6th International Conference on Applied Informatics
Eger, Hungary, January 27–31, 2004.

Content Discovery of Invisible Web

Mária Princz[a], Katalin E. Rutkovszky[b]

[a]University of Debrecen, Faculty of Technical Engineering
Department of Industrial Management and Technical Computing

[b]University of Debrecen, Institute of Informatics
Department of Informatics Systems and Network

1. Introduction

The Web recently is growing in a never experienced pace, and this process is getting faster and faster. Vast amount of information can be accessed on the net. Information needed, can be retrieved from this huge volume by the following two strategies: using services of dictionaries or search engines. Most of search engines deals with only the surface of information of the Web. By search engine, we mean the composite of several underlying technologies, including: a crawler (or spider or robots) to discover information, a page analyzer to categorize and index information (develop search engine's database) as well as the search front-end to receive queries from the user and deliver relevant results (search inside the database).

Visible Web or surface Web consists of the publicly indexable pages that have been picked up and indexed by conventional search engines, mainly consist of static HTML pages. The estimated size of surface Web is 167 terabytes.[1]

Invisible Web is the portion of the Web, which is hidden from general purpose search engines. This part consists of material that conventional search engines either cannot, or perhaps, will not include in their indexes.
The reasons that a search engine does not contain a page are technical barriers that prohibit access (typing and/or judgment are required) and choices or decisions made by search engine companies because of the validity, authority, and quality of online information to exclude.
Many authors use invisible Web and deep Web or hidden Web as synonyms, but there are differentials among them.

Deep or hidden Web is part of invisible Web. It includes information in large

searchable databases. It is somewhat hidden for crawler, but clearly available if different technology is used to access it.
Invisible Web is the largest growing category of new information and this information is more content-rich, current and up-to-date than conventional surface sites. Public information on the deep Web is currently 400 to 550 times larger then the visible or surface Web. Total quality content of the invisible Web is 1000 to 2000 times greater than that of the surface Web.
The paper is organized as follows. Section 2 describes different types of invisibility on the Web. Section 3 discusses some solutions for searching invisible Web. Finally, in Section 4 conclusions are given.

2. Different types of invisibility on the Web

2.1. Opaque Web

Opaque Web consists of data that could be, but for one reason or another is not indexed by Web search engines. This part of the Web is relatively easily accessible to the searcher.
Disconnected URLs
Spiders may never find sites that are not connected via links to other Web pages crawled or spidered, and whose URLs have not been submitted to any Web search engine.
Depth of crawling
The crawler does not crawl the entire site. The crawl depth is the number of levels into the Web site search engines will reach when looking for pages.
Frequency of crawling
Important question is how often spiders update their databases by revisiting sites they have indexed. This parameter varies significantly for different engines. Content on the pages in databases is not current that effects dead links.
Maximum number of viewable results Size limitation - each engine decides how much of the page it will crawl. (E.g. Google and AltaVista index about the first 100 KB of the page but AllTheWeb indexes the entire page.)

2.2. Private Web

Technically indexable Web pages that have been deliberately excluded from search engines.
Robots Exclusion Protocol Search engine robots check a special file in the root of each server called robots.txt. This can hold: Don't crawl and index my content.
Noindex Meta Tag
It is possible with it to specify specific page or pages the crawler is not supposed to crawl. It is in the head part of file.
Firewall is in place to observer, filter and limit of dataflow.
Password

We can use password protected pages, too.

2.3. Proprietary Web

Proprietary Web: only available to people who register with the site or organization. (Password protection, firewall.) This part includes databases witch are mainly fee-based and are produced by information providers.

2.4. Truly Invisible Web

This fraction of Web pages are not accessible for search engines mainly because of technical reasons.
Non-html text: Search engines are designed to index html text. Audio, video, images: these formats are hard for search engines to understand.
Multiple Formats - Not every format is crawled by every search engine - pdf, flash, shockwave, executable programs, compressed files - although technically indexable - until recently ignored by most of search engines. Indexing these formats is resource-intensive. Codes and frames are also difficult for some few search engines.
Registration Required
Since a search engine is incapable of filling out a username and password request, they have no way to reach the information that is hidden behind that doorway.
Dynamically Generated Pages - search engines generally refuse to crawl any material with "?"(question mark) in URL because it can be traps for engines.
Dynamic HTML pages are generated by server or client applications (cgi, javascript, asp, etc).
Real-time content - spiders do not crawl/recrawl in real-time (too much information, ephemeral and storage intensive, no real good reason to spider this type of information (finance, weather, airline flight arrival/departure information, news and magazine articles).

