ARTICLES

D-Lib Magazine
October 2002

Volume 8 Number 10

ISSN 1082-9873

Information Retrieval by Semantic Analysis and Visualization of the Concept Space of *D-Lib® Magazine*

Junliang Zhang
<junliang@email.unc.edu>
University of North Carolina, Chapel Hill [1]

Javed Mostafa
<jm@indiana.edu>
Himansu Tripathy
<htripath@indiana.edu>
Laboratory for Applied Informatics Research
Indiana University

Abstract

In this article we present a method for retrieving documents from a digital library through a visual inter based on automatically generated concepts. We used a vocabulary generation algorithm to generate a se concepts for the digital library and a technique called the *max-min* distance technique to cluster them. Additionally, the concepts were visualized in a spring embedding graph layout to depict the semantic relationship among them. The resulting graph layout serves as an aid to users for retrieving documents. online archive containing the contents of *D-Lib Magazine* from July 1995 to May 2002 was used to test utility of an implemented retrieval and visualization system. We believe that the method developed and tested can be applied to many different domains to help users get a better understanding of online document collections and to minimize users' cognitive load during execution of search tasks.

Introduction

Over the past few years, the volume of information available through the World Wide Web has been expanding exponentially. Never has so much information been so readily available and shared among s many people. Unfortunately, the unstructured nature and huge volume of information accessible over networks have made it hard for users to sift through and find relevant information. To deal with this problem, information retrieval (IR) techniques have gained more intensive attention from both industria and academic researchers. Numerous IR techniques have been developed to help deal with the informat overload problem. These techniques concentrate on mathematical models and algorithms for retrieval. Popular IR models such as the Boolean model, the vector-space model, the probabilistic model and thei variants are well established (Baeza-Yates, & Ribeiro-Neto, 1999).

From the user's perspective, however, it is still difficult to use current information retrieval systems. Us

frequently have problems expressing their information needs and translating those needs into queries. T is partly due to the fact that information needs cannot be expressed appropriately in systems terms (Bell 1980). It is not unusual for users to input search terms that are different from the index terms informatic systems use. Various methods have been proposed to help users choose search terms and articulate quer One widely used approach is to incorporate into the information system a thesaurus-like component tha represents both the important concepts in a particular subject area and the semantic relationships among those concepts (Chen et al. 1993). Unfortunately, the development and use of thesauri is not without its own problems. The thesaurus employed in a specific information system has often been developed for a general subject area and needs significant enhancement to be tailored to the information system where i to be used. This thesaurus development process, if done manually, is both time consuming and labor intensive. Usage of a thesaurus in searching is complex and may raise barriers for the user. For illustrati purposes, let us consider two scenarios of thesaurus usage. In the first scenario the user inputs a search term and the thesaurus then displays a matching set of related terms. Without an overview of the thesau — and without the ability to see the matching terms in the context of other terms — it may be difficult t assess the quality of the related terms in order to select the correct term. In the second scenario the user browses the whole thesaurus, which is organized as in an alphabetically ordered list. The problem with approach is that the list may be long, and neither does it show users the global semantic relationship am all the listed terms.

Nevertheless, because thesaurus use has shown to improve retrieval (Baeza-Yates & Ribeiro-Neto, 199 for our method we integrate functions in the search interface that permit users to explore built-in search vocabularies to improve retrieval from digital libraries. Our method automatically generates the terms a their semantic relationships representing relevant topics covered in a digital library. We call these generated terms the "concepts", and the generated terms and their semantic relationships we call the "concept space". Additionally, we used a visualization technique to display the concept space and allow users to interact with this space. The automatically generated term set is considered to be more representative of subject area in a corpus than an "externally" imposed thesaurus, and our method has th potential of saving a significant amount of time and labor for those who have been manually creating thesauri as well. Information visualization is an emerging discipline and developed very quickly in the l decade. With growing volumes of documents and associated complexities, information visualization ha become increasingly important. Researchers have found information visualization to be an effective wa use and understand information while minimizing a user's cognitive load (Card et al. 1991). Various use interface and visualization issues concerning information retrieval — such as query specification, overv of documents, and display of search results — are discussed by Hearst (1999).

Our work was based on an algorithmic approach of concept discovery and association (Mostafa et al. 1998). Concepts are discovered using an algorithm based on an automated thesaurus generation proced Subsequently, similarities among terms are computed using the cosine measure, and the associations among terms are established using a method known as *max-min* distance clustering. The concept space then visualized in a spring embedding graph, which roughly shows the semantic relationships among concepts in a 2-D visual representation. The semantic space of the visualization is used as a medium for users to retrieve the desired documents.

In the remainder of this article, we present our algorithmic approach of concept generation and clusterir followed by description of the visualization technique and interactive interface. The paper ends with ke conclusions and discussions on future work.

