

# Information Retrieval on Computational Grids

## **Abstract**

This paper describes work in progress to facilitate information retrieval (IR), particularly text retrieval, for a variety of data types. The work described here seeks to make it easier to query from multiple datasets, to merge results and to visualize information spaces. Such datasets could come from different organizations, and include diverse data types. Capable and flexible software for IR, operating within an emerging standards framework for security and interoperability, offers promise of improved searching capabilities for information seekers.

## **Introduction**

Information retrieval systems are used daily by the general public, often in the form of Web-based search engines. Specialists, including intelligence analysts and others in the intelligence community, use a variety of information retrieval systems, each of which might have different qualities and give access to different sets of data.

A long-sought goal for information retrieval (IR) is to easily query multiple datasets simultaneously, delivering merged results in relevance-ranked order. While this goal is met for some information needs such as patent searching, and some legal datasets (such as those provided by Westlaw and Lexis/Nexis), systems that support query and merging from multiple federated datasets are unavailable for many purposes. For those systems that do federate datasets, other limitations are often present, such as inability to effectively merge results from different sources, or to manage constantly updated streams of data.

This paper presents work towards the goal of querying federated datasets in secure environments using software intended for compliance with emerging standards. The next section describes some of the background and impetus for the work, and gives an overview of contemporary information retrieval systems. The following section describes GIR, a standard under development for information retrieval on computational grids. Next is a description of IRTools, a software toolkit for GIR and information retrieval research. Some performance results using the toolkit are presented for the NIST TREC conference and for visualization.

## **Background**

One of the innovations of the early Internet was Wide Area Information Servers or WAIS (Stanfill & Kahle 1986). WAIS utilized the Z39.50 standard (see <http://www.loc.gov/z3950/agency>) to transport queries and return results from multiple network-based datasets. It was quite popular, and many information providers installed WAIS servers to make their datasets searchable by others. WAIS clients had important deficiencies, however: they were not able to perform effective ranking of results from multiple datasets, and were under-capable for other purposes such as information filtering, in which a standing query is used to evaluate new documents. WAIS servers were not required to provide sufficient data about a response set, such as collection statistics, for clients to effectively rank results.

WAIS eventually disappeared from widespread use, though its technologies were incorporated into other software projects such as FreeWAIS and Isite. Instead, Digital Equipment Corporation's AltaVista and, eventually, many other centralized search engines replaced the WAIS approach.

Centralized search engines utilizing a single monolithic collection, with Google as the best exemplar in 2004, are extremely capable for some types of searches. Because of the vast number of documents accessible on the open Internet, it is possible to identify relevant documents for many short queries. However, monolithic search engines are limited in several ways:

- Meta-search across search engines is no more sophisticated than the WAIS method;
- Search engines harvest only publicly-accessible documents of particular types, thus missing vast quantities of "hidden" online information (Raghavan & Garcia-Molina 2001);
- There is little or no access to the internal processes of search engines, to perform IR research or assess weaknesses;
- Search engines lack facilities for information filtering, cross-language information retrieval, structured document retrieval (e.g., XML queries), and other features that are available in other systems;
- The indexing, searching and retrieval methods for each search engine are unique, and cannot be easily manipulated by dataset providers to enhance their data's accessibility;
- There is no security model, limiting search engine utility to open source data with queries and responses transmitted over insecure channels.

Information scientists typically measure IR system effectiveness using relevance, recall and precision. *Relevance*, or pertinence, is a measure of how closely a particular query response, or the document it links to, relates to the searcher's information need (see Schamber et al., 1990). *Recall* is the proportion of all relevant documents in a dataset that are retrieved in response to a query. *Precision* is the proportion of retrieved documents that are relevant. The research literature on IR indicates an inverse relationship between recall and precision: while small and selective search responses might have high precision, they will tend to low recall. Conversely, very large search response sets will have higher recall, but at the expense of many non-relevant documents and therefore reduced precision.