2.5. Deep Web

The biggest part of the invisible Web - currently 400 to 550 times larger than surface Web - is made up of information stored in databases. The deep Web contains 91,850 terabytes of information compared to 167 terabytes of information in the surface Web. [1]
More than half of the deep Web content resides in topic specific databases. A full 95% of the deep Web is publicly accessible information - not subject to fees or subscriptions.

The content from these Web-accessible databases can only be discovered by a direct query. When queried, deep Web sites post their results as dynamic Web pages in real-time. Though these dynamic pages have a unique URL address that allows them to be retrieved again later, they are not persistent.

3. Invisible Web Solutions

3.1. User-oriented Solutions

Specialized search tools can help the searcher locate information that, while not part of the invisible Web, is still difficult if not impossible to find using general purpose search engines and directories. These search tools include targeted, selected directories and crawlers, meta-search engines and fee-based Web services.
Knowing how to use invisible Web resources will make you a more efficient and powerful researcher. That is why very important teaching information seeking skills and enhancing users' awareness about information resources.

Invisible Web search strategies
Searching the invisible Web for information is a two-step process. First you search for the right online resources likely to hold the desired information. Next you search the site itself by using the site's own search tools, or site map.
Web search engines or site search tools are also useful to find searchable databases. Although the Web crawler cannot access data stored inside the database, it can often find the main page. Conduct a Web search on your research topic and include the terms "searchable database"OR archive OR repository in your search along with keywords for the subject you are researching.

There are some resources to help access the invisible Web:

- Selected directories of searchable databases
- Specialized resources by major Web search engines Recently some of the commercial search engines have added non-html files to their indexes.
- Meta-search engines Meta-searchers are helpful tools for searching over many Web-accessible databases at once through a unified query interface.

3.2. Technical-oriented Solutions

There are some solutions for problems of extracting content from hidden Web. Implementations affecting search engine functionality could be mentioned as search side solutions. Methods requiring modifications at Web-servers should be referred to as server side solutions.

Meta-search solutions
Many meta-search engines have special agreements with the search engines that they query for search results. These agreements allow them to access the search engine indexes through a "backdoor"and provide results at a more rapid pace than a human being could conducting the searches on their own.
Unfortunately many Web-accessible text databases are completely autonomous

and do not report any detailed metadata about their contents to facilitate meta-searching.
There are some typical database selection algorithms for summarizing of databases contents. We now briefly outline how typical database selection algorithms work and how they depend on database content summaries to make decisions.
A database selection algorithm attempts to find the best databases to evaluate a given query, based on information about the database contents. Usually this information includes the number of different documents that contain each word, to which we refer as the document frequency (df) of the word, plus perhaps some other simple related statistics, like the number of documents NumDocs stored in the database. Table 1 depicts a small fraction of what the content summaries for two real text databases might look like.

CANCERLIT NumDocs: 148,944	
Word	*df*
breast	121,134
cancer	91,688
...	...

CNN.fn NumDocs: 44,730	
Word	*df*
breast	124
cancer	44
...	...

Table 1 A fragment of the content summaries of two databases.

- Uniform Probing for Content Summary Construction

Callan et al. pesented pioneer work on automatic extraction of document frequency statistics from "uncooperative"text databases that do not export metadata.
Their algorithm extracts a document sample from a given database D and computes the frequency of each observed word w in the sample, SampleDF(w):
1. Start with an empty content summary where SampleDF(w) = 0 for each word w, and a general (i.e., not specific to D), comprehensive word dictionary.
2. Pick a word (see below) and send it as a query to database D.
3. Retrieve the top-k documents returned.
4. If the number of retrieved documents exceeds a prespecified threshold, stop. Otherwise continue the sampling process by returning to Step 2.
Callan et al. suggested using k = 4 for Step 3 and that 300 documents are sufficient (Step 4) to create a representative content summary of the database.

