

The Science of Large-Scale Information Retrieval

Gregory B. Newby

University of North Carolina at Chapel Hill

Abstract: Information scientists have investigated potentially useful methods for Web search engines and archives, yet relatively few of these methods are in active use. These scientists tend to operate without the imperative for large-scale speed and performance, and may not fully evaluate their innovations. This work presents some promising approaches to information retrieval with suggestions for the practicality of applying them to contemporary large-scale retrieval systems. It is suggested that Web search engines and archives may benefit from incorporating these approaches. For information scientists, a shift to increased concern for scalability and practicality is called for.

Introduction

Information retrieval (IR) is not a solved problem. In spite of recent advances in Web search engines, Internet applications and database technologies, the state of the art for information retrieval is relatively primitive with regards to meeting the many types of information needs.

This document presents an overview of information science methods that may be applied to large-scale document collections and archives. The goal is to assess the current and potential role of each method. It is suggested that information scientists can benefit from considering scalability issues and real-world constraints of their ideas. It is further suggested that Web search and archive developers and implementers can benefit from incorporating ideas from information science in next-generation products and services.

Background: Information Retrieval and Information Needs

Information science is the study of human interaction with information. Information may be considered as any stimulus that changes the cognitive state of a person (cf., Ingwersen 1992). Information retrieval is a discipline of information science that deals with information systems and their associated information needs. An information system may take any form - a computer database, a Web search engine, a library card catalog, etc.

Information needs may also take many forms: retrieving documents that match a topic, answering a question, providing an overview of a subject, learning how to perform a task, comparing products, etc. Information seekers are another source of variety, with different problems to solve, different personal backgrounds and different levels of knowledge.

It is the assumption of this work that the performance of an information retrieval system needs to be assessed relative to the types of information needs, information seekers, and data for which it is designed.

“Performance” is a thorny issue for information scientists, and is often tied to the “relevance” of a system’s responses to a query (where the “query” is a statement of information need – often, these terms are used synonymously. See Schamber et al., 1991). Precision is often used as a measure of retrieval performance, being the proportion of retrieved documents that are relevant. Nevertheless, other measures may also apply, such as quickness of response, currency of data presented, accuracy or authority of data presented, or degree of overlap among responses.

Priority One: Scalability

Although the increase in the quantity of information we would like to manage has recently gone hand in hand with increases in computer performance, the relationship is not one-to-one. As the quantity of information increases, the performance of many algorithms for searching the information begins to suffer.

Consider the inverted index, which is at the heart of many approaches to database and retrieval system design. In an inverted index, a term (that is, a word or word combination) is the key. The data associated with that key is the set of all documents that contain the term. As the number of documents grows, so will the number of unique terms and the relative lengths of each term’s inverted index entry. This is problematic! For most retrieval methods, the first step is to form a set of all documents that contain the search terms. Consider: all other things being equal in a retrieval system, the time to process a query will go up approximately linearly with the size of the document set (which will go up approximately linearly with the size of the data collection being indexed). Is it acceptable that a search that takes one second on a 1TB collection will take ten seconds on a 10TB collection? Probably not.

Unfortunately, many approaches to IR scale worse than linearly. Problems such as finding the nearest neighbor in an information space, sorting results, or merging results from multiple data collections may include algorithms perform exponentially or worse as the size of the collection (or response set) grows (cf. Knuth, 1998).

Information retrieval scientists have typically let scalability take a secondary role to relevance-based performance. For example, the TREC experiments (Voorhees & Harman, 1999), where contemporary research in IR is exhibited, works mostly with collections under 2GB (about 500,000 documents from news wire sources). In order for the data mining, Web searching and archiving operations of tomorrow to benefit from IR research, the research methods must have demonstrated scalability.

Some options that might offer better than linear scalability are discussed below.

The Practicalities of Large-Scale Retrieval

Today, the growth of information is outpacing our ability to keep up with it. Moore's law reliably predicts that computers will be twice as fast, with twice as much disk space, every year to 18 months. But the growth of the Internet in terms of the number of people, number of Web data and number of servers has maintained at least this pace – and far outstripped it in some sectors.

