

Exploring Document Data using InfoVis Explorer

Charles Hoffmeyer and Glen Pawloski

School of Computer and Information Systems

Grand Valley State University

Allendale, MI 49401

cchoffme@cchoffme.com, grp1625@gmail.com

ABSTRACT

In this paper, we discuss the design decisions behind InfoVis Explorer. InfoVis Explorer is a custom tool to find and discover documents deemed important to the history of Information Visualization, as described in the 2004 IEEE Information Visualization Contest dataset [1]. Through the use of effective design decisions, we feel that we have produced a tool which will aid students or researchers attempting to find quickly find relevant papers based on any number of factors in a clear and understandable format. InfoVis Explorer extends the original contest dataset by directly linking external databases to compensate for missing data elements and, in some cases, provide an electronic copy of the paper being explored.

Keywords

Document Visualization, Information Visualization, Dynamic Queries

INTRODUCTION

The InfoVis Explorer was created to efficiently explore the history of Information Visualization through the use of a large collection of topical papers deemed important by the 2004 IEEE Information Visualization Contest [1]. We felt that the best way to do this was to create a custom Graphical User Interface in Java and use JDBC to access and store our data in SQL format, using a McKoi database. This program implements the ability to dynamically create queries based on pull-down boxes which are populated from actual data within the dataset to minimize the possibility of null values.

Our anticipated audience for this visualization is people who have had experience spending several hours looking in the stacks through various conference proceedings to find any paper that met the student or researchers needs.

The goal of this visualization is to minimize the time that it

takes to locate these papers, provide the ability to explore each of them further, and to discover related papers. Each entry in the InfoVis Explorer has direct links to CiteSeer, ACM, and IEEE databases to allow the user to go beyond the dataset and find more about the document – who is citing this paper in their works, fill in missing elements from the contest dataset, and even read an electronic copy of the paper being explored!

VISUALIZATION ISSUES

To ensure that the application is effective, several issues were identified which we felt needed to be addressed in our visualization.

Experience and Skill

We do not want to limit our visualization to those which are familiar or comfortable with directly manipulating database languages or creating query strings. All available options within the database are to be created dynamically from the dataset, and not hard-coded into the application. Our expected audience is students or researchers, who are familiar with tediously searching out specific titles from various sources, and evaluating their usefulness with a somewhat standard methodology. No specific skills or prerequisite knowledge should be required to use this visualization.

Presentation

We wanted to create a visualization which will be appealing to the eye, and the user. We wanted this visualization to do more than just show the data in tabular format; the data should flow as if the user was looking at a book to determine if he should read the entire article. Screens should not be crammed with information; instead they must expand as needed to accommodate the requirements.

InfoVis Explorer is shown below. It is open, and all functionality is visible. It is clean, and flows from the top left of the screen to the bottom right in control. While the data used may change, the layout and display remains the same.

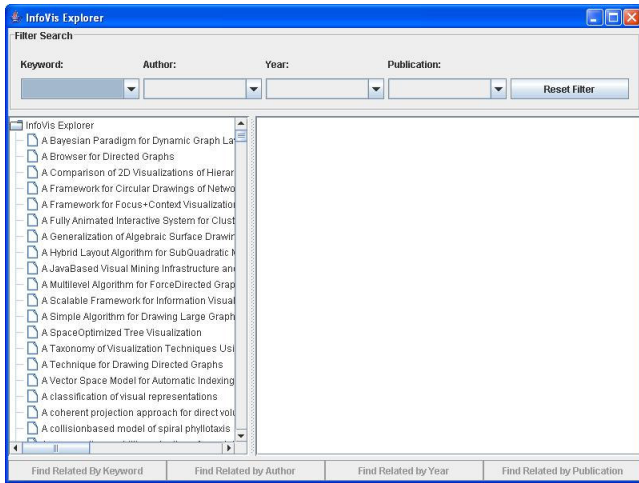


Image 1: InfoVis Explorer, in initial state.

immediately in limiting the search area and showing the result.



Image 1: Filters on top of screen to limit selection area.

Immediately below the filter pane is the result window. The search results show up on the left hand side of the screen. The right side remains empty until a result is selected. By clicking on the result, the selected result is highlighted and the data is loaded into the right side of the split panel.

