EXPLORATORY VISUALIZATION OF GRAPHS BASED ON COMMUNITY STRUCTURE

by

Yujie Liu

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2013

Approved by:

____
Dr. Jing Yang

____
Dr. William Ribarsky

____
Dr. Zachary Wartell

____
Dr. Xintao Wu

____
Dr. Min Jiang

ABSTRACT

YUJIE LIU. Exploratory visualization of graphs based on community structure. (Under the direction of DR. JING YANG)

Communities, also called clusters or modules, are groups of nodes which probably share common properties and/or play similar roles within a graph. They widely exist in real networks such as biological, social, and information networks. Allowing users to interactively browse and explore the community structure, which is essential for understanding complex systems, is a challenging yet important research topic. My work has been focused on visualization approaches to exploring the community structure in graphs based on automatic community detection results.

In this dissertation, we first report a formal user study that investigated the essential influence factors, benefits, and constraints of a community based graph visualization system in a background application of seeking information from text corpora. A general evaluation methodology for exploratory visualization systems has been proposed and practiced. The evaluation methodology integrates detailed cognitive load analysis and users' prior knowledge evaluation with quantitative and qualitative measures, so that in-depth insights can be gained. The study revealed that visual exploration based on the community structure benefits the understanding of real networks. A literature review and a set of interviews were then conducted to learn tasks facing such graph exploration and the state-of-the-arts. This work led to community related graph visualization task taxonomy. Our examination of existing graph visualization systems revealed that a large number of community related graph visual-

ization tasks are poorly supported in existing approaches. To bridge the gap, several novel visualization techniques are proposed. In these approaches, graph topology information is mapped to a multidimensional space where the relationships between the communities and the nodes can be explicitly explored. Several user studies and case studies have been conducted to demonstrate the usefulness of these systems in real-world applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Exploratory Graph Visualization Based on Community Structure

Graph analysis has become crucial to understand features in biological, social, technological, information, and other networks. Exploratory graph visualization (EGV) systems are of great importance in graph analysis, since they stimulate significant cognitive changes through learning, and thus allow users to gain improved understanding to manage large amounts of information [69]. Together with intuitive visualizations and tailored interactions, computational power is greatly leveraged in EGV systems to produce a deep change in the way graph analysis is approached. Although EGV systems have been proved to be powerful in practices, many real-world EGV missions are still in need of effective visual analytics tools. A significant one is graph community structure related analysis tasks.

In many real networks, the distribution of edges among the nodes displays big inhomogeneities. There are high concentrations of edges within special groups of nodes, and low concentrations between these groups. This feature of real networks is called community structure [20]. Communities, also called clusters or modules, are groups of nodes which probably share common properties and/or play similar roles within the graph [17]. The analysis of the relationship between these communities can greatly enhance our understanding of targeted graphs. For example, communities which are the groups of pages dealing with the same or related topics in information graphs can help researchers quickly get a general idea of a large amount of information [73]. Identifying communities of customers with similar interests in a graph depicting the purchase relationships between customers and online retailers can improve the performance of recommendation systems [17], which can enhance the business opportunities by guiding customers through the list of items. The virtual groups in the online social network also can be used to measure actors' prestige or to explore diffusion mechanisms. Obviously, the community structure of real networks has brought significant advances to the understanding of complex systems.

My work has been focused on the design of exploratory graph visualization systems based on graph community structure. To make the reader acquainted with the problem and how the exploratory graph visualizations are used to solve real world problem, in the following, I will introduce one example from Newdle, which is an EGV for large news graphs and was developed from the first project that I participated in our research group [73].

Before getting to the details of the example, I briefly introduce Newdle system. Newdle is an EGV system that targets on large online news corpora. Online news is an important yet overwhelming information source since it provides timely, ambient information to not only the masses, but also business and political policy makers, social scientists, and analysts in other application domains. Newdle stands for News Wordles, since wordles [66] are its major visual metaphor. It is built based on two news graphs: an article graph and a bipartite article and tag graph. The article graph describes the relations between the articles in a news collection. Each article is a node

Figure 1: The topic overview with the article relation threshold 3.

in this graph. There is an undirected, unweighted edge between two articles if and only if the articles share more tags than a given threshold. In Newdle, community structure is built upon the graph clustering result. An article community is considered as a news topic. The semantics of a topic is defined as the most shared tags among the articles in the community. The temporal feature of a topic is defined as the number of articles in this cluster over time. We overlay a line graph on a wordle (see Figure 1) which represents the daily number of articles published on this topic to reveal the temporal trends. In the line graph, the x coordinate represents the time period of the news collection and the height of the line represents the number of articles published on the corresponding news topic. The line graphs in different wordles use the same scale to allow users to compare the topics with regard to their strengths over time.

The relevance between topic A and topic B is the number of edges that connect the nodes in topic A and the nodes in topic B. The distance between article A and article B is the length of the shortest path between them in the article graph. They are calculated upon requests from the interactions. The article-tag graph describes the relations between tags and articles in a news collection. It is a bipartite graph in which an article is connected to each of its tags. Given a tag, its relevant articles are all the articles connecting to it.

Example: Bombing Attempt on a Christmas Day flight to Detroit

Figure 1 shows the overview of Newdle. The dataset used is online news network. I is a tag co-occurrence network with 1,200 nodes (tags) and 7,042 edges (co-occurrence of two tags in at least one article). It conveys the tag co-occurrence information of 1,078 New York Times (NYT) [2] world news articles published from February 22, 2011 to April 25, 2011. It has 36 binary node attributes carrying categorical and temporal information. Its communities reflect the major news events reported in the news corpus. First, we investigate the second to the left most topic in the top row, as it seems to be breaking news according to its line graph. We start by clicking its wordle from the topic overview to open a topic investigation view, which is shown in Figure 2. The topic of interest is in the top row. From the line graph in the wordle, we notice that it burst sometime after 12/17/2009. We drag the scrolling bar of its html

Figure 2: The topic investigation view.

box to examine the earliest news article in this topic. It was published on 12/26/2009. We click the title to access the original news article. It talks about a bombing attempt on a Christmas day flight to Detroit that was prevented by the passengers and crew members. We then examine more articles in this topic. Most of them are the follow-ups of this event and articles discussing security and warning systems and airports and airlines, as suggested by the wordle. After examining the topic itself, we browse

Figure 3: The tag comparison view.

the relevant topics presented under the top row in Figure 2. At this time, all the news communities are ranked based on their relationships to the focus community and the number of stars before each news article represent its distance to the focus community. We find that the topic in the second row discusses Yemen, terrorism, and Al Qaeda and the topic in the third row was about the reactions of Obama Barack and United States to this bombing attempt.

Why was Yemen related to this bombing attempt? To investigate this question, we select the tags Yemen and Abdulmutallab, Umar Farouk and compare the two tags. The latter is the name of a person (we learn it from its color) that appears in both topics. In the tag comparison view shown in Figure 3, several articles with both tags pop up. By reading the news with the title Yemen Says Bomb Suspect Met With Qaeda Figures, we learn that Abdulmutallab, Umar Farouk is the person who made the bombing attempt and he met with operatives of Al Qaeda in Yemen before setting out on his journey.

Through the above example, it can be seen that the community structures built inside Newdle greatly benefit users' exploration process, especially when the information is huge. More such visualizations based on graph community structure should be developed for graph exploration.

1.2 Itinerary for My Research

My research of exploratory graph visualization (EGV) started with a task of evaluating Newdle that was assigned to me after its development. The motivation for the evaluation is simple: understand how Newdle works in user exploration. During the evaluation preparation, I gained deeper understanding of EGV system and found out that there are several challenging yet important issues for community structure based EGV system design:

(1) The lack of evaluation methods of exploratory systems. Despite the popularity of EGV systems, researchers are still trying to understand the nature of users' exploratory process through EGV systems. The design space of EGV systems is still

unclear. Therefore, evaluation becomes critical throughout the development process of EGV system. Effective evaluations can help EGV system designers learn whether the interactions between an EGV system and the users promote the desired cognitive changes for a good exploration experience. This can also lead to a better understanding of how the system works and its design space. However, existing visualization evaluation methods largely ignore the internal cognitive activities of the users. A systematic approach to measuring and analyzing the cognitive process of users during exploratory visual analysis is desired for EGV system evaluations.

(2) The lack of effective visualization and interaction techniques for community structure based graph visualization. Most graph visualizations do not offer enough support for dealing with community based tasks. Specifically, several essential features are missing in most existing graph visualization systems. First, many real world tasks require simultaneous examination of lots of information, such as the labels and attributes of nodes connecting multiple communities. Such detailed information, when displayed on a large number of nodes, will clutter most graph visualization tools. A less cluttered and readable display is needed for community structure based graph visualization. Second, conducting the exploratory tasks also requires examining details within context. For example, users may want to examine how nodes with desired attributes are distributed among communities. Although it is not difficult to provide contextual information for one or two nodes, most existing visualization systems do not allow users to examine a large number of nodes within context simultaneously. Third, interactive graph exploration often requires selections and other manipulations on multiple nodes concurrently. Criteria for such selections are often

a combination of structural features and node attributes. Although a few approaches have been proposed [43], [65], [58], their functionality and flexibility are limited. New interaction techniques for complex selection criteria are desired.

(3) The lack of task taxonomy of community related graph visualization tasks. Existing efforts on visualization for exploring large graphs based on the community structure is not systematic. First, community related tasks are largely ignored in existing graph visualization task taxonomies. For example, Lee et al. [41] nicely summarize graph visualization tasks related to nodes, links, paths, and attributes, but community related tasks are merely mentioned as the task of "identifying clusters". Second, existing graph visualization systems, even those based on communities, provide very limited visualizations and interactions to conduct community related tasks. For example, one recent EGV system [72] evaluation revealed that existing approaches have poor time efficiency and correctness on very simple community related tasks such as finding bridging nodes of two disconnected communities. The above drawbacks greatly hinder existing graph visualization approaches from advancing understanding of large networks and their community structures.

These three issues draw the itinerary of my research. First, I conducted a formal user study on Newdle, both for a general methodology of measuring and analyzing the cognitive process in EGV systems and for a deeper understanding of the effectiveness of EGV systems [42]. The evaluation results were exciting. They showed the benefits and constraints of EGV systems in different information seeking tasks, as well as leveraging points in EGV system design.

Then, triggered by the evaluation, I worked on several EGV systems for graph exploration. One important finding from the Newdle evaluation is that semantic information plays an important role in users' exploration. Considering that the most common problem of large graph visualization is the clutter caused by lots of edges between nodes, one question raised: what if we abandon the links and highlight the semantic information? This idea leads to a novel visual analytics tool named PIWI which focuses on community analysis [72]. PIWI abandons the cluttered Node-Link Diagrams (NLDs) and employs uncluttered visual representations with node semantic information highlighted to support effective and efficient visual exploration of large graphs. Each community is assigned a color and represented by a row consisting of a tag cloud and a set of vertex plots. A vertex (node) belonging to the community is represented by its label in the tag cloud and dots in the vertex plots. The label and dots are displayed in the color assigned to the community. Vertex plots are used to display the neighborhood information of the communities. A set of case studies and preliminary user studies that conducted with real graphs containing thousands of nodes provide supportive evidence about the usefulness of PIWI in community related tasks.

The success of PIWI encouraged me to explore more about community analysis in large graphs. My next step was to generate a community related graph visualization task list to complete existing graph visualization taxonomies as well as the challenges facing existing graph visualization approaches in conducting these tasks. In the list, for graph objects, we consider nodes, links and community. Because paths can be considered as a special case of several links, we consider them as links. Similarly, we consider sub-graphs, connected components, clusters and groups as community, since they all can be viewed as a community of nodes with certain attributes. The task list is categorized by interactions between these graph objects and communities. Also, considering dynamic graphs, time is a special attribute. Thus, we separate it as another category.

After finalizing the community related task taxonomy, I focused on novel graph visualization techniques using multidimensional visualization. It is specifically designed to support the community related tasks that are difficult to address with existing approaches. The main idea of multidimensional graph visualization techniques is to analogize community based graph visualization to multidimensional visualization. In particular, the communities are analogized to dimensions in a multidimensional dataset, and the nodes are analogized to data items in a multidimensional dataset. Thus, the distance from a vertex to a community can be expressed by the value of a data item on a dimension, and the correlation between two communities is mapped to correlation between two dimensions. In addition, vertex attributes are also mapped to dimensions and thus the correlation between an attribute and a community and the correlation between two attributes are also mapped to dimension relationships. Therefore, existing multidimensional visualization and interaction techniques can be customized to support community related tasks, where communities, nodes, distances, attributes, and the relationships among them are explicitly conveyed and interactively explored.

The following chapters of this dissertation are organized as follows: Chapter 2 discusses the related work. Chapter 3 to 6 present the four parts of my research. Chapter 7 concludes with a summary and contributions of this dissertation as well as potential directions for future research.

CHAPTER 2: RELATED WORK

2.1 Exploratory Systems and their Evaluation

Lots of exploratory systems have been developed to assist user information seeking. For example, INSPIRE [70] projects documents onto a 2D space so that clusters of documents form galaxies or mountains. The most significant keywords of the clusters are displayed as labels of the galaxies or mountains. PaperLens [40] groups papers by their topics. Each group is displayed in a time histogram and the common topic is displayed as a label. Newsmap (http://newsmap.jp) presents news clusters generated by Google News Aggregator in a treemap style visualization. Each cluster is represented by a rectangle in the treemap. The size of the rectangle indicates the number of documents in the cluster. The title of a representative news article in the cluster is displayed in the rectangle. Google News Timeline (http://newstimeline.googlelabs.com) allows users to view news clusters on a zoomable, graphical timeline. The time stamp, title, abstract, and sometimes a figure thumbnail of a representative news article are displayed for each cluster.

To assess the performance of these exploratory systems, evaluations have been conducted. For example, INSPIRE has been tested by a set of analysts [70]. Their reports showed that INSPIRE triggered creative thinking and justified the conviction that text visualizations had to make use of the cognitive and visual process. A formal user study was conducted on PaperLens [40]. It was focused on the usability of the system using efficiency and accuracy measures. Sixteen tasks were used to test how PaperLens helped the subjects investigate research topics and their trends over time. Results showed that PaperLens assisted users to explore the data with less user effort. Pirolli et al. [52] conducted a user study to compare a clustering-based exploratory system with a simple keyword-based search. The result suggested that the clustering-based exploratory induced a more coherent conceptual image of the text collection and a richer vocabulary for constructing search queries. Zamir et al. [75] reported an empirical comparison between a standard ranked-list representation and a clustering-based representation for log files. Their study showed that the clustering-based representation influenced the number of documents the subjects read, the amount of time they spent, and their click distance. In sum, the quantitative measures in these evaluations showed the effectiveness of the systems.

As information overload continues to grow, lots of interactions between human and system are developed to assist users' sense making. In order to better understand the design space, user-centered evaluation methods are widely employed. User-centered evaluation uses both quantitative measures, such as task correctness and completion time, and qualitative measures, such as in-depth open-ended questionnaires and user comments [56]. For example, Hearst et al. [24] studied a search interface where open-ended questionnaires were used together with quantitative measures. They successfully learned the usefulness of each feature of the interface and identified leverage points where improvements could be made. Kules et al. [37] studied the change of user tactics when a categorized overview became available. They collected user comments for qualitative analysis besides recording quantitative measures. Consequently, they got a whole picture of how the categorized overview worked in shaping user tactics and a set of guidelines for exploratory search interfaces. User centered design greatly enhanced the evaluation design in these systems.

However, as pointed out in the survey paper [28],information visualizations rarely apply cognitive science and human factors other than some of the gestalt principles. Cognitive aspects are undoubtedly a subject for future research. Until now, very few usability studies that include knowledges from cognitive science have been done. Kammerer et al. [34] conducted a user study to evaluate the effectiveness of a new design of exploratory systems. In the experiment, they evaluated the cognitive load of the subjects based on the NASA work load index [22]. Anderson et al. [7] measured brain activities using electroencephalography (EEG) to study cognitive load when comparing multiple visualization techniques. Their results showed that cognitive load measures extracted from EEG data can be used to quantitatively evaluate the effectiveness of visualizations. Kang et al. [6] used NASA TLX survey to measure user cognitive load in a comparative study between single and multiple monitors.

These evaluations mentioned above provide solid knowledge background for my research. My work goes beyond the existing work by distinguishing extraneous cognitive load and germane cognitive load through integrated analysis of cognitive load measures and rich qualitative and quantitative measures.

2.2 Graph Visualizations in Practice

2.2.1 Node-Link Diagrams

Most existing graph visualization techniques are Node-Link Diagrams (NLDs) or Adjacency Matrix Representations (AMRs). NLDs usually place graph nodes on a 2D or 3D layout area and explicitly draw the edges among them as links on the display. NLD visualization systems, such as GraphVis [16] and NodeXL [58], can efficiently draw a graph with thousands of nodes in a few seconds. Recently, efforts such as visualizations based on the community structure of graphs, progressive visual exploration, and novel interaction techniques have been made on traditional NLD visualizations.

Community detection has been employed in NLDs to reduce clutter and enable the discovery of community related insights. Vizster [25] visually highlights communities in an NLD and allows users to interactively examine clustering results at different granularities. SocialAction [49] allows users to aggregate networks based on their community structure and examine communities of interest in detail. Itoh et al. [31] hierarchically cluster multiple-category graphs based on the categories and connectivity of nodes. They use a hybrid space-filling and force-directed layout to visually reveal communities and their relationships. NodeXL [58] provides an NLD layout style where communities are displayed within boxes so that the connections between the communities can be visually explored.

The progressive visual exploration strategy has been practiced by several recent graph visualization systems. Jigsaw [59] allows users to identify nodes of interest from a set of coordinated views and progressively examine their neighborhoods in an NLD. Van Ham and Perer [63] propose to "search, show context, expand on demand" when exploring large graphs. Their approach allows users to browse the immediate context graph around a specified node of interest.

A handful of interaction techniques have been integrated into NLDs. McGuffin and Jurisica [43] allow users to select a node's neighborhood by dragging out its radius using a radial menu besides the traditional rectangle and lasso selection. NodeXL [58] allows users to select multiple nodes using a rectangle. It also supports union and difference of several selections. Viau et al. [65] propose FlowVizMenu and Parallel Scatterplot Matrix, where node metrics are displayed and interactive selections based on these metrics are conducted.

However, none of the existing approaches provide enough visualization and interaction support to community related analysis tasks. Clutter is still one of the biggest problem of NLD graph visualizations. EGV systems which target at various community analysis tasks and progressive visual exploration are needed.

2.2.2 Adjacency Matrix Representations

AMRs are visual representations of the adjacency matrices of graphs. Bertin [10] first introduces visual matrices to represent networks. Matrix Zoom [5] displays large graphs as a hierarchy of adjacency matrices given a predefined clustering hierarchy using edge aggregation. MatrixExplorer [27] coordinates AMR and NLD views and supports matrix reordering, interactive filtering, and clustering. NodeTrix [26] is a hybrid visualization with NLDs showing the global structure of a network and AMRs showing communities. Ghoniem et al. [16] show that AMRs outperform NLDs for large graphs or dense graphs in several low-level reading tasks, but not in path-related tasks.

However, large graphs usually have rich attributes which come with graph node. The visualization of such rich attribute information becomes a problem in AMRs. Also, AMRs are not as intuitive as NLD to users. Tailored interactions are needed to make them understandable. On the other hand, AMRs reduce clutter by avoiding using links. This reminds us the possibility of abandoning the links in graph visualization.

2.2.3 Multivariate Graph Visualization

Many efforts have been made to visualize graphs with multidimensional node attributes. PivotGraph [68] aggregates the graph based on node attributes and uses a grid-based approach to explore the relationship between node attributes and connections. OntoVis [55] uses the ontology associated with a semantic network to conduct structural abstraction and importance filtering for manageable network visual analysis. Pretorius and van Wijk [53] use interactive attribute-based clustering for visualizing large state transition graphs. Semantic substrates position nodes in non-overlapping regions based on their attributes [55], [8]. Dynamic query sliders and filtering buttons are provided for interactive node selection and filtering. GraphDice [11] uses multidimensional visualizations to facilitate multivariate graph visualization.

Another possible approach is using multidimensional visualization techniques, which is still under its initial development. For example, hierarchical Parallel Coordinate [19] visually presents clusters of multidimensional data items using a band and a mean without displaying individual data items in them. In my research, I follow this path and try to make more use of powerful multidimensional visualization techniques.

CHAPTER 3: EXPLORATORY GRAPH VISUALIZATION EVALUATION

As information overload continues to grow, there is a dire need for information seeking systems to support iterative, opportunistic, and evolving information foraging and decision making. Such exploratory analysis stimulates significant cognitive changes through learning, and thus allows users to gain improved understanding to manage large amounts of information [69]. Targeting this need, many exploratory graph visualization (EGV) systems have been developed to support users in conducting exploratory analysis with the aid of visualization techniques.