Most recently, there has been extremely high growth in .com, with a growing proportion of dynamically generated data (as opposed to static data that are most accessible to today's searching technologies). Future high growth will almost certainly include non-English languages. Additionally, we can anticipate a higher reliance on more current data – whether from today's newspaper, this week's press release, or a television show being broadcast right now.

Current hardware limitations include the relatively slow growth of memory. In the 1990's, standard mainstream PC memory has gone from 2 to 64MB (32X), while standard disk drives have gone from 40MB to 4GB (1000X). Even the largest servers seldom exceed 32GB of memory. Disk speed has also been a problem – fast disk systems may peak at 160MB/sec (RAID3), versus 40MB/sec of 1991. Many other numbers go into computer system performance, so the picture is not as grim as it might appear. The fact is, though, that information is being produced more rapidly than it can effectively be stored or searched.

Network latency, slow remote servers, bad HTML and poorly configured Web sites all contribute their own challenges.

The challenge of maintaining a very large (multi-TB) disk farm makes us look to near- or off-line storage solutions. Yet, consider whether a tape format from as little as 5 years ago could be read today. Even the CD-ROM (poor as it is for large-scale archiving) has outlived its lifecycle

after 10 years. The only method for insuring long-term access to archived data is to store it on media you can still read. Today, the choice is between disk storage and robotic tape systems, both of which require migration to new media as systems are upgraded.

Finally, consider the archival problems of data types. Fewer than 10 years ago, people were using word processors and other applications that use file formats we can't read today. Can we hope for any better in 10 years? Archiving file format specifications and software (e.g., Netscape & MSIE, word processors, graphics editors) needs to be a requirement for long-term access, not just archiving data. A strategy helped by continued development of HTML, is to track the appropriate DTD. This gives us some hope for rebuilding the original look and content of a document in the future.

The alternative to maintaining knowledge of today's file formats and software is to constantly convert documents to the current format. This is possible, if expensive, in a closed environment (e.g., a business data warehouse), but probably not reasonable to expect of the heterogeneous large-scale Web.

The Science of Retrieval: Methods for Improved Search Performance

The sections below offer brief discussions of approaches to information retrieval that have not been widely used for large-scale Web searching and archiving. At the outset, we will consider a "typical" retrieval system to be geared towards document retrieval by topic based on Boolean methods augmented with term weighting.

"Document retrieval" means that (citations to) documents are the outcome of a search. Compare this to the answer to a question, or a summary of multiple documents. "Topic matching," means that other factors, such as searcher information need, are not considered. "Boolean methods augmented with term weighting" simply means that the retrieval process consists of identifying documents that contain query terms, then examining the frequency of terms within a collection and within these candidate documents to determine which documents to present first.

This is not a fair characterization of all retrieval systems, but offers a common starting point for the discussion.

Information Seeking and Information Needs

Above, it was mentioned that performance should be considered relative to a particular information seeker's information need. What can we know about information seekers that might aid performance? Can such knowledge also help with scalability?

Several models of the information seeking process have been developed. For example, Kuhlthau (1993) identifies six stages in the information search process:

1. Task initiation
2. Topic selection
3. Prefocus exploration
4. Focus formulation
5. Information collection
6. Search closure

As another example, consider Dervin & Nilan's (1986) approach to the information need as "gap bridging" behavior. Information seekers' gaps might include: trying to solve a problem, moving to the next stage, overcoming an obstacle, resolving conflicting information, and so forth.

IR systems can certainly incorporate knowledge about stages in the search process, gap types, or other knowledge about information seekers. There is every indication that documents can be automatically recognized based on certain traits – such as whether they answer questions, provide overviews, integrate topics, etc. Metadata offer another method of providing this type of information about documents.

Scalability opportunities for retrieval based on type of information need or type of information seeker are excellent, at least at the level of stereotypical needs and seekers. The main opportunity comes from segmenting the collection (as discussed further below): if we can identify sub-collections that, for example, offer introductory overview information for a topic domain, we only need to search that sub-collection for seekers of overviews. Recently, some search engines (including AskJeeves) have taken the approach of segmenting based on particular questions that information seekers ask – but not based on classes of information needs.