- Focused Probing for Database Classification

Another way to characterize the contents of a text database is to classify it in a Yahoo!-like hierarchy of topics according to the type of the documents that it contains. To automate this classification, these queries are derived automatically from a rule-based document classifier.

Web Site Optimization Solutions
Part of the problem with the search engines not finding dynamically created pages has been that the search engine spiders have avoid URLs containing a '?'. Such URLs are definite indicators of dynamic pages, and it is quite possible for a spider to get into a loop from which it cannot escape when following dynamic page links. The problem with URLs containing a '?' has become less of an issue in recent times, as spiders such as Google's and HotBot's will follow such a URL to the first level (i.e. they will not keep following links to dynamic pages that they find inside a dynamic page, but will follow a dynamic link from a static page).
A Webmaster may choose to optimize his or her site so that all links to dynamic pages appear as 'spider-friendly' static links. There are various URL rewriting tools that assist in doing this. Many, such as the modfirewrite module for the Apache Web Server, allows the '?' in the URL to be replaced by a different character. Cold Fusion provides an option through which the '?' is replaced with '/' and the query parameters are encoded like subdirectories.
These solutions require significant manual effort to design and maintain the Web site to help out the spiders. Not only do all URLs containing '?' have to be manually translated in the Web pages, all relative links need to be re-written as absolute links to work around the change in URL structure introduced by the translation. Absolute links in Web pages significantly increase maintenance costs and make Web site mirroring more difficult.

Content Discovery and Linking
YourAmigo's Spider Linker product offers a solution to the Webmaster which allows all of the content on a site to be automatically made available to the Internet engines such as Google with negligible on-going administrative effort. Using some of the techniques mentioned above, Spider Linker creates spider-friendly links to dynamic page URLs. Beyond what any of the other products do, however, Spider Linker also employs YourAmigo's patent pending technology for content discovery in order to identify the optimum set of dynamic pages that each configured script on the Web site can produce, and creates a table of contents (TOC) for all of the static and dynamic content on the site. This means that a Web site running Spider Linker can potentially have all of its catalogue entries indexed by Google. The TOC also serves as an administration tool.

The TOC may take one (or both) of two forms:

- A series of one or more HTML pages with links (direct or indirect) to content on a site.

- A sitelist.txt file for each virtual host, conforming to the standard for this file. The sitelist.txt file will be used by robots to determine what pages they need to get at a particular site. It will list the pages at the site which are new, updated, or deleted in order of modification date (most recent first). This way robots can only get the pages that have changed since their last

visit. The sitelist.txt file will be located at a server's html root. The format is simple. There is a header area and a body area. The two are separated by two new lines.

In order for the TOC to be effective, it must be found by the search engine spiders. There are several ways in which this can be achieved:
1. Linking from the home page
This is perhaps the simplest method. If the default home page or a site map page contains a link to the TOC page, a spider will navigate to the TOC and then be able to find all linked content.
2. Cloaking
Cloaking is a technique whereby different content can be delivered through a given URL depending on who is asking for it. Typically, the Web server will look at the User Agent field of the request, which indicates whether the request came from a Netscape browser, or Internet Explorer, or a spider, and deliver content accordingly. The TOC page may be cloaked with the home page so that when a typical user visits, they see the home page but when a spider visits, it sees the TOC. The practice is often frowned upon by the search engine companies, and could in theory cause a site to be banned.
3. Direct submission
Many search engines allow Webmaster to directly submit URLs that they wished to have indexed. For engines which use these submitted URLs as starting points for their spiders, it would be appropriate to submit the TOC as the starting URL.
4. Replace the home page
In some cases, the header and footer content which can be included in the TOC page can be used to make the TOC appear as you would like to see your home page appear. Using JavaScript, the TOC links may even be hidden to human users but left visible to spiders.
5. Using sitelist.txt
For spiders that understand sitelist.txt, none of the above methods are necessary. The spider will find the sitelist.txt file and from there read all of the content.

Building a Directory Site
BrightPlanet Corporation is the leader in deep Web research and the development of innovative ways to efficiently search, monitor and manage all Internet and internal content.

