Effectiveness of Shallow Hierarchies for Document Stores

Karen Corral
Department of Information Technology and Supply Chain Management, Boise State University, Boise, ID, United States., karencorral@boisestate.edu

Gregory Schymik
Department of Accounting and Information Systems, Emporia State University, Emporia, KS, United States., gschymik@emporia.edu

David Schuff
Management Information Systems, Temple University, Philadelphia, PA, United States., schuff@temple.edu

Robert St. Louis
Information Systems, Arizona State University, Tempe, AZ, United States., St.Louis@asu.edu

The Effectiveness of Shallow Hierarchies for Document Stores

Gregory Schymik
Emporia State University
gschymik@emporia.edu

Karen Corral
Boise State University
karencorral@boisestate.edu

David Schuff
Tempe University
schuff@temple.edu

Robert St. Louis
Arizona State University
st.louis@asu.edu

ABSTRACT

Employees spend as much as 4.4 hours every week searching for documents that they never find. Despite this cost, most managers continue to believe that there is no viable alternative to keyword search. In this paper we present the results of an experiment which uses the eight level hierarchy of ABI/Inform to test how many levels are necessary to retrieve one specific paper. Our findings demonstrate empirically that a browsable subject hierarchy of just four levels provides almost as accurate a search result at deeper layers. Therefore the cost of implementing and maintaining a browsable hierarchy is not nearly as high as is frequently estimated. This has significant implications for both researchers and practitioners.

Keywords

Search, document management, hierarchies.

INTRODUCTION

All organizations face the challenge of managing an ever-growing store of documents. An organization's collective knowledge is contained within the memos, white papers, and documentation which are most frequently stored digitally. A recent study by IDC[1] claims that digital content follows its own "Moore's Law" of growth, where content doubles every two years. The ability of employees to successfully navigate that collection of digital content is critical to their productivity and competitiveness. However, significant amounts of time are wasted on information search. Studies have shown that employees spend up to 35% of their time searching for information, but of those searches, only 50% are successful.[2]

One popular solution to the enterprise search problem is full-text keyword search. The appeal of this solution is clear. Keyword search requires little or no set up cost, and only requires the software necessary to index the organization's set of documents. However, studies have shown that the lack of precision of keyword search is a serious drawback, making it inferior to more managed solutions such as tagging documents with metadata and constructing a browsable hierarchy (Corral, Schuff, Schymik and St. Louis, 2010). Hierarchies do not require the user to recall keywords from the document. Instead, the user quickly browses within the appropriate dimension the list of words or terms, narrowing the document set until the correct item is found. These browsable hierarchies have many commercial applications. Netflix and iTunes use them to facilitate finding titles. Library databases, such as ProQuest's ABI/Inform, also use hierarchies to support navigation.

Nevertheless, enterprise search solutions based on browsable hierarchies are far less popular than full-text search. Results of IDC (2011) surveys indicated that employees spend 8.8 hours per week searching for information.[3] Despite this, "C-level and IT executives interviewed by IDC assumed that a solution was too costly or difficult to implement, or that their existing costs and consequences were insignificant."

While people perceive that the up-front cost of establishing and maintaining the metadata store is too expensive, the hierarchy may not need to be nearly as deep as is seen in applications such as library databases. This excessive complexity is one cause of overestimating the up-front effort required to establish the ontology, and also leads to an exaggerated estimate of the ongoing costs of classifying new documents. The result is a skewed view of the true cost of creating a browsable hierarchy.

---

[1] http://www.emc.com/collateral/demos/microsites/emc-digital-universe-2011/index.htm
[2] http://www.kmworld.com/articles/readarticle.aspx?articleid=9534
[3] http://www.ricoh.no/Images/IDC_Executive_Insights_January2011_t_77-4420.pdf

In this paper we demonstrate that an effective, "shallow" hierarchy can be constructed with much less effort than commonly thought. In the next section, we explain why shallow hierarchies are desirable from both the perspectives of user interface design and information retrieval. Then we detail the results of an experiment where we show that the benefits of additional complexity drop off rapidly when the hierarchy's depth exceeds four levels. Finally, we discuss the implications of our findings for practice and future research.