Knowledge of a User or User Population

Information filtering is the process of choosing new documents that match an existing profile. This is frequently seen in news services and opportunity databases, but seldom seen in Web search engines. The near-term opportunity for filtering in large-scale Web-based retrieval has to do with the use of profiles for

increasing retrieval effectiveness. Standing queries for new information seem less important for Web data, but may be effective for different data types. For example: "what's on television right now?" or "which of this month's magazines have articles that interest me?"

The role of a profile is comparable to that of both relevance feedback and segmenting a collection based on information need. For segmenting, a user profile might allow automatic identification of whether to select from, for example, documents about agriculture, foods or motorcycles given a query, "hogs." A profile, like relevance feedback or query expansion, could offer some extra terms or concepts for retrieval.

At an aggregate level, information profiles might be applied to stereotypical user groups. Collaborative filtering occurs when a group of people with similar information needs provide relevance judgments or other structured feedback to improve the utility of results.

IR performance might benefit from being able to provide documents related only to business, only to K-12, only to exercise enthusiasts, and so forth to other groups we target. Identification of sub-collections related to these profiles is straightforward, although choosing which profiles might be useful and how to characterize them could be difficult. This technique might be most useful to targeted audiences (such as investors or teachers).

Scalability of information profiles is a double-edged sword. On the one hand, the identification of sub-collections offers a strong method for avoiding search time linear with the (total) collection size. On the other hand, retrieval of a profile (e.g., by a cookie or prior search history), and integration of that profile with a query, adds a new level of complexity to searching. Furthermore, there is a possibility of profiles blocking potentially useful information (for example, what if the business person is seeking information for her 6 year old child?).

Segmenting the Collection

Dividing the possible universe of documents into multiple smaller domains is an important strategy for avoiding search times linear with the collection size. A challenge is to identify the basis for dividing the collection. With the Web, choosing of a series of topics for sub-collections seems risky. Identifying the search stage, document type, or some other criterion for a document is reasonable, but anticipating which ways of segmenting a collection for real-world searchers is more difficult.

One approach available on some Web search engines is to identify some simple characteristics. Examples include only searching .com or .edu, searching only documents updated with the last 30 days, and only searching in one language. All are found in contemporary search engines.

Another approach is to consider supra-document retrieval. For example, a business might have a collection of Web pages on one or more servers. An IR system might consider the whole business at once as a potential good match for a query. Identifying that some Web sites host multiple collections (e.g., geocities.com, ibm.com and usda.gov) is not difficult to do automatically. Link structure, shared cascading style sheets and shared terms are all indicators.

By segmenting at the collection level, the IR problem scales from that of hundreds of billions of Web pages to tens of millions of servers. The added work in identifying collections does not need to be significant.

Question answering, NLP, Sub-Document Retrieval

Historically, IR has focused on identifying document surrogates (e.g., bibliographic citations) that match a query. Today, the emphasis is on full-text retrieval of the documents themselves. However, another type of IR is gaining increased importance, particularly for purposes such as data mining: sub-document retrieval and question answering.

Sub-document retrieval involves identification of a passage (paragraph, sentence, heading, etc.) that matches a query. Question answering might go an additional step and generate a natural language response to a query, based on content from one or more documents.

Scaling of sub-documents is a major practical challenge (as demonstrated by the limited availability of “nearness” or “adjacency” operators in search engines). One document might be broken into dozens or hundreds of sub-documents, with overlapping boundaries. The benefit is precision: IR responses that identify a particular passage relevant to a query. A less obvious benefit is that queries tend to be short. Thus, matching with short sub-documents avoids some of the problems of document length normalization and term weighting that are encountered with complete documents.

At the question answering level, two aspects may require natural language processing or understanding (which is currently computationally expensive): query processing, and document processing. For a question, “where were clocks invented,” NLP could identify that the answer will be a location, region or proper noun. For documents, then, sub-documents with this type of content related to

clocks would be desirable. A further stage might take the sub-document and generate a response in correct format, rather than just presenting the sub-document.

