Hamparsum Bozdogan

Statistical Data Mining, and Knowledge Discovery

CRC PRESS
Boca Raton Ann Arbor London Tokyo

Contributors

Heinrich, Berry, Dongarra, and Vadhiyar

0-8493-0052-5/00/$0.00+$.50

Contents

0-8493-0052-5/00/$0.00+$.50

1

The Semantic Conference Organizer

Kevin Heinrich, Michael W. Berry, Jack J. Dongarra, Sathish Vadhiyar
University of Tennessee, Knoxville

CONTENTS

Overview

The organization of a technical meeting, workshop, or conference involving submitted abstracts or full-text documents can be quite an onerous task. To gain a sense of what topic each submission addresses may require more than just a quick glimpse at the title or abstract. The use of automated indexing and text mining can revolutionize the manner and speed of information assessment and organization. In this work, we demonstrate the use of Latent Semantic Indexing (LSI) for probing and labeling conference abstracts using an intuitive Web interface and client-server internal software design using grid-based middleware such as NetSolve. Automated text parsing and keyword extraction is facilitated using the General Text Parser software (C++) developed in the UTK Department of Computer Science.

1.1 Background

Creating a conference manually can be a burdensome task. After all papers have been submitted, the human organizer must then group the papers into sessions. The session topics can be decided either before or after the organizer has a feel for the material covered in the papers. If the session topics have been pre-conceived, then

0-8493-0052-5/00/$0.00+$.50

the organizer must select papers that fit the topic. The other option is to peruse the subject material covered in the papers and discern where natural clusters form and create sessions accordingly. In either case, once a paper has been assigned to a particular session, it cannot belong to another session. This exclusivity causes papers to be grouped together in sub-optimal arrangements so that each topic has a constrained number of papers assigned to it.

Since the average conference has around one hundred papers submitted to it, the organizer must shuffle these papers between topics trying to find a workable fit for the papers and the sessions to which they are assigned. Of course, one person trying to fit fifty to one hundred papers into about twenty sessions will lose context very quickly. Switching rapidly between sessions will cause confusion, and renaming sessions or assigning different topics may cause the entire conference to get reworked. Many times the human organizer will only work with document surrogates such as an abstract or simply the paper title, so often papers will be misclassified due to summarization errors. Also note that a significant amount of time must be spent reading and re-reading abstracts to remember what each paper's subject is. Manually creating a conference takes anywhere from a day to a week or longer. With such a combinatorial problem confronting the person who manually organizes the conference, the need for some sort of automated assistance is justified in hopes of reducing the hours spent in creating a conference.

1.2 Latent Semantic Indexing

In order for the Semantic Conference Organizer to be useful, it must replace the most time-consuming of tasks undertaken when creating a conference—reading. There are several techniques and algorithms used in the field of information retrieval that enable relevant documents to be retrieved to meet a specific need without requiring the user to read each document. The model used by the Semantic Conference Organizer is latent semantic indexing or LSI [1].

Once the document collection is received, it must be parsed into barewords called *tokens*. All punctuation and capitalization is ignored. In addition, articles and other common, non-distinguishing words are discarded. In effect, each document is viewed as a bag of words upon which operations can be performed. Once the bag of words has been formed, a term-by-document matrix is created where the entries of the matrix are the weighted frequencies associated with the corresponding term in the appropriate document.

The weight of a term within a document is a nonnegative value used to describe the correlation between that term and the corresponding document. A weight of zero indicates no correlation. In general, each weight is the product of a local and global component. A simplistic method of obtaining weights is to assign the local component as the frequency of the word within the document and the global component as

the log of the proportion of total documents to the number of documents in which the term appears. Such a method is known as a tf-idf (term frequency, inverse-document frequency) weighting scheme [2]. The aim of any scheme is to measure similarity within a document while at the same time measuring the dissimilarity of a document from the other documents within the collection.

The Semantic Conference Organizer uses a log-entropy weighting scheme [3]. The local component $l_{ij}$and the global component g_i can be computed as

$$l_{ij} = \log_2\left(1 + f_{ij}\right),\ g_i = 1 + \left(\frac{\sum_j \left(p_{ij}\log_2\left(p_{ij}\right)\right)}{\log_2 n}\right),\ p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}},$$

where f_{ij} is the frequency of the ith term in the jth document, p_{ij} is the probability of the ith term occuring in the jth document, and n is the number of documents in the collection [4]. The weighted frequency for each token is then computed by multiplying its local component by its global component. That is, the term-by-document matrix is defined as

$$M = (m_{ij}),\ \text{where } m_{ij} = l_{ij} \times g_{ij}\ .$$

The aim of using the log-entropy weighting scheme is to downweight high-frequency words while giving distinguishing words higher weight.