The BrightPlanet's technology consists of software for:

- Identifying deep Web sites
- Harvesting and updating deep and surface Web content
- Filtering high quality results
- Placing Qualified results into automatically generated category structures (directories)

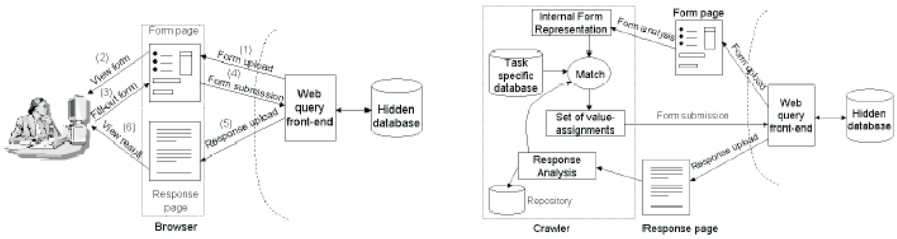
- Creating context-specific document summaries
- Publishing final results as Web sites

Hidden Web Crawler

Close to 80% of the content on the Web is dynamically generated, and that this number is continuing to increase. As major software vendors come up with new technologies to make such dynamic page generation simpler and more efficient, this trend is likely to continue.

The problem of crawling a subset of the currently uncrawled dynamic Web content. In particular, we concentrate on extracting content from the portion of the Web that is hidden behind search forms in large searchable databases. The content from these databases is accessible only through dynamically generated pages, delivered in response to user queries.

At Stanford it was built a task-specific hidden Web crawler called the Hidden Web Exposer (HiWE).



(a) User form interaction (b) Crawler form interaction

Figure 1

The basic actions of HiWE (fetching pages, parsing and extracting URLs, and adding the URLs to a URL list) are similar to those of traditional crawlers. However, whereas the latter ignore forms, HiWE performs the following sequence of actions for each form on a page:

1. Form Analysis: Parse and process the form to build an internal representation based on the above model.
2. Value assignment and ranking: Use approximate string matching between the form labels and the labels in the LVS table to generate a set of candidate value assignments. Use fuzzy aggregation functions to combine individual weights into weights for value assignments and use these weights for ranking the candidate assignments.
3. Form Submission: Use the top "N"value assignments to repeatedly fill out and submit the form.
4. Response Analysis and Navigation: Analyze the response pages (i.e., the pages received in response to form submissions) to check if the submission yielded valid search results. Use this feedback to tune the value assignments in Step 2. Crawl the hypertext links in the response page to some pre-specified depth.

4. Conclusion

Search engines use different types of crawlers and don't always index the same pages or Websites and no single engine indexes the entire Web. Policies vary on which non-html file formats to index. What is invisible to one search engine might be indexed by another. This is one of the reasons you should use more than one search engine when seeking information.
Web collection development will be more important.

References

[1] How much information 2003.
http://www.sims.berkeley.edu/research/projects/how-much-info-2003/

[2] C. Sherman, G. Price: The Invisible Web: Uncovering Information Sources Search Engines Can't See. CyberAge Books, 2001.

[3] BrightPlanet.com. http://www.brightplanet.com
The Deep Web: Surfacing Hidden Value.
http://www.completeplanet.com/Tutorials/DeepWeb/

[4] S. Raghavan, H. Garcia-Molina. Crawling the Hidden Web. Technical Report 2000-36, Computer Science Department, Stanford University
What is the "Invisible Web"?
http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/InvisibleWeb.html

[5] J. Caverlee, D. Buttler: Discovering Objects in Dynamically-Generated Web Pages, 2003
http://disl.cc.gatech.edu

[6] G. Price: Direct Search http://www.freepint.com/gary/direct.htm

[7] Robert J. Lackie, Those Dark Hiding Places: The "Invisible Web"Revealed
http://library.rider.edu/scholarly/rlackie/Invisible/Inv_Web.html

[8] Web Searching - Know Your Tools, 2003. Dialog,. http://www.dialog.com/

[9] Search Indexing Robots and Robots.txt http://www.searchtools.com/robots/

[10] L. Gordon-Murnane: The Invisible Web: what Search Engines can't Find and Why University of Maryland Libraries Digital Dateline Series, Nov. 2003

[11] P.Ipeirotis-L.Gravano: Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection, 28th VLDB Conference, 2002