Text and Image Metasearch on the Web

Steve Lawrence and C. Lee Giles
NEC Research Institute, 4 Independence Way, Princeton, NJ 08540
{lawrence,giles}@research.nj.nec.com

Abstract

As the Web continues to increase in size, the relative coverage of Web search engines is decreasing, and search tools that combine the results of multiple search engines are becoming more valuable. This paper provides details of the text and image metasearch functions of the Inquirus search engine developed at the NEC Research Institute. For text metasearch, we describe features including the use of link information in metasearch, and provide statistics on the usage and performance of Inquirus and the Web search engines. For image metasearch, Inquirus queries multiple image search engines on the Web, downloads the actual images, and creates image thumbnails for display to the user. Inquirus handles image search engines that return direct links to images, and engines that return links to HTML pages. For the engines that return HTML pages, Inquirus analyzes the text on the pages in order to predict which images are most likely to correspond to the query. The individual image search engines tend to excel at different classes of queries, and the combination of engines is surprisingly effective at finding images corresponding to a given query. Both the text and image metasearch functions of Inquirus are surprisingly fast, and we describe the parallel architecture of the engine that provides this efficiency.

Keywords: text search, image search, metasearch, parallel Web search.

1 Introduction

Limitations of the search services on the Web have led to the introduction of metasearch engines [1, 2]. A metasearch engine searches the Web by making requests to multiple search engines such as AltaVista or Infoseek. The results from the individual search engines are combined into a single result set. Advantages of metasearch engines include a consistent interface to multiple engines and improved coverage.

We have previously developed Inquirus, a metasearch engine for the Web [3]. Inquirus has a number of differences from typical metasearch engines. Inquirus is fundamentally architected around downloading and analyzing individual Web pages referenced by the search engines, as opposed to simply retrieving and merging search engine responses. This means that Inquirus is always up to date and working with the current contents of pages, and allows extensive analysis to be done on page contents. The page analysis functions performed by Inquirus include extracting query-sensitive summaries that make relevance assessment easier for the user, using a consistent relevance measure across all pages (difficult for a typical metasearch engine), filtering non-relevant documents, improved duplicate detection based on query term context, query term highlighting and quick jump links when viewing pages, query expansion using co-occurring morphological variants, and clustering using the co-occurrence of words and phrases. Inquirus returns results progressively as they are processed, and the parallel architecture of Inquirus allows it to typically return the first result faster than a standard search engine can respond.

While the use of multiple search engines improves coverage of the Web, we note that most of the advantages of Inquirus are also applicable even if only one search engine is used.

This paper provides details of the image metasearch functions of Inquirus, as well as providing details and statistics for text metasearch not covered in previous papers.

2 Image Metasearch

There are a number of image databases on the Web, e.g. Yahoo Image Surfer (http://isurf.yahoo.com/) and AltaVista PhotoFinder (http://image.altavista.com/). Many of the large Web search engines also allow searching for images using keywords, e.g. Lycos (http://www.lycos.com/), HotBot (http://www.hotbot.com/), and AltaVista (http://www.altavista.com/). However, these databases tend to index many different images and tend to excel at dif-
different queries. The same concepts that have been applied to text metasearch can also be applied to images. The following sections describe the motivation for and operation of the keyword based image metasearch functions of Inquirus.

2.1 Previous Work

Several companies [for example, PLS (www.pls.com), Lexis-Nexis (www.lexis-nexis.com), DIALOG (www.dialog.com), and Verity (www.verity.com)] have produced systems that integrate results from multiple heterogeneous databases [4]. Many text metasearch services exist such as the popular and useful MetaCrawler service [1]. MetaSEEk [5] is a metasearch engine for images. MetaSEEk targets query by example as opposed to keyword search, which is the focus of the image metasearch functions of Inquirus.

2.2 Motivation

The principle motivations behind the image metasearch functions of Inquirus are similar to the motivations behind the creation of text metasearch engines: the poor precision, limited coverage, limited availability, and limited user interfaces of the image search engines. Expanding on these points:

Poor precision. Automatically determining the relevance of images to keyword based queries is a difficult task and the various image search engines typically use different means of assessing relevance. For a given query, it is difficult to know which image database will produce the best results.

Limited coverage. The search engines that index publicly available images on the Web (e.g. Lycos) do not index the Web exhaustively [6]. As with their coverage of the Web in general, their coverage of images is limited. The different engines tend to index different images, so coverage can be improved by combining the results of multiple engines.

Limited availability. Due to search engine and/or network difficulties, we have observed that the engine which responds the quickest varies over time.

Limited user interfaces. Many of the image search engines support a different query syntax, e.g. some engines do not support the use of phrases in queries. One advantage of metasearch techniques is that queries can be modified appropriately for individual engines.