NLP tends to be most effective in small domains, and is usually language (or culture) specific. For these reasons, it may be most effective with query processing, in order to identify additional search criteria for the retrieval process.

An example of question answering may be found in the AskJeeves search engine, but with a high reliance on human intervention and editorial policy to build the collection of documents containing answers. AskJeeves might find an article about clocks, but would not yield a response such as “The first public clock that struck the hours was made and erected in Milan in 1335.”

Collection Self-Awareness

The Z39.50 protocol, as implemented by WAIS and other tools, enables simultaneous searching of distributed collections. We have yet to see if this approach implemented at the Web server level. One possible implementation would be for the Apache Web server to incorporate indexing capabilities and a Z39.50 interface (free software, such as `ht://dig`, could be fairly easily adapted to this task).

The difference between such an implementation and the generation of sub-collections as discussed above is that the server index would be customized for the content of that server (possibly for multiple sub-collections or content types). Continued developments in metadata generation and XML will make this easier to automate. Because Z39.50 does not specify a particular retrieval method for a data server, some might choose a controlled vocabulary, others might use vector-space retrieval, others might limit searching to a small class of general public documents, and so forth.

This approach is akin to metasearch engines, but at a far higher resolution.

Scalability becomes a problem of communication and coordination more than collection size. If each server could respond to a query very quickly, network latency is still a potential problem. Even if network latency were low, the query’s origination point would need to receive and somehow rank results from dozens, hundreds or thousands of data servers. Such data fusion problems are challenging at many levels. The benefits stem mainly from collection indexes and localized search engines that are optimized for local collection content and policy, but

also from course-grained parallelism and a reduced need for all-encompassing central collections.

An important problem addressed by collection-level searching is dynamic content. Many Web sites, notably commerce-oriented sites, do not use static Web pages but instead generate content in response to input parameters.

A further problem addressed is that of streaming media or other constantly changing data. It is reasonable for a particular server (or organization) to maintain a current inventory of their content. This may be too much to expect of centralized search engines, which have demonstrated great difficulty in maintaining currency.

Document Clustering

Document clustering has a long and somewhat spotty history (see Larson, 1991; Willett, 1988). Early information scientists saw clustering as a way of solving the problem of searching exhaustively through a collection for the best matching documents (perhaps using magnetic tapes or punched cards). Clustering was also suggested as a method for automatically classifying documents for indexing or cataloging.

Clustering has traditionally focused on grouping documents by topic. There are many different approaches to clustering, but most do not scale well and are sensitive to such factors as initial conditions and document ordering.

If applied successfully, topic clustering could be a good solution to avoiding linear increases in search time. Multiway search trees may offer search time proportional to $\log n$ (rather than n) to identify a topic-based cluster. Some contemporary approaches to retrieval, such as Latent Semantic Indexing (Deerwester et al., 1990) and Information Space (Newby, 1997), are addressing this possibility.

Today, clustering is often approached differently: as the study of how Web site hyperlinks interact. This type of clustering may identify overlap in collections and collection authoritativeness. The benefit of these goals over topic-based clustering is that the ambiguity of language is bypassed in favor of less ambiguous link structures. Even so, large graph problems, nearest neighbor searches and the like are typically worse than linear as the node and link count goes up.

Link-based approaches have been applied by Alexa and Google (among others), but the science and scalability of these approaches is not yet well known.

Other Areas

Additional work in information science is relevant to the development of scalable real-world information retrieval systems but has not been discussed here. Some topics of potential interest include:

- Vector and probabilistic systems
- Variations on term weighting
- Known-item retrieval
- Relevance feedback (positive and negative)
- Cross-language retrieval
- Spoken document retrieval (sounds)
- Image retrieval (moving images; photos)
- Live retrieval (from audio/video broadcast streams)
- Knowbots and other active agents
- Version control and duplicate identification
- Document structure (tags; discourse level)

For many areas, there is a disjoint between academic research underway and the actual needs of large-scale systems. Coordinated efforts need to be made to determine what techniques are feasible, and how to best implement them.

Evaluation of information systems needs to incorporate a user-based approach: what are people trying to do? How do the results of a search assist this purpose?