Despite the popularity of EGV, researchers are still trying to understand the nature of users' exploratory process through EGV system. The design space of EGV system is still unclear. Therefore, evaluation becomes critical throughout the development process of EGV system. Effective evaluations can help EGV designers learn whether the interactions between EGV systems and users promote the desired learning process for a good exploration experience, which can lead to a better understanding of how the system works and its design space. However, existing visualization evaluation methods largely ignore the internal cognitive activities of the users. A systematic approach to measuring and analyzing the cognitive process of users during exploratory visual analysis is desired for EGV system evaluations.

Specifically, I conducted a formal user study on the news graph visualization system developed by our research group, Newdle [73]. It is both for a general methodology of EGV system evaluation and for a deeper understanding of the design space for EGV system based on graph community structure. Towards a general methodology, we sought practical methods for measuring and analyzing the cognitive process in EGV system evaluations. The study also explored the benefits and constraints of community structure based EGV system in different information seeking tasks, as well as leveraging points in EGV system design. In the following parts, I will discuss several major challenges in evaluating EGV systems and propose a set of approaches to address these challenges. The theoretical foundation underlying our approaches is introduced and our practice of these approaches in the user study of graph community structure based EGV systems is presented as examples to illustrate them.

3.1 EGV System Evaluation: Challenges and Approaches

3.1.1 Understanding the Cognitive Process

When interacting with EGV systems, users are engaged in a sensemaking process to bridge a knowledge gap that prevents them from accomplishing the tasks [60]. EGV system evaluation needs to assess how EGV systems assist users in acquiring new knowledge through learning so that the gap in the cognitive process is bridged. To achieve this goal, Cognitive Load Theory [60], which consists of rich cognitive process models, procedures, and instructions, provides a solid theoretical foundation for EGV system evaluation design.

According to cognitive load theory, two parts work together in a user's cognition during an iterative exploration process: limited working memory and comparatively unlimited long term memory. The working memory is where important learning process happens, while the long term memory is where users' knowledge lies, including all the existing schemas. When new information is introduced, it is learned in the working memory to extract schemas for filling the knowledge gap. The schemas are then transmitted to the long term memory and saved. During this process, cognitive load is generated. Sweller [60] distinguished three types of cognitive load according to their sources:

(1) Intrinsic cognitive load. It is caused by the structure and complexity of the material being learned and cannot be influenced by system designers. Intrinsic cognitive load can only be reduced when needed schemas already exist in the long term memory.

(2) Extraneous cognitive load. This load is induced by system designs without sufficient consideration on the structure of information and the cognitive process. It is an overhead that interferes with the understanding of materials.

(3) Germane cognitive load. It represents users' efforts to process and comprehend the materials. It is devoted to schema acquisition and automation and thus enhances learning.

Understanding the above three types of cognitive load is essential in EGV system evaluations. First, users' prior knowledge in the long term memory affects the intrinsic cognitive load and thus user variability needs to be carefully controlled in an EGV system evaluation. Second, both extraneous cognitive load and germane cognitive load are imposed by the system design and vary from system to system. The indications of an effective EGV system are *low extraneous cognitive load* and *high germane* *cognitive load*, since the former hinders learning while the latter enhances learning. Therefore, it is critical to distinguish these two types of cognitive load during an EGV system evaluation. My evaluation methodology follows the above guidelines.

3.1.2 Conducting EGV System Evaluation in Laboratory Settings

The ideal evaluation approach for EGV system is longitudinal and in a naturalistic setting, because EGV systems are often used in the context that is open-ended, progressive and iterative. However, it is often costly and not efficient enough for system development, especially for the initial step of system design. In this section, I focus on user-centered evaluation that combines controlled lab experiments with questionnaires and interviews. In particular, I discuss how to design tasks, control user variability, and motivate subjects in laboratory settings.

3.1.2.1 Task Design

To reconstruct the multi-faceted exploration process in the laboratory setting, I suggest using existing task taxonomies to guide the task design, since they well summarize users' activities in certain domains.

Practice: The goal of our user study was to better understand how community structure based EGV system assist users in an information seeking process. The tasks used in this study were designed based on a task taxonomy of high level web information seeking [36]. It can be easily extended to cover a large portion of information seeking tasks for other kinds of document collections. In this taxonomy [36], there are five categories of information seeking tasks, namely "browsing", "fact finding", "information gathering", "transactions", and "others". I adapted the taxonomy and applied it into an online news collection information seeking scenario. In particular, I excluded "transaction", which happens in E-commerce, since there is no transaction in news data. I chose "Revisit" from the "others" category, since it happens frequently when readers want to retrieve what they met before. In the user study, one task for each category was used, namely "browsing" (Task 1), "fact finding" (Task 2), "information gathering" (Task 3), and "revisit" (Task 4). Detailed descriptions of the tasks are provided in Table 1.

Table 1: Four information seeking tasks.

Category	Description [36]	Task
Browsing	A serendipitous task where you may visit the data with no specific goal in mind.	Find as many distinct topics from the dataset as possible. Describe each of them using a few sentences.
Fact Finding	A task in which you are looking for specific facts or pieces of information.	Find as many articles as possible about humanitarian aid in Haiti earthquake. Save the links of each article.
Information Gathering	A task that involves the collection of information, often from multiple sources. Unlike fact finding, you do not always know when you have completed the task and there is no specific answer.	Summarize the activity of President Obama in Human Health Insurance.
Revisit	A task happens when you need to revisit some source that you previous met.	List as many keywords as possible that can be used to retrieve articles in the previous task.

3.1.2.2 User Variability Control

Comparing multiple systems is a common practice in laboratory evaluations. To do so, either within-subject evaluation or between-subject evaluation can be used. Here I focus on between-subject evaluation. It avoids the learning effects since each subject only uses one system. However, it is influenced by the individual differences of the subjects, such as their prior knowledge, mental models, and demographic profiles [33]. Thus, user groups should be balanced. In the following, I will discuss user variability control on prior knowledge, mental model, and writing skill.

Prior knowledge

Prior knowledge can affect subjects' performance since schemas in the long term memory can reduce intrinsic cognitive load. Unlike demographic profiles which can be collected using surveys, prior knowledge is not that easy to access. We design the prior knowledge test by bringing ideas from the Education domain. In Education practice, teachers gauge students' prior knowledge when they enter a course or a program. There are several ways to evaluate student prior knowledge, such as prior knowledge self-assessments, concept maps, and concept tests [1]. We use the self-assessments since they are easy to conduct and score and proved effective in in previous research [34].

In a self-assessment, a subject is asked to reflect and rate her/his level of knowledge and skill. A potential issue is that the subject may not be able to accurately assess their knowledge. However, accuracy can be improved when the questions clearly differentiate levels of knowledge. Identifying concepts and techniques that are needed in the exploration process can be of great help in generating effective questions. For concepts, we suggest five levels: never heard of → have heard of → have some idea → have a clear idea → can explain it. Also, we suggest five levels for techniques: never used it → tried using it but no result → can do simple interactions → can manipulate multiple functions → can easily use it and build results.

Practice: A between-subject design was used in our study. In particular, 36 subjects were assigned to four groups; one for each test bed. Each group had 9 subjects. The subjects had a prior knowledge test before the evaluation was conducted. The results were used to balance the groups. The prior knowledge test included 20 questions on concepts and 4 questions on techniques. The 20 concept questions were about the five most significant news stories in the new data (including the task-related ones), 4 for each. The questions included general ones, such as "How would you rate your knowledge regarding Haiti Earthquake?" and specific ones, such as "How would you rate your knowledge regarding humanitarian aid in Haiti Earthquake?" The technique questions evaluated the subjects' prior knowledge on computer usage, browsing and searching experience, database system usage, and information visualization system experience. For example, the question on "information visualization system" was "How familiar are you with information visualization systems?". All the questions were answered using the five level scales as described above. The test was scored 1 - 5 (5 for highly knowledgeable). The subjects were sorted by their test scores from high to low, and assigned to four groups using a "Z" style. For example, the subjects ranked 1-12 were assigned to the groups as follows: Group 1: subjects ranked 1st, 8th, and 9th; Group 2: subjects ranked 2st, 7th, 10th; Group 3: subjects ranked 3st, 6th, and 11th; Group 4: subjects ranked 4th, 5th, and 12th. After the initial assignment, the average scores for each group were calculated and minor exchanges were made to balance the groups.

The correlation between prior knowledge and user performance was analyzed to learn whether the community structure based EGV system compensated prior knowledge shortage of novice subjects.

Mental models

Mental models are the "psychological representations that aid in understanding, explaining, or predicting how a system works" [32]. A matching mental model can greatly enhance a user's experience with a system while a mis-matching mental model can bring unnecessary mental barrier into a user's exploratory process. Although it is interesting to study how the different mental models affect the effectiveness of an EGV system, the effects of different mental models need to be controlled to reduce their influence on intrinsic cognitive load when they are not the focus of a study. To do so, the evaluation designers need to: (1) make sure that subjects understand how the system works; and (2) minimize the differences in the mental models that the subjects have already built. We suggest conducting mental model control in the training process before the formal experiment, which is effective with low cost. Conceptual description and system practice [23], which have been proved effective in mental model building, can be employed.

Practice: In our study, each subject had a training session before the formal testing. In the training session, an instructor first introduced the system to the subject (conceptual description), including how the system processed the raw data, how the outcome should be interpreted, and how the interactions were processed by the system. Then, a training dataset was given to the subject. The subject was asked to freely interact with the system (system practice) and encouraged to talk to the instructor about each step she took, such as her thoughts about the interface, the aim of the interaction, and her prediction of the behavior of the system reacting to her interaction. In this way, the instructor observed how the subject learned the system and corrected her when her understanding of the system was inaccurate. Comments were recorded by the instructor for system improvement. When the subject felt ready for the formal testing, she was asked to describe the working process of the system and how the system responds to different interactions (self conceptual description). Again, misunderstandings were clarified when needed. The goal of such "exit description" was to make sure that the subject's mental model matched the system to be tested.

Writing skills

In EGV system evaluations, the subjects are often required to summarize or describe information they collected. The writing skills of the subjects vary. The differences can affect the result assessment and thus need to be controlled since they are irrelevant to the system design. However, assessing the subjects' writing skills is difficult and assigning groups based on the writing skills further increases the complexity of group construction. We propose a practical alternative: when assessing results whose qualities are affected by writing skills, ask the judgers to explicitly rate the quality of writing besides other metrics. In this way, the influence of individual differences on writing skills can be separated from the influence of the system.

Practice: The information gathering task required the subjects to summarize the information they collected. We evaluated the results based on the information quality standards of Wikipedia (http://en.wikipedia.org/wiki/Information_quality). To simplify the evaluation process, three measures were generated from these standards. They are accuracy, completeness, and quality of writing. The accuracy measure indicated whether the information was true or not. The completeness measure indicated whether a summary covered all aspects of the whole story. The quality of writing measure captured all other metrics which depend on individual's writing skills.

3.1.3 Measurements

3.1.3.1 Time

Subjects' completion time when conducting a task is often used to measure system performance. For exploratory tasks, completion time can be affected by factors other than the effectiveness of the EVS being evaluated. For example, curious subjects may spend a significant amount of time to read the raw data during the experiment. The evaluation designer should either limit such activities or exclude the time for such activities from result analysis.

Practice: A measure used in our study was *immediate completion time*. It recorded the time the subjects first found information that was useful for them. It was obtained either by self-reporting during the exploration or retrospectively by examining relevant screen captures.

Sometimes capturing immediate completion time is not practical for open-ended exploratory tasks. Also, the subjects are often overly thorough in the test situation [33]. In these cases, posing time limitations to the tasks is a good alternative. To reduce the possible stress caused by time limitation, M. Kaki and et al. [33] suggest using instructions such as "whatever can be found in the given time is acceptable". In addition, suitable time limitation getting from transaction log analysis or average task time from pilot study can make the subjects' behavior closer to the real behavior. Furthermore, we suggest allowing the subjects to exit open-ended tasks, such as browsing, at anytime they want. Thus the time the subjects spend on the tasks, named *exploration time*, can be analyzed together with the subjects' comments to further learn about user engagement or frustration. Along with the time limitation, *interactive precision and recall*, which are proposed by Veerasamy et al. [64], can be used. Interactive recall measures the proportion of relevant documents in the result that were used by the subjects whereas interactive precision states the proportion of relevant documents in the documents that the subjects made use of.

3.1.3.2 Measuring extraneous and germane cognitive load

As we discussed in Section 3.1.1, low extraneous cognitive load and high germane cognitive load indicate effective EGV systems. Therefore, it is important to distinguish these two types of cognitive load. There are lots of direct or indirect cognitive load measurements in cognitive load theory, such as the NASA task load index questionnaire [22] and direct measurement “dual-task approach” [12]. A good description of existing cognitive load measurements can be found in [47]. However, most measures, such as the cognitive load factors on mental demand and physical demand, do not differentiate extraneous and germane cognitive load. In other words, a high cognitive load measure can be either caused by high extraneous cognitive load or high germane cognitive load. Therefore, in EGV system evaluations, it is important to analyze cognitive load measures together with the subjects’ comments and other qualitative and quantitative measures to distinguish extraneous and germane cognitive load.

The comments are obtained from open questions, post questionnaires, and interviews. To reduce the complexity of cognitive load analysis, we suggest classifying comments into three categories, namely “Engagement”, “Neutral” and “Frustration”.

"Engagement" comments express excitements and encouragements when interacting with the system or describe how the subjects are motivated to explore more information and put more efforts into the exploration process. Examples are comments such as "The keyword murder caught my eyes. Okay, let's see what happened here" and "The information is well organized and already there. I would like to see more about what it will get." "Neutral" comments describe how the subjects deal with the task, such as the comment "I put in the keywords and search for what I need." "Frustration" comments report unexpected situations the subjects experience and difficulties they meet. Examples are comments such as "I need to go through each news article. I do not see a place to highlight what I want. Frustrating" and "Associating between different news articles costs me lots of time."

The high values of certain cognitive load measures, such as the mental demand factor, indicate high germane cognitive load when they appear together with comments whose majority are "Engagement" comments. On the contrary, their high values indicate high extraneous cognitive load when they appear together with comments whose majority are " Frustration" comments.

Practice: In our study, we measured the subjects' cognitive load after they conducted each task. We used a modified version of the NASA task load index questionnaire [22], which was designed to identify the cognitive variations in subjective workload within and between different types of tasks. Six workload-related factors, namely mental demand, physical demand, temporal demand, performance, effort, and frustration level, were measured using 5-point scale questionnaires to derive a sensitive and reliable estimation of workload. The detailed information is shown in

Table 2: Six cognitive factors. Created by Hart and Staveland [22].

Factors	End Points of Scale (1/5)	Description
Mental Demand (MD)	Low / High	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand (PD)	Low / High	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
Temporal Demand (TD)	Low / High	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
Performance (OP)	Good / Poor	How successful do you think you were in accomplishing the goals of the task set by experimenter? How satisfied were you with your performance in accomplishing these goals?
Effort (EF)	Low / High	How hard did you have to work (mentally and physically) to accomplish your level of performance?
Frustration Level (FR)	Low / High	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Table 2.

User comments were collected in two ways. First, after each task, we asked the subjects to write down the challenges they met in the task and how they solved them. Second, we interviewed the subjects after the study with the screen capture available to them. We specifically asked what they thought about the system and the tasks, as well as the reasons for their preference and cognitive load rating. The oral feedback from the subjects were recorded on paper. Two reviewers worked together to go through all the feedback and mark useful sentences, namely comments. In this study, 576 comments were marked. Each of the two reviewers processed comments from 24 subjects (there were 36 subjects in total) respectively, and thus 190 comments from 12 subjects were classified by both of them. Among these 190 comments, 90% of the classification results agreed by the two reviewers. The two reviewers discussed the categories of the comments with different classification results, and then refined the classification results according to the discussion.

3.2 The User Study

A formal user study was conducted on a community structure based EGV system, using the EGV system evaluation methodology discussed in previous section. Through this study, I explored the benefits and constraints of community structure based EGV systems when conducting different information seeking tasks. I also explored important design aspects that can affect the effectiveness of a community structure based EGV system, namely influence factors.

3.2.1 Test Beds

3.2.1.1 Influence Factors

In Newdle, the graph community structure is built upon the graph clustering result. To identify influence factors, we first examine existing systems which are build upon clustering result and identified several features that distinguish them from other visualization approaches. First, one distinguish feature of EGV systems like Newdle is the community structure embedded in the systems . They always groups a large collection of documents into communities by some means. For example, PaperLens [40] groups papers by their topics and Newsmap (http://newsmap.jp) presents news clusters generated by Google News Aggregator. Second, an visual representation for semantic information derived from the communities is needed in almost all EGV systems. For example, labels in INSPIRE reveal the most significant keywords of the clusters [70]. Thus, two influence factors are identified here: community analysis and visual representation of community semantics.

We pose the following questions in this study: do community structure based EGV systems really help users explore large document collections? Does the fact that relevant documents are organized into communities alone lead to the advantages of community structure based EGV systems? How important is the visual representation of community semantics to community structure based EGV system? The answers to the above questions can effectively guide further design space exploration of EGV that bases on community structure .

To answer these questions, we built three test beds teasing out community analysis and visual representation of community semantics. Since the most traditional way of news exploration is web searching, the New York Times (NYT) website was used as a baseline system in the evaluation. We compared the performance of the four test beds using a variety of information seeking tasks.

3.2.1.2 Customized Test Beds

Beside the baseline system, three test beds from Newdle (see Figure 10) [73] were created. Newdle is a EGV system which is based on community structure and created recently for interactive exploration of large news archives. Systems such as NewsMap (http://newsmap.jp) and Google News Timeline (http://newstimeline.googlelabs.com) were not selected since they have a strong bias toward the most recent documents in the collections, which is beyond the scope of this study. Other reasons why we chose Newdle were that there were clear boundaries among the communities in Newdle and that a semantic representation was explicitly provided for each news community. These features made it easier to tease out the community analysis and semantic representation factors in Newdle than in other systems such as INSPIRE [70].

Newdle has the following components:

Clustering Analysis. Newdle clusters New York Times (NYT) (http://www.nytimes.com/) news articles based on their tags. The tags are manually generated by NYT editors and thus have a high quality. A document graph is constructed where two documents with more than three shared tags are connected. Clustering is conducted using leading eigenvector community analysis [45] based on the network structure. We manually inspected the clusters visualized in the user study. The majority of them consisted of closely related documents. Noise (a small number of not so related documents) existed in some clusters. For example, the news cluster about Toyota recall event might accidently contain one article about the spokesperson in its advertisement. However, such noise is small in most clustering algorithms.

Topic Canvases. Newdle pre-attentively represents the most shared tags in a cluster in a rectangular area named "topic canvas", as shown in Figure 10a. These tags provide a high level overview of the semantics of the news articles in this community. Thus, the topic canvas provides visual representation of the community semantics. In a topic canvas, the most shared tags are displayed using Wordle, a tag cloud like visualization that packs a large number of tags with varying font sizes and colors into a small screen space [66]. The colors of the tags represent their categories assigned by NYT editors, such as people, organizations, locations, and topic descriptors. For example, all the location tags are in yellow and all the person tags are in white in Figure 10a. The size of a tag is proportional to the number of articles with it within the cluster. The overlaid line graph on a topic canvas is a time graph revealing the daily number of articles in this cluster.

Document Lists. Besides topic canvases, Newdle also presents snippets of documents within a community using a list, where the documents are ordered by their time stamps in descending order (see Figure 10b). The snippets can be displayed in a detail mode or a compact mode. In the detail mode, the titles, tags, summaries, time stamps, and authors of the documents are displayed. In the compact mode, only titles are displayed. Note that the titles are good indicators of the contents of NYT news articles since they are carefully chosen. Users can switch between the two modes. Clicking on a title leads the users to the full text. The colors of the titles range from blue to white, indicating the age of the documents.

Multiple Views. Newdle provides an overview and a detail view. The overview allows users to browse the major communities in the collection. The detail view allows users to examine communities of interest. To generate the overview, the news communities are sorted by the numbers of documents. Topic canvases of communities are displayed in a grid, as shown in Figure 10a. Users can learn the semantics of the communities at a glance and quickly drill down to communities of interest. After users select communities of interest from the overview, a detail view of the selected communities are generated. In the detail view, the topic canvases and document lists of the selected communities are displayed side by side, as shown in Figure 10b. Users can open a document (a new article) by clicking its title in the document lists.

Interactions. Newdle allows users to conduct in-depth analyses on communities, tags, and documents through a rich set of interactions. From the overview, users can select communities containing a set of tags of interest or communities related to a focus community. The search results are visually present in the detail view. From the detail view, users can open a document and examine its detail from the document list.

Table 3: Testbeds.

	Feature	Interaction
Newdle	1. Clustering. 2. Document list. 3. Topic canvas.	1. Keyword search by clicking a keyword from a topic canvas or typing a keyword from a text entry. 2. Results are grouped into clusters. Each cluster is presented with a document list and a topic canvas.
Clustering -only system	1. Clustering. 2. Document list.	1. Keyword search by typing a keyword from a text entry. 2. Results are grouped into clusters. Each cluster is presented in a document list.
NYT	1. Category. 2. News figures. 3. Document list.	1. Keyword search by typing a keyword from a text entry. 2. Results are ordered by time in a document list. Search keywords are highlighted in the results. 3. Accessing relevant documents through the hyperlinks.
Plain	1. Document list.	1. Keyword search by typing a keyword from a text entry. 2. Results are returned as a document list organized by time.