Once the $m \times n$ term-by-document matrix, M, has been created, a truncated singular value decomposition of that matrix is performed to create three factor matrices

$$M = K\Sigma D^T,$$

where K is the $m \times r$ matrix of eigenvectors of MM^T, D^T is the $r \times n$ matrix of eigenvectors of M^TM, and Σ is the $r \times r$ diagonal matrix containing the r nonnegative singular values of M [5]. The size of these factor matrices is determined by r, the rank of the matrix M. By using only the first $s$columns of the three component submatrices, we can compute M_s, a rank-s approximation to M. In this case, s is considerably smaller than the rank r. Document-to-document similarity is then computed as

$$M_s^T M_s = \left(D_s\Sigma_s\right)\left(D_s\Sigma_s\right)^T,$$

and can be derived from the original formula for the rank-s approximation to M [6]. Queries can be treated as *pseudo*-documents and can be computed as

$$q = q_0^T K_s \Sigma_s^{-1},$$

where q_0 is a query vector of the associated term weights [7].

The end result of LSI is a reduced space in which to compare two documents at a broader level. The goal is to map similar word usage patterns into the same geometric space [8]. In effect, documents are compared in a more general sense, so concepts are compared against each other more so than vocabulary.

1.3 Software Issues

The Semantic Conference Organizer is designed to assist a human organizer in creating a conference—it is not a tool for automating conference creation. As such, great care was taken to present information to the user without overloading the user with too much information at one time. The three basic actions that an organizer performs to create a conference are reading papers, creating sessions, and grouping papers together to form sessions. Therefore, after a document collection is submitted, the screen is split into three frames in which each of the three aforementioned actions can take place. The right frame is responsible for creating, deleting, and modifying session names, the bottom frame shows how papers semantically fit within a given session and give the user the ability to group papers into a session, and the left frame allows the user to browse a particular document. Figure 1.1 illustrates how splitting the window into three frames enables the user to maintain both a local and a global view of the document collection. Furthermore, it also allows only the requested information to be transmitted across the network at one time, which greatly reduces load time. As discovered in the first attempt at creating the organizer, the delays incurred through CGI can be quite significant if one is attempting to maintain a global perspective on the document collection by transferring the entire document collection with each page load.

Once a document collection is submitted, the text is parsed and keywords are extracted using the General Text Parser (GTP) [9]. Singleton words* are allowed to be keywords since abstracts themselves are small. Allowing singleton words also allows the user to query for a specific person and get the intuitive results. LSI is applied to the document collection after keyword extraction. A log-entropy weighting scheme (see Section 1.2) is used to ensure that distinguishing words within an abstract carry more weight.

Queries to the document collection are processed using the query module of GTP. Subsequent queries are only routed through the query module since there is no need to re-parse the document collection if the server has access to it. All other functions of the organizer are accomplished through scripts and simple text files.

*A singleton word is one that only occurs once across the entire document collection. Singleton words are discarded in many information retrieval algorithms since a singleton usually does not distinguish a document from a collection in a meaningful way.

