Moving More Quickly toward Full Term Relations in Information Space

Gregory B. Newby*
School of Information and Library Science
University of North Carolina at Chapel Hill

Abstract

This paper describes the ISpace retrieval system's involvement in TREC8. The main goal for this year's work was to speed up document indexing and query processing compared to previous years. This goal was achieved, but retrieval performance was not as good as for TREC7. System details for the AdHoc task, small Web task, and large Web (VLC) task are presented. The AdHoc task emphasized query expansion, while the large Web track emphasized rapid indexing and retrieval. The paper describes an implementation of a multidimensional tree structure for retrieval from information space based on the kd-tree. The larger setting for ISpace, the TeraScale Retrieval project, is summarized. A concluding section describes plans for ISpace.

Introduction

Efforts for the 8th Text REtrieval Conference (TREC) included the following:

1. AdHoc task, fully automatic.
2. Small Web track
3. Large Web track (VLC)

Throughout the work described here, the central question of interest is:

How might information space techniques achieve high performance?

The issue of performance is ambiguous, but was defined as emphasizing the following, in decreasing order of importance:

a. Performance means being able to quickly produce a ranked response set for a query topic
b. Performance means being able to handle the full variety of queries and documents – i.e., without limitations on the number of unique terms or number of documents
c. Performance means the response set has a large proportion of relevant documents

In this hierarchy, the goal of high relevance is uncharacteristically last, but not forgotten. Because post-hoc analysis of last year's non-judged TREC submissions (Newby, 1999) indicated reasonable recall-precision performance with exact precision of 0.14, the emphasis was on developing a more practical and usable system. While 0.14 is unremarkable compared to other groups' TREC submissions, it represented an order of magnitude improvement from prior years (Newby, 1998).

A description of the information space technique, system design considerations for each phase of the work and outcomes follow. A concluding section summarizes this year's TREC activities and lays out plans for the near future.

* School of Information and Library Science, Campus box 3360 Manning Hall, Chapel Hill, NC, 27599-3360.
Email: gbnewby@ils.unc.edu

The Larger Picture

ISpace, the Information Space retrieval system described here, is part of a larger research project. The project, led by the author, is called TeraScale Retrieval. The purpose of the TeraScale Retrieval project is to investigate problems of information retrieval through the development and evaluation of high-performance modular retrieval system components.

In Korfhage et al. (1999), an effort was made by leaders in retrieval research to identify major challenges to progress. At about the same time, the U.S. Presidential Information Technology Advisory Committee (PITAC, 1999) released a report intended to chart the near-term future of high-technology research and research funding. Both of these reports had similar thing to say about the needs of retrieval research. The TeraScale Retrieval project is attempting to meet some of these needs:

1. "To develop specifications for [and implement] a complete and modular set of IR tools to be made available to the IR community (Korfhage et al., p. 5)."
2. To create an infrastructure for "sharing and distribution of IR tools (Korfhage et al., p. 5).
3. To overcome limitations of many retrieval research systems, viz., "they lack modularity and do not facilitate interactive and operational retrieval experimentation (Korfhage et al., p. 5). This comment was directed particularly at the limitations of TREC.
4. To develop software "for managing large amounts of information (PITAC, p. 4)."

PITAC recognized that "transforming the way we deal with information ... requires significant improvements in data access methods, including high performance information systems and tools to help individuals locate information and present, integrate, and transform the information in meaningful ways (1999, p. 13)."

ISpace is part of a hybrid system. It uses data specifications and structures similar to that of IRIS (Yang & Maglaughlin, 2000). It is able to perform Boolean retrieval, and integrates multiple methods for basic tasks such as stemming and file access. It is being expanded to model vector space, probabilistic, and latent semantic indexing styles of retrieval. This will provide a controlled environment for retrieval experimentation that will enable IR researchers to control differences of implementation details among the various major types of retrieval systems.

