Categorizing Web Search Results into Meaningful and Stable Categories Using Fast-Feature Techniques

Bill Kules, Jack Kustanowitz and Ben Shneiderman
Human-Computer Interaction Lab and Department of Computer Science
University of Maryland
College Park, MD 20742

{wmk,kustan,ben}@cs.umd.edu

ABSTRACT

When search results against digital libraries and web resources have limited metadata, augmenting them with meaningful and stable category information can enable better overviews and support user exploration. This paper proposes six "fast-feature" techniques that use only features available in the search result list, such as title, snippet, and URL, to categorize results into meaningful categories. They use credible knowledge resources, including a US government organizational hierarchy, a thematic hierarchy from the Open Directory Project (ODP) web directory, and personal browse histories, to add valuable metadata to search results. In three tests the percent of results categorized for five representative queries was high enough to suggest practical benefits: general web search (76-90%), government web search (39-100%), and the Bureau of Labor Statistics website (48-94%). An additional test submitted 250 TREC queries to a search engine and successfully categorized 66% of the top 100 using the ODP and 61% of the top 350. Fast-feature techniques have been implemented in a prototype search engine. We propose research directions to improve categorization rates and make suggestions about how web site designers could re-organize their sites to support fast categorization of search results.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

General Terms

Measurement, Design, Experimentation, Human Factors

Keywords

Classification, Categorization, Browsing, Taxonomies, Open directory, Metadata

1. INTRODUCTION

Traditional digital libraries maintain rich metadata for their holdings, but as their holdings expand to include heterogeneous collections of semi-structured information, the available metadata dwindles, and human-generated metadata is expensive to create. External sources of digital knowledge can be integrated to provide valuable metadata. Web search is an example of this growing challenge. This paper explores an approach to add useful metadata to search results by fast-feature techniques, that is, those that utilize only features available in the search result list to extract meaningful category information from external sources.

JCDL'06, June 11–15, 2006, Chapel Hill, North Carolina, USA.

Digital library and web search engines today do a remarkably good job providing a linear list of sorted or ranked results for a query. For known-item queries, users often find the site they are looking for in the first page of results. However, a list may not suffice for more sophisticated exploratory tasks, such as learning about a new topic or surveying the literature of an unfamiliar field of research, or when information needs are imprecise or evolving [29]. In these situations users can benefit from overviews of search results based on meaningful and stable categories, such as when they see a list of music categories in a record store or news categories on CNN.com. Our studies of exploratory search tasks using US government agency hierarchies [17] and thematic categories [3] have demonstrated the benefit of meaningful and stable categorical presentations of result sets.

Categorizing search results according to meaningful and stable categories provides several benefits:

1. **Overviews**: There is value to seeing the spread of categories covered by a given search term. For example, it might be interesting or surprising to see that "soybeans" have matches in the NASA section of the government hierarchy.

2. **Navigation within search results**: Searchers review search results to predicate which web pages will be topical, authoritative and high quality [21]. If the desired item is ranked far down the list, searchers are unlikely to find it, since they rarely look beyond the top 10-20 results. If, however, it falls in a visible and meaningfully labeled category (perhaps as part of an overview), the searcher can navigate directly to the category and then to the desired item, rather than linearly scanning the entire list, which could involve requesting multiple additional pages from the server.

3. **Negative Results**: Categorization allows users to see at a glance where their search term did not yield results, for example that there is no result for "cabinet making" in the "graduate" area of the University of Maryland hierarchy. This can help them avoid examining results that are not relevant to their information need.

This paper seeks to motivate research in lightweight, rapid techniques for categorizing search results into meaningful and stable categories. We believe they hold promise for rapid development, easy deployment, and effective use in search engines and browsers. Their straightforward implementation will

facilitate maintainability and be understandable to users. For example, if the URL includes .gov or .edu then users will understand why these results were placed in the "government" or "educational institution" categories. And the freely available nature of the data makes them economically feasible. In contrast with automatic clustering techniques, the stable categories should be beneficial because the investment that searchers make in learning the categories is amortized over future searches.

In the next section, we define terminology, describe a framework for search result categorization techniques, and briefly review related work. The remainder of the paper focuses on our contribution, which is to promote the fast-feature techniques, particularly those based on rich and meaningful hierarchies. Section 3 describes six fast-feature techniques that we have investigated and our initial assessments of each technique. Section 4 briefly describes two applications of the techniques, illustrating their practical value. Section 5 discusses alternative ways to augment search results with metadata. Section 6 concludes with a summary and suggestions for future work.

2. SEARCH RESULT CATEGORIZATION FRAMEWORK

In this paper, we use the term *classifier* for any algorithm or software that maps a search result or web page to one or more categories. We next consider three dimensions of a framework for search result categorization: Lean/rich, online/offline and fast-feature/full-feature.

2.1 Lean vs. Rich Categories

Our research focuses on applying meaningful and stable categories to organize search results. We can characterize a set of categories as lean or rich. **Lean categories** are simple, readily understandable categories with modest breadth and depth. In the context of the web, they can be constructed from document attributes such as file formats, DNS top-level domains, and meaningful date or size ranges. As an example of the utility of lean categories, [18] found that using the document type (e.g., product catalog, online shop, call for papers, home page, bulletin board) in searches improved precision of the results.