To summarize, three test beds were derived from Newdle to tease out the visual representation and the community analysis. The first test bed kept the basic features of Newdle, namely the community analysis, the document lists, and the topic canvases. We refer to it as Newdle. The second test bed was derived from the first bed by removing the topic canvas. The most recent news articles are highlighted in the document list as labels for each community (see Figure 10c). We call it the clustering-only system. The third test bed was a plain system without community analysis. All documents were displayed in one document list, where they were ordered by the time stamps in descending order. Figure 10e shows the plain system. In addition, the NYT website was used as a baseline system. Table 3 summarizes the features and interactions of all four test beds used in the study.

By comparing Newdle against the plain system and the NYT website, we evaluated the performance gain of EGV systems based on community structure. By comparing Newdle and the clustering-only system, we examined the performance gain from the visual representation of community semantics. Comparing the clustering only system against the plain system and the NYT website allowed us to assess the performance gain by grouping relevant documents into communities.

3.2.2 Data

The data used in the user study are 3,640 news articles fetched from the New York Times (NYT) online RSS feeds in January 2010. We set constraints in the NYT website using the advance search function to make sure that the subjects using the NYT website accessed the same set of data as the other subjects.

Since online news is a typical text source in huge data volumes, we believe insights from this study can be extended to text exploration in many other domains, such as digital libraries and archived reports.

3.2.3 Subjects

Thirty-six UNCC students (17 male, 19 female) participated in the user study. The ages of the subjects ranged from 20 to 28 years old. Eighteen of them were Computer Science majors. Twelve students were Communication majors. Four were Electronic Engineering students. The other two subjects were a Mathematics major and a Chemistry major. All subjects reported at least one year of computer experience. One week before the study, invitation emails were sent out with an informed consent form and a prior knowledge test. Subjects who wanted to join the study replied to the email with the signed consent form and completed prior knowledge test. The test had a maximum score of 120. There were 5 subjects whose scores were within the range 62∼58, 24 subjects within the range 55∼50, and 5 subjects within the range 50∼46. The subjects were assigned to four groups according to their prior knowledge test scores using the method described in Section 3.1.2.2. Each group worked on one of the four test beds in the user study.

The average prior knowledge test scores for Newdle, Clustering-only, NYT, and plain group were 53.875 (Highest:62, Lowest:46), 53.875 (Highest:61, Lowest:48), 53.5 (Highest:58, Lowest:48), and 53.625 (Highest:58, Lowest:48). The standard deviations of the four groups in the same order were 4.55, 3.91, 3.07 and 3.50.

Twenty-six subjects were native speakers of English; the other ten subjects also spoke and wrote fluent English. The numbers of native speakers in Newdle, clustering-only, NYT, and plain systems were 6, 7, 6, and 7 respectively.

3.2.4 Procedure

The subjects took the study in a laboratory setting one by one. The study lasted about two hours for each subject. Before the study, the instructor explained the goals of the user study to the subject. The background information about extraneous cognitive load and germane cognitive load was explained. The subject was informed to pay attention to the cognitive load she/he experienced during the experiment.

A twenty minute training session was first conducted. The capabilities of the test bed and the upcoming tasks were explained to the subject by an instructor who supervised the experiments of all the subjects. Details of the training session were introduced in Section 3.1.2.2.

The test session followed the training session. The subject was asked to conduct the four information seeking tasks presented in Table 1, one by one using the test bed. 15 minutes, 5 minutes, and 5 minutes were given to the browsing, fact finding, and revisit tasks, respectively. There was no time limit on the information gathering task and the completion time and the immediate completion time were recorded. All the questions were given in a text editor. Subjects' screen activities were recorded. After each task, the subject answered the cognitive load questionnaire (see Table 2) and open-ended questions, such as how they felt about the system and how they rated the cognitive load ratings. This session was concluded with an interview about the user experience with the system and a subjective questionnaire (see Table 8).

3.3 Results

The collected data include (1) prior knowledge test scores; (2) effectiveness measures; (3) cognitive load measures; (4) classified comments; and (5) subjective ratings. They were analyzed together as follows:

1. The correlation between the prior knowledge test scores and the effectiveness measures was calculated to learn whether a test bed can compensate knowledge shortage in novice users. T-test was used in the correlation analysis.

2. The cognitive load measures were analyzed together with the classified comments and effectiveness measures to reveal the strengths and types of the cognitive load experienced by the subjects. We computed the means and standard deviation of the six cognitive load factors for the four tasks and compared the performance of the systems using ANOVA.

In the following, we report the results for each of the four tasks: browsing, fact finding, information gathering, and revisit.

3.3.1 Task 1: Browsing

Task: Find as many distinct topics from the dataset as possible. Describe each of them using a few sentences.

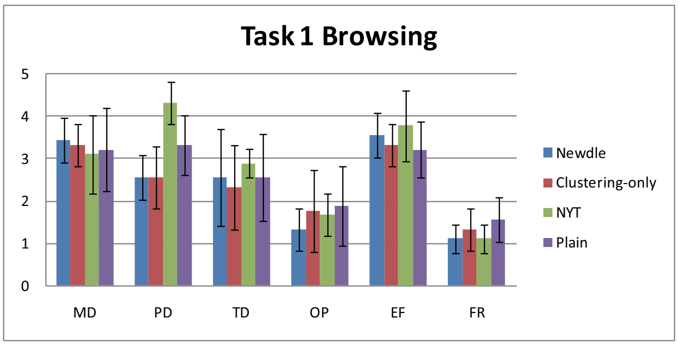

Figure 4: Means of six cognitive load factors' 5-scales rating, Task 1.

From the cognitive load analysis (as shown in Figure 4 and Table 4), we noticed the order of mental demand (MD) is: Newdle > Clustering-only > Plain > NYT ($F = 2.67, p = 0.05$). Newdle users had the highest MD. NYT users had the lowest MD. Clustering-only system users had slightly higher MD than base line system users. In contrast, NYT users had the highest physical demand (PD, $F = 16.5, p = 0.0000012$). Newdle and the clustering-only system have the same lowest PD. The time demand (TD) exhibited a different trend. NYT and Newdel users (NYT > Newle, $F = 0.72, p = 0.4$) had higher TD than other two systems (Plain > Clustering-only, $F = 0.22, p = 0.64$). Newdle users had more confidence than users of the other three systems ($F = 7.2, p = 0.04$). The plain system users had the lowest confidence of their answers. NYT users gave the highest rating of effort (EF, $F = 2.3, p = 0.02$). Newdle users had the second highest EF rating. Surprisingly, the plain system users had the lowest EF rating. The order of frustration level was opposite to EF: NYT <

Newdle < Clustering-only < Plain ($F = 2.17, p = 0.01$).

Table 4: Task 1: Browsing.

	Mean(SD)			
	Newdle	Clustering -Only	NYT	Plain
Effectiveness				
♯ of Topics	5(0.67)	4(0.71)	4(0.60)	3(0.44)
Exploration Time (min)	11.39(3.51)	8.13(1.06)	10.86(2.72)	7.62(1.14)
Comments				
	Category	Subjects	Mentions	Mean
Newdle				
	Engagement	9	20	2.22
	Neutral	9	15	1.66
	Frustration	6	10	1.66
Clustering-only				
	Engagement	1	1	1
	Neutral	9	26	2.88
	Frustration	9	9	1
NYT				
	Engagement	4	6	1.5
	Neutral	9	20	2.22
	Frustration	1	1	1
Plain				
	Engagement	0	0	0
	Neutral	9	17	1.88
	Frustration	7	8	1.14

Table 4 shows Newdle users had the largest number of "Engagement" comments among the four test beds. For example, there were comments such as " The keyword cloud is interesting. I realized some news that I did not know before and explore them. Funny." The comments suggested that Newdle users had high MD since they were intrigued to explore more unknown news topics and became more engaged in the exploratory process. Clustering-only and NYT users had more comments related to "Neutral". For NYT, four users commented on the "Engagement" while clustering-only had only one user. The comments showed that NYT users liked the website in their exploration. They enjoyed navigating through hyper links on the webpages. This explained the high physical demand for NYT users. Users commented that "There is no challenge. I am familiar with this kind of news website. Most of my time was moving from one page to another and got what I need." The plain system had almost the same number of comments in the "Neutral" and "Frustration" categories. According to the comments, it was hard to associate different news articles using the plain system.

The browsing task's effectiveness was measured by "♯ of topics" and "exploration time". As shown in Table 4, on average, Newdle users found 5 news topics; clustering-only and NYT users found 4 news topics, respectively; and the plain system users only found 3. "Exploration time" grew as the "♯ of topics" increased. It indicated how the subjects were engaged in this free exploration task. Four Newdle users and three NYT users reached the 15 min time limit. All the clustering-only and plain system users exited before 15 mins. Together with the cognitive load data and comments, these two effectiveness measures suggested that Newdle and NYT users were more engaged in their task than clustering-only and plain system users.

The correlations between prior knowledge and "♯ of topics" were analyzed using t-tests. The t-tests were positive for NYT users ($r = .51$,$p < 0.05$) and the plain system users ($r = .42$,$p < 0.05$). In contrast, there was no positive correlation for Newdle users and the clustering-only system users (Newdel:$r = -.55$; Clustering only: $r = -.32$). The correlation analysis between the effectiveness measure and the prior knowledge of subjects suggested that the performance difference between novice users and experienced users in the browsing task can be reduced by community structure based EGV systems.

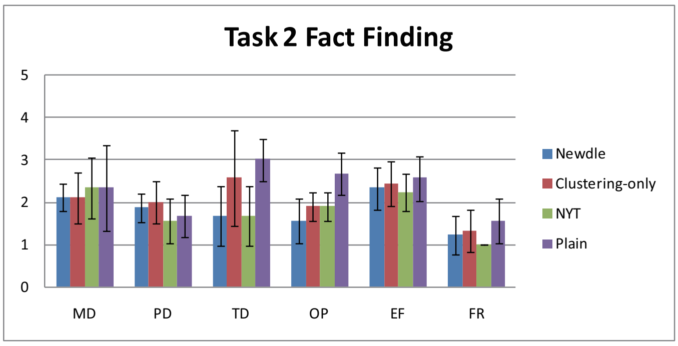

Figure 5: Means of six cognitive load factors' 5-scales rating, Task 2.

3.3.2 Task 2: Fact Finding

Task: Find as many articles as possible about humanitarian aid in the Haiti earthquake. Save the links of each article.

Overall the cognitive load of task 2 was low (as shown in Figure 5). All the ratings were below 3. There were no big differences among the four test beds in mental demand (MD, $F = 0.3, p = 0.82$). NYT and the plain system users' MD ratings were slightly higher. On the other hand, Newdle and clustering-only system users had slightly higher physical demand (PD, $F = 1.66, p = 0.19$). Clustering-only and plain system users experienced higher time demand (TD) than Newdle and NYT users ($F = 6.33, p = 0.001$). Newdle users were the most confident in their performance while the plain system users were the least confident ($F = 10.67, p = 0.0000513$). The performance ratings were almost the same between clustering-only and NYT users. NYT users experienced the lowest frustration level in task 2 ($F = 2.67, p = 0.06$).

Most comments were in the "Neutral" category (see Table 5). For NYT, one user expressed engagement in this task. Other systems did not have comments in "Engagement". According to the comments, most NYT and the plain system users

Table 5: Task 2: Fact Finding.

	Mean(SD)			
	Newdle	Clustering -Only	NYT	Plain
Effectiveness				
Interactive -Precision	0.95(0.08)	0.91(0.09)	0.77(0.11)	0.78(0.09)
Interactive Recall	1(0)	1(0)	1(0)	1(0)
Comments				
	Category	Subjects	Mentions	Mean
Newdle				
	Engagement	0	0	0
	Neutral	9	27	3
	Frustration	3	5	1.66
Clustering-only				
	Engagement	0	0	0
	Neutral	9	19	2.11
	Frustration	7	12	1.71
NYT				
	Engagement	1	1	1
	Neutral	9	16	1.77
	Frustration	8	9	1.12
Plain				
	Engagement	0	0	0
	Neutral	9	19	2.11
	Frustration	9	9	1

worked on identifying relevant news articles from the returned search results and refining their search strategy. Most Newdle and the clustering-only system users benefited from the clustered results and thought "It is clear and organized well".

"Interactive recall" and "Interactive precision" were used to measure effectiveness of task 2. As shown in Table 5, all the four test beds had 1 on "interactive recall", namely that all the subjects were able to find relevant articles. Newdle and the clustering-only system users had higher "interactive precision" than NYT and the plain system users. This suggested that NYT and the plain system users included more un-related articles in the answer.

Correlation analysis was conducted between "interactive precision" and prior knowledge. It showed a positive correlation for all the four test beds in this task ($r > 0.5, p < 0.05$).

Table 6: Task 3: Information Gathering.

	Mean(SD)			
	Newdle	Clustering -Only	NYT	Plain
Effectiveness				
Immediate -Completion Time(s)	37(5.29)	56(1.81)	61(2.97)	65(2.71)
Completion Time (min)	13.5(0.37)	15.45(0.68)	14.24(0.48)	12.89(2.5)
Result Quality	11.44(1.01)	11.33(1.32)	10.67(1.22)	9.78(1.30)
Comments				
	Category	Subjects	Mentions	Mean
Newdle				
	Engagement	2	3	1.5
	Neutral	9	28	3.11
	Frustration	3	6	2
Clustering-only				
	Engagement	0	0	0
	Neutral	9	20	2.22
	Frustration	5	12	2.4
NYT				
	Engagement	2	2	1
	Neutral	9	26	2.88
	Frustration	9	9	1
Plain				
	Engagement	0	0	0
	Neutral	9	10	1.11
	Frustration	9	19	2.11

3.3.3 Task 3: Information Gathering

Task: Summarize the activities of President Obama on Human Health Insurance.

Figure 6 showed that the trends of mental demand ($F = 7.23, p = 0.0007$), physical demand ($F = 10.44, p = 0.0000607$), performance ($F = 21.12, p = 0.000000009$) and effort ($F = 12.35, p = 0.0000156$) among the four test beds were similar: Newdle

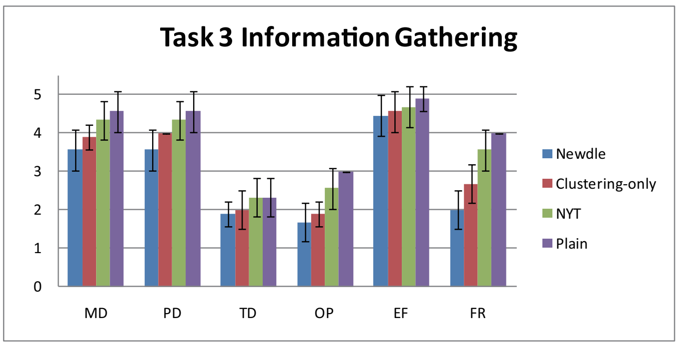

Figure 6: Means of six cognitive load factors' 5-scale ratings, Task 3.

< Clustering-only < NYT < Plain. The task did not have a time limit, so the time demand ratings of all four test beds were low ($F = 2.19, p = 0.11$). The plain system users experienced the highest frustration level. Newdle users had the lowest frustration level ($F = 37.14, p = 0.0000000000155$). NYT users had a slightly lower frustration level than the clustering-only system users did.

Table 6 showed that for NYT and Newdle, 2 users who provided "Engagement" comments for either system. Newdle users described their excitement about quickly locating the information they needed. NYT users liked the highlighting of searched keywords in the result list. For NYT, 9 users who gave "Frustration" comments as opposed to 3 for Newdle. NYT users provided comments such as "Lots articles are returned to me. I can see why they are returned, but how are they related? It is hard to form a story.", "It is terrible to see so many articles that I need to summarize. Organizing them and finding association among them cost me most of time", and "sometimes I am lost when jumping through links in the article. Too much information". This together with the cognitive load measures suggested that NYT users had higher extraneous cognitive load than Newdle users.

It seemed that the semantic representation in Newdle reduced extraneous cognitive load. For the clustering-only system, no subjects gave "Engagement" comments, 5 subjects expressed "Frustration", and the clustering-only system's cognitive load measures were higher than of those of Newdle. Some clustering-only system users complained that it was not easy to understand the relationship between the searched keywords and the returned results and extra reading was needed.

The plain system received the largest number of "Frustration" comments and the

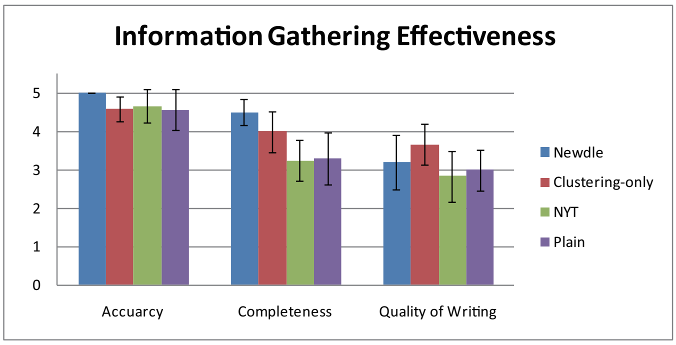

Figure 7: Means of three result quality scores, Task 3.

highest cognitive load measures. Obviously it caused the highest extraneous cognitive load on the subjects. According to the comments, the reason might be the lack of cues on the associations among the news articles.

The effectiveness measures (see Table 6) were mostly consistent with the cognitive load analysis. Newdle users had much shorter immediate completion time than other users, namely that they were able to locate desired information much faster than other users. The task completion time's order was: Plain $<$ Newdle $<$ NYT $<$ Clustering-only ($F = 5.85, p = 0.002$). However, the plain system users got the lowest overall score on result quality ($F = 110.78, p = 0.000000000554$). From the cognitive load analysis above, we knew that plain system users had the highest frustration. These together might suggest that plain users were so frustrated that they exited the task much quicker. The detailed quality measures (see Figure 7) indicated that the results from Newdle users were more accurate and complete than that from other subjects. According to Figure 7 and the immediate completion time, the clustering-only system performed better than NYT and the plain system, but there was a significant performance boost from the clustering-only system to Newdle, which indicated the importance of the visual representation of cluster semantics. Overall, Newdle outperformed the other three test beds in this task.

Correlation analysis was conducted between result quality and prior knowledge. It showed a positive correlation for the plain system ($r = .65$, $p < 0.05$) and no strong correlation for other three test beds ($r < 0$, $p < 0.05$).

3.3.4 Task 4: Revisit

Task:List as many keywords as possible that can be used to retrieve articles in task 3.

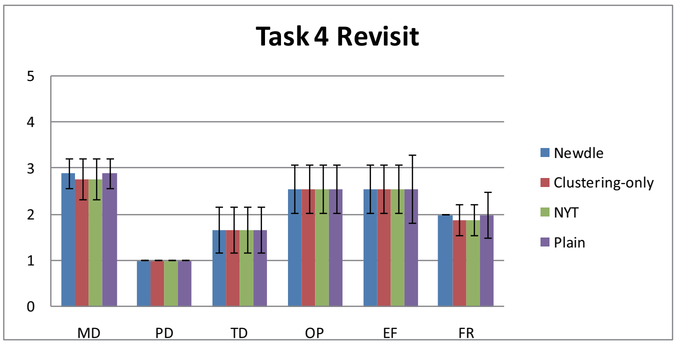

Figure 8: Means of six cognitive load factors' 5-scales rating, Task 4.

In this task, the plain system and Newdle had slightly higher mental demand ($F = 0.24, p = 0.86$) and frustration level ($F = 0.31, p = 0.81$) than the other test beds (see Figure 8). According to Table 7, no user expressed "Engagement" in this task. Five Newdle users expressed their feelings of "Frustration". The comments showed that they tried to remember the keywords in the "topic canvas", which resulted in unnecessary cognitive load.

The effectiveness of task 4 was measured by the number of keywords listed. As shown in Table 7, NYT users performed better than the other users. Correlation analysis showed that effectiveness and prior knowledge were positively related for all the test beds ($r > 0$, $p < 0.05$).

Table 7: Task 4: Revisit.

	Mean(SD)			
	Newdle	Clustering -Only	NYT	Plain
Effectiveness				
♯ of Keywords	5(2.23)	5(2.6)	6(1.51)	4(1.73)
Comments				
	Category	Subjects	Mentions	Mean
Newdle				
	Engagement	0	0	0
	Neutral	9	16	1.77
	Frustration	5	7	1.4
Clustering-only				
	Engagement	0	0	0
	Neutral	9	10	1.11
	Frustration	0	0	0
NYT				
	Engagement	0	0	0
	Neutral	9	12	1.33
	Frustration	0	0	0
Plain				
	Engagement	0	0	0
	Neutral	9	14	1.55
	Frustration	0	0	0

3.3.5 User Preferences

Table 8: Subjective questions.