One of the most important tasks of the TeraScale Retrieval project is to address scaling issues. Future information retrieval systems will be aimed at terabytes of raw data: millions of unique terms, billions of documents, and peta-scale quantities (quadrillions) of sub-documents. In spite of the continuation of Moore's law for doubling CPU power and processing speed every 18 months, the quantity of information we would wish to access is growing more quickly.

Many IR methods, including the most common implementations of vector spaces, probabilistic and Boolean models, use search and indexing algorithms that scale approximately linearly with the size of the collection. With modern hardware, these produce acceptable performance with collections the size of TREC AdHoc (2GB). But consider: if a search that takes 1 second on a 2GB collection takes 100 seconds on a 200GB collection, we can't wait for Moore's law to bring this longer search time back to 1 second (two orders of magnitude). Instead, we need to develop methods that scale better than linearly to speed up performance. The multidimensional tree described for the Large Web task below is one such method.

The TeraScale Retrieval project is also intended to address the shortcomings of large Web search engine's contributions to the scientific knowledge of information retrieval. Although these giants have made significant contributions to people's ability to find information on the Web, they generally have not shared their particular methods with the scientific community. This is consistent with PITAC's finding, "the PITAC members from industry were unanimous in their opinion that it is not feasible for the private sector to assume responsibility for long-term, high-risk research, in spite of the success of the information technology industry (p. 6)." The TeraScale Retrieval project will specifically address large-scale Web-based retrieval issues, and share findings, software and systems with the community of scientists interested in information retrieval.

A Brief Tour of Information Space

This section introduces the approach to information retrieval employed by the author. The Information Space approach to information retrieval is comparable to Latent Semantic Indexing (LSI, see Deerwester et al., 1990), with some differences. These differences are:

- Information Space starts with the term by term correlation matrix while LSI starts with the term by document co-occurrence matrix;
- Information Space performs eigensystems analysis while LSI performs a Singular Value Decomposition (SVD);
- Information Space has not made use of eigenvalues for scaling document vectors while LSI does make use of singular values (the square root of the eigenvalues);
- The Information Space approach does not assume that higher-dimensioned eigenvectors are without merit while LSI historically has sought to discard higher-dimensioned eigenvectors

Techniques that have been applied to various types of IR have also been applied to the Information Space system described here, called ISpace. These include: query expansion, part of speech tagging, document length normalization, term weighting, and stemming.

Like many other IR systems, ISpace may be compared to the Vector Space Model (VSM). Although variations in the VSM have proliferated, fundamental differences between Information Space and the VSM are:

- Information Space measures relations among terms while VSM treats term vectors are unrelated (orthogonal)
- A fair approximation of an Information Space may be visualized in 2 or 3 dimensions while a vector space does not have a clear visual interpretation

For TREC8, the ISpace programs from prior years were largely rewritten. The goal of rewriting was to increase modularity and enable incorporation of multiple IR techniques. For example, the same data structures, term weights, etc. used for an Information Space retrieval experiment with ISpace may also be used for a VSM or Boolean retrieval experiment.

The specific steps taken for ISpace retrieval are described for the different tasks, below.

The AdHoc Task

Experimentation with query expansion was the main goal for the AdHoc task. The steps taken for the task were as follows:

1. Read terms from all AdHoc documents, building an inverted index and term frequency list. About 367,000 Porter-stemmed terms were pre-identified from various word lists, past TREC topics, and the AdHoc document set. A test run demonstrated that all possible terms could be indexed, but would include many single-use terms not likely to be in queries.
2. 3136 non-stoplist terms were selected based on IDF values for inclusion in the information space. (The SMART stoplist was used.)
3. A term co-occurrence matrix for the 3136 terms was generated from the inverted index.
4. A Pearson product moment correlation matrix of term relations was generated from the co-occurrence matrix.
5. Eigensystems analysis was performed on the correlation matrix, resulting in 3062 eigenvalues ranging from 2871 to nearly zero.
6. Each AdHoc document was then re-analyzed and assigned a location vector at the geometric center of the eigenvectors of the terms it contained.
7. These document vectors were weighted with *tf * idf* using simple formulas from Frakes & Baeza-Yates (1992). Then, vectors were normalized to unit length.
8. Topics were expanded by either 25 or 50 terms. Term expansion added the most highly correlated terms with each topic term.
9. The topic vector in the information space was then weighted and normalized at the center of its (expanded) term vectors.
10. The closest document vectors to the topic vector were retrieved in rank order