Rich categories are extensive classifications, taxonomies, ontologies, or other knowledge structures, often professionally developed, that provide "semantic roadmaps" of an area of knowledge that can be useful for searchers [25]. Examples of rich categories include the ACM Computing Classification System, West Publishing's Key Numbers classification of legal topics, Library of Congress Subject Headings, and the US Government organizational hierarchy. Web directories like Yahoo! and ODP organize web sites into thematic hierarchies. They are of interest to us because they cover a small but important portion of the web with high quality. Taxonomies such as MeSH also have been used to organize search results in specialized (non-web) search applications [14, 20].

2.2 Online vs. Offline Categorization

Much work has been done on how to categorize web pages. Figure 1 shows a typical data flow for the process. Categorization can be done either completely online (at query time), or it may require prior processing (offline).

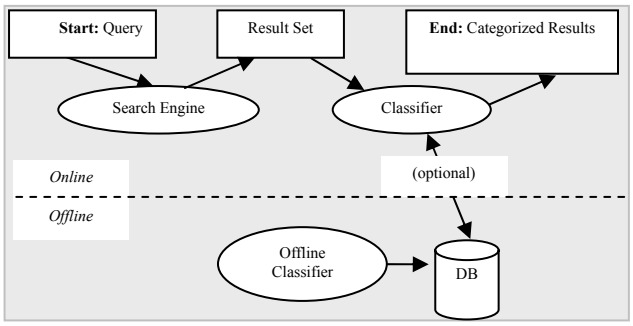

Figure 1. Components used to categorize web search results. A set of search results returned from a search engine is categorized by a classifier. The classifier may optionally reference previously acquired information or knowledge, such as a database of rules or training data.

Online categorization can be done when the search results are generated if the mapping of page to the hierarchy is trivial (for example, grouping by the DNS domain suffix such as .gov, .com, .edu, etc.), or if it comes "for free" with the result set (search engines may provide one or more topical categories for each result), or if it is a function of the result set (such as grouping by document size, where the size ranges depend on the result set). Online categorization can be done from a database, either local or remote (such as querying the Open Directory Project (ODP) web directory (dmoz.org) if the topical category is not provided with the query result set).

Offline categorization is required if no database exists to map search results to the desired categories. In that case, an agent such as a web crawler looks at URLs (fast-feature) or actual web pages (full-feature), potentially creates a hierarchy or reads an existing one, and places that page into the appropriate place in the hierarchy, storing the resulting mapping in a database. Run-time activity is then simply looking up the URL in question in the database and returning the appropriate mapping. Web page classifiers may require offline training to learn statistical models of the categories.

2.3 Full-feature vs. Fast-Feature Techniques

We will distinguish two techniques for categorizing search results. A search-result categorization technique is referred to as **fast-feature** if it requires only information provided in the search result set, and therefore does not require the full text of each link destination. In contrast, a **full-feature** technique is one that *requires* the full text of the link destination (or possibly other documents, e.g., if it uses structural information such as hyperlinks).

Typically information returned includes URL, date, size, and perhaps summary and/or topical category). Thus, for example, a technique such as a text match on the URL would be considered fast-feature, but one that does textual analysis of the body of the HTML page pointed to by the link would not. Table 1 summarizes how these distinctions may divide up the space that describes how search results are analyzed.

Full-feature online techniques would consist of reading a list of links returned from a search engine, and then at runtime, downloading each destination, performing some analysis on each page, and then doing some kind of categorization. This is not easily scalable to large result sets, because it requires N network

calls for N results and is largely dependent on remote sites for correct functionality. While it might be feasible on a set of pages with reliable links and guaranteed fast network performance, or when pages are available on the local machine, it is not practical in general.

Much research has been done on *full-feature offline* techniques by information retrieval classification researchers. In general, these require downloading and analyzing the full contents of each page, whether it is using link data to automatically build site maps as in MAPA [8], or machine-based learning techniques that can categorize pages based on statistical analysis of word counts. Manual categorization, in which page designers are requested to categorize their respective pages can also be seen as a full-feature technique, as it also requires knowledge of the page contents.

Table 1: Techniques for Search Result Categorization, our contribution in black border

	Online (at query time)	**Offline** (requires prior setup or background processing)
Full-feature	Accessing each web page in a search result and doing extensive analysis (not addressed here; often impractical due to performance)	Extensive text processing, manual, link analysis, machine learning (Work done by information retrieval and classification researchers)
Fast-feature	Uses only features in result set, such as title, snippet, URL, domain, size, ODP, pre-existing database map	Web crawler for URL directory hierarchy parsing, search engine mining (query probing)

2.4 Related Work

2.4.1 Online Fast-feature Techniques

Zamir and Etzioni [30] argued that search result clustering algorithms must work well with just the snippets returned by the search engine. They found that clustering on snippets was almost as effective as using the full-text of the document. URLs are often human-readable and can be used for webpage categorization via a two-phase pipeline of word segmentation/expansion and classification, without downloading the entire document [16]. The Clusty web search engine (clusty.com) allows users to organize image search results according to file format, which is readily identifiable from the results. Clustering methods group pages on-the-fly, generating by automated clustering of the title and text snippet. Although these techniques are useful for organizing search results [15], the clusters (and the associated labels) are not stable, and can be ambiguous or confusing [22].

2.4.2 Offline Fast-feature Techniques

In the simplest case, pages can be manually placed into categories by human editors (e.g., LookSmart, Yahoo! or ODP). Rule-based or knowledge engineering systems allow users to construct classification rules [13] for documents. Commercial knowledge management systems such as DataHarmony (dataharmony.com) support both automated rules and manual assignment.