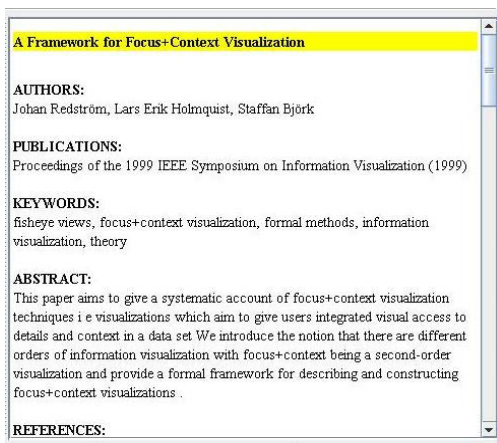


Image 2: Image result pane, shows when paper is selected from the left center pane, shown in Image 3.

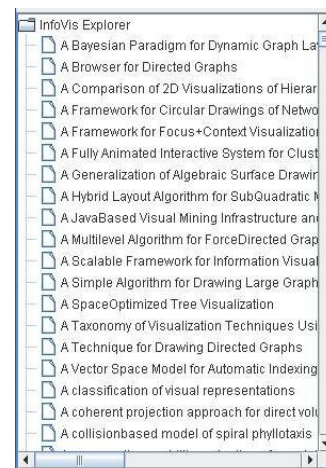


Image 3: Left Side of Center Pane – Shows available papers based on the filtered criteria selected in Image 1.

The result showing on the right portion of the screen is ordered in a consistent manner for each record. If the display is not large enough to handle the complete dataset, a scroll bar will appear on the right hand of the pane, allowing the user to see more information (if desired).

Affordance

The visualization should be easy to use, and lend itself to be understood without explanation. We want to have the control and use of the application flow from left to right, top to bottom. The purpose of any control should be obvious and any action of that control should be immediate.

InfoVis Explorer addresses this concern by ensuring that software design patterns are followed.

The user starts at the top of the screen with four filters and a reset button. Each filter is labeled as to its use (Keywords, Authors, Year, and Publication) and in order from most useful to least. When each filter is selected, it takes effect

Once a paper is selected and shown in the right panel, the row of buttons at the bottom of the screen is enabled, turning from grey to black. Each of these buttons causes a search for related items to be performed, based on the content of the selected result.



Image 4: These buttons activate when a paper is selected, allowing for the user to find related papers based on either keywords from the paper, authors of the paper, years the paper was published, or all publications that paper is in.

Selection

This dataset is rather large, and the visualization should assist the user in narrowing the search criteria to one which produces a reduced set of desirable papers to select from.

InfoVis Explorer uses filters to limit the search criteria, and “Related Item” searches to expand the selection space.

The resulting set of records will always show in the same result panel on the left hand side of the screen.

Interaction and Exploration

Through the use of these selection controls, the user will be able to interact with the database dynamically, to locate the desired information and expand to find related articles. The user must also be free to explore beyond the data of one paper to find related articles, or to directly seek more information than is available using external sources.

Each selection or de-selection of an object in InfoVis Explorer causes a dynamic query to be generated, to refresh the results window. This ensures that the user does not have to think about what to push next to explore the data further.

A unique feature of this visualization is that it allows for external access to trusted databases, including CiteSeer, IEEE, and the ACM. This can be used to fill in data that may be missing from our dataset, discover who has referenced the paper we are looking at, and much more. It also links to DogPile, an internet search engine, so that the user can obtain unbiased comments on the article in question from outside sources.



Image 5: Icons at the bottom of each result that the user can select to cause automatic search results to appear for the selected internet site.

Through this feature, the user can even obtain an image of the paper from CiteSeer (when available).

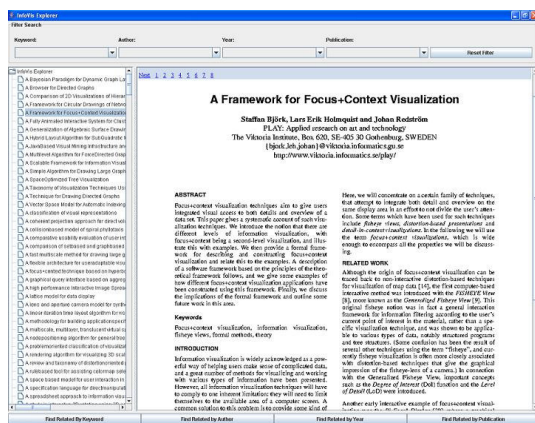


Image 6: Copy of Selected Paper through InfoVis Explorer

DISCUSSION OF VISUALIZATION CHOICES

Textual Representation vs. Graphical Representation

In [2], Spence states the following in regards to Document Visualization: “We are concerned primarily with the task of gaining insight into information which exists in the form of text... without reading those documents.” While we considered some of the graphical representations which have been discussed in several other works [3], we felt that they were not obvious in their usage and that they hid some of the real detail which we sought to see immediately.