Four AdHoc runs were submitted. Two judged runs utilized topic titles and descriptions, expanded by 25 or 50 terms (isa25, isa50). Two un-judged runs utilized titles only and expanded by 25 or 50 terms (isa25t, isa50t). Results were considerably poorer than comparable AdHoc ISpace runs from TREC7, yielding average exact precision scores under 0.025.

There are two likely explanations for this poor showing. One is that part of speech tagging (Brill, 1994) and query processing, used last year, are important. This is supported by the observation that expanded topics expanded all topic terms, including terms without much discriminatory value. This may have served to bring in useful terms, but it also increased the noise in topics. For query processing, all topics had been pre-analyzed for sentence structure in TREC7 so that phrases about non-relevance were eliminated. Thus, only "desirable" terms were kept.

The other likely explanation is that the 3136 terms chosen were not well suited to the TREC8 AdHoc topics. Of the 283 unique stemmed terms in topics 401-450, 14 topic terms that casual interpretation suggests were important were not among them.

The stemmed missing terms were: Burma, comet, decai, estonia, hurrican, legionnair, lockerbi, milosev, parkinson, potassium, saharan, salvag, scotland, and typhoon. From this list, it is evident that low performance would be expected from the topics that include them: 403, 405, 406, 408, 409, 411, 415, 423, 429, 434 and 443.

However, topics 403 ("osteoporosis") and 408 ("tropical storms") both performed relatively well, as mentioned below. It appears that the missing term from 403, "potassium," was not

overly injurious, and the presence of both "tropical" and "storm" may have offset the missing "hurricane" and "typhoon" from topic 408.

Exact precision scores for the AdHoc task ranged from 0 to .59. Table 1 presents a summary of scores. Due to a scoring error, scores for the title-only 50-word expansion are not considered here.

Topics with exact precision or average precision greater than 0.10 on the AdHoc task included:

1. Topic 403, "osteoporosis." This topic had the highest median average precision of all topics, possibly due to a large number of fairly specific terms in the expanded topic. Even the title-only run yielded good scores (Exact precision = .42). Term expansion appeared to work well, with a better exact precision for expansion by 50 than by 25 (.38 vs. .19).
2. Topic 407, "poaching, wildlife preserves." Good performance on the title-only run, but not title plus description. .13 average precision and .17 exact precision for title-only.
3. Topic 408, "tropical storms." Average precision scores of about .9 on all runs were unexceptional, but exact precision scores of .22 and above for all but the expand by 25 condition indicate reasonable early precision.
4. Topic 441, "Lyme disease." This seems to be a good example of a topic well-represented by two terms. Title-only runs yielded exact precision scores in excess of .52, and average precision of over .58, while title plus description runs were very poor, below .01 on both exact and average precision.
5. Topic 444, "supercritical fluids." This is the one topic where title plus description heavily out-scored title-only, but only for the expansion by 50 case. 3 of the 17 (total) relevant documents across participants' systems were retrieved in the top 10 documents, indicating some good terms were brought in by expansion that were missed in other runs.

Table 1: Summary of AdHoc Task Retrieval Performance

	Expand by 25 (isa25)	Expand by 50 (isa50)	Expand by 25 title only (isa25t)
Exact precision	0.0081	0.0349	0.0515
Average precision	0.0026	0.0203	0.0273
Number over median average precision	0	0	0

The easiest interpretation to make from these results is that query term expansion is valuable, but only when applied to useful terms. Future efforts will return to a reliance on part of speech tagging and term weights to identify "good" terms to expand (i.e., terms with good discrimination potential), while avoiding expanding the rest.