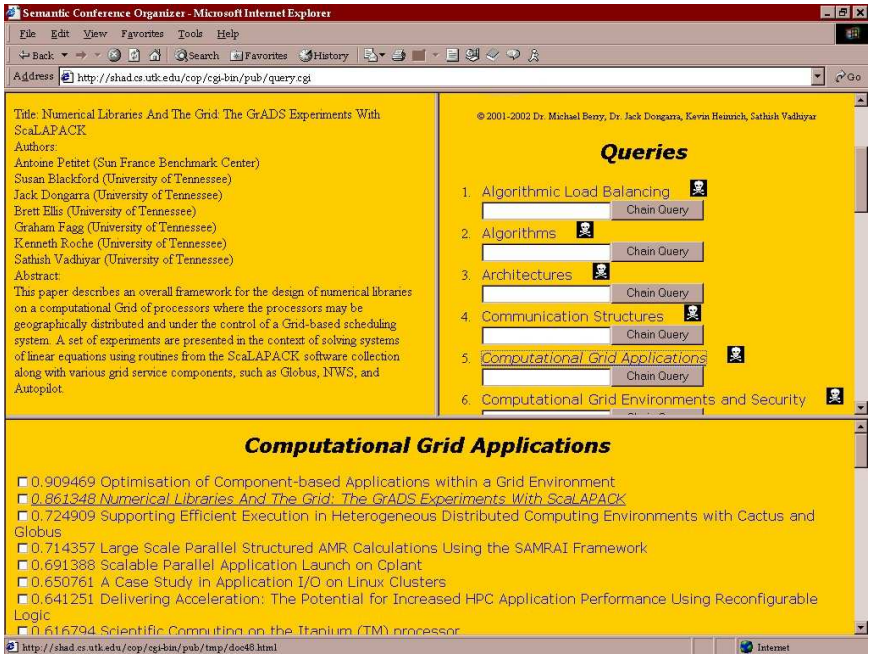

FIGURE 1.1
Sample layout of the Semantic Conference Organizer.

1.4 Creating a Conference

1.4.1 A Simple Example

A working version of the Semantic Conference Organizer can be found at http://shad.cs.utk.edu/cop. A simple query of *weather* on the documents from the Supercomputing 2001 Conference† will produce the output shown in Figure 1.2. Once the document collection has been submitted, three frames should appear. The right frame has the list queries. A new query can be added by placing them in the textbox at the bottom of the frame. Clicking on a skull next to a session will delete the entire session, while clicking on a skull next to the most recent query will only delete that query. The ability to delete intermediate queries is not provided. Clicking on a query will show the ranked list of documents for that query in the bottom frame. Clicking on a specific document title will show the entire document in the left frame. The checkboxes next to the document titles are used to lock documents to a query, i.e., assign a paper to a specific session. Once a document has been locked to a specific

†http://www.sc2001.org/techpaper.shtml

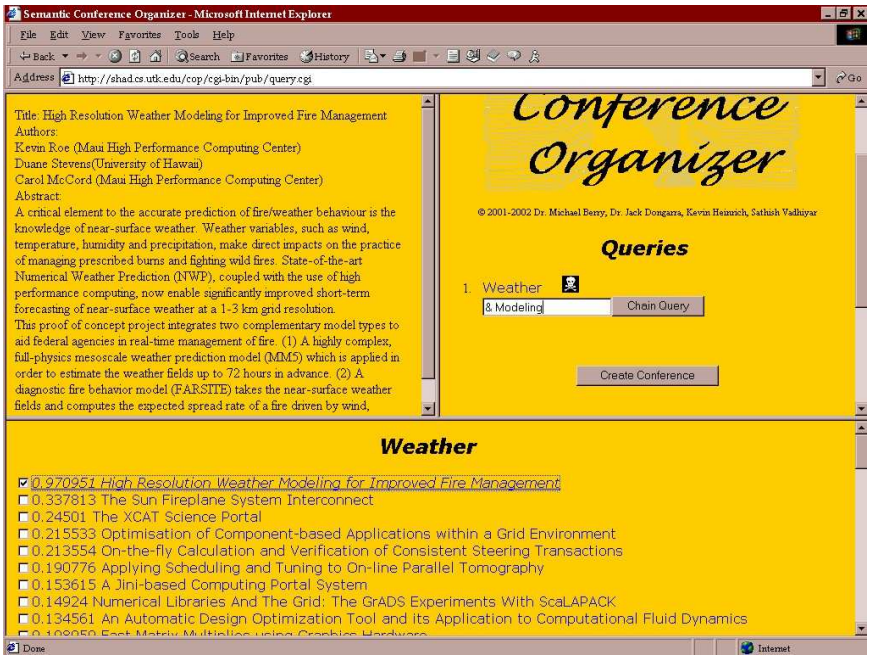

FIGURE 1.2
Return list generated for the query *weather*.

session it cannot be locked to another unless the original lock is released. Locked documents will appear in white font if the document is locked to another session or black if the document is locked further up the query chain in the same session.