2. Concept Generation

The document set we used in our research came from an online archive of *D-Lib Magazine*

(http://www.dlib.org/) from July 1995 to May 2002, which contains approximately 350 articles. From tl archive, we extracted and parsed out the full text of articles for concept generation and clustering purpo The metadata of the archive, including title, author, volume number, issue number, and URL, were also parsed out and were used for the display of search results.

After parsing the full texts, we generated concepts for the *D-Lib* articles using a vocabulary generation (VG) algorithm. The following steps were adopted to generate the concepts:

1. *Elimination of stop words and word stemming.* A stop word list was developed to remove those word that are too general to be useful such as "the", "an", "unless", "versus", etc. After the elimination of stop words, a stemming algorithm was used to identify the word stems for the remaining words (for use in st 2). The stemming algorithm applied was developed by Porter (1980). With an explicit list of suffixes, th algorithm strips the suffixes of words to leave a valid stem based on some heuristics. Since it does not u any stem dictionary, the algorithm is very fast. (The detailed description of this algorithm is available at <http://www.tartarus.org/~martin/PorterStemmer/def.txt>. It can also be downloaded free from <http://www.tartarus.org/~martin/PorterStemmer>.)

2. *Unique word identification and weighting.* After application of the stop word list and stemming, uniq words were identified independent of case. Each unique word in the document set was then weighted pe document using a tf*idf weighting formula. (See Appendix.) Even though other more sophisticated term weighting and refinement techniques are available in IR, it was found that the tf*idf approach is almost effective as other more sophisticated ones (Lewis, 1992). Weighted words were ranked per document based on ascending weight order.

3. *Term selection.* Instead of extracting all the unique words, we only selected a subset identified as important concepts. Term selection was controlled by two parameters: *R* and *D*. The parameter *R* contro the rank of a particular term in documents. The parameter *D* controls the distribution of a particular term the document set. The algorithm selected those terms as relevant concepts that were ranked in top *R* am all terms and present in at least *D* documents. Here, *R* and *D* values were determined empirically accord to the needs of the application (retrieval in our case) and corpus size.

3. Semantic Analysis

Given the generated concepts, we explored the semantic relationship among these concepts, and further divided the concepts into clusters that were to be treated as the different sub-topics of the document set. The similarity between two terms was measured by the Salton & McGill cosine formula. (See Appendix Using that measure, a term by term similarity matrix was produced, where each cell represents the similarity value between two given terms.

Based on the similarity matrix, we used a technique called *max-min* Distance Identification (Tou & Gonzalez, 1974) to divide the concepts into several clusters to generate the sub-topics in the concept sp In the algorithm, concepts were initially divided into two sets of terms: centroids and centroid candidate At the beginning, all terms were centroid candidates and the centroid set was empty. The first term in th set was then selected as the initial centroid. The second centroid was built by selecting the candidate wi the maximum distance from the centroid set, which also must be at least larger than the value of control parameter, *theta* (between 0 and 1). The distance of the candidate from the centroid set is defined as the minimum distance of the candidate from all the members of the centroid set. The cluster building proce was repeated until the maximum distance failed to exceed *theta*. For each remaining term, the nearest centroid was found and the term was grouped under that centroid as its cluster member. The *theta* permitted us to control the granularity of the cluster space. A high *theta* produces a smaller numbers of

centroids than a low value.

A concept space was thus established representing a set of generated concepts for *D-Lib Magazine* cont and the semantic relationships among those concepts. A spring embedding algorithm was then used to visually depict this concept space. The following three sections contain the description of our visualizat algorithm and interface design.

4. Spring Embedding Algorithm

The spring embedding algorithm is also called the force-directed placement algorithm. The algorithm simulates a physical model in which the nodes are represented by rings and the edges connecting nodes represented by springs. Initially, all the rings are randomly displayed. The forces exerted on the rings fo them into an equilibrium configuration, where the sum of the forces on each ring is zero. (See Appendi The spring embedding algorithm is especially useful for showing the semantic similarity relationship among the nodes, since the natural length of the spring between two nodes can be determined by their similarity. The more similar two nodes are, the shorter the natural length of the spring is, and vice versa After the spring graph reaches an equilibrium configuration, it nicely represents the semantic similarity relationships among the nodes.

5. Interface and Interaction

A Java applet [2] was developed to implement the spring layout algorithm and interactive functions. Fig 1 shows the visual interface for the *D-Lib Magazine* concept space, which contains three areas: visualization (on the left with the colored concept bubbles), search (at the bottom with the search box), concept space manipulation functions (three tabbed panels on the right). The three tabbed panels are the control panel, the attributes panel, and the concept list panel.