Finally, note that no runs were made with non-expanded queries. This would be a valuable comparison. Future experiments should therefore compare:

- Expansion versus no expansion
- Different levels of expansion (bringing in 25 terms versus 50 terms, or other values)
- Methods for choosing terms to expand (*tf* weights, part of speech, presence in `<title>` field, etc.)

The Small Web Track

The steps taken for the small Web track (a 2 gigabyte subset of the large Web track's Very Large Corpus or VLC) were nearly identical to the steps for AdHoc. The only difference was that the co-occurrence matrix was not based on the small Web track corpus, but rather the AdHoc corpus. This mismatch should theoretically be detrimental to the recall-precision statistics, but with an average exact precision of 0.02 from the AdHoc task, such differences would be difficult to see. (Performance at exact precision values of 0.02 or below is approximately equal to that expected from a random sample of documents, although higher early precision than later precision indicates the retrieved set is actually non-random.).

Four small Web track runs were submitted. As with the AdHoc task, two title plus description runs were judged, but two title only runs were not judged. Due to a scoring error, scores for the title-only 50-word expansion are not considered here. See Table 2 for a summary.

Table 2: Summary of Small Web Track Retrieval Performance

	Expand by 25 (isw25)	Expand by 50 (isw50)	Expand by 25 title only (isw25t)
Exact precision	0.0048	0.0451	0.0524
Average precision	0.0018	0.0291	0.0291
Number over median average precision	0	2	2

Topics yielding exact precision or average precision over 0.10 on one or more runs included:

1. Topic 403, "osteoporosis." Presumably for the same reasons the topic performed well for the AdHoc task (above).
2. Topic 408, "tropical storms." Interestingly, performance was somewhat less than for the AdHoc task, but still better than most other topics.
3. Topic 415, "drugs, Golden Triangle." This topic performed well with title-only, but not well with title plus description, presumably due to the ambiguous terms in the description (e.g., "organizations," "international.").
4. Topic 432, "profiling, motorists, police." This topic also performed well only on title-only runs.
5. Topic 433, "Greek, philosophy, stoicism." One or more good expansion terms were presumably identified in title plus description expansion by 50, which yielded .18 exact precision but less than .01 for the other runs.
6. Topic 448, "ship losses." Similarly to topic 433, exact precision on title plus description expansion by 50 yielded an exact precision of .18, but very low scores on other runs.
7. Topics 441, 445, 446 and 450 all had comparable patterns of exact precision scores near or over .20, but unexciting average precision scores (except for topic 450, with an average precision of .10 or above for all runs). These topics are characterized by useful title terms and, except for 441, specific description terms.

The small Web track data set seemed to be fairly well suited for topics 401-450. Average precision scores across all TREC participants were sometimes higher for the Web track, and

other times higher for the AdHoc task. The mean of median average precision scores for the AdHoc task and small Web track across all TREC8 participants was identical to 2 decimal places at .24.

It seems reasonable to conclude that the somewhat better performance for the small Web track versus the AdHoc task for the ISpace system is genuine. Some possible explanations are that an increased variety in Web documents resulted in fewer highly-ranked false hits, and that the 3136 terms pre-identified for information space document location calculation helped to eliminate a larger proportion of small Web track documents than AdHoc task documents.

The Large Web Track (VLC)

The large Web track was the main emphasis of this year's work, with the primary goal of achieving reasonable speed and efficiency in processing documents and topics. As discussed below, this goal was achieved, but at the expense of retrieval performance. This section describes the system considerations for the large Web track, including a discussion of the tree data structure employed for retrieval. However, because results for the VLC were submitted after the deadline, they were not officially judged. *Because almost none of the VLC documents retrieved by ISpace were judged, no useful retrieval performance statistics were generated.*

As with the small Web track, the large Web track made use of the information space pre-computed from the AdHoc task. Thus, the steps for retrieval were nearly identical, except that a multidimensional tree was used for retrieval, rather than a brute-force search for the closest document vectors for each query vector.