Web search engines generally emphasize high early precision. Because the dataset of the Web is so large, an effective strategy employed by Google and other search engines is to look for documents with the highest probability of relevance (judged separately from the query), and present them first. Specifically, this means ranking documents highly when they come from highly regarded sites or are linked by other highly regarded sites (Page & Brin, 1998). While this is an effective strategy, it makes the search engines less effective for high-recall information needs, in addition to the shortcomings mentioned above. High recall might be critical for analysts seeking to get complete data about a topic, rather than just the most highly ranked hits.

There is a rich body of research in IR that informs alternative approaches to monolithic search engines. While information scientists seldom have the resources to work with datasets on the scale of the entire Web, a number of research systems operate with reasonable subsets. The annual Text REtrieval Conference (TREC, see <http://trec.nist.gov>), sponsored by NIST and DARPA, is a showcase for state of the art systems offering differing capabilities and innovations. Other DARPA events, such as the Document Understanding Conference (DUC) and Translingual Information Detection Extraction Summarization (TIDES) workshop, offer additional venues for presentation of innovation by researchers, industry and government.

Research systems at these forums are often lacking in practical utility due to insufficient scalability, poor resiliency to error, or weak computational performance. One of the crucial ingredients lacking is a common base development system to “plug in” new IR system components in a modular fashion. The work of the authors and colleagues is specifically geared towards creating such a software base through the development of general-purpose IR test bed software, and by working on the development of standards and systems for IR on computational grids.

### ***Grid Information Retrieval***

Grid Information Retrieval or GIR is information retrieval on computational grids. Specifically, it is a standard being developed by the author and his colleagues under the auspices of the Global Grid Forum (GGF). Grid computing is at the heart of future high-performance computing (Foster, 2004), as demonstrated by its prevalence in current and National Science Foundation (NSF) initiatives and funding for the TeraGrid (<http://www.teragrid.org>) and National Middleware Initiative (<http://www.paci.org>).

Fundamentally, grid computing supports the sharing of computational resources among networked computers. Sets of systems work together as part of virtual organizations (VOs). These VOs may be ad hoc, public, or private. The grid infrastructure provides end-to-end data security by exchanging authentication certificates and tunneling data over encrypted channels. Grid computing is similar to parallel or cluster computing, in that different computational units may assume particular levels of resources will be available from other members of a VO. It is also similar to distributed computing, except that the grid offers a security model and a far higher level of computational integrity.

Systems connected to a VO offer particular services ranging from simple file transfer to distributed parallel processing for complex applications. The grid infrastructure offers methods for the systems to communicate using protocols that are generic or application specific. GIR will operate as a set of services that provide a protocol for communication among systems in a VO to perform IR tasks. As a formal standard within the GGF, it is the intent that a variety of legacy systems, special-purpose systems, and research systems will be able to work together to form virtual collections, easily searchable by members of the VO.

## **GIR Functional Overview**

GIR is intended to offer a formal set of requirements in specifications documents published by the GGF. The GGF standards-formation process is similar to that of other open standards organizations such as the W3C and IETF (Catlett, 2001). The GIR-WG (Grid Information Retrieval Working Group) was formed in 2002 to develop the standard and to deliver reference implementations for GIR. As of late 2004, the Requirements document has been published by the GGF (Gamiel et al. 2004) and the Architecture document draft is available for review via the GGF's Web site (<http://gir-wg.org>).

The architecture for GIR is designed around three components. Each component can interact with others using simple protocols and event notifications to accomplish IR tasks. The modules are described in the following subsections.

### **Indexers**

Indexers offer core retrieval functionality. They receive data from Collection Managers and provide responses to Query Processors, as described below. Generally, IR systems have two major activities. One is to build an index, which may be thought of as a machine-searchable representation of the documents in a collection. The other is to match documents against a query to produce a response set. Additional functions, such as index optimization or query expansion, might also exist.