	Questions Description
Q1	Presentation and structure of the news topic canvas visualization were clear (Newdle). Presentation and structure of news collections were clear(clustering -only, NYT and Plain).
Q2	It is easy to navigate around news collections using this system.
Q3	This system is easy to use for querying.
Q4	Training is required for effective use.
Q5	Clear relationship between query and visualization (Newdle). Clear relationship between query and retrieved result (clustering -only, NYT and Plain).
Q6	Understand the visualizations' organization with articles (Newdle). Understand retrieved results' organization with articles (clustering -only, NYT and Plain).
Q7	Visual representation of results was clear.
Q8	I like to use the system to do 'Fact finding" tasks.
Q9	I like to use the system to do 'Information Gathering" tasks.
Q10	I like to use the system to do 'browsing" tasks.

Figure 9 showed the subjects' answers to the post study subjective questions. Newdle users were more enthusiastic on using the system to conduct "Information gathering" and "Browsing" tasks than other users (Q9 and Q10). They also liked the presentation and structure of the visualization (Q1), and thought it was easy to navigate in the news collection using the system (Q2). It seemed that Newdle and the clustering-only system required more training than the other systems (Q4). According to the subjects, NYT was the best system among the four test beds for querying and fact finding (Q3 and Q8). NYT also received a high preference rating for using the system to conduct browsing tasks. Interestingly, NYT had better ratings than the clustering-only system on all the questions, while Newdle had better ratings than NYT on most questions. These results revealed the importance of visual representation in community structure based EGV systems.

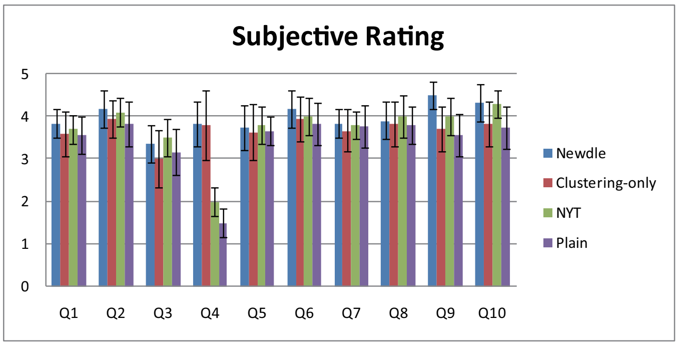

Figure 9: User subjective rating.

3.4 Insights and Leverage Points

3.4.1 Insights for EGV System Evaluation

The most important contribution of this evaluation methodology is applying cognitive load theory, especially Sweller's three types of cognitive load theory [60], to guide the design and result analyses of EGV system evaluation. We analyzed detailed cognitive load measures together with classified user comments. This approach measures and distinguishes germane cognitive load and extraneous cognitive load, the former being beneficial while the latter not. For example, the high mental demand experienced by Newdle users in the browsing task, together with lots of comments about engagement, indicates that Newdle encouraged the subjects to explore more information, which is positive. On the contrary, the high mental demand experienced by the plain system users in the information gathering task is negative, and triggers many comments of frustration. Subjects indicated that the plain system hindered their information gathering task by increasing their extraneous cognitive load. Without linking cognitive load measures and qualitative comments, such in-depth knowledge about the exploration process would have been hard to obtain. We recommend using this systematic yet practical approach to evaluate subjects' cognitive process in EGV system evaluations.

3.4.2 The Role of Community Structure Based EGV System in Information Seeking

The study reveals that Newdle, the community structure based EGV system, performs better than the other test beds in the browsing and information gathering tasks. These two tasks are typical exploratory tasks. The two tasks share the following features: (1) there is a large amount of information to be accessed; (2) subjects have vague targets before starting the visual exploration; and (3) subjects have drifting information need during the exploratory process.

In the browsing task, Newdle triggered the subjects to explore more information, which enhanced their learning. NYT website offered lots of multimedia resources like pictures and videos. Such visualizations, together with the hyper links provided, also triggered the subjects to explore information. The performance of the clustering-only system was superior to the plain system while inferior to Newdle. It indicates that clustering helps information seeking, while its results are much more useful when they are leveraged by visualization, such as the pre-attentive word cloud provided by Newdle.

Information gathering is the most complex task among the four tasks. Subjects not only need to find the information, but also have to associate, digest, and represent the information. The results show that the performance gain from using Newdle is the most significant in this task. It encourages the subjects to explore unknown topics, increased their enthusiasm in the exploratory process, compensates their shortage of prior knowledge, reduces their frustration level, and increases their confidence in the complex task. We thus recommend using community structure based EGV system in applications where complex tasks are frequently conducted.

Meanwhile, clustering-based EGV system designers need to carefully consider how to assist simple tasks such as fact finding. Although there was no big difference among the four test beds in the fact finding task, Newdle users did have a higher frustration level than NYT users. It needs to be studied how to give users appropriate information for the task at hand without overwhelming them with unnecessary information. This is consistent with a previous study in [21], which showed that visualization is a superior interface for complex, spatial, and inferential learning, and not so much the case for hunt-and-find simpler tasks.

3.4.3 What Users Need at Different Stages of Information Seeking

The detailed examination of the comments reveals that users have different information needs at different stages of an information seeking process: (1) At the beginning of an exploration, an overview is of great help. Users appreciated the fact that information is well organized and a global picture of article collection is provided in Newdle. According to user comments, "organization" was the most essential requirement at this stage. Users would like to see unrelated information separated, core information highlighted, and relationships between different information units easy to access. Designed in such a way, community structure based EGV systems can help users get an overview of a large document collection with reduced efforts.

Newdle offered two types of overviews to users (as shown in Figure 10). At the end of the training section, we asked the subjects which one they would like to use as the starting point. All the subjects selected the second one (Figure 10b). They thought the first one (Figure 10a) was "kind of overwhelming", "too much information", and "busy for eyes". In contrast, the list view in the second one made the visualization more "organized" and "understandable" to them. While after they used the system for a while and become familiar with the system, most of them began to make use of the first view. In the interview, they explained that "At first, I like the one with list because it seems more organized to me. However, after playing with the system, the other view is not that overwhelming and I can see more information when I want.", "Because I would like to see more news events at one time and I know what are under those tags. It is compact.", and "Ever since the last task, using the visualization became easier, I grasped it quicker." These comments suggest that the ability of users to digest information is evolving in the learning process. EGV sytems design can take it into consideration. A visualization with raw data presented in the way users are most familiar with is easier to understand without overwhelming novice users. During the learning process, options can be given to the users so that they can switch to a compact view with more information.

(2) During the exploration, users interacted with the system. At this stage, the most desired feature of the system was to emphasize the "association" between the subjects' actions and the results returned to them. For example, the clustering-only system frustrated users because they needed to read the article titles to evaluate the relationship between the returned results and the query in Tasks 2 and 3, whereas the search keywords were not highlighted in the results. On the other hand, NYT users in our experiment appreciated the highlighting of query keywords in the returned results.

3.4.4 Information Organization

Users provided lots of useful feedback on the information organization in clustering-based EGV systems. They are summarized in the following sections.

3.4.4.1 Revealing relationships among clusters

The study shows that the relationships among the clusters, if any, should be explicitly presented to users. Our experiment subjects called for better organization of the clusters in Newdle. A subject commented "It is difficult to see the relationships between different topics. For example, I saw several clusters about Toyota recall. But it is hard to know what the relations are between these Toyota news." User feedback indicates the need for better layout of the clusters in the design space to explicitly convey the relationships among them. We also suggest providing interactions to allow users to compare and associate clusters.

3.4.4.2 Using document lists

Interestingly, in this study, our subjects liked the view in Newdle where the topic canvases and the document lists were displayed side by side. It seems that although the cluster semantic representation helped users locate topics of interest, there still was ambiguity in the representation. The document list helps diminish this ambiguity by providing detail information. In the document list, we provide document snippets such as titles, tags, and summaries. Note that clustering-based visualization systems often provide other snippet information such as the logo of the information source, images, or even a few sentences from the documents. Most users commented that the snippets were useful. All the users agreed that the titles were very useful. In addition, the document list allowed the subjects to quickly access individual document, which was preferred by the subjects.

3.4.4.3 Organizing documents within a cluster

Our experiment subjects expressed interests for the hottest, oldest, and latest news in a cluster. This suggests clustering-based visualization systems could provide flexible ways to organize the documents within a cluster to support different tasks. For example, they can be grouped by key persons or locations and ordered by time stamps, hotness, and similarities. Their effectiveness for different exploration tasks needs to be further explored and evaluated.

3.5 Discussion

In this chapter, I introduce a systematic approach to evaluating EGV system and present our utilization of this approach in evaluating community structure based EGV system. Rooted in cognitive load theory, the methodology provides practical means to control the variability among human subjects and measure the complexity in underlying cognitive process. These methods can be feasibly adapted in EGV system evaluations, as the user study has demonstrated.

Our user study provides in-depth insights about how community structure based EGV systems worked. Our results indicate the benefits, limitations, and circumstances of community structure based EGV systems in supporting different information seeking tasks. The results show that they can benefit complex information seeking tasks such as browsing and information gathering. Moving forward, such insights and leverage points from our study should be made use of in the EGV system design.

Figure 10: Evaluation test beds.

CHAPTER 4: EXPLORING GRAPHS WITHOUT CLUTTER

Newdle evaluation greatly enhances our understanding of community structure based EGV system. As mentioned in Chapter 1, the support from existing graph visualization systems for community related tasks is fairly limited and there are several challenges for EGV system. Our first EGV system PIWI [72] , developed by Dr. Jing Yang, me, and our other collaborators, is targeting at filling this gap. PIWI enables users to conduct community related tasks effectively. It employs uncluttered, intuitive, and yet meaningful visual representations and interactions to support effective and efficient visual exploration of graphs with thousands of nodes.

In the following part of this chapter, I will present the visualizations and interactions of PIWI and exemplify its usefulness by the visual exploration of a real graph. This graph, named the NYT graph, is the same news graph we used in Newdle. It conveys the tag co-occurrence information of 1,078 New York Times (NYT) [2] world news articles published from February 22, 2011 to April 25, 2011. In this graph, each node is the tags from articles and there is one edge between two nodes only if the two tags both appear in at least one article. It also has 36 binary vertex (node) attributes carrying categorical and temporal information. Its communities reflect the major news events reported in the news corpus.

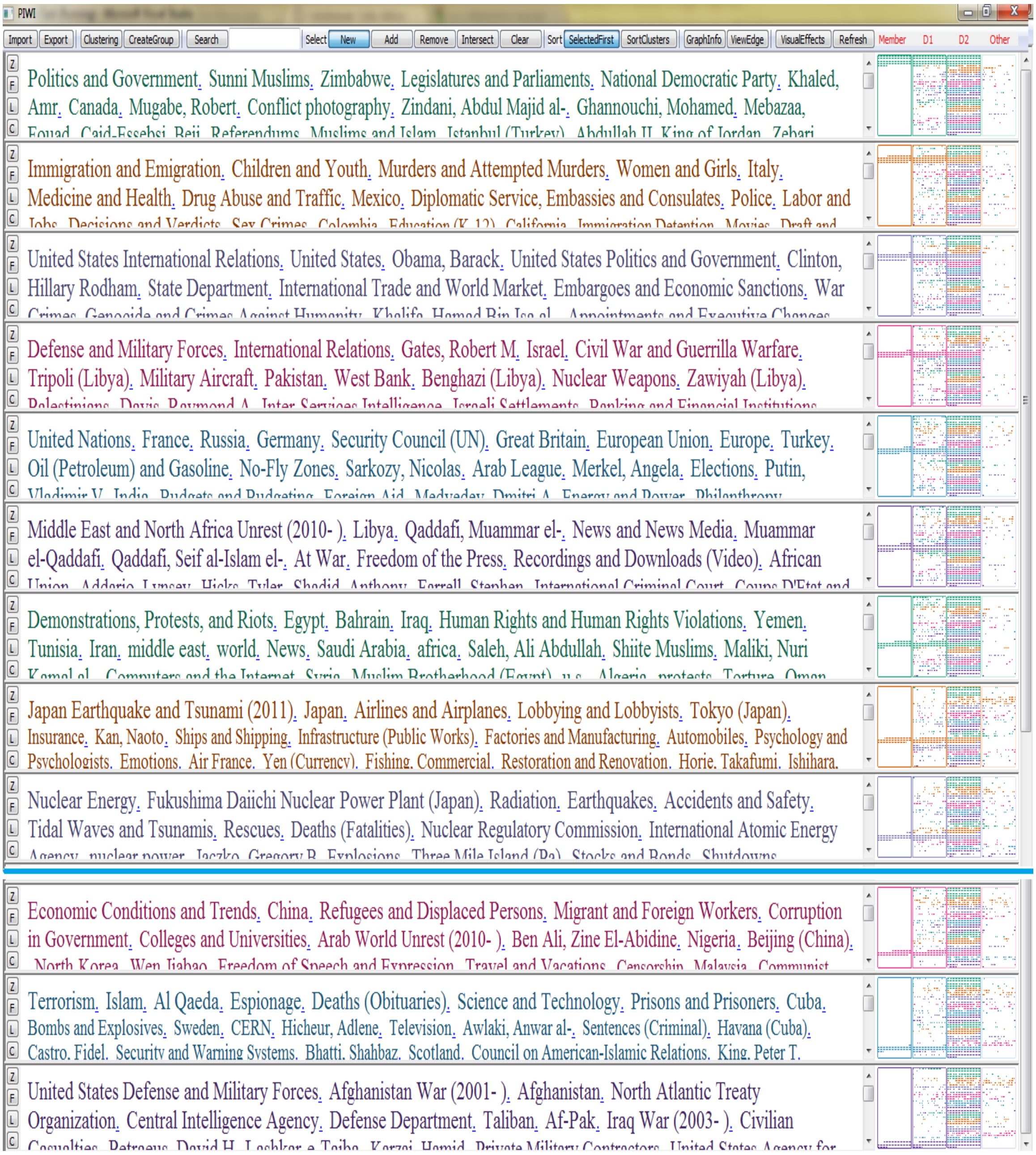

Figure 11: The initial view of the NYT graph (1,200 nodes, 7,042 edges).

4.1 Community Detection and Visual Representation of Communities

Following the suggestion by Fortunato [17], PIWI visually depicts communities and their relationships to provide a coarse-grained overview for a large graph. The communities can be detected using one of the community detection algorithms integrated into PIWI or imported from external programs. PIWI integrates multiple algorithms, including Girvan-Newman [20] and Clauset-Newman-Moore [15] from the SNAP library [44] and MCL [62]. PIWI allows overlapping communities [48]. In the initial view of a graph, the communities are visually presented to users. Figure 11 shows the initial view of the NYT graph. Twelve communities are detected using an external graph partition algorithm named AdjCluster [71]. The three communities under the blue line are accessed using the scrollbar on the right of the window; this removes the limitations on how many communities can be displayed and how much space can be assigned to each of them.

Each community is assigned a color and represented by a row consisting of a tag cloud and a set of vertex plots. A vertex belonging to the community is represented by its label in the tag cloud and dots in the vertex plots which are introduced later. The labels and dots are displayed in the color assigned to the community unless the vertex belongs to multiple communities. In the latter case, the color of a vertex will be the color assigned to the community it is most related to, which is determined by the number of member nodes it connects to or other measures. In Figure 11, several distinct colors are assigned to adjacent communities. The colors are reused when there are more communities. They help users distinguish adjacent communities in the vertex plots. Users can interactively change the number of distinct colors or display all labels in gray through the visual effect setting dialog triggered by the "Visual Effects" button on the toolbar.

Tag Clouds: A tag cloud is a list of tags where the sizes of the tags reflect their popularity [9]. The tags can be sorted alphabetically or by size. In PIWI, a tag cloud is used to display the labels of all member nodes of a community; the nodes in the central position of the community are emphasized. They may have an important function of control and stability within the community and are often looked for in real applications [17]. In particular, the tag sizes are decided by a user selected centrality metric. For example, the degree centrality can be used if nodes with large numbers of neighbors are of interest, while the betweenness centrality can be used if nodes essential for communicating are of interest. The tags are sorted by the same metric in descending order. To save space, the tag cloud is placed in a scrollable html window. Labels of the nodes with the highest selected centrality metric are visible in the initial view and users can access other labels by scrolling the html window. A label can appear in multiple tag clouds if the vertex belongs to multiple communities.

Example 1.1: Browsing Communities. A user browses the communities (Figure 11) to get an overview of the NYT graph. This is an easy task since she can read the labels of each community following her daily reading manner. She finds that this corpus contains news about Middle East and North Africa Unrest (2010-), Japan Earthquake and Tsunami (2011), Fukushima Daiichi Nuclear Power Plant (Japan), Terrorism, and etc.

Vertex plots: PIWI uses vertex plots to display the neighborhood information of the communities. A vertex plot displays nodes with desired features as colored dots without overlap. To coordinate the many vertex plots in PIWI, each vertex occupies the same position in all vertex plots. A dot is drawn in that position if the vertex has the attribute/structural feature specified in a plot. Otherwise the position is left blank. The dots are placed line by line from top to bottom, in the order from large communities to small communities. Nodes belonging to multiple communities are placed within the communities they are most related to.

As shown in Figure 11, each row, representing a community, contains multiple vertex plots. The first plot (from left to right) displays all nodes belonging to the community and is named the member plot. The second plot displays all nodes directly connected to any vertex in the community and is named the Degree-1 neighborhood (D1) plot. The third plot displays nodes whose shortest distances to the community is less than or equal to 2. It is named the Degree-2 neighborhood (D2) plot. Similarly, Degree-3 and larger neighborhoods can be defined. Users interactively set the number of neighborhoods to be displayed according to their exploration tasks. The last plot displays nodes that are not displayed in other plots in the row and is named the "Other" plot.

Example 1.2: Comparing community detection algorithms. A graph mining expert wants to compare the results of multiple community detection algorithms. Through the vertex plots displayed in Figure 11, she notices that the communities resulting from the AdjCluster algorithm have similar sizes (according to the member plots) and each community can connect to most nodes in the graph in two steps (according to the D2 plots). She re-clusters the graph using the MCL algorithm and examines the new vertex plots (please watch the supplementary video for the new display). She finds that the new results contain communities with much more variation in size and connectivity.

4.2 Interactions

Interactions play an essential role in PIWI. Users can effectively select nodes with desired structural features and vertex attributes for complex graph analysis tasks. Progressive visual exploration can be conducted by iteratively constructing User-Defined Vertex Groups (U-groups).

4.2.1 Selection

PIWI maintains a selection set. Nodes in this set are called selected nodes. The selection set is interactively modified by users through mouse clicks. We use the term hits to refer to all nodes displayed in a vertex plot a user just clicked or a vertex whose label was just clicked. According to the selection status, which is defined by the users through the toolbar, hits are used to modify the selection set as follows: (1) “New”: the selection set is cleared and the hits are added into the set; (2) “Add”: the hits are added into the selection set; (3) “Remove”: the hits are removed from the selection set; (4) “Intersect”: nodes not in the hits are removed from the selection set. Users can click the “Clear” button to clear the selection set.

Complex selection tasks can be conducted by a sequence of clicks. For example, intersecting two vertex plots, namely to click one plot under the “New” status and then click another plot under the “Intersect” status, selects all nodes displayed in both plots. It can be used for many purposes: intersecting the D1 plots of two communities selects all nodes connecting to both communities; intersecting the D1 plot of community A with the member plot of community B selects all nodes in community B that connect to community A; intersecting the member plots of two communities

Figure 12: Selection illustrations.

selects the nodes shared by these communities. For another example, users can create a union of two communities by clicking the member plot of one community under the "New" status and then clicking the member plot of the other community under the "Add" status. Since PIWI allows users to select a large number of nodes with desired features simultaneously, we say that PIWI supports bulk selection.

Selected nodes are highlighted in all displays. In the tag clouds, labels of selected nodes are highlighted by underscores. To further emphasize them, users can (1) gray out the labels of unselected nodes (see Figure 12 (b)); (2) place selected nodes before unselected nodes in the tag clouds (see Figure 12); and (3) hide unselected nodes. In the vertex plots, selected nodes are displayed in bigger dots than unselected nodes. Users can manually adjust the sizes for both selected and unselected dots. The users set the above options through the visual effect setting dialog.

Sort Clusters: Selected nodes may be distributed in multiple communities. To effectively examine them, users can click the "Sort Clusters" button on the toolbar to bring the communities with a large number of selected nodes to the top of the display.

View Edge: To investigate the relationship among the selected nodes, users can click the "View Edge" button on the toolbar. A popup window will display all edges connecting the selected nodes. In this example application, the titles of the articles where the two tags co-occur are listed under each edge. Users can access the original articles by clicking the titles.

Example 1.3: Investigating the relationship between a community and other communities. A community in the NYT graph with nodes Nuclear Energy and Fukushima Daiichi Nuclear Power Plant (Japan) triggers the interest of a user ((Figure 12 (1)). She wants to examine how this community is related to other communities. To do so, she sets the selection status to "New", and clicks the D1 plot of this community (Figure 12 (5)). Now all nodes connecting to this community are selected. She clicks the Sort "Cluster" button to sort the communities by the number of selected nodes they contain in descending order. Now the communities most relevant to the community of interest are brought to the top of the display (Figure 12 (a)).