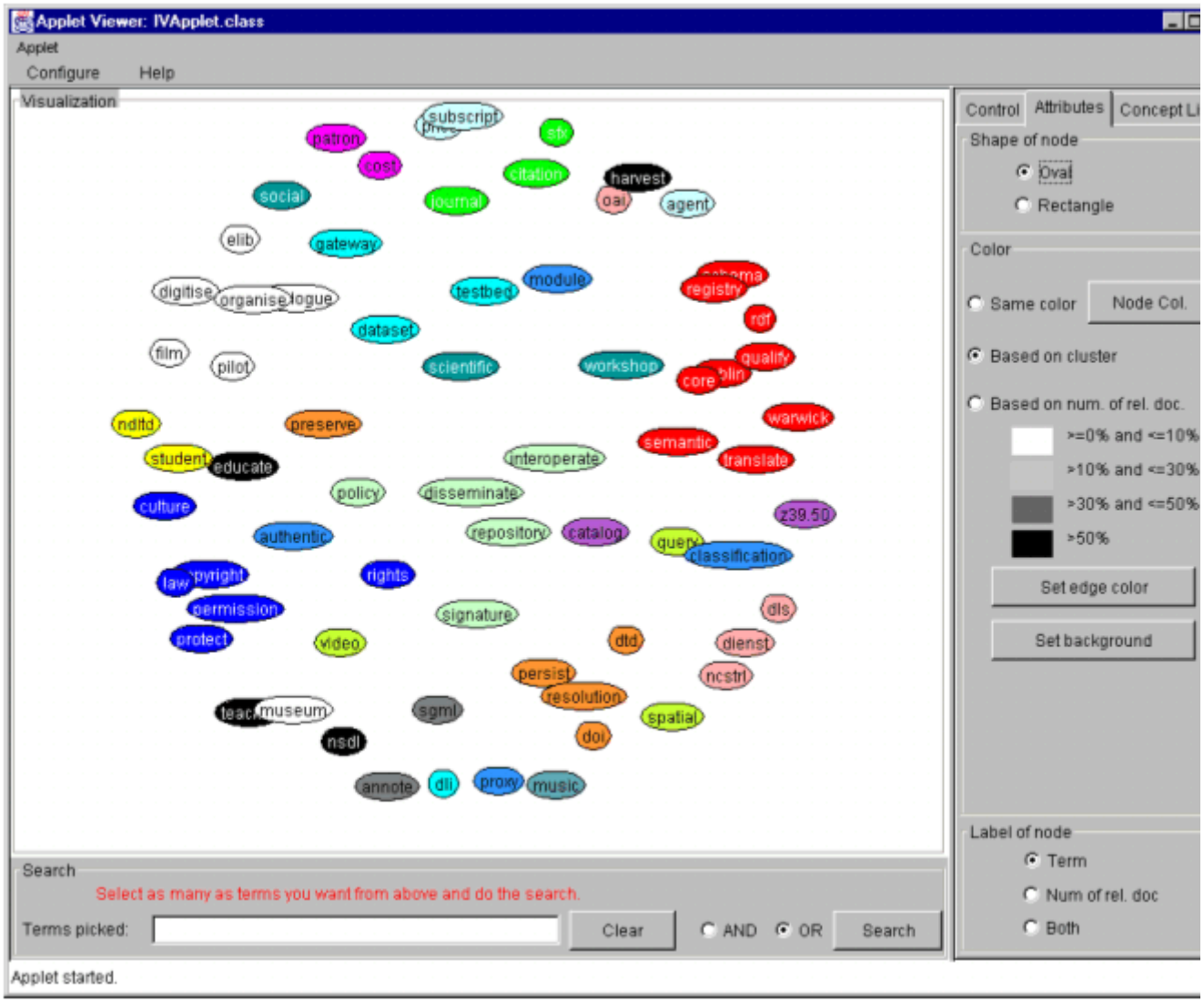

Figure 1. Visual interface of the concept space for the *D-Lib Magazine* online archive.

In Figure 1, the visualization area displays 69 nodes that represent the 69 generated concepts [3]. Simila concepts are close to each other in the physical space, for example, "copyright" is near "permission", an "journal" is near "citation". Nodes are labeled with the concept name by default, and moving the cursor over a node will pop up the tool tip showing both the corresponding concept name and number of the documents related to it. All the nodes are color-coded based on cluster analysis conducted during the semantic analysis process. Distinctive colors are used for different clusters of concepts. For example, th cluster in Figure 1 containing the terms: "protect", "permission", "law", "copyright", "culture" and "righ is shown in blue and the cluster of "journal", "citation", and "sfx" is shown in green. Notice that some o the concepts from the same cluster that represent a strong similarity relationship are not necessarily clos each other visually. One of the reasons is that a particular node experiences spring force exerted by all t other nodes in the concept space, and the total force of other nodes may pull the node away from its clu Also, during the process of node movement, nodes can be trapped in a local minimum energy position.

Boolean searches can be done on the displayed concepts in the search area. Users can select as many concepts as they want from the visualization panel and conduct "AND" or "OR" searches by choosing t appropriate Boolean options. When the search button is pressed, another browser is invoked to present t

search results, which includes article titles, authors, volumes and issue numbers. The articles on the resu page are hyperlinked to the original articles in *D-Lib Magazine*. Users can also input their own keyworc the text field or can use both their own keywords and the displayed concepts as query keywords.