In GIR, Indexers are able to exist autonomously from the data they collect and queries they receive. While Indexers are usually thought of as holding a single collection, they may themselves federate multiple collections. Indexers may be specialized for particular data types, ranging from plain text, to formatted text (SGML, XML, HTML...) to multimedia. The only fixed requirement for an Indexer is that it be able to respond to a query with a response set, within the security guidelines described below.

### **Collection Managers**

Collection Managers are GIR components that provide documents to Indexers. In the case of Web harvesting or other sorts of dynamic data, it is unreasonable to expect that a sophisticated Indexer is also, for example, a sophisticated harvester. The Collection Manager handles gathering data for the Indexer and, to the extent they agree on delivery methods, can pre-process the data for the Indexer to use.

Collection Managers can also act as proxy servers or transformation engines. In GIR, we envision scenarios where data may not be freely distributed. In such a case, a Collection Manager can perform data retrieval or transformation as required for a particular Indexer (or Query Processor). We believe this will be an especially important characteristic for enabling fuller access to the hidden web, as well as for providing for limited data sharing among organizations.

### **Query Processors**

Query Processors are components that gather response sets from Indexers and present them to end-users. In the simple case of a query being delivered to a single Indexer, Query Processors might not be necessary. But with GIR, querying many federated datasets over time is envisioned, and sophisticated merger and presentation of results will occur. Thus, Query Processors are envisioned as the solution to some of WAIS'

shortcomings, as well as a repository for standing queries (information filtering) and a tool for data presentation.

Query Processors, like other GIR components, are designed to operate in the grid's event-driven control scenarios. We envision Query Processors that are able to receive events that signal updates in Indexers they are monitoring, and then to assess whether new relevant documents have become available.

Using these three GIR components, task-specific IR systems can be built for *ad hoc* or ongoing use. For example, an indexer might take input from a query processor that filters desired information. Another processor would seek new relevant documents from that indexer, and several others. Collection managers can operate in both a traditional "pull" model (i.e., by querying a remote dataset for updates) or event model (where subscribed-to collections send notifications of updates). If desired, a configured IR system may be shared with other members of the VO, or kept to oneself.

### **Security Infrastructure**

While GGF protocols offer system- and service-level authentication, IR on the grid requires additional security that is not otherwise available. Through cooperation with other working groups in the GGF, it is likely that some or all of the security infrastructure for GIR will become part of other grid-based applications. The Security Infrastructure for GIR operates at the query, index, collection and user level, providing for authentication, authorization and logging at every step of the process.

Authentication on the grid is based on open standards as described in the work of the Open Grid Security Architecture group (OGSA, see Foster et al. 2004). The model for authentication rests on digital certificates. These certificates determine whether particular systems or services on them may interact with other services. This is the coarsest level of security within the grid model.

In order to provide user-level authentication at different levels within GIR, access control lists at the system, user, collection, query and record level must be consulted as events occur. The details of what levels of authentication are required are up to the members of the VO who create instances of GIR services. Essentially, every time an index is changed or queried, or a query is sent, or a collection manager receives a request for some data, they must receive an authorization token before proceeding.

Because GIR relies primarily on events, rather than client-server interaction, it is possible for an authorization token to be delivered asynchronously. This is intended to allow for human decision-making or other out-of-band communication as part of the authorization process. For example, if a user from one organization wishes to send a particular query to another organization, personnel in the other organization might be required to process the request before allowing an authorization token.

## ***GIR Use Scenarios***

The GIR-WG members envision uses of Grid information retrieval in which information seekers arrange the different elements of GIR into a system responsive to different types of information needs. One typical need might be an alternative to monolithic search engines for when such engines do not offer sufficiently sophisticated capabilities, or when security constraints limit their use. Within organizations, there may be diverse data sources (such as different units or departments, each of which produces sets of data with different types on different timescales).