The user notices a relevant community with nodes France, Russia, and Germany (Figure 12 (3)). How is this community related to the community of interest? She

Figure 13: A progressive visual exploration on terrorism events.

sets the selection status to “Intersect” and clicks the D1 plot of this community (Figure 12 (6)). Now only nodes connecting to both communities are selected, as shown in Figure 12 (b). Some of them belong to other communities. To focus on the selected nodes in these two communities, the user removes nodes in other communities from the selection set (set status to “Remove” and click the member plots of other communities) and clicks the “View Edge” button. From the html window, she finds interesting articles such as one titled Japan’s nuclear crisis strokes fear in Europe ((Figure 12 (7)). She clicks the titles to read the articles to get a better understanding of the relationship.

4.2.2 U-Groups

U-groups are user defined vertex clusters. They are visualized and interactively explored in the same way as automatically detected communities. To distinguish U-groups from automatically detected communities, the former are displayed on top of the latter and their left buttons are red (see Figure 13 (1), (2), and (3)).

To create a U-group, users first select nodes of interest and then click the "Create Group" button from the toolbar. A new U-group will be created which contains all the selected nodes when the button is clicked. Users can create any number of U-groups and delete a U-group at any time. A vertex can be included in multiple U-groups and communities and hence allows users to examine its context from different viewpoints.

Example 1.4: Investigating nodes of interest using U-groups. A user is interested in terrorism events reported in the NYT corpus. When browsing the communities, she notices a community with the vertex Terrorism. She wants to examine nodes connecting to Terrorism within and outside this community. To do so, she selects this vertex by clicking its label and creates a U-group for it (Figure 13(1)). By clicking the D1 plot of this U-group, all nodes connecting to Terrorism are selected and highlighted. She sorts the communities by the number of selected nodes they contain, as shown in Figure 13. It is not surprising that many nodes from other communities are connected to Terrorism. When browsing these nodes, she finds an interesting vertex Youtube.com. Why is Youtube.com related to Terrorism? She decides to examine the common neighbors of these two nodes. She creates a U-group

Figure 14: A U-group which is a union of (1) and (2) in Fig. 12.

for Youtube.com (Figure 13 (2)). Then, she intersects the D1 plots of the U-groups (1) and (2). Now, the common neighbors of Terrorism and Youtube.com are selected. To examine the selected nodes easier, she creates another U-group (Fig.3 (3)) to accommodate them. The tag Awlaki, Anwar altriggers her interest. She selects this vertex together with Terrorism and Youtube.com and examines the edges among them using the "View Edge" function (see Figure 13 (4)). The news article Radical Cleric Still Speaks on YouTube gives her the answer why these nodes are related.

In the above example, a progressive visual exploration has been conducted. Intermediate results are recorded in the U-groups, which can be kept for future use or be removed if they are of no further interest.

Example 1.5: Merging closely related communities. The two communities in Figure 12 (1) and (2) are closely related. The user decides to examine them together.

She selects both communities and creates a new U-group to accommodate them (see Figure 14).

4.2.3 Other Interactions

Import: Users can load a graph with or without community detection results. If community detection results are not provided, PIWI will cluster the graph automatically using the community detection algorithm selected by the user.

Export (Save sub-graphs): Users can save the sub-graph consisting of nodes in the selection set and those edges connecting them as a new graph. The new graph can be loaded into PIWI for further analysis.

Clustering: Users can select a different community detection algorithm and then re-cluster the graph.

Search: PIWI allows users to search nodes containing text entered in the “Search” box on the toolbar. To make the results visible, the nodes will be selected and a U-group will be created to hold them.

Sort: Besides sorting communities by the number of selected nodes they contain, users can manually move a community to the top, to the bottom, or remove it from the display by clicking the small “F”, “L”, or “C” button on the left of a row.

Zoom: Users can click the “Z” button on the left of a row to examine its labels in a large popup html window.

4.3 Visualization of Multivariate Graphs

Vertex plots can represent vertex attributes as well as structural information. A binary attribute is represented by a vertex plot that displays nodes whose values are

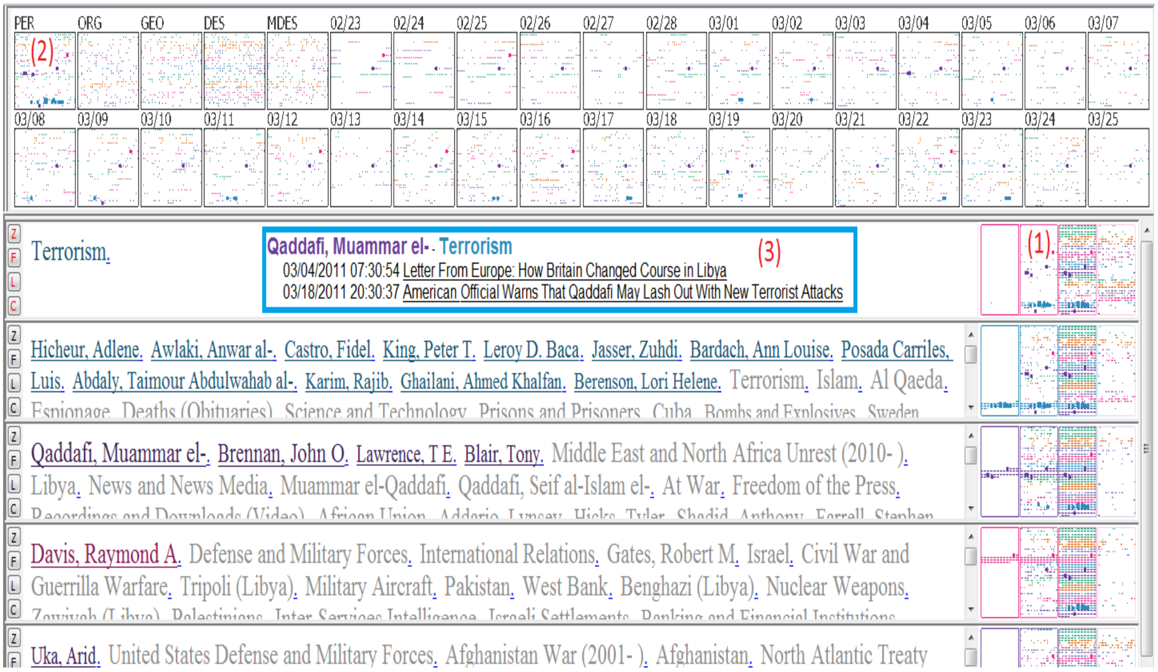

Figure 15: The highlighted nodes are persons related to Terrorism.

1. Categorical attributes with N distinct values are presented by N vertex plots, each of which displays all nodes with a distinct value. Numeric attributes are discretized and then visualized. Bulk selections can be conducted on vertex plots representing vertex attributes in the same way as on vertex plots representing neighborhoods.

Example 1.6: Exploring multivariate vertex attributes. Nodes in the NYT graph carry 36 binary attributes, as shown on the top of Figure 15. Five of them record the categories of the nodes, such as PER (person), ORG (organization), and GEO (geospatial information). Each of the other 31 attributes is a time stamp used to record whether a vertex was used to tag any NYT articles published on the date it represents.

First, a user wants to find nodes in the person category that are related to Terrorism. To do so, she creates a Ugroup for Terrorism and intersects its D1 plot (Figure 15

(1)) with the PER vertex plot (Figure 15 (2)). Now all nodes in the person category connecting to Terrorism are selected. These nodes and their context are brought to the top of the display after the user clicks the “Sort Cluster” button (see Figure 15).

Second, the user wants to investigate how the Japan earthquake was reported by the New York Times. To do so, she clicks the vertex plots of the time stamps (see Figure 15) one by one under the “New” status heading to examine how nodes are selected in the U-group she created in Example 1.5. Figure 14 shows a few screenshots she sees. The progress of how this news event was reported by NYT is clearly revealed in this way.

4.4 Case Study

In this case study, a user investigates the InfoVis co-author network which has 1,462 nodes (authors) and 6,075 edges (co-authorships). The network was generated using 4,526 bibliography entries from the DBLP Computer Science Bibliography. All of them are authored/coauthored by 36 researchers who served on the program committee of the IEEE conference on Information Visualization. From this network, the user wants to identify significant author communities and explore the relationships among them. She also wants to identify collaboration leaders and investigate their collaboration patterns.

Example 2.1: Browsing communities. The user loads this network into PIWI and clusters it using the Clauset-Newman-Moore algorithm [12]. Twenty-one communities are generated and visualized. Figure 16 shows the most significant communities. Collaboration leaders in these communities, whose names are displayed in large fonts

Figure 16: Most significant communities in the Infovis co-author network.

on top of the tag clouds, are easily identified.

Example 2.2: Finding disconnected communities. The user is curious to know if there is any community having no collaborations with the biggest community on the top. She selects its neighbors by clicking its D1 plot and sorts the communities by the number of selected nodes they contain. She then drags the scrolling bar to the bottom. At the bottom she finds three communities containing no selected nodes, which indicates that these communities are disconnected from the large one at the top.

Example 2.3: Finding bridging nodes. The user wonders if there are any researchers in other communities who bridge the biggest community with a community that is not directly connected to it. She intersects the D1 plots of these two communities.

The selected nodes are the possible bridging persons.

Example 2.4: Investigating the relationship between a vertex and multiple communities. The user wants to know the collaboration patterns of a research leader John T. Stasko. She creates a U-group for John T. Stasko and clicks its D1 plot to select his collaborators. She sorts the communities by the number of selected nodes they contain. She immediately notices that there are six communities besides John's own community whose members collaborate with John.

4.5 User Study

4.5.1 Setup

A preliminary user study has been conducted to evaluate the effectiveness of PIWI in helping sophisticated users conducting community related tasks on a real graph. In this study, PIWI was compared with the state-of-the-art graph visualization tool NodeXL (version 1.0.1.196 released on December 2011, the newest downloadable version when the study was conducted in February 2012) [2]. NodeXL displayed graphs whose edge and vertex lists were stored in an Excel 2007 or Excel 2010 workbook (Excel 2007 was used in the study). It uses coordinated NLDs and Excel worksheets to display the graphs and their vertex, edge, and group details.

Subjects: Ten subjects participated in this study. They were graduate and senior undergraduate students of CS and other engineering majors. They were not familiar with PIWI were divided into two groups balanced by the number of subjects, major, and education level. One group used PIWI and the other group used NodeXL in the study.

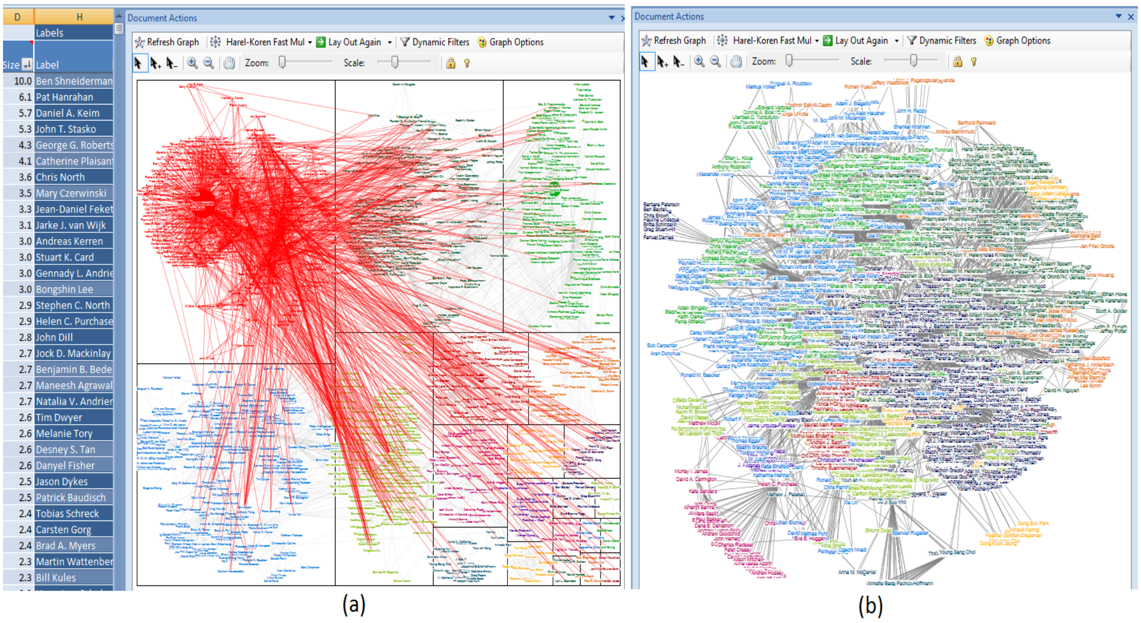

Figure 17: NLDs (Infovis co-author network) in NodeXL [58].

Datasets and their Communities: Two graphs were used in the study. The Polbooks graph (105 nodes and 441 edges) [67] was used in the training. The Infovis co-author Network (1,462 nodes and 6,075 edges) was used in the test. Four communities were detected from the Polbooks graph and twenty-one communities were detected from the Infovis co-author network using the Clauset-Newman-Moore algorithm [15]. The same community detection results were used in NodeXL and PIWI.

NodeXL Setup: (1) NLD layout: each of a graph's community was laid out in a box. The boxes were sorted by community size (see Figure 17 (a)). This layout was chosen since it was the best layout for community related analysis tasks in NodeXL (Figure 17 (a) and 7 (b) compare the layout used and another layout in NodeXL). (2) Visual effects: Vertex size helped the subjects identify nodes with high degree centralities. The edges were set to be semitransparent to reduce clutter. The subjects were allowed to change the opacity interactively during the study.

NodeXL Interactions: The subjects were trained to use the following interactions in NodeXL: (1) Selection: Selecting nodes would highlight them and all edges connecting to them in both the NLD and the Excel worksheets. Clicking a vertex on the NLD or the Excel worksheet would select it. Drawing a rectangle on the NLD would select all nodes in it. This technique was used to select all nodes in a community. Figure 17 (a) shows the results of such a selection. A right click on a vertex would trigger a menu from which all neighbors of the vertex could be selected. Users could add/remove newly selected nodes to/from the selection set, but intersection was not available in NodeXL. (2) Clutter reduction: Dynamic filters would be used to remove nodes with small degree centralities from the NLD. Scaling the nodes and labels and zooming in would reduce overlap among the labels. In addition, if a vertex was selected from the NodeXL, it would be automatically brought to the center of the Excel worksheet, where its label could be read feasibly. (3) Search: Excel search functions could be used to find nodes whose labels contained a given string. There were more interactions available in NodeXL, but they were not relevant to this study.

Procedure: The subjects conducted the study one by one on a Lenovo T420 laptop (Intel(R)Core(TM)i5-2410M CPU@2.30GHZ). The laptop was connected to an external display set with a 20 inch LCD monitor with a resolution of 16801050 pixels. Each subject had a training session followed by a test session. In the training session, the instructor first briefly introduced the visualization and the interactions of the system to a subject. Then the subject conducted four training tasks on the Polbooks graph, which were highly similar to the tasks to be conducted on the Infovis co-author graph in the test session. Hands-on help was provided by the instructor, including exemplifying strategies for conducting the task and answering questions. The training session ended when the subject finished the training tasks successfully. The total time spent in the training session was recorded for each subject. In the test session, the subject conducted four tasks on the Infovis coauthor graph independently. The time she/he spent on each task was recorded. After each task, the subject was asked to rate the effectiveness of the tool on a 5 point scale. After the test session, the subjects provided preference ratings and comments.

The four tasks used in the test are presented as follows. They were significant community related analysis tasks identified from the case study. The best strategies for using NodeXL to conduct most tasks, figured out by the authors in the pilot study and taught to the subjects in the training, were also presented. The strategies taught to the PIWI users are presented in Examples 2.1, 2.2, 2.3, and 2.4 in Section 4 and are not repeated here. The subjects were free to use any other strategies they wanted. In all tasks the subjects only needed to write down the first names of the authors to reduce the effect of their writing speed on performance.

Task 1: Browsing communities. Identify four big communities. From each of them, find three persons with high degree centralities and write down their first names. NodeXL Strategy: Apply the dynamic filter to remove nodes with small degree centralities. Scale the labels and zoom into big boxes for labels of large nodes. Alternative NodeXL Strategy: Sort the nodes in the Excel worksheet by their degree centrality in descending order. Select all nodes of a community displayed in a large box. Copy the labels of the three topmost selected nodes from the worksheet. This was a more efficient strategy than the first one for large graphs. The first strategy, rather than the second one, was taught to the subjects to test the capability of the subjects in reading labels from an NLD.

Task 2: Finding disconnected communities. Given the label of a vertex (Ben Shneiderman), find three groups that are not directly connected to its community. Write down the first names of three persons from each group. NodeXL Strategy: Search and select the given vertex. Identify its community and select all its nodes. The nodes and all edges connecting to them are thus highlighted (see Figure 17 (a)). Scan the NLD for communities where there is no vertex connecting to the highlighted edges.

Task 3: Finding bridging nodes. Labels of two nodes are given (Helen C. Purchase and Ben Shneiderman). Their communities are not directly connected. Are there any persons in other groups who connect these two communities? If so, write down their first name. NodeXL Strategy: Select nodes in both communities and manually look for bridging nodes. The answer had three bridging nodes.

Task 4: Investigating the relationship between a vertex and multiple communities. There are six communities connecting to John T. Stasko besides his own community. Write down the first name of a person connecting to John T. Stasko from each of these communities. NodeXL Strategy: Select the vertex and its adjacent nodes. Look for selected nodes from different boxes.

4.5.2 Results

The average training time was about the same for NodeXL (9.7 minutes) and PIWI (9.3 minutes). The average completion time and correctness of PIWI users and

Table 9: User study results.

		PIWI	NodeXL
Task 1	Time (minutes)	1.8	4.7
	Correctness (%)	100	100
	Effectiveness Rating (1-5)	5	3.6
Task 2	Time (minutes)	2.7	4.9
	Correctness (%)	100	47
	Effectiveness Rating (1-5)	4.8	2.6
Task 3	Time (minutes)	2.6	5.6
	Correctness (%)	100	33
	Effectiveness Rating (1-5)	4.6	2.2
Task 4	Time (minutes)	1.9	3.7
	Correctness (%)	100	97
	Effectiveness Rating (1-5)	5	2.8
Easy to Use	(1-5)	3.2	2.2

NodeXL users on each task are reported in Table 9. After each task, the subjects answered the question "Is this tool effective in conducting this task?" on a five point scale (5-strongly agree, 1-strongly disagree). After all tasks, the subjects also gave preference ratings on "easy to use" (5-very easy, 1- very difficult). The average ratings from PIWI users and NodeXL users are reported in Table 9. As shown in Table 9, the 5 PIWI users had faster average performance times than the 5 NodeXL users for all tasks. PIWI users also had more average correct answers than NodeXL users for tasks 2 and 3. PIWI users thought their tool was more effective for the four tasks than NodeXL users. In general, PIWI users thought their tool was easier to use than NodeXL users. We expect that if the alternative strategy was taught in Task 1, the NodeXL users would have had better performance in Task 1. The current result of Task 1 revealed the difficulty of label reading in NLDs if no other facilitated views were provided.

After the test, the subjects wrote down their comments on the advantages and disadvantages of the system they used. PIWI users listed the following advantages: "I felt that the system was well organized which made detail operations really easy to do." "Easy for tasks. Enable me to quickly access the connected nodes without cluttering the display." "This system is excellent for searching through and determining relationships within graph data." "Easy to find neighbors of a given vertex and common neighbors of several nodes." "I like the vertex plots for selection." "The ability to quickly construct new groups gave a lot of additional information on the relation between these groups." "The sort cluster function was very helpful." All of the disadvantages listed for PIWI were mild suggestions, such as adding a search group function and adding a right click menu to avoid having to move to the toolbar when changing the selection status. These suggestions will be applied to the new version of PIWI.

NodeXL received positive comments such as the interactions were simple, community relationships were revealed to some extent, the integration with Excel was good, and the highlighting and selection were helpful. The negative comments revealed that multiple NodeXL users experienced difficulties in selection: "Sometimes I need to click nodes one by one to find the answers. Crazy." "Difficult to select nodes due to the label clutter." "This system is easy to understand, but not effective. I have to search over the interface to find answers." The subjects also complained about the clutter in the display: "Labels are overlapped. Hard to recognize." "When a vertex has too many neighbors, it is hard to distinguish them."

4.6 Exploratory Study for Utility

In the user study reported in previous section, the subjects conducted graph analysis tasks following given strategies. To learn how PIWI was used in a more complex visual reasoning process, another exploratory study was conducted with sophisticated users. In this study, eight computer science graduate students were asked to use PIWI (CPU version) to gather information from the NYT news corpus. Two tasks were given to each subject. The first task was to identify as many news events about Libya as possible in 10 minutes. The second task was to identify three terrorism events in the shortest time possible. For each subject, the instructor first showed him/her how to use PIWI to explore the NYT graph, identify nodes and edges of interest, and access the original news articles through the edges. The subject then tested the interactions with the help of the instructor. This free form exploration lasted about 10 minutes. After that, she/he conducted the tasks independently. After conducting the tasks, she/he rated PIWI in "usefulness", "ease of use" and "awareness of context" on five point scales and answered open questions.