LITERATURE REVIEW

Hierarchy Depth

A keyword hierarchy is typically arranged with a set of broad keywords at the top level, with lower levels containing greater specificity. Figure 1 shows an excerpt from the ABI/Inform subject keyword hierarchy which is seven levels deep.

Business & Industry

- Accounting & taxation
  - Accounting
    - Accounting methods
      - Cost accounting
        - Activity based costing
        - Activity based management
        - Cost accounting
        - Cost allocation
        - Depreciable assets
        - Depreciation
        - Management accounting
        - Recapture rules
      - Financial accounting
        - Financial statements
          - Accounting changes
          - Accounting consistency
          - Accounting irregularities
          - Asset retirement obligations
          - Audited financial statements
          - Balance sheets
          - Cash flow statements
          - Consolidated financial statements
          - Contingent liabilities

Figure 1. Keyword Hierarchy of ABI/Inform

The tradeoff between depth (the number of levels) and breadth (the number of items on a particular level) of menus has been extensively researched. In a summary of the literature (Jacko, Salvendy and Koubek, 1995), menus with greater breadth were found to be more effective than those with greater depth in facilitating navigation and the location of information. The review concluded that the optimal menu depth is two levels. Shneiderman (1998) suggested that users risk getting lost in hierarchies four to five levels deep.

More recently, Galletta, Henry, McCoy and Polak (2006) found that the negative effects of web site navigation delays on performance were worse when users were required to navigate a web site with greater depth. They proposed that this is partially due to increased costs of information foraging (Card, Pirolli, Van Der Wege, Morrison, Reeder, Schraedley and Boshart, 2001; Pirolli 2003), as deeper hierarchies increase uncertainty in the user's navigation path.

Cataloger/Retriever Conflict

In information retrieval, there is an inherent conflict between the cataloger and the retriever. The cataloger would like to put the full text of all documents in a single file. The retriever would like to have easily recognizable metadata. Clearly these are conflicting objectives.

The total costs of a retrieval system include the cost of the search engine itself, the effort required to index documents to a set of subject terms as documents are made available for search, and the cost of developing and maintaining the index of subject terms. This raises the following research question:

> *How does the granularity of the hierarchical structure of the subject metadata used to describe the contents of the documents impact the likelihood of successful search in enterprise-oriented search?*

We conduct an experiment to manipulate the hierarchical structure of the subject index used in the searches so that the impact of the index structure on search results can be evaluated. Specifically, we collapse the depth of the hierarchy and observe the impact on the precision and reliability of the results. The relevance of this research is driven by the need to understand the true cost of cataloging documents in the subject index. This is a primary concern of those facing the decision to implement such a system.

HYPOTHESES DEVELOPMENT

Single item search queries are performed using query terms selected from a subject index whose structure is manipulated to test the impact that depth level has on results. A single-item search was chosen because most enterprise searchers seek specific information or a specific document. For the purposes of the experiment, a single-item search avoids relevance issues, as only one document is relevant to the search. No judgments need to be made about which documents are relevant to which search tasks.

In experiments, retrieval effectiveness usually is determined by evaluating the relevance of the results returned by searches to the intent of the searcher. Two broad measures are used to quantify relevance: recall and precision. Recall is defined as the proportion of relevant documents contained in the result set or, the number of relevant documents in the result set divided by the number of relevant documents in the collection (Voorhees, 2007). Applying that definition to the proposed single-item search experiment will result in a binary measure: the document sought will either be in the result set or it will not. Such a measure will not allow for comparisons of effectiveness beyond a very coarse measure: success or failure. Such a coarse measure will not allow for an adequate answer to our research question. Therefore, recall will be operationalized as a measure of the position – or *rank* – of the sought-after document in the result set. Blair's futility points (2002) and Fagin, Kumar, et al.'s (2003) research indicate that users rarely look beyond the first two pages of search engine results. This shows the need for a rank-based measure of recall.