For example, a business might have a technical product support unit that has textual documents in XML markup or other formats, and a manufacturing unit that tracks part numbers, inventories and their relations to manufacturing plant facilities. In this type of organization, an information seeker might want to find out whether some technical documentation was outdated based on changing manufacturing processes or materials. GIR offers two important capabilities over alternatives: federation, and security. Because the datasets (technical documents and manufacturing information) are not available to the outside world, they cannot be harvested by public search engines. By creating two separate IR indexes, perhaps with different IR systems or settings, search is enabled. GIR offers the ability to search across both datasets and merge results, based on the capabilities of each IR system. Because GIR operates within the secure Grid computing environment, desired access controls will be built in, both for outsiders and within or across organizational units.

A different use case scenario is information filtering. GIR will offer the ability to set up long-running queries against dynamic datasets. This capability is known as information filtering. Information filters can be complex representations of information need, and can be modified, tuned or trained over time. For example, environmental analysts might monitor a wide variety of data sources. GIR will include elements to continuously monitor these sources such that when change events occur, indexing events may be triggered. If newly added items exceed a threshold in the filter, they are presented to the information seeker. Because the filter would be an active long-running process, it could combine or re-present data items. For example, if the environmental analyst were interested in threats to endangered and threatened species, a news item from one source about a wildfire in California might be combined with an item from another source about legislation protecting bald eagles that live in that part of the United States.

## ***IRTools, the Information Retrieval Toolkit***

With support from the NSF (grant # 0352029), the author and his colleagues are developing a high-performance software toolkit for information retrieval research, IRTools. IRTools is intended as a test bed for a variety of IR applications, and is currently being developed for deployment at a GIR reference implementation. This toolkit, which is open source software under the GNU General Public License (GPL), is intended for evaluation of retrieval methods in practical environments. It was developed

in direct response to the relative lack of configurable software with well-known methods for retrieval evaluation.

There are a few other software toolkits with similar intent, but IR research software, for the most part, cannot handle numerous data types for many millions of documents. Such a toolkit is needed in order to develop and evaluate new approaches to IR. The alternate is to use special-purpose systems – for example, one for HTML and another for XML. This approach makes experimental evaluation difficult because the differences between such systems are unknown or difficult to quantify. With IRTools, the full implementation details are available via the source code, and experimenters can select which factors to change during evaluation.

IRTools, like most IR software, performs two main actions: to *build* a representation of a dataset, by reading in documents and generating various indexes to them; and to *query* the dataset by taking words as input and generating a ranked list of documents as output. About a dozen program classes in C++ are used to deliver this core functionality in about 12,000 lines of code. Some support software is written in Perl. The Gnu compiler collection is used to build the software, which has been tested in Solaris, Linux and MacOS X environments. The software relies on the BerkeleyDB and MySQL packages for back-end database files, on libwww (from the World Wide Web consortium) for document parsing, and on the C++ Standard Template Library or STL for internal data representation. The software is available via <http://sourceforge.net/projects/irtools>.

IRTools is designed to be easily extensible, and is a work in progress. The author has used it for several years of TREC experiments (see Newby, 2003), for both HTML and plain text documents. XML support is also included. Depending on options chosen, the software can index up to 1GB of data per hour, generating index files, which are approximately 100% of the size of the input data (again, depending on options chosen).

Query time depends on whether a Boolean AND is chosen or Boolean OR, and what particular results ranking scheme is requested. Two main ranking schemes are available, the Vector Space Model (Salton & McGill, 1986) and Latent Semantic Indexing or LSI (Deerwester et al., 1990).

### **Sample Results from TREC 2003**

This section briefly describes some experimental outcomes and results from the 2003 TREC. See Newby (2003) for more details. One of the experiments that IRTools was utilized for is the “Web named page” track. In this experiment, systems were presented with a list of short queries (usually 2 or 3 words) and expected to find the Web page corresponding to the organization, group, business, or similar entity. As such, it was a high-precision task where the goal was to identify one or more appropriate pages among the first IR search results. The document collection is TREC’s 20GB .GOV collection, with about 1.5 million Web pages harvested from US government Web sites (.gov sites). Four runs, plus a combined run, were submitted under in two different scenarios.