On average, 3.75 news events were identified by the subjects in the first task. The subjects completed the second task in 7.5 minutes on average. It was observed that the subjects developed different strategies to make use of PIWI. For example, some of them first browsed the communities in the initial view and then created U-groups for nodes of interest; some of them started by searching nodes of interest and then examined communities most relevant to these nodes; some of them started by bulk selection (such as selecting all nodes connecting to terrorism), followed by exporting and exploring sub-graphs. These strategies showed that PIWI was a flexible graph exploration tool.

With regard to preference ratings, PIWI received average ratings of 4, 3.75, and 4.38 on "usefulness", "ease of use" and "awareness of context", respectively (1 is the lowest and 5 is the highest). The post-study comments from the subjects showed that they liked the vertex plots. There were comments such as "The vertex plot is powerful. I can quickly erase the unnecessary nodes. I like it." and "The vertex plot can help me find the relationship among different types of information". The subjects also commented that using the vertex plots for selection was much quicker and intuitive than using menus or check lists. In addition, the subjects commented that the tag clouds made sense to them. There were comments such as "Perfect. The tag clouds provide direct description of what I am working at!"

4.7 Discussion

PIWI is effective in various community related tasks. Its "tag clouds + vertex plots" visualization presents the semantics, attributes, and communities of a large graph in an uncluttered way. Labels of a large number of nodes can be examined within their communities and neighborhood context simultaneously. PIWI provides bulk selections to facilitate the selection of a large number of nodes according to desired structural features and vertex attributes. Moreover, user-defined vertex groups, which can be constructed efficiently with bulk selections, allow users to use the output of a previous step as the input to further visual exploration, and then to compare, refine, and integrate results from multiple visual exploration steps. Our case studies and user studies demonstrated that PIWI facilitated rich community related graph analysis tasks and iterative visual reasoning.

During PIWI's development, lots of ideas come to our mind and make our future direction more clear. First of all, note that NodeXL is a general purpose graph visualization system, while PIWI is specially designed for community related tasks. All the tasks used in the study were community related tasks and thus they might favor the design features of PIWI. Thus the tasks at hand provide unique user experience of the system. However, the community related tasks in visualization domain have not been systematically collected and summarized yet. It would be much better if we have had a task taxonomy at the beginning of PIWI design. Second, if the vertex plots of a vertex group (a community or a U-group) are considered a visual presentation of the distances from all nodes to that group, PIWI actually maps a graph from a high dimensional space to a much lower dimensional space. This space consists of dimensions that summarize the structural information (communities and their neighbors), dimensions that reflect user interests (U-groups and their neighbors), and vertex attributes. In this way, a graph can be mapped into a multidimensional visualization space. If we figure out a suitable way to do the mapping, lots of useful multidimensional visualization techniques can be made use of in graph visualization.

CHAPTER 5: COMMUNITY RELATED GRAPH TASK TAXONOMY

Since the community structure exists in most real world networks and understanding it is essential for the understanding of these networks, there is a dire need for supporting community related tasks in graph visualization systems. A systematic approach to community-based graph visual exploration is desired: major tasks need to be discovered; significant gaps and challenges need to be identified; effective visualization techniques need to be developed. In this chapter, I present our preliminary efforts along this direction. First, I surveyed 225 research papers about graph communities from research domains including data mining, sociology, and visualization. Based on this survey, a taxonomy of community related graph exploration tasks is constructed. Then, a set of users of real networks were interviewed to validate the task taxonomy. Concrete examples of the tasks with solid application backgrounds were collected from these interviews. Four categories based on the graph objects' interaction with community are identified. They are node, link, community and time. Next, 17 state-of-the-art graph visualization systems were examined against the task taxonomy, which revealed a large number of community related exploration tasks ignored in existing visualization efforts. Then, challenges in existing approaches are discussed and where future research efforts need to be conducted are identified.

5.1 Taxonomy Construction

Two hundred twenty-five research papers related to graph communities were collected from references of recent community detection [17] and graph visualization [38] survey papers, as well as top cited research papers returned by Google scholar using the search term *community*. They include visualization papers (46%), social network analysis papers (28%), and graph mining papers (26%). Although the collection does not include every paper relevant to graph community analysis, it is a good sample that reflects recent common practice. I reviewed these papers to identify community related graph exploration tasks. The tasks are often mentioned in introductions, case studies, and user studies of these papers.

Moreover, a group discussion about community related graph analysis tasks was conducted in a lecture of Information Visualization. Thirty-two college students were divided into four groups. Within each group, the students spent about 20 minutes to list as many tasks as possible and then the results from all groups were merged. A task taxonomy was generated based on the tasks collected from the literature review and the group discussion results. Section 5.3 presents this task taxonomy in detail.

5.2 Taxonomy Validation

To examine how the task taxonomy fits real world applications, my teammate Craig Mason and I conducted a set of face to face interviews with expert users of real world networks. Each interview typically lasted one hour or longer. In each interview, an interviewee was asked to describe the real network she met in her daily work, provide concrete examples to the tasks listed in the taxonomy, and rate their importance; she was also asked to provide tasks that do not fit into the taxonomy. The profiles of the interviewees and the networks they deal with are described as follows:

- Interviewee 1 is a vice president of Brotherhood of Railroad Signalmen (BRS). He deals with a Trade Union Network (TUN) consisting of many sub-unions, such as local lodges made up of members who work for a particular railroad, or in the case of large railroads, on a particular division of the railroad.
- Interviewee 2 is an associate professor in the research area of bioinformatics. He studies a Protein Similarity Network (PSN) formed abstractly after taking measurements on different proteins, where proteins and protein residues (small pieces of proteins) are nodes and their similarities are edges.
- Interviewee 3 is an employee of a game designing company. He analyzes a mobile social network for gaming (GAME), where players are nodes and they can interact with messages to their friends, game recommendations, and giving each other in-game items.
- Interviewee 4, 5, and 6 are senior PhD students conducting research on visualization, biology, and networking, respectively. They are familiar with co-author networks (Co-author) where researchers from different research communities collaborate with each other. In these networks, researchers are connected by co-authorships on research papers.

Most example tasks collected from the interviews fit into the initial task taxonomy very well. For the few exceptions, I slightly modified the taxonomy to fit them in.

The interviews revealed that the importance of each task heavily depends on specific domains, namely that a task important in one domain may be unimportant in another domain. Since the task taxonomy should be domain independent, I decided not to provide importance evaluations to the tasks in the taxonomy.

5.3 Resulting Task Taxonomy

In this section, we present the task taxonomy in detail. Inspired by Lee et al. [41], the tasks are classified into four categories according to the objects interacting with communities. They are nodes, links, communities, and time. A general description and a few examples from our interviews are presented for each task.

Node Related (N) Tasks

N1: Identifying nodes in a central position of a community.

Example: (1) (TUN) Identifying important elected officials who could be helpful to some piece of legislation the BRS is interested in passing. Finding general chairmen who are leading or spearheading a particular issue. (2) (PSN) Finding the key protein residues governing enzyme activity.

N2: Identifying nodes at the boundary between a community and other communities or common nodes of overlapping communities.

Example: (1) (TUN) Identifying elected officials who both support railroad development and labor to help with legislation. (2) (Co-author) Identifying researchers in a mobile visualization research group that have collaborations with people from the text visualization group.

N3: Browsing the connection between a vertex and the communities.

Example: (1) (TUN) Examining how a particular member of railroad management connects to members of unions and others in railroad management. (2) (Co-author) Finding out how one researcher collaborates with researchers that work on data mining.

N4: Finding nodes within a given distance to a community.

Example: (1) (TUN) Identifying political activists who could help with a movement in a district which is far away. (2) (PSN) Building phytogenetic trees and finding what are several steps away. This helps understanding the protein evolution process.

N5: Identifying common node attributes of community members.

Example: (1) (TUN) Finding all the union members in a particular locality. (2) (GAME) Finding out which age/gender demographics enjoy which games. This helps developing new games and features which appeal to different demographics.

Link Related (L) Tasks

L1: Finding the proportion of links in a community that go to different node categories.

Example: (Co-author) For a given community, what fraction of collaborations go to different research categories?

L2: Identifying common edge attributes of a community.

Example: (1) (GAME) Game developers need to design features in a game that meet the interests of players. This relates to the interests of players and what connects them. (2) (Co-author) Identifying the reason that people get together and form a community. Common interest or near location?

Community Level Tasks (C)

C1: Estimating the basic information about a community, such as its diameter, number of nodes, link density, or other features.

Example: (1) (GAME) It is important to know the diameters of communities to learn the scalability of games. The faster one can propagate new features and events across the network, the more money that is made. (2) (Co-author) Identifying the patterns of collaboration in one community. Are there one or more significant scientists leading this collaboration or is collaboration more widespread?

C2: Understanding the connections between a community and other communities.

Example: (1) (TUN) Finding how one union communicates with other unions. (2) (Co-author) Examining how a research community collaborate with other research communities.

C3: Examining the distribution of nodes with certain attributes among all communities.

Example: (1) (TUN) Examining the distribution of age and the length of service for members within one union. (2) (GAME) Finding out what communities like in games helps developing new features/games. Game developers are interested in demographics.

C4: Identifying central and/or marginal communities in a graph.

Example: (1) (GAME) Identifying the central/hardcore gammers and appealing to their needs. (2) (Co-author) Which research group has the widest collaboration? Which is a relatively new and small group?

C5: Finding commonality among multiple communities.

Example: (1) (GAME) Node groups that are involved in multiple games help identify which common features are good, and also help with understanding network spread. (2) (BIO) Finding commonality between communities to study protein evolution.

Tasks about Time and Changes (T)

T1: Examining how nodes emerge and die out in the communities.

Example: (TUN) Identifying emerging popular candidates for an election within the BRS or in a political office.

T2: Identifying emerging and dying communities.

Example: (GAME) Finding emerging popular game groups and their features is very important. Are players moving from a specific game to another one? Is a game getting more or less popular after a new feature is implemented?

T3: Examining how the relationships between node attributes and communities evolve over time.

Example: (1) (TUN) Spotting an ideological trend among new voters or new union members. (2) (GAME) Examining what happens when a community gets older. Are the players still interested in similar games or do their interests shift?

T4: Examining how the relationships among the communities evolve over time.

Example: (1) (GAME) Examining cross-group interactions over time is somewhat important. One game community might like a new feature while another community might not. (2) (Co-author) At which points in time do people in a research community collaborate more often with people in another research community?

Table 10: Conducting community related tasks in 17 visualization systems.

	N1	N2	N3	N4	N5	L1	L2	C1	C2	C3	C4	C5	T1	T2	T3	T4	Evaluated by
SocicalAction [50]	✓	+	+		-		✓	+	-		-		✓	-	-		P + V
NodeXL [58]	-	-	-	+	-		+	-	-	+	✓		✓	-	-		P + V + D
Apolo [13]	✓		-		-												P + V
Tulip [3]	✓							+									V
OntoVis [55]	✓					+	✓		-		-						P + V
Vizster [25]	✓		-	-	-	-		-	-	-			+				P + V
GraphDice [11]	+	-	+		+			+	-	+	-	-					P
Multi-categoryGraph [31]	-				-												P + V
FlowVisMenu [65]	+	-	+		+			+	-	+	-	-					P + V
EulerDiagrams [54]		+	+						-			-					P + V
NodeTrix [26]	+	-	+					-	-		-	-					P + V
DissertationBrowser [14]									-			+				✓	P + V + D
GMap [29]									-		+						P
CGroup [35]		+	+														P
NSSV [57]			+	-	+	✓		-	-	-			✓	-	-	-	P + V
PivotGraph [68]					+	✓			-	-							P
PIWI [72]	✓	✓	✓	✓	✓	+	-	-	✓	✓	✓	-	-	-	-	-	P + V + D

5.4 Discussion: State-of-the-art and Challenges

To learn how the community related exploration tasks are supported in existing visualization systems, we sampled 17 graph visualization systems from recent literature (see Table 10) and reviewed their publications, videos, and downloadables if they were available. We examined each system against each task in the taxonomy bearing the following questions in mind: 1. Does the system support the task? 2. If it does, can users conduct the task with ease? 3. What are the strategies employed by the system to support the task? 4. If the system doesn't support the task with ease, what are the major challenges? The answers to the first two questions are summarized in Table 10. In the table, "✓" represents the task can be conducted with minimal efforts (namely the task can be done with less than a few interactions and without any sequential scans). "+" represents the task can be done with medium efforts namely the task can be done with a few interactions and a few sequential scans. "-" represents the task can be done with significant efforts namely the task can be done with a large number of interactions and a large number of sequential scans. "P" represents paper.

"V" represents video. "D" represents demo. Note that the questions are answered heuristically by me and the answers to an individual system may vary by a different reviewer. However, the purpose of this table is an estimation of the state-of-the-arts and some deviations can be tolerated.

In order to make the system evaluation clearer to the readers, I briefly introduce the selected systems in the following. Each system has a description of its design goals and visualization approaches.

SocialAction [50]: It is a social network analysis tool that integrates visualization and statistics to improve the analytical process. It visualizes social members as the nodes and their relationship as links. Statistics information is used to filter nodes, find outliers, and identify subgroups of interest. Link types are also used to find link patterns.

NodeXL [58]: It is a free, open-source template for Microsoft Excel 2007 and 2010 for graph analysis. It has a node link diagram visualization with node, link, and group information listed in a worksheet. Various layout, filtering, and grouping methods are offered.

Apolo [13]: It combines user interaction and machine learning together. It uses node link diagram as its major visualization. Users identify the node that they are interested in as exemplars. Then, Apolo uses a machine learning method called Belief Propagation to infer which other nodes may be of interest. In this way, users form the interested subgraph.

Tulip [3]: It is an information visualization framework dedicated to the analysis and visualization of relational data. A node link diagram is the major visualization.

Various layout, filtering, and grouping methods are offered.

OntoVis [55]: It is designed for understanding large, heterogeneous social networks, in which nodes and links could represent different concepts and relations, respectively. OntoVis visualizes these concepts and relations through an ontology. The ontology is used to guided the graph exploration.

Vizster [25]: It is an interactive visualization tool for online social networks, allowing exploration of the community structure of social networking services. Vizster focuses on ego-centric social networks, or network views centered on a single individual and her direct linkages. A node link diagram is its major visualization.

GraphDice [11]: It is a multivariate network visualization system for exploring the attribute space of social networks. GraphDice uses multidimensional visualization techniques to support complex visual analysis tasks on large networks and provides a simple and consistent interface for interacting with network data.

Multi-categoryGraph [31]: It targets at graphs used in real-world applications consist of nodes belonging to more than one category. It uses node link diagrams as major visualization. It constructs hierarchical clusters of the nodes based on both connectivity and categories and places the nodes to display both connectivity and category information.

FlowVisMenu [65]: It develops new visualizations and interactions based on multidimensional visualization techniques. It focuses on supporting various attribute manipulations on the graph.

EulerDiagrams [54]: It is focused on set visualization and uses Euler style diagrams as visual representations. Two approaches were proposed to simplify a complex collection of intersecting sets into a strict hierarchy. From the visualization,users can quickly find clusters of nodes and overlapping communities.

NodeTrix [26]: It targets at community analysis in social networks. NodeTrix uses matrices to reveal community and node link diagram to reveal relationships among the communities.

DissertationBrowser [14]: In this visualization, each Stanford department is displayed as a circle, colored by school and sized by number of PhD students graduating from that department. After an interesting department is identified, it becomes the focus of the browser and every other department moves to show its relative similarity to the centered department.

GMap [29]: This system fauces group visualization. It visualizes relational data with geographic-like maps. Each group is visualized as a region in the map.

CGroup [35]: C-Group is a tool for analyzing dynamic group memberships in social networks over time. The system is focused on pair relationships between groups, particularly highlighting the similarities and differences between node groups.

NSSV [57]: This system uses node semantic information to generate the graph layout. Users can get a general idea about node attributes through the graph layout. Combined with rich filtering and selecting functions, users can do lots of complex graph exploration.

PivotGraph [68]: The system is designed specifically for multivariate graph. PivotGraph uses a simple grid-based approach to reveal the relationship between node attributes and connections. The axes of the scatter plot are node attributes and organizations. Through the visualization, the multidimensional comparisons are straightforward to users.

As shown in the table, there are several relatively "easy" tasks (N1, L1, L2, T1) that have been supported in a few existing systems. All these tasks do not require examining more than one community in detail at the same time. On the contrary, other tasks, which typically involve two or more communities, are largely ignored or require medium or significant efforts in most systems. There are much more "difficult" tasks than "easy" tasks, which reveals the big gap between the need and the current practice. The following challenges facing existing systems have been identified:

Challenge 1: Poor Readability. Graph readability is always a challenge in graph visualization when semantics of nodes and links matter. This is especially true when conducting community related exploration tasks [72]. For example, when examining the connection between a community and other communities (C2), most systems allow users to select all nodes in one community, which will highlight their direct neighbors. However, the labels of these highlighted nodes can easily clutter the display and thus make a detailed examination of these nodes very difficult. The clutter would be worse if nodes connecting to the community in more than one step are examined. The same challenge is facing other tasks such as N2, N3, N4, L2, C3, and T4. They may return a large number of nodes/links whose labels need to be examined.

A common practice to overcome this problem is to filter out less important nodes and thus labels of more important nodes can be readable [58]. However, the labels of the nodes remaining on the display may still overlap and the filtering criteria may be difficult to define for tasks such as C2. PIWI [72] presents a promising direction to address the readability problem. The idea is to shift the display focus from topology to semantics of the graphs. PIWI assigns the majority of the screen to node labels, which are organized according to the community structure. Users can conduct interactions on the labels directly or through compact displays attached to the label groups which convey community level topology information. This approach is effective when users are more interested in semantic information and community level interactions than the inner structure of the individual communities.

Challenge 2: Difficulties in Conveying Heterogeneous Information. Conducting tasks such as N5, C3, T3 and T4 often requires users to digest heterogeneous information. Such information may include multivariate node and link attributes/statistics, distances from nodes to communities, and temporal information. Visualizations and interactions of such heterogeneous information are complex. First, the available visual channels within traditional graph visualizations are quite limited. For example, the traditional approach of mapping node attributes to node colors or shapes can only convey one or two attributes at the same time. Second, users often desire to conduct interactions such as selections according to heterogeneous criteria. Such criteria are difficult to be intuitively expressed. As a consequence, it is different to examine and manipulate heterogeneous information in complex community related exploration tasks.

A recent trend to address this problem is using multidimensional visualizations closely coordinated to traditional graph visualizations to present multidimensional node attributes/statistics. Explicit links are often used to connect nodes in the graph view and the attribute view to help users coordinate the views. Examples include FlowVisMenu [65] and GraphDice [11]. However, the distances between nodes and communities, which are essential for addressing tasks N2, N3, N4, C2, and C4, are not explicitly conveyed in the above approaches. This drawback prevents them from supporting many community related exploration tasks. There is a dire need for approaches effectively conveying the distance information. PIWI [72] uses pixel-oriented displays to convey the distance information, but it has a drawback that it is difficult to track individual nodes in the pixel-oriented displays.

Challenge 3: Disoriented Users. In a previous user study [72], we observed that users frequently conducted zoom in, zoom out, and panning operations when conducting community related exploration tasks using Node-Link Diagrams. They zoomed out to get a global view for further navigation and zoomed in to retrieve details such as node labels. This was time consuming and required lots of efforts from the users. Moreover, the users could easily lose the context of their exploration after a sequence of zoom in and zoom out operations. Why were so many zoom in and zoom out operations conducted? A fundamental reason was that when users examine a community in detail, most other communities are excluded from the display to give more room to the community being examined. To overcome this problem, new approaches that simultaneously visualize multiple communities with enough details are needed. In addition, interaction results should be well organized so that users can effectively examine nodes of interest no matter they are in central communities or marginal communities.

To invoke more research efforts towards supporting community related graph exploration tasks, in next chapter, I will tailor two multidimensional visualization techniques into graph visualizations, which aim at addressing a large number of community related graph visualization tasks in the taxonomy.

CHAPTER 6: MULTIDIMENSIONAL GRAPH VISUALIZATIONS

As discussed in the previous chapter, two features are desired in graph visualizations to support community related tasks. First, the distances between nodes and communities need to be visually presented and employed in interactions. Second, multiple communities need to be examined with enough details on the same screen. Their relationships also need to be visible to users with enough details.

In this chapter, I propose two visualization techniques with the above desired features. The main idea is to analogize community based graph visualization to multidimensional visualization. In particular, the communities are analogized to dimensions in a multidimensional dataset, and the nodes are analogized to data items in a multidimensional dataset. Thus, the distance from a node to a community can be expressed by the value of a data item on a dimension, and the correlation between two communities is mapped to correlation between two dimensions. Therefore, existing multidimensional visualization and interaction techniques can be customized to support community related graph exploration tasks.

Pirolli and Card [51] identified several leverage points in the analytic process. When developing the proposed approaches, I particularly focused on one leverage point: the cost structure of scanning and selecting items for further attention. The purpose of the visualization is to provide an interactive visual index of a graph and guide users toward data of interest for further attention. Particularly, I focused on tasks C1 ∼ C5, which are not well supported by current approaches as shown in Table 10.