Precision is defined as the proportion of relevant documents in the result set, or the number of relevant documents in the result set divided by the total number of documents in the result set (Voorhees, 2007). Unlike the definition of recall, the definition of precision does not fail in the context of a single-item search: a ratio of the number of relevant documents (always equal to 1) to the total number of documents in the result set can be calculated whenever the document is found (and is equal to zero otherwise). This definition of precision will be applied but, since the numerator of the ratio will always be equal to 1, precision will be operationalized as the size of the result set or the number of documents returned by the search engine for a given query. In order to answer the research questions posed above, we test the following hypotheses:

> *H1: The depth of the subject thesaurus can be reduced without significantly affecting the size of the result set returned in subject-aided searches.*

> *H2: The depth of the subject thesaurus can be reduced without significantly affecting the rank of the sought-after document in the result set returned in subject-aided searches.*

EXPERIMENTAL DESIGN

The experiment consisted of attempts to find randomly selected documents from a collection using alternative structures for the subject indexes. Rank and result set size are the measures of the quality of the search results obtained via each query. A collection of 6025 documents was used in the experiment. The documents in the collection were taken from six publications: three academic journals (*MIS Quarterly, Information Systems Research,* and *Management Science*) and three magazines (*ComputerWorld, InformationWeek,* and *Newsweek*). Articles were selected from both academic journals and magazines in order to include both long and short articles. Overall, the documents ranged in size from half-page news snippets to 25+ page academic papers.

Microsoft Excel was used to randomly select 288 individual documents to be used as search targets. Figure 2 is a plot of the number of unique, semantically relevant words (those remaining after the application of a stop list) for each of the 288 documents used as search targets. The vertical axis shows the number of unique words in the article.

The horizontal axis labels the documents from 1 to 288. The documents are plotted according to size (number of unique words) from left to right. The plot shows two fairly prominent inflection points at approximately 500 and 1250 unique words. These points were used to segment the results into small documents (those containing no more than 500 unique words), and large documents (those containing at least 1250 unique words) to control for the influence of document size on the results of the experiment.