- Run 1 searched only the HTML *title* tag for query terms. Weighting for this and all other runs favored the exact query phrase, but was essentially based on the Vector Space Model (VSM).

- Run 2 looked for the exact query phrase within the same *paragraph* (where paragraph is defined as any block-level set of text, including HTML tags such as p, table, and ul).
- Run 3 ranked documents containing all query terms *within 10* terms from the first query term.
- Run 4 was a “bag of words” approach, in which any document with all query terms was considered for ranking.
- The fifth set was the combination of all prior sets.

It was speculated that the four runs could operate in a “fall through” manner: if run 1 yielded no results for a particular topic, run 2 would ensue. Similarly, run 3 would only occur for a topic if run 2 yielded no results. The bag of words approach in run 4 was, essentially, a move of desperation. Thus, the “Combined” column of Table 1 is the result of all 150 topics in which one or more of the runs were completed, only the last of which produced results. This is the fall through scenario.

**Table 1: TREC 2003 Named Page Task Results Summary for Fall Through Queries**

Recall	Precision				
	Run 1	Run 2	Run 3	Run 4	Combined
		(If Run 1 fails)	(If Run 2 fails)	(If Run 3 fails)	(Complete set)
0	0.5392	0.2461	0.1447	0.4048	0.2839
0.1	0.5392	0.2461	0.1447	0.4048	0.2839
0.2	0.5392	0.2461	0.1447	0.4048	0.2839
0.3	0.5392	0.2461	0.1447	0.4048	0.2839
0.4	0.5392	0.2461	0.1447	0.4048	0.2839
0.5	0.5392	0.2461	0.1447	0.4048	0.2839
0.6	0.451	0.2333	0.1447	0.3095	0.2483
0.7	0.451	0.2333	0.1447	0.3095	0.2483
0.8	0.451	0.2333	0.1447	0.3095	0.2483
0.9	0.451	0.2333	0.1447	0.3095	0.2483
1	0.451	0.2333	0.1447	0.3095	0.2483
# of Queries	34	13	77	21	145
Retrieved	142	104	541	75	862
Relevant	39	17	82	27	165
Relevant retrieved	20	6	22	11	59
Exact precision	0.49	0.23	0.1	0.33	0.24

Table 1 shows that relatively high precision was achieved in Run 1 (title only), but at the expense of many queries for which no results were submitted. 34 topics resulted in 142 documents identified as potentially relevant, 20 of which actually were. Most topics only produced a few documents, with only 64 (“work/life center map”), 88 (“export import bank”) and 137 (“endangered species picture book”) in the double digits with 31, 31, and 14 documents each, but none was relevant. Those topics are in contrast to topics that hit exactly, with one to three documents found, all of which were relevant.

The practical consideration from these brief results is that it is possible to “tune” a system such that, for a given query, a sequence of processing will occur leading to higher and higher recall – but at the expense of precision. To allow an information seeker to select his desired precision level (say, with a slider or set of check boxes) is therefore trivial with IRTools. Such functionality is not easily accessible in large search engines we have seen.

### **Information Visualization with IRTools**

There are relatively few visual interfaces for IR. This is at least partially due to the difficulty in determining an appropriate metaphor for visual presentation of textual documents. One of the metaphors under investigation is “information space,” as described in Newby (2002). Visualized information spaces are essentially 3D point clouds, where relations among documents and the terms they contain are represented geometrically, with closer distances representing closer relationships.

The means by which IRTools is able to offer 3D coordinates for documents and terms derive from the use of a statistical technique similar to LSI (Deerwester et al. 1990), based on an eigensystems analysis of the term-by-document matrix for a collection. This is similar to such statistical techniques as principal components analysis. Figures 1 and 2 show screen shots of displays for terms and documents in the interactive information space. A separate system, called Yavi, has been developed for visual interaction with information space. It operates by tying to an IRTools daemon and displaying query results in near real-time.