Machine learning techniques train classifiers using labeled example data [23]. These techniques require extensive offline configuration, but can effectively categorize 70% or more of search results [7]. Query probing approaches have been used to categorize databases by issuing queries and analyzing the results. QProber classified Web-accessible databases by issuing a set of queries (query probes) to each database and analyzing the counts of the number of results to classify each database into a set of thematic categories [11]. Wang, Meng and Yu [27] used a similar approach, starting with the top 2 levels of the Yahoo! hierarchy.

2.4.3 Offline Full-feature Techniques

Web page categorization researchers have investigated the use of many features as input for classifiers. Dumais and Chen [7] and many others [23] used the full-text of documents as a vector ("bag of words") representation. Sun, Lim and Ng [26] used elements of HTML pages such as <TITLE> and hyperlinks as the features to classify. They found that the use of context features, especially hyperlinks, could improve the classification performance significantly. Hyperlinks contain semantic cues that can be used as features [4, 26]. The anchor text surrounding a hyperlink can be used in citing documents as one of the features to classify cited documents [10]. The web page structure can be used by considering the URLs and their visual placement instead of the textual content of a page [24]. Extracting these features requires analysis of the full-feature of the target document, and in some cases, analysis of additional (e.g., referring or neighboring) documents, which limits their utility for online processing. Hybrid classification approaches can incorporate supervised and unsupervised classification to build and train machine classifiers, e.g., for the Bureau of Labor and Statistics (BLS) web site. [9]. In this project, classifying using keywords (an uncontrolled vocabulary) was found to be an effective compromise between title-only and full-text classification.

3. FAST-FEATURE CATEGORIZATION

This section discusses three kinds of fast-feature classifiers. We briefly consider the online lean techniques before focusing on our primary interest, the online rich techniques. We close out the section by describing an offline technique that uses search engine query probing to develop a classifier for specialized web sites.

The fast-feature techniques draw on meaningful relationships between a feature in the search result and some external database or knowledge structure. If the relationship exists, that is evidence of membership in the category. The converse, however, is not true. If no relationship exists, that simply means we do not know. When analyzing these techniques, an important characteristic is what proportion of search results can be categorized. To assess the potential utility of these methods, we implemented examples of each of the three kinds of classifier. We then measured the percentage of search results that each categorized or analytically determined the coverage. Each classifier was targeted to a specific domain, so five representative queries were constructed for each target domain. For each query, the top 100 search results were retrieved from the Google search engine, and the number of results categorized by the classifier was measured. We performed an additional analysis on the ODP classifier.

3.1 Online Lean Techniques

A fast-feature online categorization technique is one that does not require the offline creation of a database, and also does not require the full text of the link destinations. The lean techniques often draw on surface features of the URL, such as the top-level domain to classify documents into simple categories. Table 2 describes lean classifiers. We do not claim this is a complete list, but it illustrates the breadth of classifications available using only the data returned from the search engine and any freely available, pre-existing databases.

Table 2: Online lean classifiers can provide simple categories to help users locate relevant information. The three classifiers that were initially implemented are highlighted in bold.

Name	Description
Top-level DNS Domain	This classifier extracts the final part of the hostname, which typically indicates either a country code (e.g., us, jp, uk, de, etc.), or one of {com|edu|org|gov|…}. This provides a simple way to provide a flat (non-hierarchical) categorization. A search for "chip manufacturers", for example, could be usefully organized according to country code.
Last Time Visited	The web browser history can be used to categorize documents by how recently they were visited (e.g., today, yesterday, this week, this month, never).
Document Format	The file format of the document (e.g., HTML, PDF, PS), can often be determined from the suffix of the filename in the URL or from a format indicator in the search results.
Document Language	The document language can be inferred from the title and snippet using dictionary lookup, yielding a flat categorization.
Document Size	This classifier groups results into similar size classes. Size categorization may be useful for image search.
Document Indexing Date	Search engines sometimes provide the date the document was indexed (or "crawled") in search results. This can be used to categorize documents by how recently they were indexed, using values similar to the previous example.

3.1.1 Top-Level DNS Domain Classifier

The domain classifier is the simplest of the online classifiers we implemented, and places URLs into a flat set of about 110 categories based on the domain suffix {com|edu|gov|int|mil|net|org|arpa|nato}, or the appropriate country code. A simple lookup table maps the country code to country name, so that the categorization text can use the actual country name. For example, the following two URLs would be categorized as follows:

- www.whitehouse.gov/ -> GOV
- http://www.corriere.it/ -> Italy

A user interface showing this categorization would allow quick navigation to all educational institution web sites, for example. Because the domain is available in every search result, this has the desirable property of 100% coverage, that is, no results are left "uncategorized." Country codes may not be immediately recognizable to searchers, and at least one country (Tuvalu) has used its top-level domain (.tv) to host television websites, which could be initially confusing or mislieading.