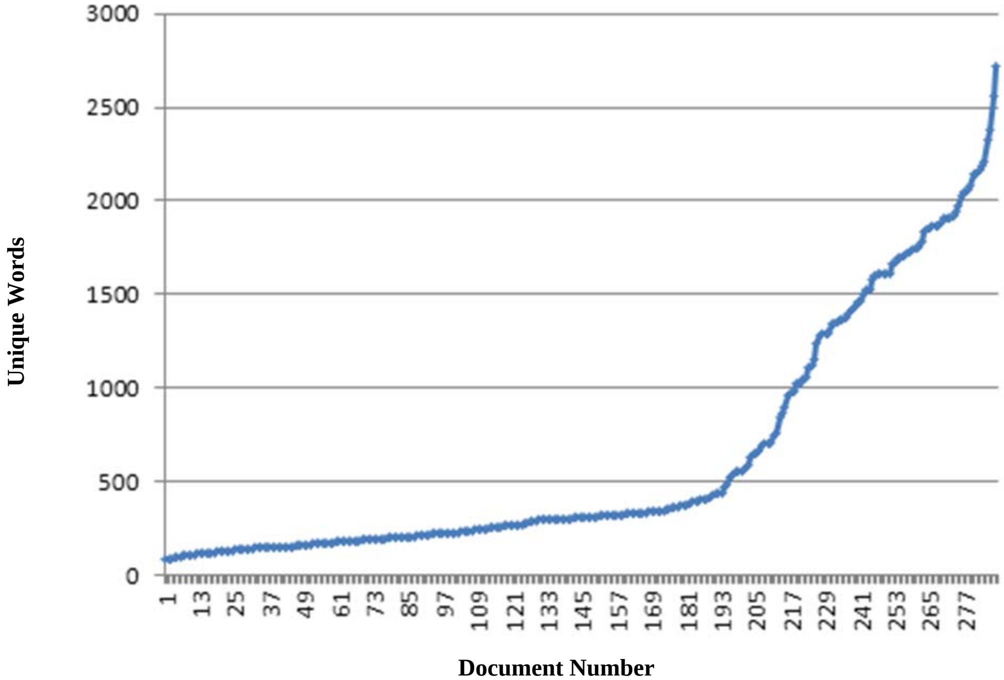

Figure 2. Number of Unique Words in Target Documents

The subject metadata for each document in the collection was downloaded from bibliographic records found for each document in ProQuest's ABI/INFORM research collection. Scripts were written and executed to insert the metadata into each document so that it could be found, indexed, and ultimately searched by the search engine. The IBM Omnifind Yahoo! Edition (OYE) search application was used to run the searches on the collection. The OYE search application uses the Apache Project's open-source Lucene search engine as its core. The OYE search application was configured so that it indexed the full-text of the document and the metadata as two separate fields so that searches could be run on the two fields individually or in combination.

Query Types

To show that queries on subject terms perform better than queries on keyword terms, and to show the performance of queries on subject terms does not decrease as the depth of the subject index is decreased, 8 different query types were defined and run. Two queries made up of either one or two keyword terms are the control for the experiment. Six queries made up of combinations of one or two keyword terms and one or two subject terms are the treatment set for the experiment. The keyword and subject terms are the same in the control and treatment query sets. The eight queries have the following structure:

KW1

KW1 KW2

SU1

SU1 SU2

KW1 SU1

KW1 SU1 SU2

KW1 KW2 SU1

KW1 KW2 SU1 SU2

The queries used in the experiment were generated from the documents themselves in an attempt to avoid the introduction of any bias on the part of the experimenter. The keyword terms were selected based on word frequencies in an attempt to model how users select keywords. Subject terms used in the search of the subject metadata of each document were randomly selected from the subject terms provided by ABI/INFORM. Keyword terms were single-word terms and subject terms were single or multiple-word terms as defined in the subject thesaurus.

Determining how to select the subject terms was very straight forward. The subject thesaurus provided by ABI/INFORM was the source for the subject terms used in the experiment and provides a controlled vocabulary against which the documents in the collection are indexed. It is unique to this specific collection. Each term in the thesaurus is cross-referenced with associated terms in the thesaurus. Subject matter experts are used by ABI/Inform to index each document against the subject thesaurus. Articles in the collection generally are indexed to several subject terms. The subject terms were randomly selected from the terms associated with the article. Because the subject terms are known to have been associated with the sought-after article, this may have overstated the precision and reliability of the results for subject searches. However, the overstatement will be the same for all levels of the index, and hence should not bias our findings on the impact of reducing the depth of the index.

Determining how to select the keyword terms was more difficult. Very little research has been done to understand how searchers select the keyword query terms they submit when searching. Markey (2007) provides a review of the literature on search behaviors. However, beyond identifying a fairly typical broad-to-narrow approach, why searchers choose the terms they use remains an unanswered question. We therefore needed to identify and justify a keyword selection heuristic to be used in this research.

The psychology literature suggests that term frequency will impact an individual's selection of search keywords. Since recall relies on recency and repetition, enterprise searchers seeking documents they've seen before will likely remember the more frequently occurring words in a document and use them as search terms. This leads to the conclusion that an appropriate keyword selection mechanism is one that relies on term frequencies. Several heuristics were identified and examined to determine what would be the best term-frequency-based keyword selection heuristic.

Document Ranking and Best-Case Keyword Selection

Enterprise search engines do not have the benefit of the HTML metadata used to perform the popularity-based analyses (e.g., Google's PageRank) used by web search engines, and are therefore limited to using only term frequency-inverse document frequency (tf-idf). This enables the search engine to rank the documents in the collection that match the search terms according to this tf-idf score and display them to the searcher in descending tf-idf order on the results page.

Various search engines apply various weights to these scores to account for things like document size and collection size and often allow those implementing the search engine to set weights for certain parameters in the formula. Some search engines have been implemented that allow searchers to set weights on search terms. The Apache project online documentation available for the Lucene search engine provides a detailed explanation of the specific tf-idf document ranking formula applied by the OYE search engine used in the experiment.