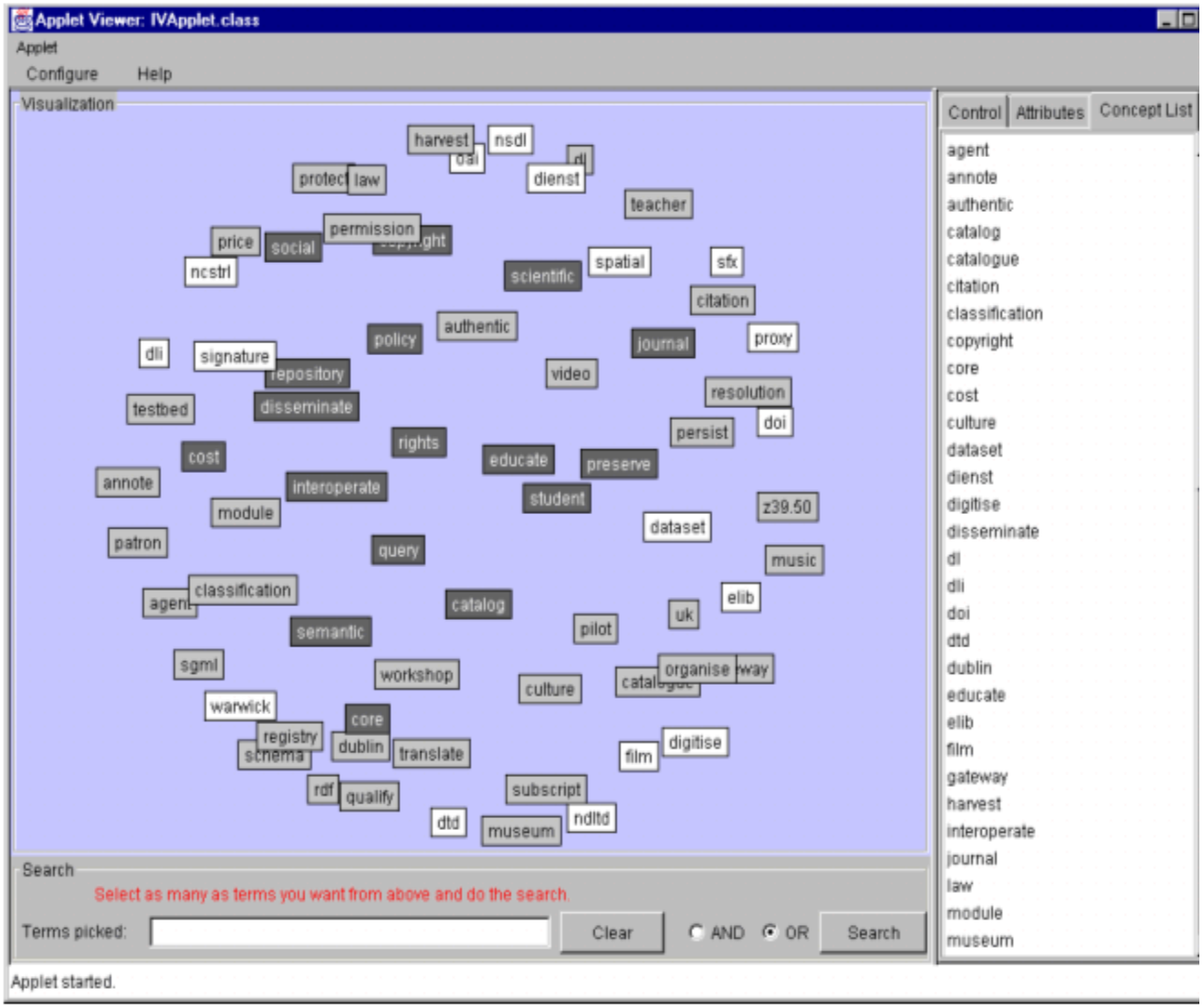

Figure 2. Color-coding based on the frequency of documents associated with concepts.

The tab panels were designed to save screen space. The attributes panel allows the user to manipulate different visual attributes of nodes such as their shape, color, and label. Since color is considered superi to size, brightness, and shape in visually identifying items (Christ, 1975), we used color to encode node based on the cluster analysis discussed above. We also coded the nodes by color based on the number o documents in the archive related to each concept. For example nodes with fewer than 10% of related articles in the *D-Lib Magazine* archive are shown in white, and concepts that appear in more than 30% the articles in the archive are shown as gray nodes. Hence, from the visualization in Figure 2 we can lea that concepts such as "disseminate", "interoperability", "repository" are more heavily discussed topics i *D-Lib Magazine* than the concepts appearing as white nodes. Based on this feature of the interface, user can get a quick impression of relative concept distribution or occurrence of concepts in the online collection.