3.1.2 Last Time Visited Classifier

Categorizing search results by when they were last seen can be useful in certain situations. Although users attempt to re-access previously found documents via search engines, they have trouble remembering the specific query and/or navigation sequence that they originally used [1, 28]. Integrating these categories into a search interface could help searchers more readily find previously visited pages. Alternatively, these pages could be excluded from search results if the searcher wished to find new material. Personal browse histories maintained by a web browser can be used to indicate whether a web page or its web site has been visited and if so, when it was last visited. Our classifier categorizes web pages into five categories: Today, Yesterday, Within a Week, Before Last Week, and Never Visited. This classifier depends on the existence of a complete browse history, which introduces the issues of privacy and data storage size. The initial implementation works with the Firefox web browser, using. It uses an external script to read the web browser history file, which is only updated when the browser exits, so sites visited in the current session are not immediately visible. If a complete browse history is available, this technique will provide 100% coverage, because any page not in the history can accurately be placed in the "Never Visited" category. If the browse history is limited, however, the "Never Visited" category cannot be used, because the absence of a page in the history file could either mean the page was never seen, or that it was seen but subsequently removed from the history.

3.1.3 Document Size Classifier

Since search engines return size information for pages, a dynamic categorization of sizes can be built automatically, and this classifier can thus also run online. This could be useful when searching for images or multimedia documents. Categorization may be done uniformly (which may yield many categories with 0 results), or by online defining ranges that contain matches within the result set. If the negative result is desirable, for example if the user may want to notice that there are no results with file size between X and Y KB, then the first approach would be appropriate, whereas if a good visual of the distribution is desired, the second would work better. Our implementation defines a constant number of groups, divides the range of page sizes by the number of groups, and then places the results into one of those groups. This is useful for visualizing a uniform distribution of page sizes. An alternate implementation could choose categories of fixed intervals, such as 100-200k, 200-300k, etc., even if the categories were not a uniform size. This would be useful for seeing, for example, that no results were between 100k and 3MB for a given query. If both of these implementations were published and adhered to the common interface, a user could choose which size classifier to use based on the desired visualization or search. Note that this classifier will trivially yield 100% coverage.

3.2 Online Rich Techniques

The fast-feature rich techniques typically use a pre-existing database to map a URL to one or more categories. Table 3 describes several rich classifiers. As in the previous section, this illustrates the breadth of classifications available.

Table 3. Online rich classifiers can provide meaningful and stable categories that add context to the search results.

Name	Description
US Government	This classifier uses a pre-existing database that maps URLs to a government hierarchy. For example www.whitehouse.gov/president maps to the second-level category "Executive/Executive_Office_of_the_President".
Open Directory Project (ODP)	This classifier uses the Open Directory Project category information that is returned with the query results to build its hierarchy. The ODP is a human-edited web directory (www.dmoz.org).
Musical Genre	This classifier parses search results from the AOL Music search engine to categorize songs according to a two-level musical genre. (A similar classifier categorizes songs by period.)

3.2.1 U.S. Government Classifier

The government classifier uses an existing database that maps government web pages into a government hierarchy, for example mapping *http://www.af.mil/* to the hierarchy node "/Executive/Executive_Agencies/Department_of_Defense/Department_of_the_Air_Force". Since the lookup is done locally, this can be done online at query-time. On its own, this classifier has coverage that is limited to the list of URLs in the database. We extended coverage by using prefix matching, i.e., *any* URL beginning with *www.af.mil* would be mapped to this node, unless a more detailed match was found. Five representative queries were constructed by selecting the most commonly asked questions reported by the First.Gov web site (http://answers.firstgov.gov/cgi-bin/gsa_ict.cfg/php/enduser/std_alp.php), removing obviously navigational questions, as described in [2], and creating short queries from keywords in the questions. The results are shown in Table 4. For both the "new passport" and "foreign embassy" queries, many of the uncategorized pages were from the domain "usembassy.gov", whereas the database had "usembassy.state.gov". This slight difference illustrates the sensitivity of this approach to URL variations, and suggests that additional heuristics could be developed to make it more robust.

Table 4. Percent of top 100 results categorized by US Government classifier for five representative queries.

Query	% Categorized
new passport site:gov	39
start business site:gov	58
gasoline prices site:gov	100
foreign embassy site:gov	43
obtain grant site:gov	72

3.2.2 Open Directory Project Classifier

The category classifier uses ODP information to place search results into categories within the ODP hierarchy. Even though web directories cover only a small fraction of the web, popularity follows a power law [6]. That is, a few sites receive much use. We conjectured that the highest ranking pages in search results would often be cataloged in the ODP. To categorize a search result into the ODP hierarchy, the web site is looked up in the ODP using prefix matching as in the US Government classifier. Since web sites can be cataloged in multiple categories, this yields a list of categories for the result. For example, a web page from the web site of the University of Maryland Human-Computer Interaction Lab would be categorized into the following three ODP categories:

- /Computers/Human-Computer_Interaction/Academic
- /Computers/Computer_Science/Academic_Departments/North_America/United_States/Maryland
- /Reference/Education/Colleges_and_Universities/North_America/United_States/Maryland/University_of_Maryland/College_Park/Departments_and_Programs

The classifier used a web service provided by Alexa.com. The Alexa service only categorized a single web page per HTTP request, so we implemented a cache to minimize processing time for large sets of search results.

Five queries representative of general web search were selected from the most common searches reported by AskJeeves search engine (http://sp.ask.com/docs/about/jeevesiq.html), after removing navigational queries. In addition, the five government queries described above were also evaluated (Table 5).

The preliminary tests were promising, but we wished to measure coverage for a more extensive set of searches. We particularly wanted to measure coverage rates for the ODP when used for general web search, because we would be building a categorizing search prototype for this purpose. We chose the TREC 2004 Robust Topics to provide a set of queries because it was created as a set of realistic, but difficult topics for information retrieval.

