PaSE's crawling time is substantially smaller than 22 sec. As shown in Table 2, about

65% of the total citations take only 2 sec. of average crawling time regardless of the chosen crawling algorithms.

Table 2. The average crawling time

PaSE schemes	65%	14%	21%
BFS	2.33 sec	6.82 sec	90.58 sec
Random	2.18 sec	7.02 sec	99.24 sec
DFS	2.18 sec	6.93 sec	87.25 sec

For 21% of citations which took 80-100 sec, a large amount of time was wasted because of abnormal conditions (e.g., web server down). Such case can be avoided by setting some threshold on waiting time or by implementing more sophisticated error handing. We leave this as future work.

5 Conclusion and Future Work

We have developed PaSE to find online copies of scientific papers more effectively, and studied the heuristic-based crawling and distance-based title matching algorithms. Our preliminary experiments show that PaSE can deliver better accuracy than conventional approaches.

There are many rooms for future research. First, more thorough study on the effects of different crawling and citation matching algorithms (and their interplay with other factors such as domain) are needed. Second, by extending the PaSE with web services based interface, machine programs can communicate and retrieve online copies of target documents, making PaSE available for more applications.

References

1. G. Hoff, M. Mundhenk, "Finding Scientific Papers with HPSearch and Mops", SIGOC '01 (2001)
2. Z. Nie, S. Kambhampati, T. Hernandez, "BibFinder/StatMiner: Effectively Mining and Using Coverage and Overlap Statistics in Data Integration", 29th VLDB (2003)
3. A. E. Papathanasiou, E. P. Markatos, S. A. Papadakis, "PaperFinder: A tool for scalable search of digital libraries", WebNet 98, (poster paper) (1998)
4. J. Cho, H. Garcia-Molina, L. Page, "Efficient Crawling Through URL Ordering", 7th WWW (1998)
5. P. De Bra, R. Post, "Information Retrieval in the World Wide Web: Making Client-based Searching Feasible", 1st WWW (1994)
6. M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalheim, S. Ur, "The Shark-Search algorithm – an application: tailored Web site mapping", 7th WWW (1998)
7. J. Cho, H. Garcia-Molina, "The evolution of the web and implications for an incremental crawler", 26th VLDB (2000)
8. S. Lawrence, C. L. Giles, K. Bollacker, "Autonomous Citation Matching", The 3rd Int'l Conf. on Autonomous Agents, Seattle Washington (1999)

9. B. J. Jansen, A. Spink, "An Analysis of Web Documents Retrieved and Viewed", The 4th Int'l Conf. on Internet Computing, Las Vegas Nevada (2003)
10. Google Web APIs, http://www.google.com/apis
11. Google normal service, http://www.google.com
12. Google advanced service, http://www.google.com/advanced_search?hl=en
13. DBLP Bibliography, http://www.informatik.uni-trier.de/~ley/db/
14. CiteSeer.IST Scientific Literature Digital Library, http://citesser.ist.psu.edu
15. arXiv.org e-Print archive, http://arxiv.org
16. F, Burchsted, "Finding Personal Papers in United States Repositories", http://www.people.fas.harvard.edu/~burchst/FPPiUSR.htm
17. Amazon.com, http://www.amazon.com/
18. The Sherlock Plagiarism Detector, http://www.cs.usyd.edu.au/~scilect/sherlock
19. Lawrence, S., "Online or Invisible?," Nature 411(6837):521 (2001)
20. Homepage Search, http://hpsearch.uni-trier.de/hp/