The control panel (right part of Figure 3) provides several useful functions to help the user explore the semantic relationships in the concept space. By setting the value in the "set minimum similarity value" field, the user can generate a filtered visualization of corresponding concepts, the similarity among whi is larger than the value specified by the user. The visualization area of Figure 3 shows the snapshot of t interface with the 0.6 minimum similarity value, on which only concepts with relationships satisfying 0 or larger similarity are visualized. The radio button, labeled "line", is used to display edges between noc Each edge comes with an integer numbers above it indicating its natural length [4]. The "scramble" butt restarts the moving of the nodes by randomly placing the nodes on the visual panel. The "Increase the gravity power" slider is used to proportionally increase or decrease the natural length among concepts, actual effect of which is to spread out or shrink the graph.

The concept list panel shows all the concepts in an alphabetically ordered list (see the right part of Figu 2). When a concept in the list is selected, the corresponding concept displayed in the visualization panel highlighted so that users can quickly locate the concept and view it in the context of other concepts wit the concept space.

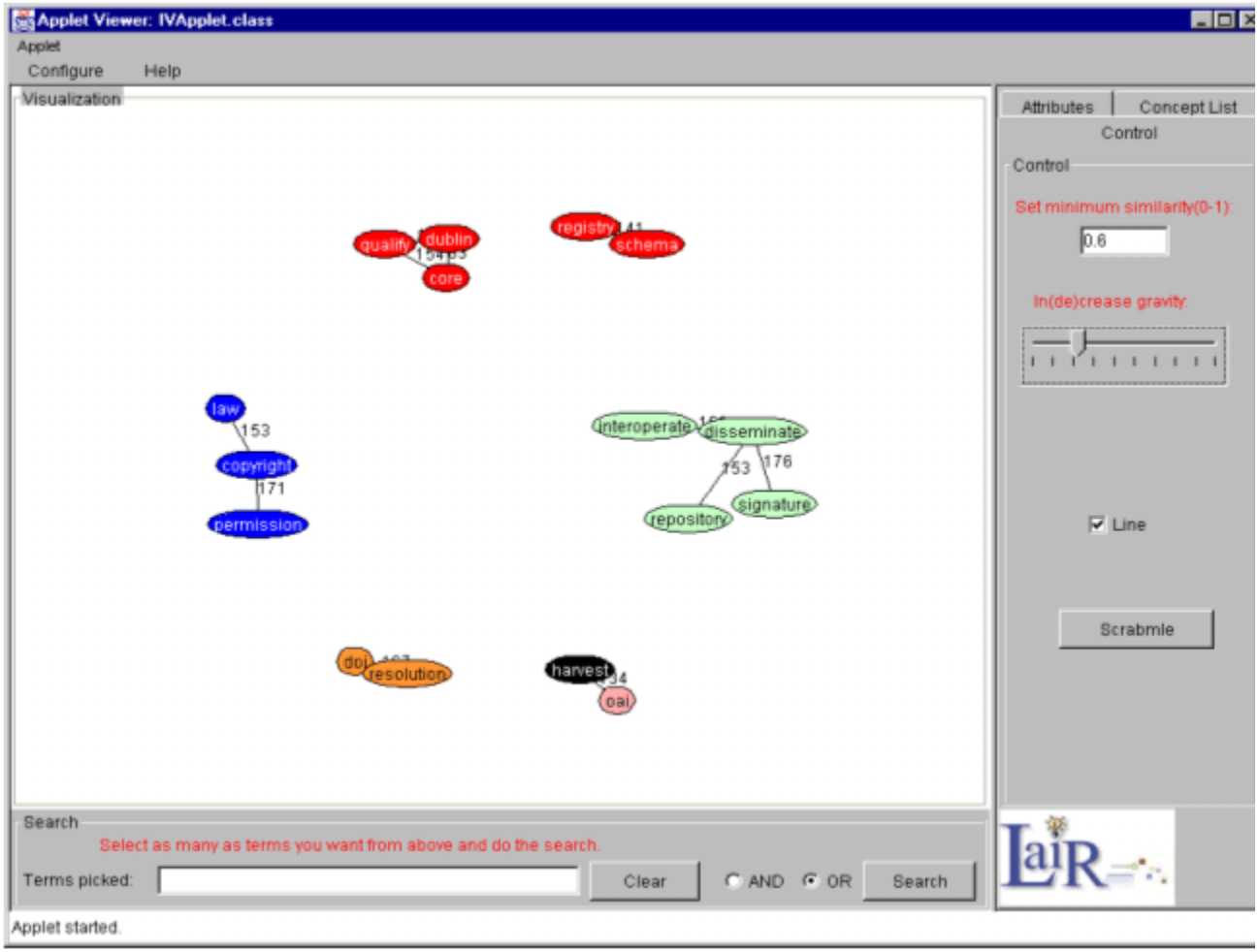

Figure 3. Visualization of filtered concept space.