For each of the 250 topics, we submitted the contents of the Title field to a Google search and requested the top 350 results. This yielded 86,900 results. Because of the quantity of results, it was not practical to use the Alexa service to categorize them. We downloaded the ODP data and imported it into a MySQL database, and processed the results using PHP scripts. We then checked to see if each result could be categorized in the ODP. We measured the number of results categorized within the top 100, 250 and 350 results (Table 6). The average coverage for the 246 queries successfully processed and categorized was 66.0%, 62.9% and 61.6% for the top 100, 250 and 350 results, respectively.

We briefly compare our work with work by Chirita, et al. [5]. They used ODP data to re-rank Google search results, boosting the rank of preferred categories, which were selected in advance by the searchers. The found that the top 5 re-ranked results were judged better than the original top 5, which illustrates the value that a large-scale knowledge resource can provide. Our use of the ODP differed in that we wished to expose the structure of the search result to the user in the form of an overview, thus avoiding the need to pre-specify categories of interest. We observed higher coverage results in our tests, and we can consider two possible causes for this. They elicited specific types of queries (ambiguous, partially ambiguous, and unambiguous) from their

Table 5. Percent of the top 100 results categorized by the Open Directory Project classifier for five representative queries in each of two domains: general web search and government web search.

Query	% Categorized
General web search	
music lyrics	76
Games	83
Maps	90
real estate	82
Poems	76
Government web search	
new passport site:gov	69
start business site:gov	73
gasoline prices site:gov	90
foreign embassy site:gov	68
obtain grant site:gov	88

Table 6. Coverage for the top 100, 250 and 350 search results from 246 queries based on the TREC 2004 Robust Topics.

	Range	Mean (SD)	% Categorized
Top 100	36-87	66.0 (7.68)	66.0
Top 250	87-194	157.2 (16.00)	62.9
Top 350	110-257	215.6 (21.11)	61.6

test participants, who were research colleagues, whereas we used a set of TREC topics. It is possible that their queries were focused more narrowly to yield the desired level of ambiguity. It is also possible that the prefix matching strategy allowed our classifier to categorize a larger fraction of pages. The evaluation of the working system (see Section 4) lends support to the prefix matching approach, although it has not been fully evaluated.

3.2.3 Music Genre Classifier

The Music Genre classifier was constructed to categorize search results from the AOL Music search engine into a two-level musical genre. It uses the open source Freedb.org CDDB database, which contains entries for 1.9 million CDs. At query time, the song title, artist, and album are used to index into the CDDB data and find the entry for that song. The genre is then extracted from the entry. A similar classifier was built to categorize songs by song era. It illustrates how search results with limited meta-data (in this case, song title, artist, and album title) can be augmented by integrating large-scale knowledge resources in a simple, yet novel way.

3.3 Offline Techniques

When pre-existing mappings from URL to a topical hierarchy such as ODP are not available for web sites, they must be generated offline in order to guarantee fast query-time performance. The techniques outlined here are fast-feature in that they do not require text analysis of the link target. The assumption is that a database can be quickly built using these techniques without resorting to more complex algorithms, in the general case of a set of web pages without a pre-existing categorization.

3.3.1 Directory Hierarchy Parsing

One simple way to build a page hierarchy from a web site is to look at the directory hierarchy. For example, a search result for "taxes" in the whitehouse.gov domain could be categorized as "Vice President" if the URL is in a subdirectory of www.whitehouse.gov/vicepresident, or as "President" if the URL is in a subdirectory of www.whitehouse.gov/president.

For websites with well-defined directory hierarchies, this approach could prove fruitful. However, there is no rule that requires these directories to keep their names, and a restructuring of the website could destroy the entire inferred hierarchy. Since the web is based on links and not absolute paths, a completely flat directory structure with well defined links is perfectly legal on the web. It is therefore unreliable in general to depend on directory structure unilaterally, even though in many cases it would seem a reasonable way to proceed. Shih and Karger [24] presents an in-depth discussion of the problems inherent in using URLs semantically, along with areas in which URL parsing does have some success.

If implementers want to support fast-feature-offline creation of a directory hierarchy, they should enforce policies on their website that require the directories to have hierarchical meaning, so that an automated categorization using that data will provide meaningful results.

3.3.2 Search Engine Mining (Query Probing)

Search engine mining is the process by which queries are made to a search engine in order to collect a set of related URLs that the search engine has amassed through its web crawling. Since the mined classifier relies on a separate background process that is doing the mining and keeping its database up to date, it can be seen as an offline implementation. It can be used as a tool in building lightweight classifier, in the following manner:

1. A hierarchy or classification is first defined. This may already exist in the form of a site map, or a grouped list of links on a site's home page.
2. Each term in the hierarchy is used to construct a query to a search engine, which returns a predefined number of resulting URLs.
3. These URLs are stored in a database and mapped to the node in the hierarchy whose descriptive text generated them as hits.
4. Online categorization consists of searching for each URL in the database (possibly using pattern matching to support inexact or partial matches), and placing that URL at the corresponding place in the existing hierarchy.