Although 3062 eigenvectors (dimensions) were available for use from the AdHoc collection, the VLC documents were only processed with 300 eigenvectors per term. The 300-dimensional information saved to disk was not utilized for the actual retrieval, however: only 100 dimensions were utilized. This is beneath the threshold recommended by Deerwester et al. (1990) and others, but was necessary to fit as many documents as possible into the tree, which was memory-based.

Table 3: Summary of Large Web Track Non-Retrieval Performance

Measure	*Performance*	*Notes*
Time to index 100GB	5 clock hours	0.0012 seconds per document
Index size, all files	11GB, 18 files	~640 bytes per document
Index size, eigenvectors only	8GB, 3 files	300 dimensions per document
Time to start retrieval engine (prior to running queries)	20 clock minutes	
Time to run 10,000 queries	52 seconds	0.0052 seconds per query

The 100GB VLC (Very Large Corpus) was stored on a disk/tape array at UNC Chapel Hill. This enabled rapid staging from tape robot to disk without requiring more than about 10GB of disk space at a time to stage documents. See Table 3 for a summary of non-retrieval performance measures

Most of the large Web track was performed on UNC-CH's Sun ES-10000 server with 36 processors and 16GB of main memory. Disk access on the disk/tape array was found to be quite fast, cutting indexing time by 60% versus a comparable server with local SCSI RAID disk. No

parallelization was utilized (all programs ran on a single CPU), and the ES-10000 was shared with many other user processes.

The overhead to start the retrieval engine consisted mainly of reading in all document locations (eigenvectors) from disk to a modified kd tree in memory. The kd tree is a data structure for matching data on a large number of keys – in this case, the keys corresponded to the relatively large number of dimensions. Kd trees are a type of multidimensional tree developed by Friedman, Bentley & Finkel (1977), intended to solve the challenge of quick searching of multi-keyed data. This is a similar problem to that of multi-way trees such as B-trees (cf. Knuth, 1998), but whereas B-trees are split on a single key (such as a filename), kd trees are split on multiple keys (such as dimensions in a multidimensional space). The particular memory-based implementation of the kd tree utilized for ISpace was derived from Weiss (1999).

Although up to 300 dimensions were available on disk, as mentioned above, only 100 were used for the tree. Even so, only about 7 million of the 14 million VLC documents were considered for retrieval at one time on the ES-10000, using about 4.5GB of memory. (There was no opportunity for dedicated access to the ES-10000.) Note that only 14 million of the 18 million VLC documents were assigned locations in the information space, because the other 4 million had none of the 3136 terms (many of these were in non-English languages).

Efforts are underway to store the kd tree on disk, rather than storing the entire tree in memory. This is a required development to grow ISpace beyond 7 million documents in 100 dimensions.

The general purpose of the kd tree is to minimize the number of document vectors that need to be compared by brute force (one at a time) to the query. Consider that a key difference between ISpace (and other LSI-like approaches) and the VSM or Boolean systems is that a document can be "close" to a query without having any (or many) terms from the query.

For example, a document, "pig farmer" might be close in ISpace to a query, "swine domesticator." In order to determine and rank the closest documents to a query, it is necessary to consider ALL documents, not just those with query terms. This is an important drawback of non-orthogonal term vectors. For up to a few thousand documents with coordinates in main memory, an exhaustive comparison and ranking is reasonable. For millions of documents, as in the large Web track, it is desirable to group documents in some sort of structure so that a relatively small proportion of documents need to be exhaustively compared.

There are only two differences between a typical kd tree and the modified kd tree used here (tentatively called a green-black or gb tree for its similarity to a red-black tree). One difference is that there are three children per parent, instead of two. The other is that the decision of which branch or leaf to select at each level is based on the coordinates at the corresponding dimension, instead of on the dimension with the highest remaining variance. These changes are anticipated to be helpful with a highly multidimensional space (hundreds or thousands of dimensions), as compared to the more typical case from the literature on algorithms where only a dozen or so dimensions are used.

- Insertion time for a kd tree is proportional to n log (n) where n is the number of items to be inserted (i.e., the number of items in the completed tree).
- Cell identification time, to find single cell with the best match, takes time proportional to log n.