6. Limitations and an Alternative Interface

There are two limitations of the current visual interface. First, the initial visualization may appear crow and complex when the number of nodes reaches 100 or higher. Second, there are some ambiguities in t

way the concept clusters are displayed. Due to the way the spring embedding algorithm functions over a large term set, in certain instances, the concepts within the same cluster are not necessarily near each ot visually. To solve these two problems, we developed an alternative two-layer hierarchical visualization In this implementation, the initial visualization only displays the clusters instead of showing all the nod The clusters are color coded as small square panels. The panels are labeled, and in each panel, cluster concepts appear as small dots (see Figure 4). A zoom-in operation is available, which enlarges a cluster panel and its corresponding concepts, and this feature helps the user see the details of individual cluster The concepts on the enlarged cluster panel are visualized as a spring embedding graph representing the concept space of individual clusters (see Figure 5). The other clusters are maintained as smaller square panels around the spring graph to allow the user to quickly select alternative clusters. All the major interface functions mentioned previously (in Section 5) are supported in this alternative interface. The alternative interface can effectively handle up to 32 clusters and at least 400 concepts, and the visualiza of clusters is clearer. However, when it comes to a small set of concepts, the previous interface may provide an advantage because the user can view all the relevant concepts associated with an online collection in one display. A usability test needs to be conducted to compare these two visualizations.

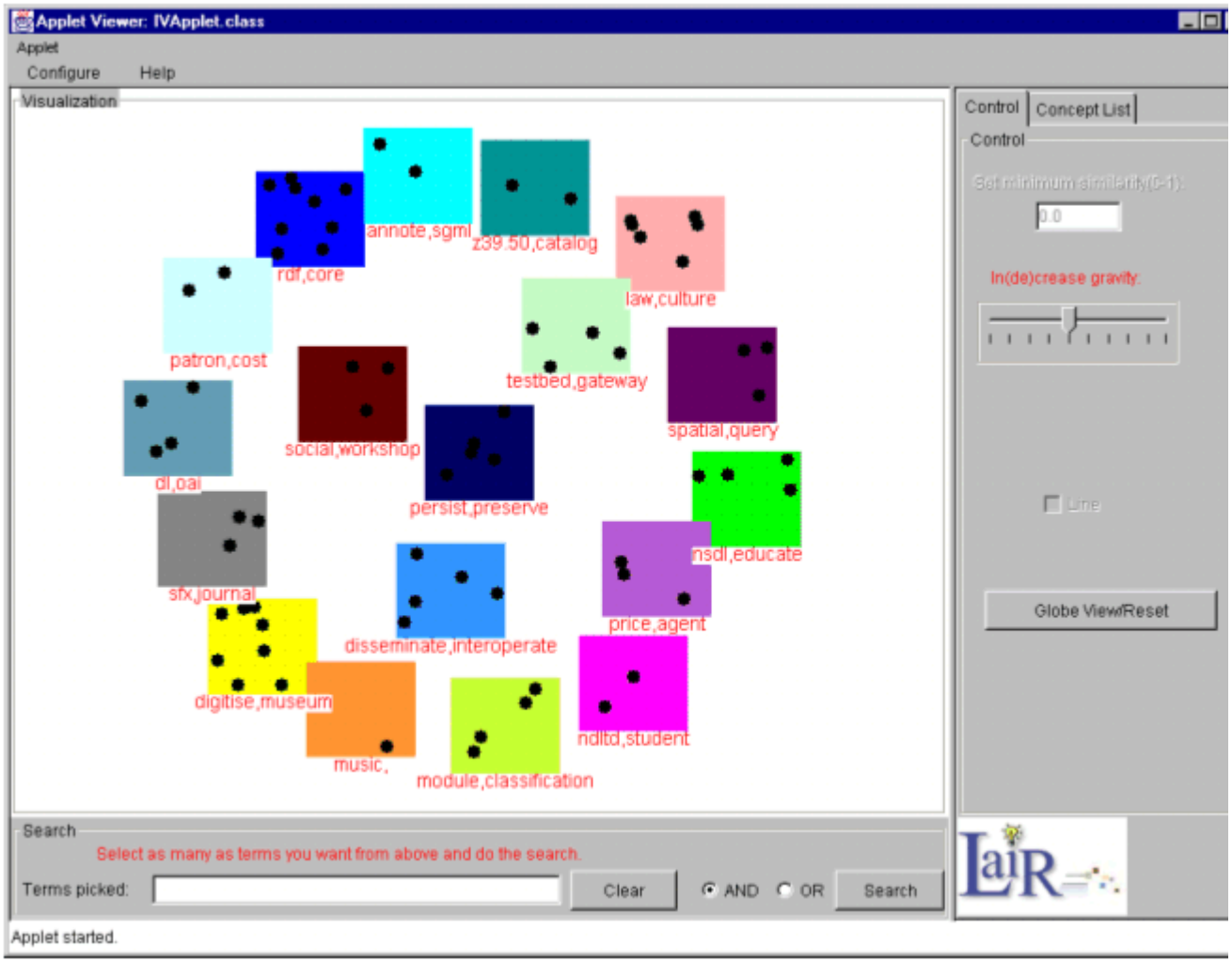

Figure 4. Overview of the concept space in the alternative interface design.