Figure 1: Screen shot from Yavi for TREC topic, “Generic Drug Substitutions.” Topic terms and related terms are in white with blue dots, while documents are in orange. Colored crosshairs are for orientation only. Users employ a mouse or keyboard to move through and rotate the space with six degrees of freedom.

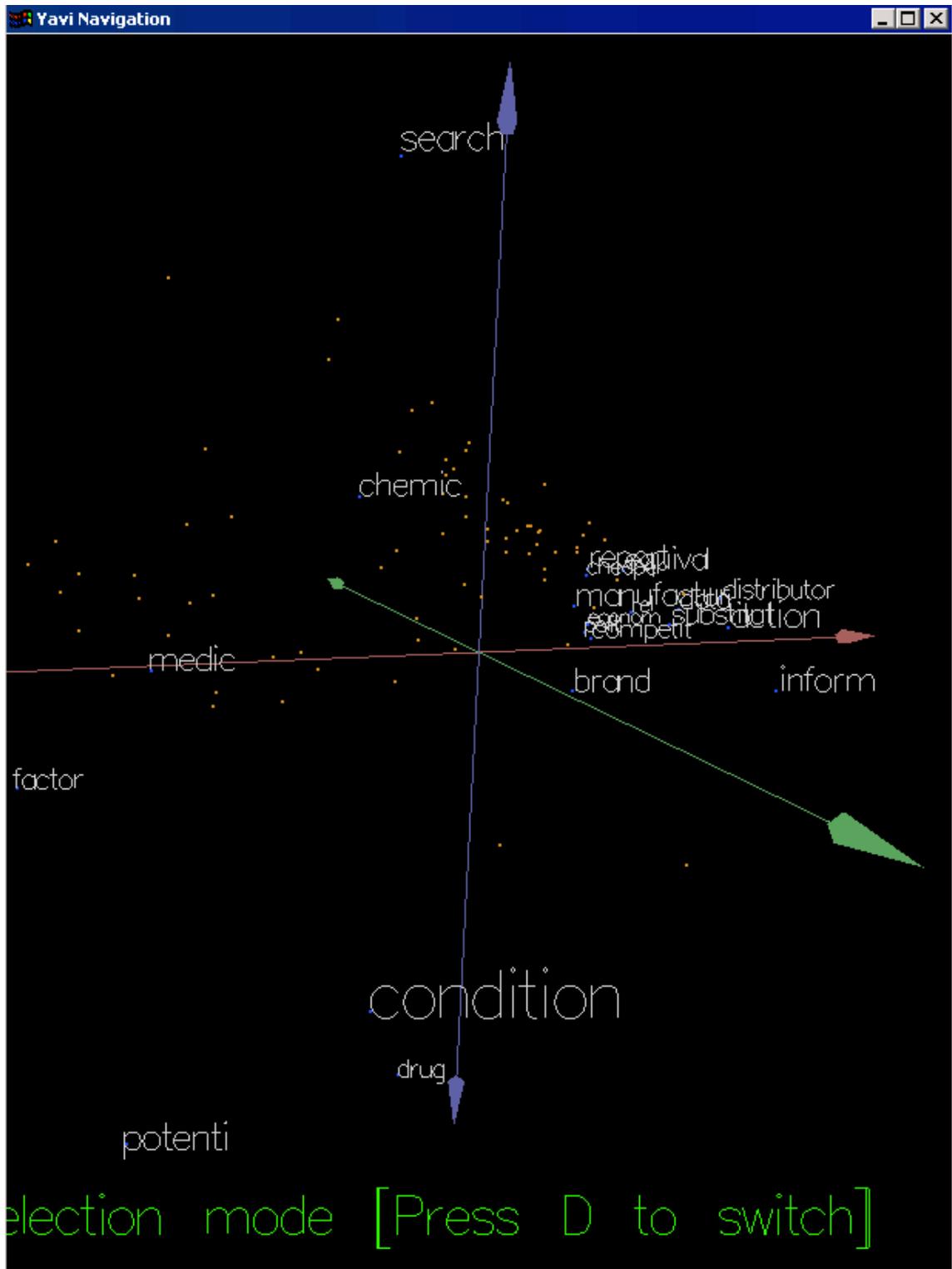
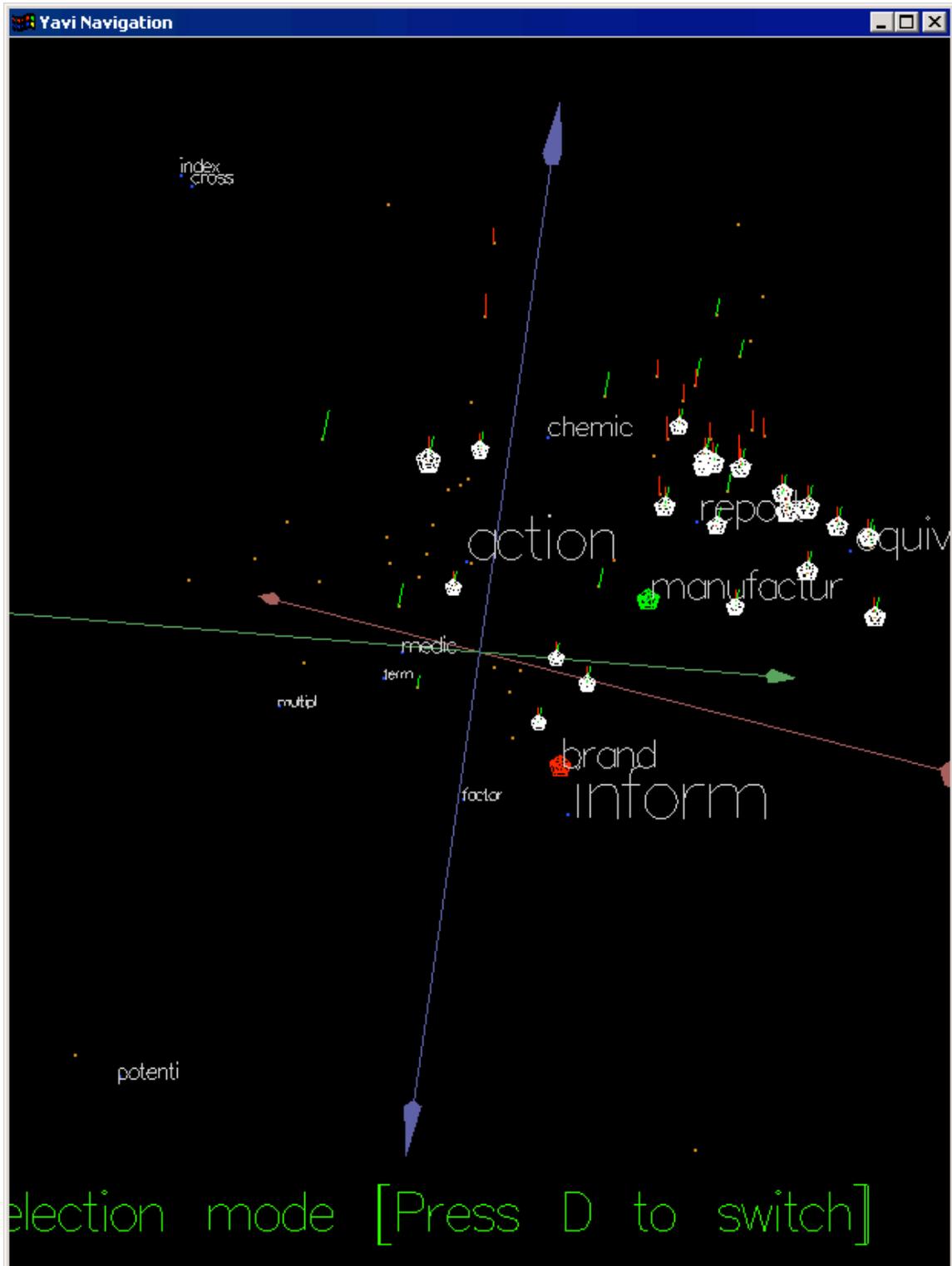