Online categorization using this technique may be difficult to assess, because the results vary widely depending on, for example, the quality of the original hierarchy used for mining, the number of queries performed, which search terms were used, and the number of words in each query. We identified seven factors that impact the quality of the results:

- **The "magnitude".** The "magnitude" refers to the number of results we request during population of the database. The optimal magnitude should be a function of the size of the website, although it may only be meaningful up to a given number, since it is not clear at what point the quality of the result degrades so much as to be meaningless.
- **The number of pages in the website**. Fewer pages increase the probability that the mining will turn up *all* results that map to the given node in the hierarchy; i.e., that the average result set during mining is less than the magnitude.
- **The average number of words in the hierarchy terms.** If the hierarchy consists of terms like "students" and "faculty", a much larger magnitude would be needed than a hierarchy with terms like "producer price indexes" or "wages by area and occupation", since there would be more results to mine from the search engine.
- **The number of nodes in the hierarchy.** More terms implies a smaller magnitude, since each node would be more specific and would return better/fewer results.
- **The depth of the hierarchy**. A deeper hierarchy also implies a smaller magnitude due to greater specificity.
- **The type of hierarchy.** Can internal nodes contain URLs, or only leaf nodes? It is also domain-dependent whether hierarchical terms are ANDed in the query, or if just the lower-level text is used. For example, some hierarchies could have "Baseball" with subnodes "Teams", "History", "Hall of Fame", and "Trading Cards", which would require ANDing the subnodes with their parent to get reasonable results. On the other hand, if the parent node were "NY/NJ/CT Traffic Patterns" and the child nodes were "NY Patterns", "NJ Patterns", and "CT patterns", it would be preferable to just look at the child text.
- **The distribution and scope of the website's content.** The distribution and scope of a website's content will likely affect quality of results.

We used the generic engine to construct a custom categorizer for the Bureau of Labor Statistics (BLS) web site, which contains approximately 123,000 pages. The categories (15 top-level and 87 second-level) were taken from the BLS home page. For each category, the top 500 results were requested (a magnitude of 500), although the actual number of results returned varied. This yielded a modest size database of 23,000 entries. Note that a single web page can appear in multiple categories. Five queries representative of web search on the BLS web site constructed by selecting the most commonly asked questions reported (http://www.bls.gov/dolfaq/blsfaqtoc.htm), removing obviously navigational questions, and creating short queries from keywords in the questions (Table 7). The classifier successfully categorized 48-94% of the results into a category.

This technique is limited in important ways. The relationship between the category and the categorized item is not as well-defined for this method as for the others, because it relies on the search engine to compute relevance to the category via the constructed query terms. Thus accuracy, which we could expect to be very high with the other classifiers, must be carefully evaluated. Moreover, it is very sensitive to the many parameters described above. These limitations constrain the practical application of this technique as described, however it does illustrate that such a classifier can yield high coverage rates. This approach and the BLS classifier mentioned earlier [9] illustrate two mid-points on the continuum between automated metadata extraction and manual annotation.

Table 7. Percent of the top 100 results categorized by the Search Engine Result Mining classifier for five representative queries in the BLS website (www.bls.gov).

Query	% Categorized
consumer price index site:www.bls.gov	94
es-202 site:www.bls.gov	77
civilian noninstitutional population site:www.bls.gov	48
seasonal adjustment site:www.bls.gov	63
employee benefits survey site:www.bls.gov	94

4. APPLICATION OF FAST-FEATURE TECHNIQUES

We implemented these fast-feature classifiers in our SERVICE web search prototype that displays a ranked list of search results with an interactive overview based on topical, geographic, government, and last-time-seen categories. The SERVICE system sends user queries to the Google search engine, retrieves the top 100 results, and categorizes them using fast-feature classifiers. The ODP categories are restructured slightly to extract the geographic categories for the overview.

The SERVICE prototype allows the categorized overview to be enabled (as shown in Figure 2) or disabled (which simply shows the ranked list of results. Clicking on a category filters the displayed results to just the pages within that category. Moving the pointer over a category highlights the results in that category, and vice versa. We have conducted a comparative evaluation, asking users to perform an exploratory search task. Preliminary analysis of the results indicate that with the overview users explored deeper within their search results while remaining more organized, yielding a more stimulating and satisfying experience.

A second application of fast-feature classifiers was built for the AOL Music search engine. This prototype allows users to search for songs, and categorizes the results according to genre and era (Figure 3).

5. ALTERNATIVES TO FAST-FEATURE CLASSIFIERS

The fast-feature techniques are useful when limited metadata is available, but an explicit approach will yield much more precise categorization, which is our ultimate objective. Specifically, if sites were to publish a machine-readable site map (call it "sitemap.xml" for example), and place it in a standard location, categorization engines would be able to classify pages precisely as the authors intended. Such engines might periodically read sitemap.xml to update their internal categorization of pages, and page authors would just need to specify where in the sitemap their page should be located, possibly in multiple locations. Google SiteMap (https://www.google.com/webmasters/sitemaps/) provides a very similar service, which allows web site maintainers to directly submit a sitemap to Google, and it is easy to envision adding category information to this. Future work could focus on