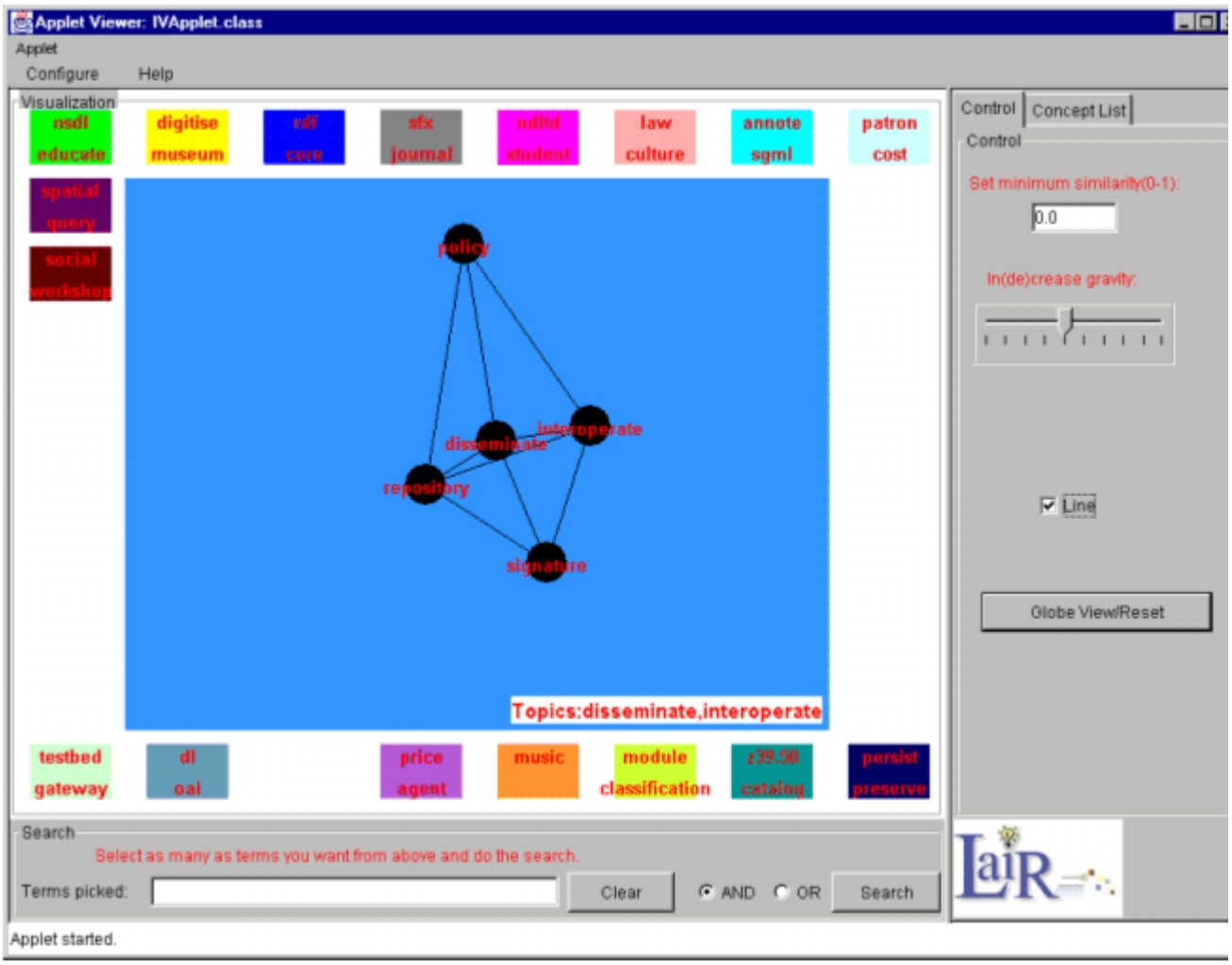

Figure 5. Zoom-in function applied on a specific concept cluster.

7. Conclusions and Future Work

We developed automatic techniques for term set extraction, association generation and visualization of concept spaces to aid retrieval from online document collections. Based on automatic concept discovery and clustering, the interface we developed visually depicts the generated concepts and their semantic relationships by using a spring embedding graph. The visualization provides the user a clear and attracti overview of what is available in a document collection. It shows the major sub-topics appearing in the document collection as concept clusters. The visual cues used in the visualization, such as color, label, a distance among nodes, are designed to be visually distinctive, and the design takes advantage of a huma powerful visual perception. The three tab panels also provide users with additional power to control and customize the interface based on different usage conditions. To support improved search, the interface a allows the user to conduct searches on the visualized collection based on terms specified by the user or concepts chosen directly from the visualization. We believe that this tool can be applied in many differe domains to help users attain clearer understanding of a complete document collection and to minimize users' cognitive load during search formulation and execution.