6.1 Mapping a Graph to a Multidimensional Space

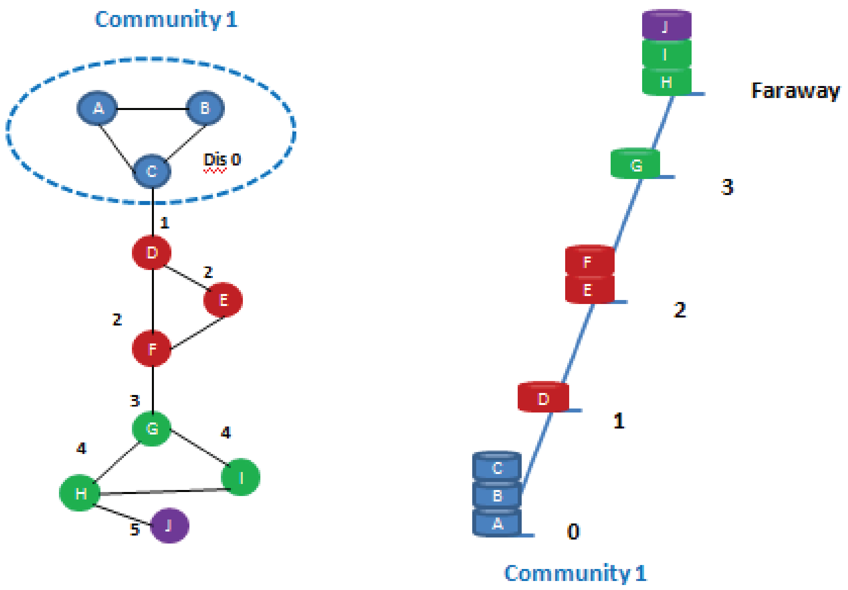

Figure 18: Topology information mapping.

Fig. 18 illustrates the topology mapping from a graph to a multidimensional dataset using a simple example. It shows a graph with four communities. Nodes within the same community have the same color. Taking *Community 1* which is in blue for example, we illustrate how nodes are mapped to an axis representing *Community 1*. In Fig. 18, each node's topology distance to *Community 1*, namely the shortest distance from the node to any node in Community 1, is labeled above itself. For example, the topology distance from Nodes A, B, and C to Community 1 is 0; and the topology distance from Node D to Community 1 is 1. The nodes are mapped to the axis according to the topology distance values.

To reduce the cardinality of the dimensions, a user-defined threshold is used. All nodes whose topology distances to a community are larger than the threshold are mapped to a position on its axis marked Far away. In Fig. 18, the threshold is 3. Such a mapping compresses topology information of a graph into a multidimensional space based on the community structure.

6.2 Stacked Parallel Coordinate

6.2.1 Visualization

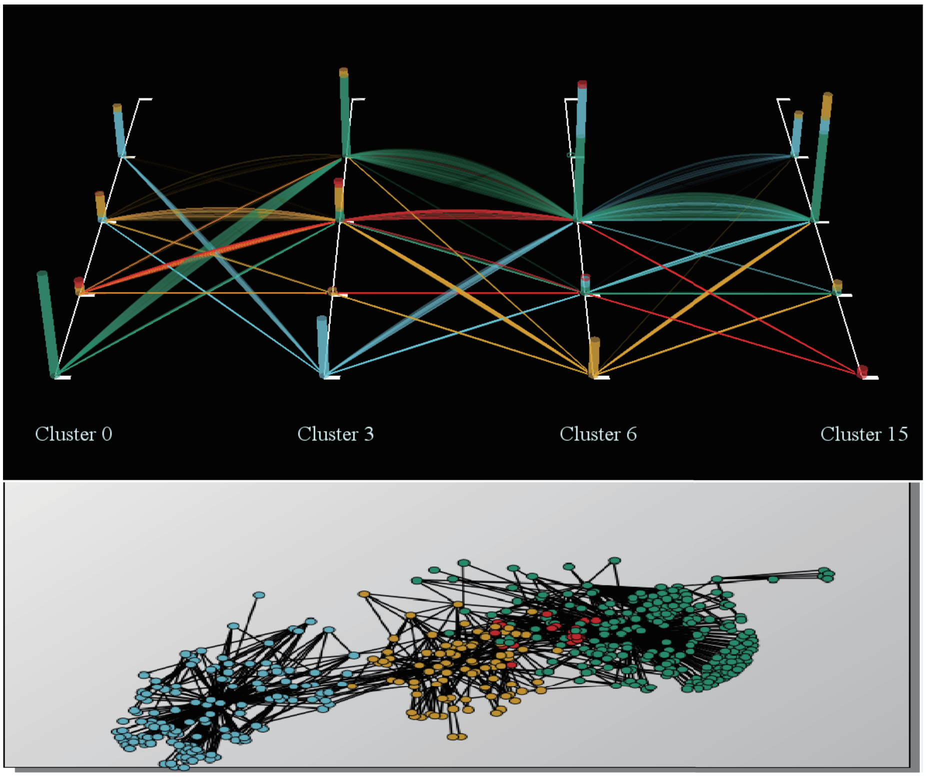

Figure 19: A subgraph of co-author network containing 4 communities.

Parallel Coordinates [30] is a popular multidimensional visualization technique with many merits. For example, it helps users trace the values of individual/groups of data items in the multidimensional space, identify clusters and outliers, examine relation-

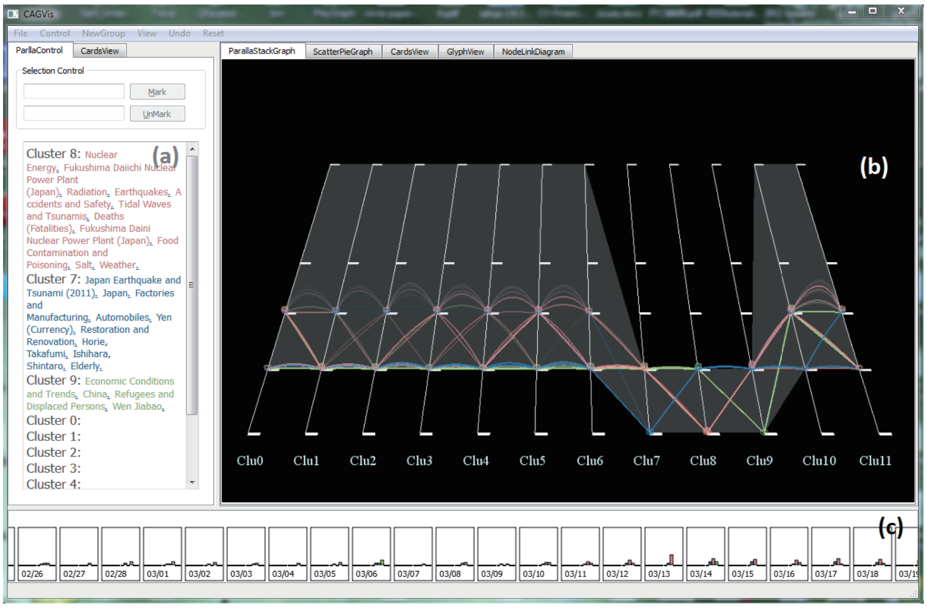

Figure 20: (a) Semantic box; (b) Visualization window; (c) Attribute panel.

ships between adjacent dimensions, and conduct N-dimensional brushing. These merits can benefit many community related graph exploration tasks, such as learning the distances from individual/groups of nodes to communities and identifying common neighbors of multiple communities.

However, the discrete nature of the topology distance causes serious clutter on traditional Parallel Coordinates when visualizing the multidimensional dataset mapped from a large graph. To address this challenge, we adopt a recent Parallel Coordinates variation [46] where the stacking method is used to reduce the clutter caused by discrete values. Its visualization is customized for graph visualization and a set of new interactions are developed for community related exploration tasks. We name this approach Stacked Parallel Coordinates.

Fig. 19 shows a subgraph of a co-author network (1,462 nodes and 6,075 links) [72] in the Stacked Parallel coordinates. Seventeen communities are detected using an external graph partition algorithm named AdjCluster [71] and four of them are visualized in Fig. 19. Each axis represents a community. A node is mapped to each community as a small cylinder positioned according to its topology distance to that community. The mappings of the same node on two adjacent communities are connected using a curve. Using the stacking method [46], the cylinders positioned at the same distance value on an axis are stacked and the curves with the same starting and ending points are also stacked. The cylinders and curves of nodes belonging to the same community have the same color. They are also placed adjacent to each other in the stacked cylinders and curves to better reveal community behaviors. The transparency of each link is proportional to the centrality of the node. In the initial view, the axes are ordered by the sizes of the communities they represent to help users examine large communities. Users can interactively change the order during their visual exploration. The view is drawn in a 3D space and distance 0 is facing the readers.

Several insights can be directly gained from Fig. 19. First, the heights of the stacked cylinders at distance 0 reveal that the sizes of the four communities vary a lot (C1). *Cluster 0* is the largest community and *Cluster 15* is the smallest community. Second, the connection among the communities can be examined (C2). For example, *Cluster 3* directly connects to *Cluster 6* and indirectly connects to *Cluster 0*. Most nodes in *Cluster 0* are far away from *Cluster 3*. Third, central and marginal communities can be identified (C4). For example, although *Cluster 15* is a small community, it connects to the rest of the graph well; it reaches most nodes with one or two edges. At the bottom of Fig. 19 is a Node-Link Diagram whose layout is generated using the force directed algorithm [18]. The same color encoding is used and the same communities are displayed. The overlapping nodes and links make connection patterns and quantitative differences between the communities difficult to find.

6.2.2 Interaction

Node Selection. In Stacked Parallel Coordinates, nodes can be interactively selected and highlighted. A selection set contains ids of selected nodes. It can be modified by the following operations:

(1) Stack selection. Users can select nodes with the same topology distance to a community, which is useful for tasks such as N4. In particular, the users can click a stack of cylinders to select all nodes they represent. These nodes have the same topology distance to a community. The users can also hold a function key or change the selection status through the menu to add, intersect, or remove nodes they clicked from the selection set. Thus the selection set becomes the result of boolean operations among multiple sets of nodes. This enables complex selections such as selecting common neighbors of multiple communities (N2).

(2) N-dimensional Brushing. N-dimensional brushing is a powerful interaction in Parallel Coordinates [39]. In Stacked Parallel Coordinates, users can manipulate a brush by defining its start and end points on each axis using mouse clicks. Nodes falling into this range on each axis will be selected. For example, in Fig. 20 (b), nodes close to cluster 7, cluster 8, and cluster 9 are selected by the brush. This operation is useful for tasks C1 $\sim$ C5.

(3) Selection by name. Users can type the label of a node into an edit box and

choose “mark” or “unmark” to select or unselect it. This operation is necessary for supporting tasks N1 $\sim$ N4.

Dimension Management. A large number of axes will clutter Parallel Coordinates. Thus the following interactions are provided so that users can control the axes displayed and their linear order: (1) Dimension reduction. Users can select communities to be displayed as axes through a community control dialog. For example, Fig. 19 is a visualization of the co-author network where only a subset of selected dimensions are displayed. Dimension reduction reduces the clutter caused by a large number of communities resulting from automatic analysis and helps users better focus on communities of interest.

(2) Dimension Ordering. In the initial view, axes are sorted by community sizes. After users identify a community of interest, they can place it to the left most of the display and sort the other communities according to their correlations to it. There are two options for sorting: (1) sorting by important nodes, assuming that users are most concerned about direct neighbors. In particular, communities are first sorted by the numbers of nodes directly connecting to the community of interest in descending order from left to right. Then, the communities with the same number of direct neighbors are sorted by the number of nodes that connect to the community of interest through two links. After that, the three step and four step neighbors are considered in turn; (2) sorting by the angles between the centroid of the community of interest and the centroids of other communities' in the graph's adjacency eigenspace. In [71], the authors presented an effective graph partition algorithm, *AdjCluster*, which projects node coordinates (along the orthogonal lines) onto the unit sphere in the spectral

space and then applies the classic k-means algorithm to find clusters. The angles between centroids of the identified clusters in the spectral space can then be used to capture the distances of communities. The smaller the angle is, the closer the two communities are.

6.3 Scatter Pie Matrices

6.3.1 Visualization

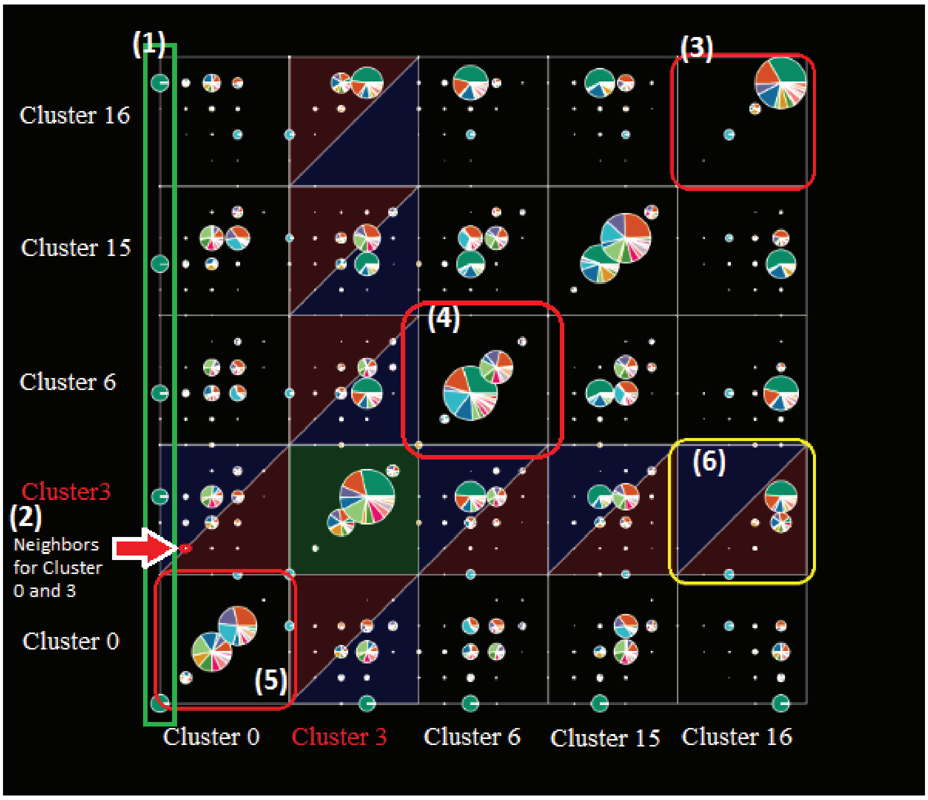

Figure 21: A subgraph with 5 communities in Scatter Pie Matrices.

Scatter plot matrix [61] are powerful by making it easy to examine all pairwise dimension relationships in one place. To better assist visual analysis of pairwise community relationships, we display the multidimensional mapping of a graph using a scatterplot matrix. Each axis represents one community and each data item represents a node. The values on an axis encode the topology distance from the nodes to the community. For unweighted graphs, the distance values are discrete values. Thus many nodes will be mapped to the same position in a traditional scatterplot matrix. To address this problem, we propose Scatter Pie Matrices, which is a variation of scatterplot matrices.

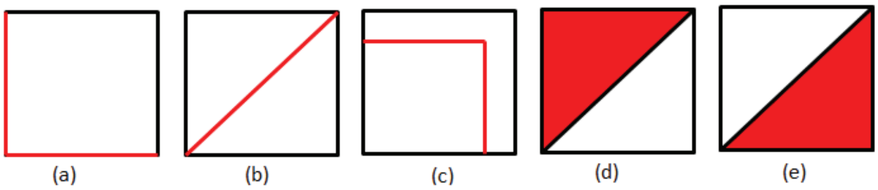

Figure 22: Regions' definition in Scatter Pie Matrices.

Fig. 21 shows an example of Scatter Pie Matrices. In Fig. 21, the X-axis and Y-axis of each plot represent two communities, named Community X and Community Y, respectively. We use $disX$ and $disY$ to denote the distances from a node to Community X and Community Y, respectively. The origin of both axes is at the left bottom corner of the plot. Nodes with $disX = 0$ and $disY = 0$ are projected here. Nodes with the same $disX$ and the $disY$ will be projected to the same position in the plot. They are aggregated and represented using a pie chart, whose radius is proportional to the number of them. Nodes from the same community are represented by a piece in the pie, whose color is a unique color assigned to that community and whose angle is proportional to the percentage of them in all nodes projected to that position.

In Scatter Pie Matrices, different types of insights can be obtained from different regions in a plot. Fig. 22 illustrates these regions. In Fig. 22 (a), the left and bottom border lines of a plot are highlighted. They indicate the positions where the distributions of the distances from Community X/Y members to Community Y/X can be observed. For example, by scanning the left borderlines circled in the green box (see Fig. 21 (1)), we can learn that Cluster 0 members are far away from Cluster16. Also, Cluster 0 is closer to Cluster 6 and Cluster 15 than Cluster 3 and Cluster 16. Thus this region conveys information about how one cluster is connected to other clusters (C2).

In Fig. 22 (b), the diagonal line of a plot is highlighted. Nodes with the same distance to Community X and Community Y are projected here. For example, the common direct neighbors of Cluster 3 and Cluster 0 (N2) are explicitly represented by the pie highlighted in red (see Fig. 21 (2)). In each plot on the bottom left to top right diagonal line of the matrix, the X and Y axes represent the same community. The diagonal line in such a plot provides a clear picture of how the distances between the nodes and this community are distributed. For example, from Fig. 21 (3), we observe that Cluster 16 is a marginal cluster (C4) since it is far away from most nodes. On the contrary, Cluster 6 and 0 are central communities (C4) since they can access a large number of nodes in two steps (see the Fig. 21 (4) and (5)). It is interesting that Cluster 6 is a small community according to the pie at the bottom left of the same plot (C1).

Fig. 22 (c) highlights the “Far away” distance. This information helps users identify nodes and communities that are far away from the X or Y community. In Fig. 22 (d), the red region displays all nodes that are closer to Community X than Community Y. Similarly, in Fig. 22 (e), the red region displays all nodes that are closer to Community Y than Community X. By comparing these two regions, users can compare two communities with regard to their relationships to other communities (C2). For example, Fig. 21 (6) reveals that Cluster 3 has a more central position than Cluster 16.

6.3.2 Interactions

Users can select nodes with desired distance values by directly clicking the pies. For example, clicking the pie highlighted in red (Fig.Fig. 21 (2))selects all common direct neighbors of Cluster 3 and Cluster 0. Boolean operations on the selection set can be conducted in a similar way as in Stacked Parallel Coordinates.

According to our pilot study, Scatter Pie Matrices is so information rich that users can be overwhelmed. To address this problem, Scatter Pie Matrices provides:

Region selection. Each figure in Fig. 22 is displayed as an icon in the interface. Users can click an icon to display its region while hiding all other regions. Thus users can focus on the information helpful to their current task.

Community selection. When one community is selected by clicking the community label, all related plots will be highlighted as follows: nodes displayed in a red region are closer to this community than to the other axis of the plot and blue region means the opposite; green means that it is the plot showing the distribution of distances between this community and all nodes. For example, Cluster 3 is selected in Fig. 21.

6.4 Prototype

We have implemented Stacked Parallel Coordinates and Scatter Pie Matrices in a fully working prototype for community based graph visual exploration. The prototype is named CAGVis. Beside the distance information convey by these two approaches, other information useful for community related tasks, such as labels and attributes of the nodes, are also displayed in CAGVis, using coordinated auxiliary views. As shown in Fig. 20, CAGVis has three windows:(a) a semantic box; (b) a main window; and (c) an attribute panel. The main window is where Stacked Parallel Coordinates or Scatter Pie Matrices are displayed. Labels of selected nodes are displayed in the semantic box. Binary node attributes are visualized in the attribute panel. Each attribute plot displays nodes whose values are 1 in the attribute. The nodes are aggregated into histograms, each bar representing a community, in the same color as the community in the main window. In this way, users can examine the distribution of communities in the attributes in a glance. Users can click an attribute plot to select all nodes displayed in it.

Besides the aforementioned interactions, the following interactions are provided in CAGVis to support iterative user exploration. Note that the first three interactions were proven very useful in PIWI [72]:

(1) *New communities.* This function allows users to create new communities by their own criteria during an iterative visual exploration. A new community is constructed when clicking the “NewGroup” button from the menu bar. It includes all nodes in the current selection set. It is visualized and interactively explored in the same way as automatically detected communities. Users can create any number of new communities and delete them at any time.

(2) *View edges.* To learn the semantic information carried by the links, a “ViewEdge” button is offered on the menu bar. It will invoke a window displaying labels of all

edges connecting the selected nodes.

(3) *Save subgraphs.* Users can save sub-graphs consisting of selected nodes and reload them into the system for further investigation.

(4) *Undo* and *Reset* allow users to undo previous operations and recover the initial view.

CAGVis supports the following visual exploration pipeline: CAGVis conducts community detection and distance calculation automatically; users interactively explore the communities and their relationships and select nodes of interest; users interactively create new communities for further visual exploration.