Given the reliance upon the tf-idf scores for document ranking inside the enterprise, the best possible pair of keywords s to use in the search for a specific document would be the two terms in the document being sought that have the highest tf-idf score. Choosing the two best-performing tf-idf keyword terms as a keyword selection heuristic would introduce a very strong bias towards successful queries. These two keywords (without the benefit of searching the subject metadata stored with a document) should, by definition, return the sought-after document ranked very close to the top of the results.

Given the weakness of the idf component of the tf-idf scores in the enterprise search context (Sarnikar, 2007), a reasonable proxy for the two best tf-idf keywords would be the two most frequently occurring words in the document. A run of queries seeking the 288 documents to be searched for in this research was submitted using the two most frequently occurring – post-stop-list – keywords. These queries returned the sought-after document ranked in the top 20 results in over 89% of the 288 KW1KW2 queries. This result solidified the notion that using the best performing keywords in the experiment would introduce unrealistic bias in favor of good search results because they are not representative of the 50% failure rate reported by several enterprise search surveys.

Another argument against limiting the choice of search keywords to the most frequently occurring words in the document is that the most frequently occurring terms may not be terms a searcher would think are relevant to the search they are performing. That is, the most frequently occurring terms in a document frequently are not semantically relevant to the search.

Realistic Heuristic for Keyword Selection

Since approximately half of all enterprise searches fail, and since research has shown that between 50% and 85% of searchers look only at the first page of results (Jansen and Spink, 2006) a realistic heuristic might be one that returns the sought-after document in the first 10 results in approximately 50% of queries.

To account for the fact that searchers are most likely to remember words that occur most frequently in a document, we used a proportionally weighted random selection process to select the keywords. That is, the two full-text search terms (KW1, KW2) chosen for each document were randomly selected from a keyword distribution weighted according to the frequency of occurrence of each word in the document. Thus if the frequency of occurrence of KW1 was twice that of KW2 for a given document, then the probability that KW1 would be selected also was twice the probability that KW2 would be selected. Any word occurring only once in a given document was excluded from the selection process because such words are unlikely to be remembered by the searcher and could be non-words created as a result of errors in the word frequency counting software or the optical character recognition software used to convert image PDF files to text PDF files.

Examining the experimental results, it appears that this heuristic is a reasonable operationalization of search term selection. The process resulted in a set of queries that contained the sought-after document in the top 10 documents 67.36% of the time, and in the top 20 documents 74.65% of the time. These results are higher than the typical 50% success rate. However, searchers frequently choose keywords that do not appear in the document they seek. Our experiment never chooses such keywords as search terms, and thus our results are biased towards higher success rates. Taking this bias into account, our observed success rates roughly approximate the 50% failure rates reported in surveys of enterprise searchers.

ABI Informs' Subject Index

One organization that has incurred the up-front costs of developing a subject index is ABI Informs. This organization estimates that it takes about 10 minutes per article to determine relevant keywords. The resulting subject terms are placed in a thesaurus and arranged in a tree hierarchy as follows:

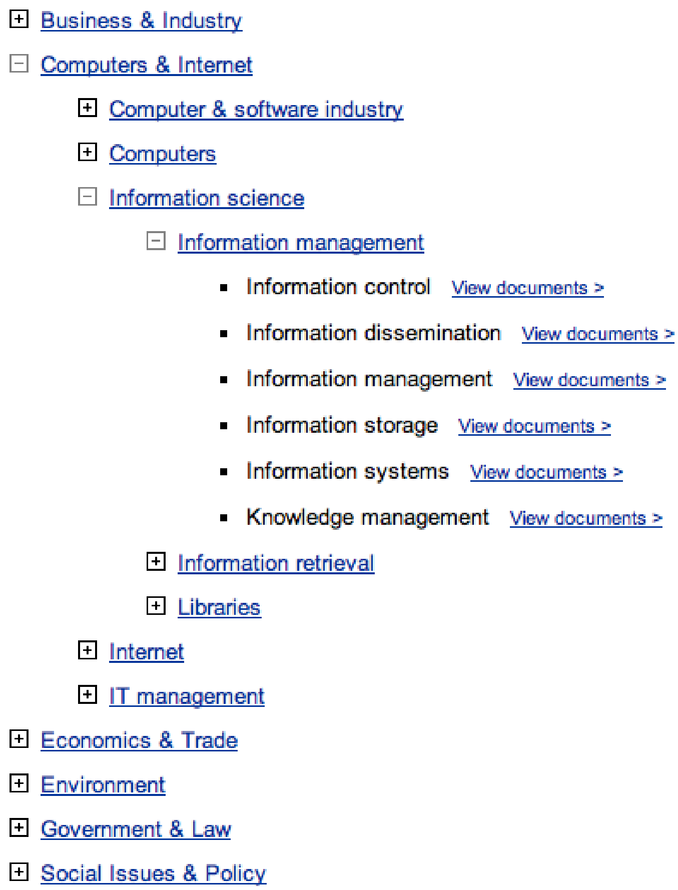

Figure 3. Image of ABI/INFORM Subject Index Structure

As depicted in Figure 3, the documents are indexed only to subject terms at the leaf nodes of any branch of the tree. Because of this, the measures of both depth and breadth occur at these leaf nodes. Depth is the level of the leaf node in the tree with the top level – the level that defines the 8 major categories of subject terms – being level 0, and the level increasing by one until the leaf node is reached (e.g. - "information control" in Figure 3 would be at level 3 in the tree). The measure of depth used in this experiment is the maximum depth of the index.

The breadth of the tree is represented by the average width of the leaf nodes. The width of a leaf node is the number of documents indexed to the subject term at that node. The measure of breadth of the index used in this experiment is the average number of documents indexed to a subject term (i.e. the average width of the leaf nodes in the tree). Our objective was to determine what impact manipulating the structure of the index has on search results. The manipulation of the index in this experiment was achieved by "rolling-up" the index to maximum depths ranging from a depth of zero to a depth of nine.

Rollup Process

In order to test how search results vary at different depths of the index, a single-subject term query was run using the subject term associated with each node of the tree between the leaf node and the root node of the tree. In order for this experiment to work properly, the subject metadata for all of the documents in the collection had to be updated to include the subject terms along the branches of the index associated with every subject term originally indexed to each document.

Using the image of the ABI/INFORM index provided in Figure 3 as an example, if a document in the collection was indexed to the term Information Systems, the subject metadata for that document had to be updated so that it contains the terms Information Management, Information Science, and Computers & Internet. This list of subject terms represents all of the subject terms between the Information Systems node and the root. Adding this information to the metadata allows for the rolling-up of the index along a given branch of the tree. Similar updates had to be made for every subject term to which ABI/INFORM indexed the document. By doing this, querying at any node will return all of the documents indexed to subject terms below that node in the tree.

By running queries at each node above the originally indexed subject term, the experiment, in effect, manipulates the depth of the index one level at a time; rolling-up the indexed terms at each level. This concept can be described using the screen shot of the ABI/INFORM subject index in Figure 3. Assume that a searcher is seeking a document thought to be indexed to the "Information control" subject term. Starting at the Information control leaf node, a subject field query for "Information control" will return all of the documents indexed to that term. Rolling-up the tree one level would limit the searcher to using the next lowest available subject term: "Information management". At this depth in the tree, the child nodes of "Information management" would not be visible to the searcher. The subject metadata for the document being sought would no longer contain the "Information control" term. The searcher would be limited to choosing "Information management" as the subject term for the query. This query would return all of the documents indexed to "Information management". This set of documents would include all of the documents originally indexed to the child nodes of "Information management". This is a larger set of documents than the set returned by the "Information control" query. Rolling-up the tree yet another level would result in a query containing the "Information science" subject term. A further increase in result set size would be expected similar to that just described for "Information management". This rolling-up behavior continues until the top level of the index is reached. For the branch in this illustration, that would result in a subject field query using the term "Computers & Internet".