If a given topic does not produce the expected results, the user may wish to modify the topic slightly. To accomplish this, we have added the ability to *chain queries*. Chaining queries is a quick way to compare the results of two different queries. In the context of the organizer, a chained query is a query viewed over time. That is, documents that have appeared in the top twenty over the last several queries will be marked to inform the user that that particular document has done a fairly good job of matching all the previous query terms. In the case of the organizer, all documents are initially colored blue. After a query has been chained, the results new to the top twenty will be colored red. After multiple chains, the documents will have a number in parentheses next to the title indicating the number of consecutive queries that the document has appeared above the threshold.

Chaining is particularly useful to see the effects a single word has on the return list. Typing an ampersand (&) at the beginning of the chained query will append the new query to the previous one. The power of chaining can be seen in Figures 1.2-1.4. Chaining has allowed the user to append *modeling* to the initial query *weather*. Not seeing desired papers appearing in the top of the return list, the user has switched the query to weather-related words. In Figure 1.3, the user has misspelled the word

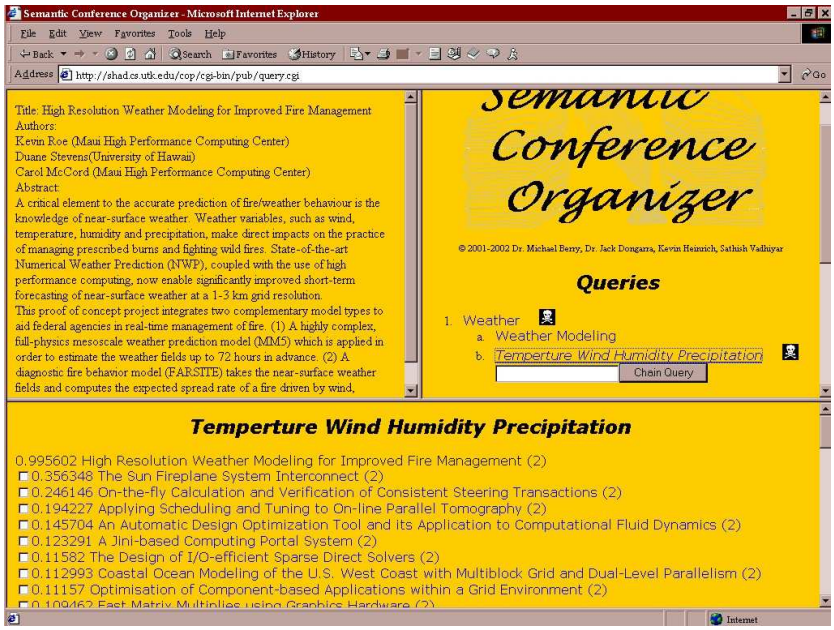

FIGURE 1.3
Note the misspelling of *temperature*.

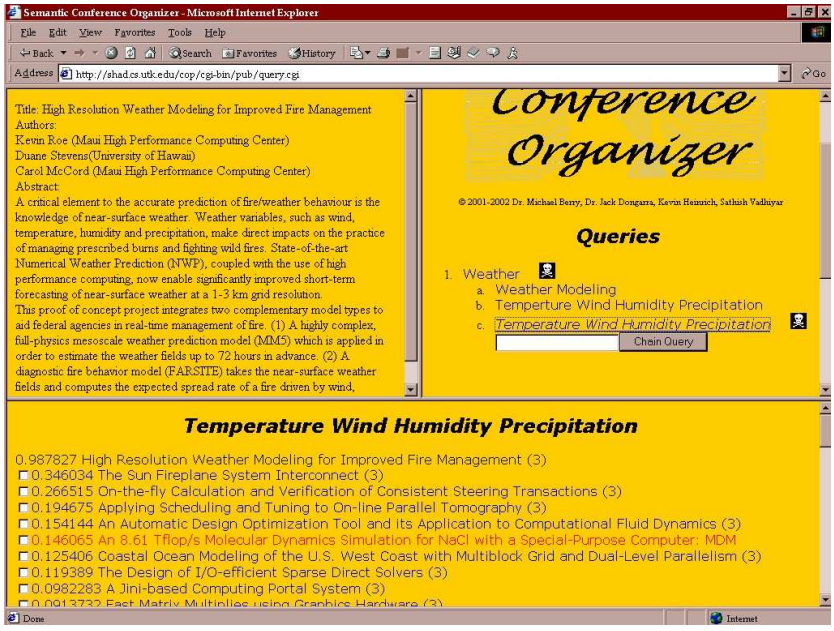

FIGURE 1.4
Notice the sixth document is new to the top 20.

temperature. By chaining, one can quickly notice the impact that the correct spelling of *temperature* in Figure 1.4 has in the return list (i.e., the sixth document returned).