- Target identification time, to evaluate all items in a cell (leaf) to find the document vector closest to the query vector, takes linear time with respect to the number of items in the cell.

For ISpace, the goal was to minimize the number of linear time comparisons by minimizing the number of cells that had to be examined. The challenge, as might be expected, is that the tree depth does not nearly reach the total number of dimensions. For example, a tree with 7 million items might reach a depth of 60 levels (that is, the most dimensions examined to separate documents into separate leafs – known as buckets – is 60). This is for a relatively large bucket size of 1000 (a limit of 1000 documents per leaf, before the leaf is split and the tree descends a level).

Smaller bucket sizes would result in deeper levels, but bucket creation and splitting the parent bucket's contents is relatively expensive.

The modified kd tree used for ISpace in TREC8 yielded strong system-based performance, with time per query well under 1 second. Further work with this type of tree structure will determine the extent to which the performance win on search time can also yield good retrieval performance.

Conclusion

This year's implementation of ISpace incorporated query term expansion. A modest-sized information space of only 3136 terms was built, consisting of term eigenvectors derived from a term correlation matrix.

ISpace retrieval performance was not as good for TREC8 as for TREC7. However, the indexing and speed performance was increased by at least an order of magnitude. Implementation of the following should result in regained retrieval performance. All of these features were present in earlier versions of ISpace, but not implemented for TREC8:

1. Allow more terms in the information space. At least 40,000 term eigensystems should be achievable.
2. Incorporate part of speech tagging to identify terms that should be expanded.
3. Implement sentence parsing for TREC-like queries so that terms in phrases identifying unwanted concepts may be bypassed.
4. Multiple options for term weighting schemes
5. Term co-occurrence relations measured at the sub-document level (e.g., within an N-term window or within the same sentence.
6. Add awareness and differential term weighting for SGML/HTML tags such as headings.

Finally, it should be noted that this year's ISpace is an IR system without an interface. Adding a Web-based front end, enabling relevance feedback and an overall higher level of interactivity will present no problems. Incorporating the navigable fly-through system, Yavi (used for TREC7) is also desirable in the near-term.

Post-TREC development has included implementation of the following:

1. Enabling every term to be indexed. Consistently with Witten et al. (1999), simple techniques were developed to insure that the large number of terms with low occurrences

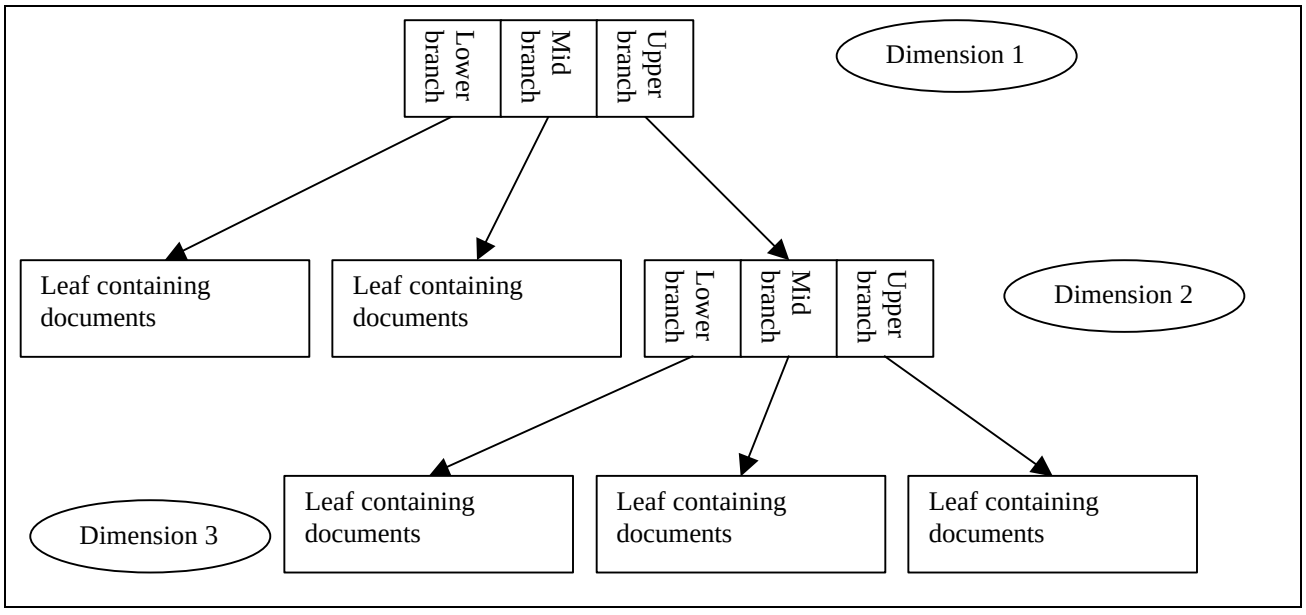
did not take a disproportionate amount of disk space or memory. An index with 982K terms was derived from the AdHoc collection.
2. More term eigenvectors. An information space with 17,583 unique stemmed terms was computed.
3. Further portability and fewer limitations. ISpace runs under Solaris, Linux and Irix. It uses files of any size permitted by the underlying operating system.