RESULTS

Table 1 provides descriptive statistics for the nine rolled-up versions of the subject (SU) term index. The Index Level indicates the maximum depth of the tree. Note that the two deepest versions of the index have the same descriptive statistics. This is an indication that the 6 terms that were extended to Level 8 of the index probably should have been incorporated into Level 7 as each was the only child of the node above.

Index Level	Total SU Terms	Mean	Std. Dev.	Min	Lower Quartile	Median	Upper Quartile	Max
8	3587	8.30	39.30	1	1	2	6	1865
7	3587	8.30	39.30	1	1	2	6	1865
6	3503	8.49	39.79	1	1	2	6	1865
5	3232	9.15	41.72	1	1	2	7	1865
4	2655	10.99	46.74	1	1	3	8	1865
3	1558	18.01	68.25	1	1	4	11	1865
2	423	59.91	192.75	1	2	6	25	2062
1	50	445.46	633.39	1	65	144	731.25	2654
0	8	1916.38	1535.64	1	691.75	1621	2549	5021

Table 1. Number of Documents per Subject Term per Level

No major changes occur until the roll-up from Level 4 to Level 3, and the most meaningful changes occur in roll-ups from Level 4 to Level 3 and from Level 3 to Level 2. The average number of documents indexed to a subject term undergoes its first major change between Levels 4 and 3 and then undergoes a much larger change from Level 3 to Level 2. This same impact can be seen when examining the median number of documents per subject term.

Table 1 suggests that the reliability and precision of query results should not deteriorate very much until the depth of the index is reduced by 50% or more. Tables 2 and 3 show that this is the case. For queries using two subject terms (row SU1SU2), fewer than 20 documents are returned until the tree is rolled up to level 4, and even then only 21 documents are returned on average. This provides support for hypothesis 1. Moreover, the sought-after document is among the top 10 documents returned until the tree is rolled up to level 3, and it is among the top 20 documents until the tree is rolled up to level 2. This provides support for hypothesis 2. We constructed similar tables for small documents versus large documents, but do not report the results here because they differed very little, and because of space limitations.

Query Type	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8
KW1	1270.2	1251.2	1261.3	1251.7	1250.8	1250.0	1250.1	1250.1	1250.1
KW1KW2	353.9	348.4	352.1	349.5	348.6	348.2	348.2	348.2	348.2
SU1	2885.2	1384.9	752.2	285.4	233.9	223.2	219.9	219.0	219.0
SU1SU2	2121.9	601.5	232.3	38.2	21.2	19.7	19.6	19.6	19.6
KW1SU1	761.0	383.1	213.3	101.8	88.1	85.1	84.4	84.3	84.3
KW1SU1SU2	583.5	186.1	70.2	18.1	11.2	10.8	10.7	10.7	10.7
KW1KW2SU1	239.9	121.1	62.8	34.6	30.6	29.6	29.4	29.3	29.3
KW1KW2SU1SU2	196.4	56.9	16.4	6.0	4.6	4.3	4.3	4.3	4.3

Table 2. Average Number of Documents Returned By Query Type for All Documents (n=288)

Query Type	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8
KW1	183.8	183.1	185.1	182.6	182.3	182.0	182.5	182.5	182.1
KW1KW2	42.1	41.6	42.1	41.2	41.0	41.0	41.0	41.0	41.0
SU1	605.0	432.6	275.9	112.5	96.9	89.7	88.2	87.9	87.2
SU1SU2	408.5	141.1	44.8	12.5	8.7	7.8	7.9	7.9	7.7
KW1SU1	159.3	93.7	65.2	30.3	26.8	25.2	24.8	24.8	24.9
KW1SU1SU2	110.2	43.0	16.1	5.8	4.7	4.2	4.2	4.2	4.2
KW1KW2SU1	39.6	23.5	15.2	8.4	7.3	7.1	7.0	7.0	7.0
KW1KW2SU1SU2	29.2	11.0	4.2	2.4	2.2	2.0	2.1	2.1	2.1

Table 3. Average Rank of Sought-After Document By Query Type for All Documents (n=288)

CONCLUSIONS AND IMPLICATIONS

In this paper, we provide evidence for the viability and practicality of browsable, subject-based search. The fixed cost associated with the time it takes to encode (or index) documents is a major impediment to the implementation of this type of navigable document structure. The immediacy of this encoding cost, when compared to the somewhat more abstract future benefit of finding documents more quickly, causes organizations to forgo constructing subject indexes.