An example of what we had hoped to avoid can best be seen in VisDB. This type of visualization does not give any of the detail that can be easily used by a researcher (or someone else from the target audience) to determine if it is indeed something that would be useful to them. Instead, we wanted to show the title and abstract immediately so that the user could determine with speed if the result was close to what they were looking for.

While some of the other visualization techniques for showing the results visually were interesting, such as showing several images of publications to choose from, we did not believe that they would add value to the user. However, we did consider for a time creating several parent nodes for Author, Title, Keyword, and Year so that the user could drill down through the articles in a similar manner. We ruled this out because the filters at the top of the screen simplified this process and reduced the possibility of confusion among users who may not realize the differences.

Command Line Query vs. Dynamic Query

At first, we had thought that a Command line query would be ideal for users, providing them with the ability to dig through the data themselves and return only those sets which they requested. In looking into it further though, it would require too much prior knowledge and training on the users’ part, require a large amount of error handling, added delay, and reduce cognitive understanding. As such, it would not achieve our project goals. Also, we found that any of the tasks that most of the users would ask to perform could be achieved with the dynamic queries. Also, since the data is stored in SQL format, advanced users can bypass our visualization and query the data directly (if they chose to do so).

Document Visualization vs. Information Retrieval

In [2], Spence states that “Document visualization is not information retrieval”. He makes a distinct point that a Document Visualization does not “retrieve” anything for the user; it only helps the user identify relevant papers. We did not agree fully with him. We felt that he was correct that the visualization should help identify relevant material, but we disagreed that it should not be allowed to retrieve the document for the user.

It appears that Spence made this statement to force the point that the user should be involved in constant refinement of the results, and not have to wait for information to be retrieved by the system *before* refining the results.

We chose to implement a feature that links to outside databases for information beyond what our dataset provided. This allows the user to easily retrieve the document if they so desire, but at the same time it allows the user to constantly refine or expand their search before the retrieval occurs.

Scroll Bars vs. Combo Boxes (Drop-Down Lists)

We discussed the use of scroll bars instead of combo boxes for use with the filters. The number of keywords and authors in our dataset was more than three times the number of papers we had in our system. Using the scrollbar, we would have only seen one item at a time, and this may cause the user to overlook a nearby value. In using the Combo Box (Drop-Down lists), the user can see eight items at a time, and still be able to scroll through the items, as they would have if the scroll bar had been implemented.

An application similar to the “Attribute Explorer” [2] had also been discussed, but the items in our dataset are qualitative and could not have been used effectively to allow a range of variables. Instead, we use the “Related Items” buttons at the bottom of the screen to expand the search from the selected paper.

Future Work

The visualization currently has a performance issue. The performance seems to be with the SQL database. Currently the connection is closed after a query is done so that if it was used as a web-based service, bandwidth and the number of users would not be an issue. This could be changed if it is only to be used as a desktop application.

Summary

We believe that we have created a useful tool, which allows for a user to query a large number of papers on Information Visualization and produce valuable results. Because this is based on an SQL database, it will work with any set of data without change. The application is incredibly useful to be able to tell quickly from a title or abstract if a given paper is relevant to the user, perhaps more-so than many of the graphical displays we have considered throughout the semester.

REFERENCES

1. Fekete, J.-D., Grinstein, G., Plaisant, C., IEEE InfoVis 2004 Contest, the history of InfoVis, www.cs.umd.edu/hcil/iv04contest
2. R. Spence, *Information Visualization* (Essex, England: ACM Press, 2001).
3. S. Card, J. Mackinlay & B. Shneiderman, *Readings in Information Visualization: Using Vision to Think* (San Francisco, CA: Academic Press, 1999).