The goals of the TeraScale Retrieval project, as mentioned above, are supported by these ISpace developments. For TREC9, these goals have been identified:

1. To seek higher retrieval performance while maintaining high speed and efficiency.
2. To (re-) incorporate retrieval techniques from prior years.
3. To provide further abstraction in the ISpace system, so that fusion of results from Information Space, VSM, probabilistic and Boolean methods may be achieved.
4. To clarify the relationships among LSI, Information Space, VSM and other methods, to identify their common mathematical themes and seek out opportunities for mutual benefit.

Information retrieval is not a solved problem. ISpace, as described here, is a system that attempts to expand the choice of retrieval techniques available to information scientists, while drawing heavily on prior achievements.

Figure 1: Diagram of gb tree structure. The tree consists of leafs or buckets, which contain documents, and branches, which are used to determine where a document should be placed.



References

Brill, Erik. (1994). "Some advances in rule-based part of speech tagging." Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94). Seattle, Washington.

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.

Frakes, William B. & Baeza-Yates, Ricardo (Eds.). (1992). Information Retrieval Data STructures &Algorithms. Englewood Cliffs, New Jersey: Prentice-Hall.

Friedman, Jerome H.; Bentley, Jon Louis; Finkel, Raphael Ari. (1977). "An Algorithm for Finding Best Matches in Logarithmic Expected Time." ACM Trans. On Math. Software 3(3): 209-226.

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

Korfhage, Robert R.; Rasmussen, Edit M.; Belkin, Nicholas; Harman, Donna. (1999). "Invitational Workshop on Information Retrieval Tools." Pittsburgh: School of Information Science. (Available online: http://www.sis.pitt.edu/%7Eerasmus/workshop.html).

Witten, Ian H.; Moffat, Alistair; Bell, Timothy C. (1999). Managing Gigabytes. San Francisco: Morgan Kaufmann.

Newby, Gregory B. (1998). "Context-Based Statistical Sub-Spaces." Text REtrieval Conference (TREC-6) Proceedings, pp. 735-746. Gaithersburg, MD: National Institute of Science and Technology.

Newby, Gregory B. (1999). "Information Space Gets Normal." Text REtrieval Conference (TREC-6) Proceedings, pp. 567-571. Gaithersburg, MD: National Institute of Science and Technology.

PITAC (President's Information Technology Advisory Council). (1999). "Information Technology Research: Investing in Our Future." Washington, DC: National Coordinating Office for Computing, Information, and Communications. (Available online: http://www.ccic.gov/ac/report/).

Weiss, Mark Allen. (1999). Data Structures and Algorithm Analysis in C++ (2nd ed.). Reading, Massachusetts: Addison-Wesley.

Yang, Kiduk & Maglaughlin, Kelly. (2000). "IRIS at TREC8." In this volume.