2.3 Architecture

Figure 1 shows a simplified control flow diagram for the image metasearch algorithm we have used. The engine consists of two main logical parts: the image metasearch code and a parallel page retrieval daemon. Pseudocode for (a simplified version of) the search code is as follows:

Process the request to check syntax and create... regular expressions which are used to... match query terms
Send requests (modified appropriately) to all... relevant image search engines
Loop for each page retrieved until maximum number.. of images or all pages retrieved
  If page is from a search engine
    Parse search engine response...
    extracting hits and any link for...
    the next set of results
    Send requests for all of the hits
    Send request for the next set of...
    results if applicable
  Else if page is an image
    If image meets display criteria...
      then add image to the display queue
  Else
    Analyze query term locations in the...
    page and predict which (if any) of...
    the images on the page corresponds...
    to the query - send a request to...
    download this image
Endif
If n images are in the display queue
  Create a single image montage of the...
  images in the queue
  Display the montage as a clickable...
  image where each portion of the...
  image corresponding to the...
  original individual images shows a...
  detail page for the original image
Endif
If any images are in the display queue
  Create a single image montage of the images...
  in the queue
  Display the montage as a clickable image...
  where each portion of the image...
  corresponding to the original individual...
  images shows a detail page for the...
  original image
Endif
Print summary statistics

As indicated in the pseudocode, for those engines that return a list of Web pages, the engine analyzes the pages in order to predict which (if any) of the images on the pages correspond to the query. The mechanism for doing this is currently relatively simple: the engine first looks for query terms in the filename of image files. If none are found then it locates the image closest to the query terms.
Very small or very thin images are filtered out – these images are typically icons or separators.

Note that the engine displays images in groups of five, creating a montage of the five thumbnails. An image map is used to display the appropriate page when the individual thumbnails are clicked on. The montage technique was used because some browser implementations were prone to crash when presented with a large number of the smaller thumbnail images.

A few notes about the architecture of the engine:

1. The engine does not rely on the validity of URLs provided by the image search engines – all images are downloaded and processed, thereby filtering out links which are no longer valid.
2. The engine presents thumbnails of all images. Because of the low precision of the search engines this helps users to determine the relevance of an image more easily and more rapidly, when compared to engines that only return a URL referencing an image or a page containing an image.
3. Results are returned progressively as the images are downloaded and analyzed, rather than after all images are downloaded.
4. When clicking on a thumbnail, the full image is displayed in another window. For images contained on Web pages, the pages are shown with the query terms highlighted.
5. All images are cached locally, which results in a substantial improvement in browsing speed, and a reduction in frustration waiting for images to load.
6. The engine detects and filters out identical images using a checksum on the files.

Figure 2 shows the response of Inquirus to the image query `sunset`.
2.4 Efficiency

A first impression of the image metasearch technique presented here may be that the technique would be too slow. However, it is actually surprisingly fast. The first results from the engine typically display within a few seconds and the remaining images typically display faster than the user browses the currently displayed images (using a T1 connection).

Part of the perceived speed is due to the parallel architecture of the engine. Figure 3 shows the median time for the first of \( n \) engines to respond when queries are made simultaneously to \( n \) Web search engines (as happens in Inquirus). Figure 4 shows the median time for the first of \( n \) arbitrary Web pages to be downloaded when queries are made simultaneously to all \( n \) pages. We use the median because the distribution of response times is significantly skewed, as shown in Figure 5, which shows a histogram of page retrieval times for arbitrary Web pages. It can be seen that downloading pages in parallel can result in the first pages being retrieved significantly faster than the average page retrieval times, which helps explain why the image metasearch engine can start displaying results quickly.

One potential drawback of this image metasearch technique is that it uses a significant amount of bandwidth. We note simply that bandwidth is increasing rapidly and that the bandwidth issue may become less important in the future. Certainly, the bandwidth requirements are far less than brute force search of the Web.

3 Text Metasearch

Details of many of the text metasearch features of Inquirus have been provided before [3]. This section provides details of the use of links by Inquirus, and statistics on the usage of the service and the performance of search engines.
3.1 Using Links to Improve Metasearch

Standard search engines are making increasing use of the links between pages. For example, Google [7] uses PageRank [8] as part of its method for ranking pages. PageRank considers the numbers of links to pages (pages that are linked to more often or by more highly linked pages get ranked higher). Inquirus does not know the number of links to all pages and hence does not use the same technique. However, Inquirus does make use of link information.

Often a query can return pages that link to a desired page before returning the desired page. These pages may be found because the links can contain better descriptions of pages than the pages themselves. For example, a researcher might have a homepage that does not contain their name in the title, but another page might link to their page and include the researcher’s name in the link. As shown in Figure 6, Inquirus displays links in page summaries, so it is easy to see and follow these links.