Research Directions

This work has attempted to specify out useful avenues for development of real-world IR systems. Challenges include the relatively small-scale approaches of many research projects, and the limited topic- and document-based focus of most contemporary search engines.

Application of the scientific method to evaluation of search engines, data warehouses and archival systems has been lacking. A primary role of information scientists needs to be the development of large-scale experimental test beds where different retrieval mechanisms may be compared. Lycos, Yahoo and Google all emerged from academic environments where IR testing occurred, but opportunities for scientific experimentation are limited in production environments.

Information scientists know how to assess information needs, understand situational and individual differences in information seeking situations, and negotiate a search session. This type of knowledge has not yet been widely applied to the practice of information systems, partially due to the expense of collecting such data, and due to uncertainty about the payoff for real-world systems. This

uncertainty can only be overcome through empirical research and evaluation.

Scholars, practitioners and the public have some shared notions of what IR should be: somewhere between HAL 9000, Brent Spinner's "Data," and C3PO. The idea of very large collections of different data types from multiple data sources constitutes only one component of this vision, as does the need to maintain archival access. We also need to consider multiple information needs, user models and histories, and non-document level retrieval. Fostering academic research that integrates multiple approaches to IR and keeps an eye towards real-world scalability is a key to achieving our future goals.

References

Deerwester, Scott; Dumais, Susan T.; Furnas, George W.; Landauer, Thomas K.; Harshman, Richard. (1990). "Indexing by Latent Semantic Analysis." J. Amer. Soc. for Information Science 41(6): 391-407.

Dervin, Brenda; Nilan, Michael S. (1986). "Information Needs and Uses." In Williams, Martha E. (Ed.). Annual Review of Information Science and Technology. Medford, New Jersey: Learned Information.

Ingwersen, Peter. (1992). Information Retrieval Interaction. London: Taylor Graham.

Knuth, Donald. (1998). The Art of Computer Programming Volume 3: Sorting and Searching. Reading, Massachusetts: Addison-Wesley.

Kuhlthau, Carol C. (1993). The Information Search Process. NJ: Arbex.

Larson, Ray R. (1991). "Classification Clustering, Probabilistic Information Retrieval and the Online Catalog." The Library Quarterly 61(2): 133-173.

Newby, Gregory B. (1997). "Context-Based Statistical Sub-Spaces." In Voorhees, Ellen & Harman, Donna (Eds.). TREC-6 Conference Proceedings pp. 735-746. Gaithersburg, Maryland: National Institute of Science and Technology.

Newby, Gregory B. (1998). "An information access model with a unified approach to data type, retrieval mechanism and information need." Proceedings of the American Society for Information Science Annual Meeting, pp. 475-484. Medford, NJ: Information Today.

Schamber, Linda; Eisenberg, Michael; Nilan, Michael. (1991). "A re-examination of relevance: Toward a

dynamic, situational definition." Information Processing and Management 26(6): 755-776.

Voorhees, Ellen; Harman, Donna. (1999). "Overview of the 7th Text REtrieval Conference (TREC-8)." In Voorhees, Ellen & Harman, Donna (Eds.). TREC-8 Conference Proceedings pp. 1-25. Gaithersburg, Maryland: National Institute of Science and Technology.

Willett, Peter (1988). "Recent trends in hierarchic document clustering: A critical review." Information Processing & Management 24: 577-597.

About the Author

Gregory B. Newby is an assistant professor in the School of Information and Library Science at the University of North Carolina at Chapel Hill. His research focuses on information retrieval. He has been a participant in TREC since 1996. Dr. Newby is working on a large-scale information retrieval tested, ISpace. Information about ISpace is available at his TeraScale Retrieval Web site. In 1999, Newby participated in TREC's "very large corpus" track, which involved running 10000 (real) Web queries against a 100GB corpus of Web documents. ISpace ran the 10K queries in 52 seconds, which was two orders of magnitude faster than the next fastest participating system.

Author Contact information:

School of Information and Library Science
University of North Carolina at Chapel Hill
CB 3360 Manning Hall
Chapel Hill, NC < 27599-3360
gbnewby@ils.unc.edu

Copyright © 2000, Gregory B. Newby.