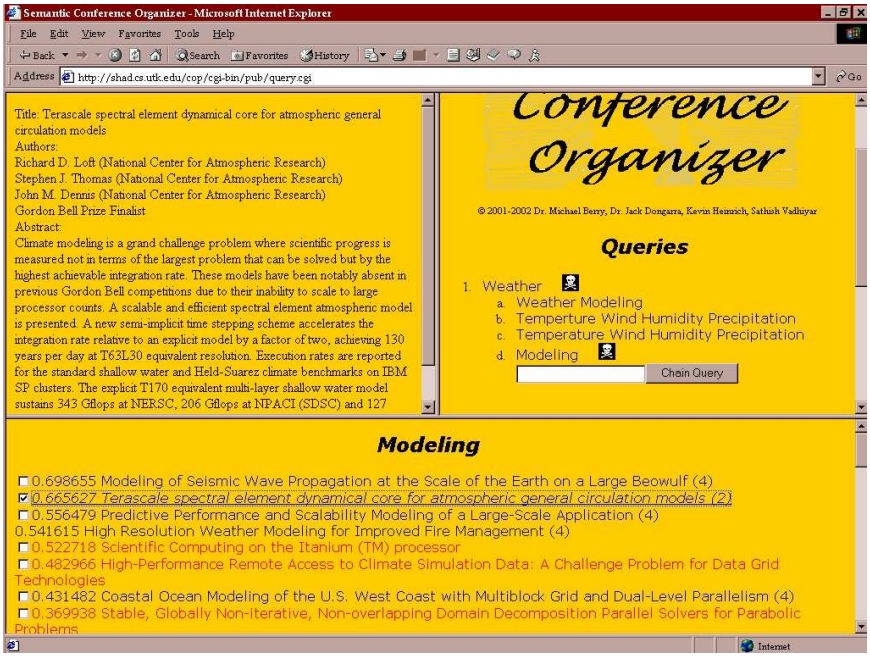

FIGURE 1.5
The second document is locked to the session *weather*.

Another useful function of chaining occurs when trying to find a session title. After all documents have been locked, one can chain queries until all documents are found high in the return list. If found high in the return list, then the session title has some semantic tie to the documents returned and hopefully will be a helpful start to finding a session title germaine to the topic. Note that in Figures 1.5 and 1.6, the user has locked two more papers to the *weather* session. As seen in Figure 1.7, the user initially tries *Global modeling* as a session title to unsatisfactory success. Changing the chained query to simply *modeling*, one notes that all three documents appear in the top three of the return list. Ideally, the three papers will be separate in a similar or (hopefully) more distinguishing way when trying to create a session title.

To create the final conference, simply click on the "Create Conference" button. The list produced is the list of sessions with the list of locked documents under each appropriate session. In parentheses next to the document name is the chained query under which the document was locked. The session title is the most recent chain query followed by the initial session title given in parentheses.

FIGURE 1.6
The third and final document is locked to the session

1.4.2 Benchmarks

Benchmarking the effectiveness of the conference organizer is a difficult task because all session groupings are highly subjective. Of the three documents that were assigned to the session *weather* in the previous example[‡], one appeared in a session titled *Groundbreaking Applications* while the other two appeared in *Sea, Wind, And Fire* in the Supercomputing 2001 Conference.

Continuing with the same document collection, two test conferences were created. In both cases, query chaining was not used. In the biased approach, the Supercomputing 2001 conference was re-created by using the same session titles and locking the corresponding documents if they appeared in the top twenty. Using such an approach, 34 out of the 60 documents were successfully locked to a session. To simulate an unbiased approach, a simple algorithm was used. First, the same session titles used in the unbiased approach were listed in alphabetical order. Next, any documents in the first session that had a score of .9 or higher were locked. This process was iterated for all twenty sessions. After that, the process was repeated for scores higher than .8 and continued decreasing the threshold by .1 until no document could

[‡]Please note that a person who did not specialize in computer weather applications created the example session.