Inquirus also analyzes the graph created by links between pages in the result set, in order to identify authoritative pages (those with many links to them), and pages that have links to many authorities (“hubs”). The results are similar to the hubs and authorities computation performed by Kleinberg and others [9, 10], although the algorithm currently used by Inquirus simply looks at the number of links to and from pages within the result set. Figure 7 shows an example. The algorithms used by Kleinberg and others are typically iterative and augment the result set with neighboring pages in the link graph of the Web. Inquirus does not do this because it would add substantially to the computational requirements. We plan to compare our algorithm with the other proposed algorithms.

3.2 Usage and Performance Statistics

Inquirus is used by employees of the NEC Research Institute. We analyzed the last 5000 queries sent to Inquirus in April 1999. Users performed an average of 7.7 queries each. On average, 2.5 pages are viewed per query. Since Inquirus shows query-sensitive summaries, this suggests
that users are finding valuable documents. There was an average of 2.9 words used per query, which is higher than typically reported values for Web search engines [11]. 51% of queries used phrases, which is much higher than typically reported for Web search engines [11]. The higher number of words per query and the increased use of phrases may be due to the user population (primarily scientists) and/or the query suggestions that Inquirus makes [3]. For example, when a two term query without phrases returns many pages that contain the terms as a phrase, Inquirus suggests the use of phrases. Many users may follow the suggestions and use phrases in the future if they obtain improved results. Few users read help information in our experience, and we believe that query-sensitive suggestions can be a good way to teach users about query syntax, query formulation, and features of the search service.

Figure 8 shows the median response time of various Web search engines as a function of time. The median is used because the distribution of response times is significantly skewed. We can see that the response time varies significantly across engines and across time.

4 Summary

We have discussed the text and image metasearch functions of the Inquirus search service. Inquirus provides a number of advantages for text metasearch by analyzing the current contents of pages. Inquirus demonstrates that real-time text and image metasearch, including downloading and analysis of the matching pages and images, is feasible. For image metasearch, Inquirus queries multiple image search engines on the Web, downloads the actual
images, and creates thumbnails for display to the user. For image search engines that return a list of pages, Inquirus analyzes the text on the page in order to predict which image is most likely to correspond to the query. The individual image search engines tend to excel at different classes of queries, and the combination of engines is surprisingly effective at finding images corresponding to a given query. Implementation of Inquirus shows that it is surprisingly fast, given that it downloads and processes all of the images.

References


---

**Figure 7.** Sample of hubs and authorities computation for the query "financial news".

**Figure 8.** The median time for search engines to respond.

---

**Tables**

<table>
<thead>
<tr>
<th>Hubs and authorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorities</td>
</tr>
<tr>
<td>10: CNNfn - the financial network</td>
</tr>
<tr>
<td>7: Barron's (<a href="http://www.barrons.com/">http://www.barrons.com/</a>)</td>
</tr>
<tr>
<td>5: <a href="http://www.quote.com">www.quote.com</a> (<a href="http://www.quote.com">http://www.quote.com</a>)</td>
</tr>
<tr>
<td>5: Bloomberg Personal (<a href="http://www.bloomberg.com/">http://www.bloomberg.com/</a>)</td>
</tr>
<tr>
<td>4: PC Quotes (<a href="http://www.pcquote.com/">http://www.pcquote.com/</a>)</td>
</tr>
<tr>
<td>4: NYSE (<a href="http://www.nys.com/">http://www.nys.com/</a>)</td>
</tr>
<tr>
<td>4: The Economist (<a href="http://www.econ">http://www.econ</a>...)</td>
</tr>
<tr>
<td>4: Data Broadcasting Corporation (<a href="http://www.dbc.com">http://www.dbc.com</a>)</td>
</tr>
<tr>
<td>4: Briefing.com (<a href="http://www.briefing.com/">http://www.briefing.com/</a>)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hubs</th>
</tr>
</thead>
<tbody>
<tr>
<td>303: News and Financial Information</td>
</tr>
<tr>
<td>134: Yahoo! Finance (<a href="http://quote.yahoo.com">http://quote.yahoo.com</a>)</td>
</tr>
<tr>
<td>126: The Computerplaza-Finance Links</td>
</tr>
<tr>
<td>106: The Search Beat's Business News Cube</td>
</tr>
<tr>
<td>81: Financial News Online, Financial Links for Investors...</td>
</tr>
<tr>
<td>81: Financial News Online, Financial Links for Investors...</td>
</tr>
<tr>
<td>78: Active Lynx Showcase: Investing: Financial News -...</td>
</tr>
<tr>
<td>75: A s i a n N e t - Finance Forum</td>
</tr>
</tbody>
</table>

---

**Tables**

<table>
<thead>
<tr>
<th>Median Engine Response Time (Seconds)</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>11 Apr 99</td>
</tr>
<tr>
<td>12</td>
<td>18 Apr 99</td>
</tr>
<tr>
<td>10</td>
<td>25 Apr 99</td>
</tr>
<tr>
<td>6</td>
<td>2 May 99</td>
</tr>
</tbody>
</table>

**Graphs**

- Median Engine Response Time
- Date