We are planning further improvement of the visualization algorithm implemented in this study. For example, the spring embedded algorithm should be fine tuned to better reflect the semantic relationship among concepts. One refinement may be to prevent the nodes from being trapped at a local energy

minimum position. Additionally, an evaluation study is being planned with two goals: 1) to compare th two alternative interface designs discussed in this article, and 2) to compare the effectiveness of the interfaces proposed in this article with the current search interface of *D-Lib Magazine* [6].

Acknowledgement

The authors wish to acknowledge Dr. Katy Börner for her valuable guidance and many constructive comments for this article. The research was partially funded by the NSF Digital Libraries Phase II grant IIS-9817572.

Notes

[1] The research reported in this article was completed while Junliang Zhang was at Indiana University.

[2] The JAVA applet is available at <http://ella.slis.indiana.edu/~junzhang/dlib/IV.html>.

[3] The number of generated concepts was controlled by the R and D parameters. Here, we selected R=D=4, restricting the extracted concepts to those ranked in top 10 and those that appears in at least 4 documents.

[4] The natural length is proportional to the similarity value between two concepts represented by nodes

[5] A prototype of this interface has been developed and is available at <http://ella.slis.indiana.edu/~junzhang/dlib/IV.html>.

[6] The D-Lib search interface is available at <http://www.dlib.org/Architext/AT-dlib2query.html>.

References

[Baeza-Yates] Baeza-Yates, R. & Ribeiro-Neto, B. (Eds.). (1999). *Modern Information Retrieval.* MA: Addison-Wesley.

[Battista et al.] Battista, G.D., Eades, P., Tamassia, R. & Tollis, I.G. (1999). *Graph drawing algorithms the visualization of graphs.* New Jersey : Prentice Hall.

[Belkin] Belkin, N.J. (1980). Anomalous states of knowledge as a basis for information retrieval. *Canad Journal of Information Science,* 5, 133-143.

[Card] Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991). The information visualizer: An information workspace. *Proceedings of CHI'91* (New Orleans, Louisiana).

[Chen et al.] Chen, H., Lynch, K.J., Basu, K., & Ng, T. D. (1993). Generating, integrating, and activatin thesauri for concept-based document retrieval. *IEEE Expert,* 8(2), 25-34.

[Christ] Christ, R. (1975). Review and analysis of color-coding research for visual displays. *Human Factors,* 17, 542-570.

[Hearst] Hearst, M. (1999). User interfaces and visualization. In R. Baeza-Yates & B. Ribeiro-Neto (Ed *Modern Information Retrieval* (pp. 257-325). MA: Addison-Wesley.

[Lewis] Lewis, D. D. (1992). Text representation for intelligent text retrieval: A classification-oriented view. In P. S. Jacobs (Ed.), *Text-based intelligent systems: Current research and practice in informatio extraction and retrieval* (pp. 179-197). Hillsdale, NJ: Erlbaum.

[Mostafa et al.] Mostafa, J., Quiroga, L.M., & Palakal, M. (1998). Filtering medical documents using automated and human classification methods. *Journal of the American Society of Information Science*, 1304-1318.

[Porter] Porter, M. F. (1980), An algorithm for suffix stripping. *Program,* 14(3), 130-137.

[Salton & McGill] Salton, G. & McGill, M. (1983). *Introduction to Modern Information retrieval.* New York: McGraw Hill.

[Tou & Gonzalez] Tou, J. & Gonzalez, R. (Eds.). (1974). *Pattern recognition principles.* MA: Addison-Wesley.

Appendix

1) The tf*idf weighting formula (Salton & Mcgill, 1983):

$$W_{ik} = t_{ik} * \log(\frac{N}{n_k})$$

Where t_{ik} is the number of occurrences of term t_k in the document i, N is the total number of documents, and n_k is the number of documents in the training set that contain the given term t_k.

2) The cosine formula (Salton & Mcgill, 1983):

$$\sum_{i=1}^{t} v_i z_i \Big/ \sqrt{(\sum_{i=1}^{t} v_i^2)(\sum_{i=1}^{t} z_i^2)}$$

In the above, v and z_i are the i th elements of term vectors V and Z respectively, representing the term weights in the i th documents.

3. The spring embedding, or force-directed placement, algorithm (Battista et al. 1999):

The force on a ring v can be represented as:

$$F(v) = \sum f_{uv}$$

In the above equation f_{uv} is the force exerted on v by the spring between u and v. u can be any other ring except v itself. The force f_{uv} follows Hooke's law, that is, f_{uv} is proportional to the difference between the distance between u and v and the natural length of the spring. The computation equation i

$$f_{uv} = k_{uv}(d_{uv} - l_{uv})$$

k_{uv} is the stiffness of the spring between u and v, which is empirically determined. d_{uv} is the distanc between u and v. l_{uv} is the natural length of the spring between u and v.

Top | Contents
Search | Author Index | Title Index | Back Issues
Previous Article | Conference Report
Home | E-mail the Editor

DOI: 10.1045/october2002-zhang