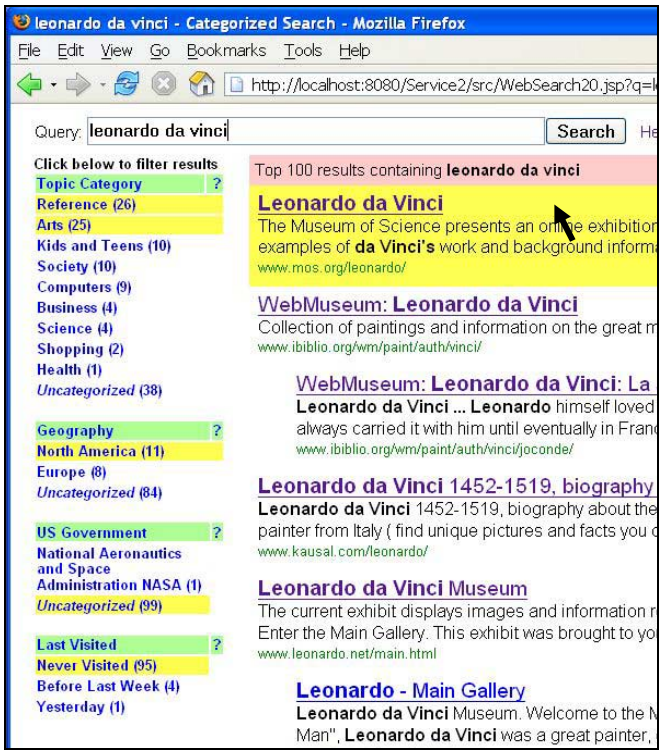

Figure 2. The SERVICE web search prototype uses fast-feature techniques to categorize search results and generate interactive overviews based on meaningful and stable categories. This detail shows that the top result is categorized by topic (Reference and Arts) and geographic region (North America), and it has never been visited by this searcher.

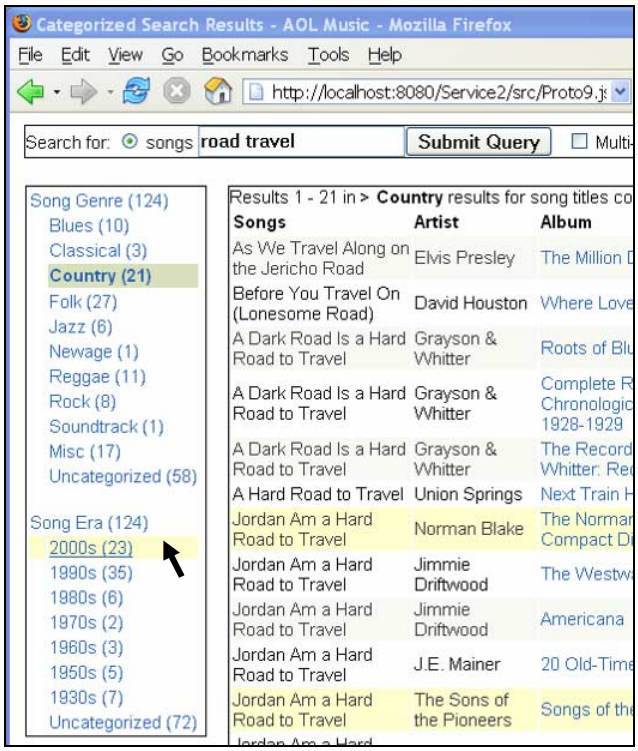

Figure 3. The AOL Music search prototype categorizes song search results by genre and decade. In this screenshot, the search results have been filtered to the top 21 results in the Country genre. The pointer has been placed over the 2000s song era category, highlighting the two visible songs from that era.

what the sitemap should contain, whether it should be a forward or backward index, and the feasibility of making the name of the sitemap file and its relative location a web standard.

The Semantic Web could also be mined for categorical information, in a manner similar to that described in Guha, McCool and Miller [12], which augments web search results with information extracted from RDF stores. Semantic Web standards such as the Simple Knowledge Organisation System (SKOS) could be used to share and distribute classifications [19]. Even as standards emerge, however, fast-feature classifiers that work on a limited domain could benefit from sites that publish such information.

6. CONCLUSIONS

To better support search in digital libraries and web resources, we have proposed a three component framework (lean vs. rich, full-feature vs. fast-feature, and online vs. offline) for search result categorization techniques and implemented six "fast-feature" techniques that utilize features available in the search result list. Five techniques employ readily available, credible knowledge resources (the Open Directory thematic hierarchy, a US government organizational hierarchy, DNS domain, document size and personal browsing histories) to produce meaningful categorizations. This helps overcome the metadata challenge posed by the growth of semi-structured and unstructured digital documents. Initial implementations of fast-feature, online techniques showed promise for quick categorization at query-time. Additionally, we propose and implement an offline method of web-page hierarchy-building that can be performed on a per-site basis. We applied our software to three search domains and measured the percent of results categorized for five representative queries in each domain: general web search (76-90%), government web search (39-100%), and the Bureau of Labor Statistics website (48-94%). An additional test submitted 250 TREC queries to a search engine and successfully categorized 66% of the top 100 in the ODP and 61% of the top 350. These initial results are encouraging and warrant a more comprehensive evaluation. Based on our implementation experience and initial evaluation, we propose approaches to improve categorization rates and offer suggestions that web site designers could apply to their sites to support fast categorization of search results.

This work was motivated by our research on user exploration and understanding of large sets of web search results. We have incorporated the fast-feature online techniques into our SERVICE web search prototype to enable user-controlled reorganization of search results using multiple categorizations.

Interactive overviews of web search results can support user exploration of large result sets. The growth of semi-structured data, epitomized by the web, requires techniques to work with limited metadata. The techniques described in this paper begin to satisfy this requirement, complementing more traditional classification techniques. They are straightforward to implement and easy to deploy. More importantly, their use of meaningful and

stable categories can support informed exploration and better understanding of search results. Fast-feature categorization of search results is a promising research direction, and could emerge as a valuable strategy for improving search result categorization.