We contend that while this choice is tempting, it is a mistake, often made due to overestimation of the cost to construct an effective index. Through an experiment simulating document retrieval using a browsable subject hierarchy, we demonstrate the marginal benefits of additional granularity drop off significantly after four levels. After five levels, there is practically no marginal benefit from additional granularity.

These findings have strong implications for both research and practice. The cognitive costs of additional levels of menu depth are well-understood (see Shneiderman, 1998; Galletta et al. 2006). Our findings complement this research by showing that there is little benefit from an information retrieval standpoint to additional depth. Interestingly, our findings show the drop off in benefit of this additional depth occurs at around four levels, the same point as where the cognitive costs have been shown to become suboptimal. Our findings can also be incorporated into previous research on the time cost of keyword versus dimensional search (i.e., Corral et al., 2010) to create a more accurate model of the relative cost of the two approaches.

For practice, our results imply that organizations can create much simpler hierarchies without a meaningful reduction in precision. For example, the subject hierarchy created by ABI/Inform can be reduced by as much as 50% without impacting the quality of the search results. This has practical value, as reducing the granularity of the index structure will greatly reduce the up-front cost, mitigating the inherent conflict between cataloger and retriever that needs to be balanced when defining the index. We believe this makes a compelling case for organizations to construct these indexes, enabling them to take advantage of the greater precision afforded by browsable search hierarchies.

REFRENCES

1. Blair, D. C. (2002) The challenge of commercial document retrieval, part ii: A strategy for document searching based on identifiable document partitions, *Information Processing and Management*,38,2,293-304.
2. Card, S., Pirolli, P., Van Der Wege, M., Morrison, J., Reeder, R.,Schraedley, P., and Boshart, J. (2001) Information scent as a driver of web behavior graphs: results of a protocol analysis method of web usability, *ACM Conference on Human Factors in Computing Systems*,498-505.
3. Corral, K., Schuff, D., Schymik, G. and St. Louis, R. (2010) Strategies for document management, *International Journal of Business Intelligence Research*,1,1,64-83.
4. Fagin, R., Kumar, R., McCurley, K. S., Novak, J., Sivakumar, D., Tomlin, J. A., et al. (2003). Searching the workplace web. *Proceedings of the 12th International Conference on International World Wide Web*, Budapest, Hungary.
5. Galletta, D., Henry, R., McCoy, S. and Polak, P. (2006) When the wait isn't so bad: the interaction of website delay, familiarity, and breadth, *Information Systems Research*,17,1,20-37.
6. Jacko, J., Salvendy, G. and Koubek, R. (1995) Modeling of computer design in computerized work, *Interacting with Computers*,7,3,304-330.
7. Jansen, B. J. and Spink, A. (2006) How are we searching the world wide web? A comparison of nine search engine transaction logs, *Information Processing and Management*,42,1,248-263.
8. Markey, K. (2007) Twenty-Five Years of End-User Searching, Part 2: Future Research Directions, *Journal of the American Society for Information Science and Technology*, 58,8,1123-1130.
9. Pirolli, P. (2003) Exploring and finding information, in J.M. Carroll (Ed.) *Toward a Multidisciplinary Science of Human-Computer Interaction*, San Francisco, CA, Morgan Kauffmann Publishers,157-191.
10. Sarnikar, S. (2007) Automating Knowledge Flows by Extending Conventional Information Retrieval and Workflow Technoglogies in *Department of Management and Organizations*, University of Arizona, Tucson, AZ
11. Shneiderman, B. (1998) Designing the User Interface, Addison-Wesley, Boston, MA.
12. Voorhees, E.M. (2007) Trec: Continuing information retrieval's tradition of experimentation, Communications of the ACM, 50,11,51-54.