Figure 2: Screen shot from Yavi for TREC topic, “Generic Drug Substitutions.” Terms “brand” and “manufacture” are selected, and documents that contain each term have a white ball (indicating Boolean “AND”). Clicking on a white ball opens the corresponding document in a new window.



## Conclusion

This paper has presented an overview of information retrieval (IR) with near-term plans for grid-based IR (GIR) and standardization. IRTools has been presented as an implementation of GIR, as well as a general-purpose engine for IR experimentation. IRTools is intended for technically capable researchers who want to avoid re-inventing the fundamentals of IR, but desire a modern and capable code base which to develop their experimental software.

Through the support of the NSF and the KDD project, IRTools development has continued. Areas of emphasis not mentioned here include cross-language information retrieval (CLIR), including mapping English, Chinese and Arabic to the same information spaces.

Evaluation of IRTools and components is ongoing. Newby (2002) demonstrated that the Yavi visual interface to IR is usable, but requires learning new modes of querying and navigating query results. Continued participation in TREC experiments allow for practical testing of IRTools capabilities in controlled environments, while the availability of source code and several online prototypes offer real-world opportunities for feedback and ongoing development.

## References

- Catlett, Charlie. 2001. "GGF Document Series: GFD.1." Lemont, Illinois: Global Grid Forum. Online: <http://www.gridforum.org/documents/final.htm>
- Deerwester, Scott; Dumais, Susan T.; Furnas, George W.; Landauer, Thomas K. & Harshman, Richard. 1990. "Indexing by Latent Semantic Analysis." J. Amer. Soc. Information Science 41(6): 391-407.
- Foster, Ian. 2004. The Grid: Blueprint for a New Computing Infrastructure. San Francisco: Morgan Kaufmann.
- Foster, Ian et al. 2004. "Open Grid Services Architecture Use Cases: GFD.29." Lemont, Illinois: Global Grid Forum. Online: <http://www.gridforum.org/documents/final.htm>
- Gamiel, Kevin; Newby, Gregory B.; Nassar, Nassib. 2004. "Grid Information Retrieval Requirements: GFD.27." Lemont, Illinois: Global Grid Forum. Online: <http://www.gridforum.org/documents/final.htm>
- Newby, Gregory B. 2002. "Empirical Study of 3D Visualization Information Retrieval Tasks." J. of Intelligent Information Systems 18(1): 31-53.
- Newby, Gregory B. 2003. "Document Structure with IRTools." In Voorhees, Ellen (Ed.). TREC Proceedings. Gaithersburg, Maryland: NIST.
- Page, Lawrence; Brin, Sergey; Motwani, Rajeev & Winograd, Terry. 1998. "The Page Rank Citation Mapping: Bringing Order to the Web" Raghavan, S. & Garcia-Molina, H.. "Crawling the Hidden Web." In VLDB 2001: 27th International Conference on Very Large Data Bases, September 2001.
- Salton, Gerard & McGill, Michael J. 1986. Modern Information Retrieval. New York: McGraw-Hill.

Schamber, Linda; Eisenberg, Michael B. & Nilan, Michael S. "A re-examination of relevance: Toward a dynamic, situational definition." Information Processing and Management 26(6):755-776.

Stanfill, Craig & Kahle, Brewster. 1992. "Parallel free-text search on the connection machine system." Communications of the ACM 29(12): 1229-1239.