7. ACKNOWLEDGEMENTS

We would like to thank Abdur Chowdhury, Ryen White, Craig Murray, and the anonymous reviewers for their helpful comments. This research was supported by an AOL Fellowship in Human-Computer Interaction and National Science Foundation Digital Government Initiative grant (EIA 0129978) "Towards a Statistical Knowledge Network."

8. REFERENCES

[1] Aula, A., Jhaveri, N. and Käki, M. (2005). Information search and re-access strategies of experienced web users. *Proceedings of the 14th International Conference on the World Wide Web*. 583-592.

[2] Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, *36* (2). 3-10.

[3] Ceaparu, I. and Shneiderman, B. (2004). Finding governmental statistical data on the Web: A study of categorically organized links for the FedStats topics page. *Journal of the American Society for Information Science and Technology*, *55* (11). 1008 - 1015.

[4] Chakrabarti, S., Dom, B. and Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *ACM SIGMOD Record*, *27* (2). 307-318.

[5] Chirita, P.A., Nejdl, W., Paiu, R. and Kohlschütter, C., Using ODP metadata to personalize search. in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (Salvador, Brazil, 2005), ACM Press, 178-185.

[6] Cunha, C., Bestavros, A. and Crovella, M. (1995). *Characteristics of WWW client-based traces* (No. TR-95-010). Boston University. Retrieved January 24, 2005, from http://cs-www.bu.edu/faculty/crovella/paper-archive/TR-95-010/paper.html.

[7] Dumais, S. and Chen, H. (2000). Hierarchical classification of Web content. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 256-263.

[8] Durand, D. and Kahn, P. (1998). MAPA: A system for inducing and visualizing hierarchy in websites. *Proceedings of the Ninth ACM conference on Hypertext and Hypermedia*. 66-76.

[9] Efron, M., Elsas, J., Marchionini, G. and Zhang, J. (2004). Machine learning for information architecture in a large governmental website. *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries*. 151-159.

[10] Glover, E.J., Tsioutsiouliklis, K., Lawrence, S., Pennock, D.M. and Flake, G.W. Using web structure for classifying and describing web pages *Proceedings of the 11th International Conference on World Wide Web*, ACM Press, Honolulu, Hawaii, USA, 2002. Retrieved, from

[11] Gravano, L., Ipeirotis, P.G. and Sahami, M. (2003). QProber: A system for automatic classification of hidden-Web databases. *ACM Trans. Inf. Syst.*, *21* (1). 1-41.

[12] Guha, R., McCool, R. and Miller, E. (2003). Semantic search. *Proceedings of the 12th International Conference on World Wide Web*. 700-709.

[13] Hayes, P.J., Andersen, P.M., Nirenburg, I.B. and Schmandt, L.M. (1990). TCS: A shell for content-based text categorization. *Proceedings of the Sixth Conference on Artificial intelligence Applications*. 320-326.

[14] Hearst, M.A. and Karadi, C. (1997). Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 246-255.

[15] Hearst, M.A. and Pedersen, J.O. (1996). Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 76-84.

[16] Kan, M.-Y. (2004). Web page classification without the web page. *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters*. 262-263.

[17] Kules, B. and Shneiderman, B. (2004). Categorized graphical overviews for web search results: An exploratory study using U.S. government agencies as a meaningful and stable structure. *Proc. Third Annual Workshop on HCI Research in MIS*. 20-24.

[18] Matsuda, K. and Fukushima, T. (1999). Task-oriented world wide web retrieval by document type classification. *Proceedings of the Eighth International Conference on Information and Knowledge Management*. 109-113.

[19] Miles, A. and Brickley, D. SKOS Core Guide, 2005. Retrieved 4/5/2006, from http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/.

[20] Pratt, W., Hearst, M.A. and Fagan, L.M. (1999). A knowledge-based approach to organizing retrieved documents. *Proceedings of the 16th National Conference on Artificial Intelligence*. 80-85.

[21] Rieh, S.Y. (2002). Judgment of information quality and cognitive authority in the Web. *Journal of the American Society for Information Science and Technology*, *53* (2). 145-161.

[22] Rivadeneira, W. and Bederson, B.B. (2003). A Study of Search Result Clustering Interfaces: Comparing Textual and Zoomable User Interfaces. *University of Maryland HCIL Technical Report HCIL-2003-36*.

[23] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, *34* (1). 1-47.

[24] Shih, L.K. and Karger, D.R. (2004). Using URLs and table layout for web classification tasks. *Proceedings of the 13th International Conference on World Wide Web*. 193-202.

[25] Soergel, D. (1999). The rise of ontologies or the reinvention of classification. *Journal of the American Society for Information Science and Technology*, *50* (12). 1119-1120.

[26] Sun, A., Lim, E.-P. and Ng, W.-K. (2002). Web classification using support vector machine. *Proceedings of the 4th International Workshop on Web information and Data Management*. 96-99.

[27] Wang, W., Meng, W. and Yu, C. (2000). Concept hierarchy based text database categorization in a metasearch engine environment. *Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00)*. 283-290.

[28] Wen, J. (2003). Post-valued recall web pages: User disorientation hits the big time. *IT & Society*, *1* (3). 184-194.

[29] White, R.W., Kules, B., Drucker, S.M. and schraefel, m.c. (2006). Supporting exploratory search. *Communications of the ACM*, *49* (4). 36-39.

[30] Zamir, O. and Etzioni, O. (1998). Web document clustering: a feasibility demonstration. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 46-54.