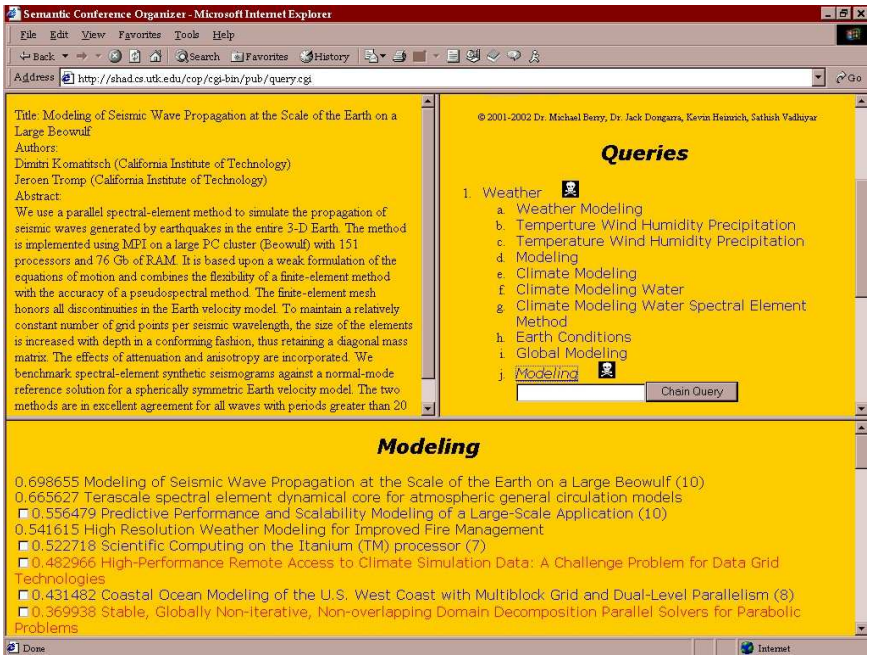

FIGURE 1.7
The three locked documents are ranked in the top four.

be locked to a session with less than three documents already locked to it. Using the unbiased approach, 49 out of the 60 documents were locked to sessions. Between the two approaches, only 7 papers were assigned to the same session. Such a disparity in results reemphasizes that a human organizer is essential to oversee conference creation.

1.5 Future Extensions

Currently, all processing and storage of the document collection is done on the webserver itself. Since the Conference Organizer only deals with document surrogates, i.e., abstracts, performing the SVD is not computationally intensive. The natural extension of this is to allow remote processing as well as remote storage on a grid which will enable the possibility of increasing the size of the document collection to include full documents. Grid-based middleware such as NetSolve[§] can be used

[§]http://icl.cs.utk.edu/netsolve/

to factor the larger term-document matrix used by LSI. Thus, the server will not be as burdened performing computationally intensive tasks and response time will inevitably improve. Given the temporary nature of the information used by this tool, distributed storage software such as the Internet Backplane Protocol[¶] is an ideal way to store the matrix and document collections themselves.

Other small adjustments are also possible for the sake of convenience. The ability to index full documents while only viewing the abstracts is one of these small future conveniences. Alternate methods to transfer the document collection such as IBP or some other method would also be nice extensions. Giving the user more flexibility with the weighting scheme and factors used by LSI is another possible future addition.

Acknowledgements

Research supported in part by the Los Alamos National Laboratory under Contract No. 03891-001-99 49.

References

[1] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41:391-407, 1990.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Boston, MA, 1999.

[3] M. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. SIAM, Philadelphia, PA, 1999.

[4] M. Berry, Z. Drmač, and E. Jessup. Matrices, Vector Spaces, and Information Retrieval. *SIAM Review* 41:335-362, 1999.

[5] G. Golub and C. V. Loan. *Matrix Computations*. Johns-Hopkins, Baltimore, Third ed., 1996.

[6] M.W. Berry. Large Scale Singular Value Computations. *International Journal of Supercomputer Applications* 6:13-49, 1992.

[¶] http://loci.cs.utk.edu/ibp/

[7] M. Berry and J. Dongarra. Atlanta Organizers Put Mathematics to Work for the Math Sciences Community. *SIAM News* 32:6, 1999.

[8] M. Berry, S. Dumais, and G. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review* 37:573-595, 1995.

[9] J.T. Giles, L. Wo, and M.W. Berry. GTP (General Text Parser): Software for Text Mining. In *Proceedings of the C. Warren Neel Conference on The New Frontiers of Statistical Data Mining and Knowledge Discovery*, Knoxville, TN, June 22-25, 2002.