6.5 Case Studies

6.5.1 Datasets

The following real datasets are used in the case studies:

(1) The NYT graph [72]. It is a tag co-occurrence network with 1,200 nodes (tags) and 7,042 edges (co-occurrence of two tags in at least one article). It conveys the tag co-occurrence information of 1,078 New York Times (NYT) [2] world news articles published from February 22, 2011 to April 25, 2011. It has 36 binary node attributes carrying categorical and temporal information. Its communities reflect the major news events reported in the news corpus.

(2) The Infovis co-author network [72]. It has 1,462 nodes (authors) and 6,075 edges (co-authorships). The network was generated using 4,526 bibliography entries from the DBLP Computer Science Bibliography in 2011. They are authored or coauthored by 36 researchers who served on the program committee of the IEEE

conference on Information Visualization. Its communities are research groups within which researchers collaborate intensively. The node attributes are keywords extracted from the authors' paper and can be considered as the research areas that the authors work in.

(3) The web spam graph. It is a subgraph extracted from the Web Spam Challenge 2007 data [4]. The original data set contains 114,529 hosts in the .UK domain and 1,836,228 edges among them. It also contains 341 hosts labeled as spam. We extracted a subgraph by starting from one of the spam hosts and crawling all its neighbors within 2 steps. The extracted subgraph contains 2,601 nodes (7 of them are spam hosts) and 17,561 edges.

Community detection has been conducted on these three networks using AdjCluster [71].

6.5.2 Case Study 1: The NYT Graph

Users load the NYT graph into Stacked Parallel Coordinates. By examining the node labels of the significant communities from the semantic box, they identify that Cluster 5 consists of tags such as "Libya" and "Freedom of the Press". It is about Libyan civil war, which is interesting to the users. They click the stack of cylinders at distance 0 of Cluster 5 to select its members and hide other nodes. They then sort the communities according to the numbers of their direct neighbors in Cluster 5. As shown in Fig. 23 (a), most nodes in Cluster 5 can reach other clusters in 2 steps or less (C2). In addition, Cluster 3 and Cluster 6 have the largest numbers of direct neighbors in Cluster 5.

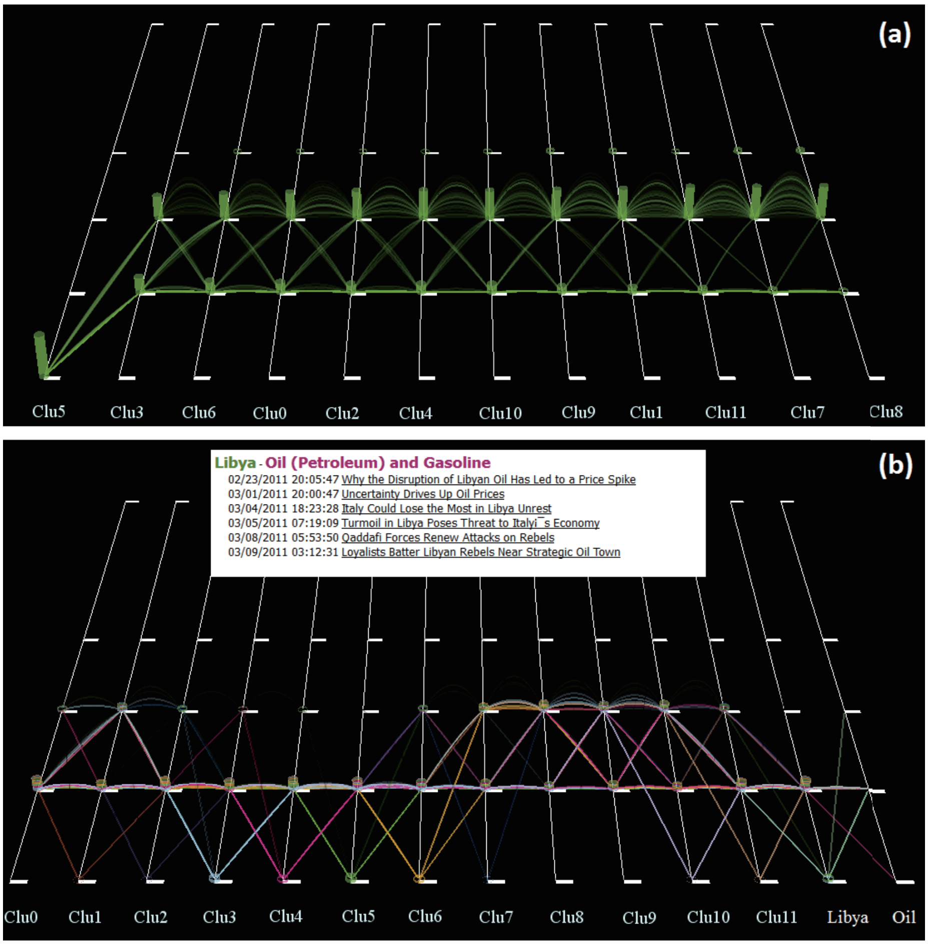

Figure 23: (a) Dimension ordering. (b) Bridging nodes identification.

The users examine Cluster 5 and how its relationships to other communities evolve (T2, T4) using the attribute plots conveying temporal information. They show histograms of tags which were used in particular dates. The temporal attribute plots show that this event has a burst on 02/27 and 03/04. The users select members of Cluster 5 appearing in 02/27 news by intersecting the plot for 02/27 with the original selection set. They then examine the news articles where the selected nodes co-occurred using the view edge function. They find out that the rebels in Libya gained their power on that day. Similarly, they examine Cluster 5 members day by day to learn how the Libyan civil war was reported by NYT over time (T1).

The users also know that Libya is famous for its oil. Did the war affect the oil market and, if yes, how (C2, C3)? They create two communities, one containing the tag Libya and the other containing the tag oil (as shown in Fig. 23 (b)). Nodes within distance 1 to the new communities and Cluster 5 are easily selected using N-dimensional brushing. Examining the edges among these nodes, the users learn that the war in Libya caused the price spike of oil on February 23rd, 2011 (see Fig. 23 (b)).

Two other news events that trigger the users' interest are Japan's earthquake (Cluster 7) and nuclear leakage (Cluster 8). Stacked Parallel Coordinates reveals that these two communities are closely related to several other communities (C2), such as a community on Economic conditions and trends (Cluster 9). The users select nodes closely related to these three clusters using N-dimensional brushing (see Fig. 20). By viewing their edges, they identify news articles such as "Japan's Industrial Heart Escapes Heaviest Blows" and "Disruptions of Power and Water Threaten Japan's Economy".

6.5.3 Case Study 2: The InfoVis Co-author network

Users load the Infovis co-author network into Stacked Parallel Coordinates and examine the labels of its significant communities. Several important research communities quickly catch their attention (C4), such as one with Ben Shneiderman from HCIL of University of Maryland, one with Daniel Keim from University of Konstanz, and one with Mary Czerwinski and George G. Robertson from Microsoft Research. How is the HCIL community (Cluster 0) connected to the rest of the graph (C2)? To get the answer, the users select members of Cluster 0 using stack selection, as shown

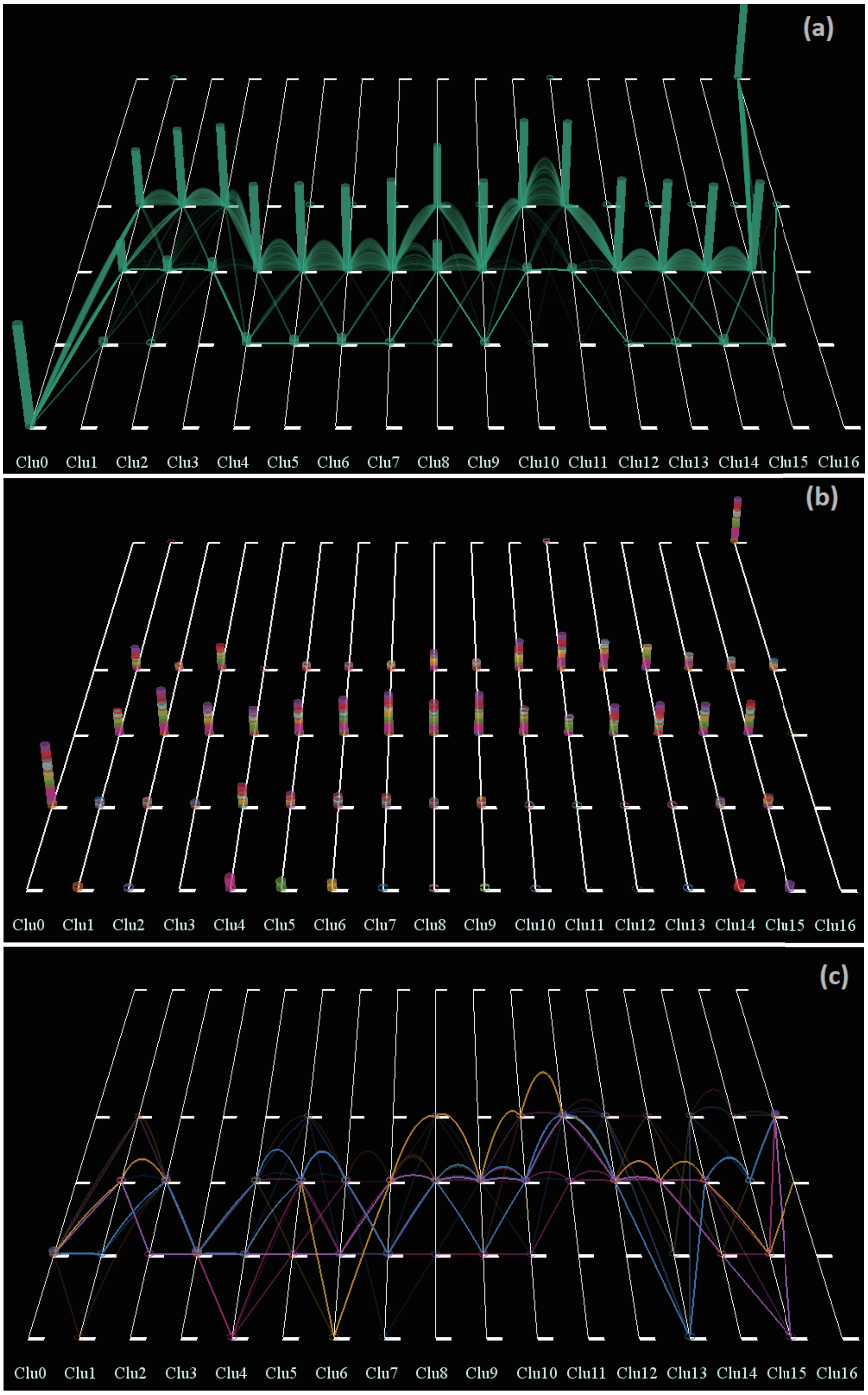

Figure 24: Analysis of the HCIL research community's connectivity.

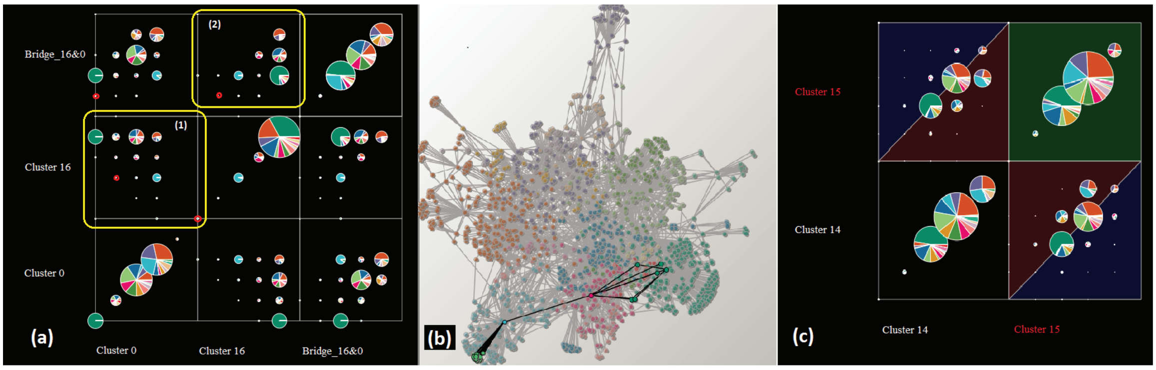

Figure 25: (a), (b) Bridge nodes identification. (c) Cluster comparison.

in Fig. 24 (a). They notice that the majority of its members are 3 or less steps away from all other communities except Cluster 16. From the semantic box, the users know that many people in Cluster 16 are from Intel Research.

They further select the direct neighborhood of Cluster 0 by clicking the stack on distance 1 of the Cluster 0 axis. They turn off the curves to see the distribution of the cylinders more clearly, as shown in Fig. 24 (b) (C2). It can be seen that many direct neighbors of Cluster 0 are from Cluster 4 (Microsoft Research) and Cluster 5 (Georgia Tech Information Interfaces Research Group). Interestingly, although Cluster 3 is not directly connected to Cluster 0, these two communities have several common neighbors, as indicated by the stack at distance 1 of Cluster 3. Intersecting this stack with the original selection set highlights these common neighbors, as shown in Fig. 24 (c). Names such as Mary Czerwinski and John T. Stasko pop up in the semantic box (N2).

The users then display the graph in Scatter Pie Matrices to examine common neighbors (see Fig. 25). They notice that there is no common direct neighbors of Cluster 0 and Cluster 6 (Fig. 25 (a) (1)). How to bridge these two communities?

They find that some nodes directly connecting to Cluster 0 can reach Cluster 16 in two steps, as indicated by the pie highlighted in red in Fig. 25 (a) (1). The users select them by clicking that pie. Switching back to Stacked Parallel Coordinates, they identify that it is a node in Cluster 6. How to reach Cluster 16 from this node? To answer this question, the users create a new community called "Bridge16and0" for this node. Common neighbors between it and Cluster 16 can be selected from the highlighted pie in Fig. 25 (a) (2). By looking at the coordinated node-link diagram (Fig. 25 (b)), where the selected nodes are highlighted in red in Fig. 25 (a), the users confirm that they bridge Cluster 0 and Cluster 16.

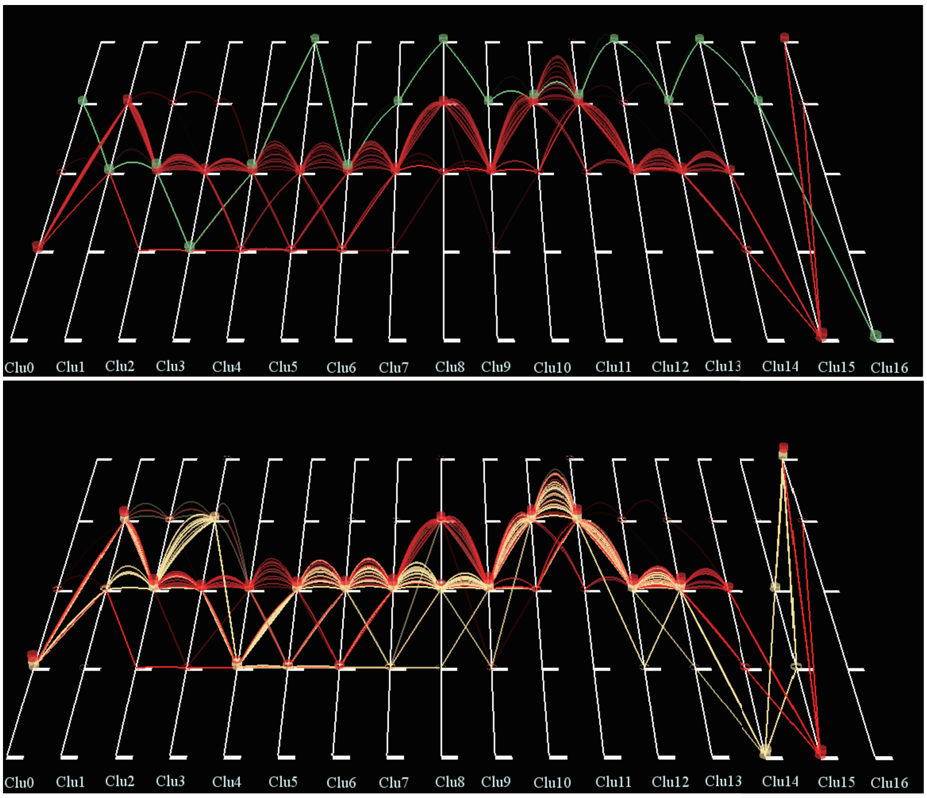

Figure 26: Connectivity comparison among Cluster 14, 15, and 16.

From the initial view where communities are sorted by size, the users notice that Clusters 14, 15, and 16 are relatively small communities. The users want to compare their connectivity to the rest of the graph (C2). In Fig. 25 (c), by selecting Cluster 15 in Scatter Pie Matrices, they quickly learn that Cluster 15 is more central than Cluster 14, since there are more nodes displayed in the red region than in the blue region. The comparison can also be conducted in Stacked Parallel Coordinates. For example, the top figure in Fig. 26 clearly reveals that Cluster 15 is better connected to the rest of the graph than Cluster 16. The bottom figure in Fig. 26 compares Cluster 14 and Cluster 15. Comparing it with Fig. 25 (c), the users find that Scatter Pie Matrices provide a better overview while Stacked Parallel Coordinates provides finer details in the comparison task.

6.5.4 Case Study 3: Fraud detection verification

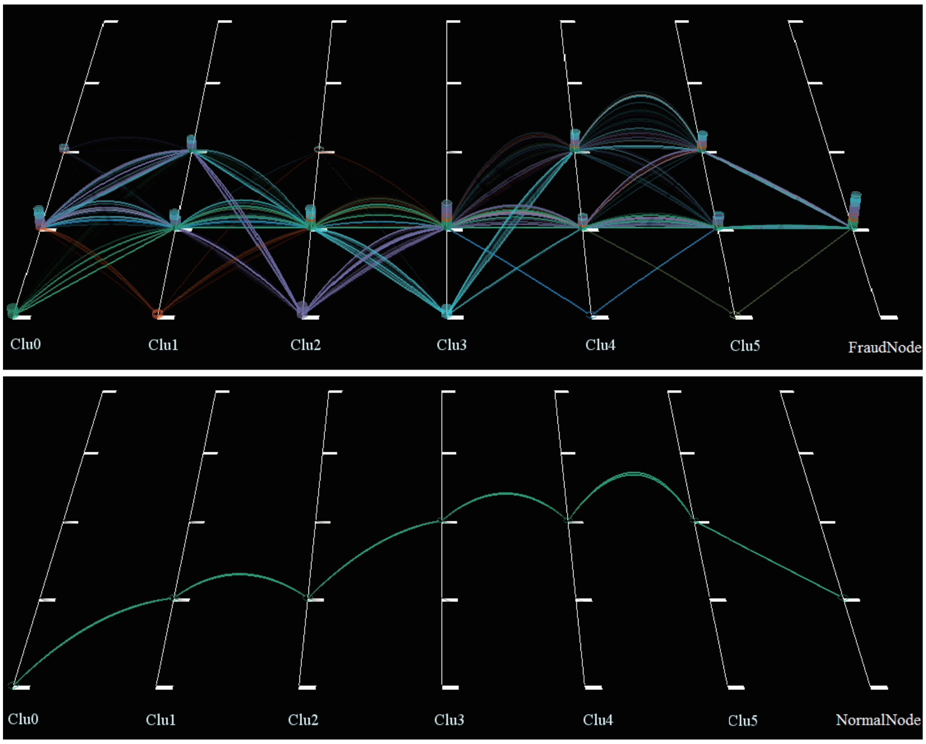

Figure 27: Verifying fraud nodes using Stacked Parallel Coordinates.

Social networks are vulnerable to various attacks such as spam emails, viral marketing, Sybil attacks, and denial of service. Algorithms based on topological structures have recently been designed to discover the perpetrators of these attacks (e.g., [74]). In many attacks, fraud nodes often randomly connect to users in many different communities. On the contrary, regular nodes (normal users) mainly connect to users from one or a few communities. A challenge is to evaluate the result from an automatic fraud detection algorithm if there is no ground truth. This case study shows how to conduct this task using CAGVis.

Users visualize the web spam graph using Stacked Parallel Coordinates to examine whether the fraud nodes exhibit the properties as they should be. The fraud/normal nodes are easily selected using the attribute plot. The users create a new community named "fraudnode" for a fraud node and highlight its direct neighbors by clicking the stack at distance 1 on its axis (see the top figure in Fig. 27). The users then create a new community named "normalnode" for a normal node and also highlighted its direct neighbors (see the bottom figure in Fig. 27) (N3). Comparing the top and bottom figures in Fig. 27, the users can clearly distinguish a fraud node from a normal node.

6.6 User Study

6.6.1 Setup

To understand the usability and utility of our techniques, I conducted a user study following the methodology proposed in [42]. In this study, CAGVis and a state-of-the-art graph visualization tool NodeXL (version 1.0.1.196 released on December 2011) [58] were compared using a within subject user study design. NodeXL uses Node-Link Diagrams and coordinated Excel worksheets to display graphs and their

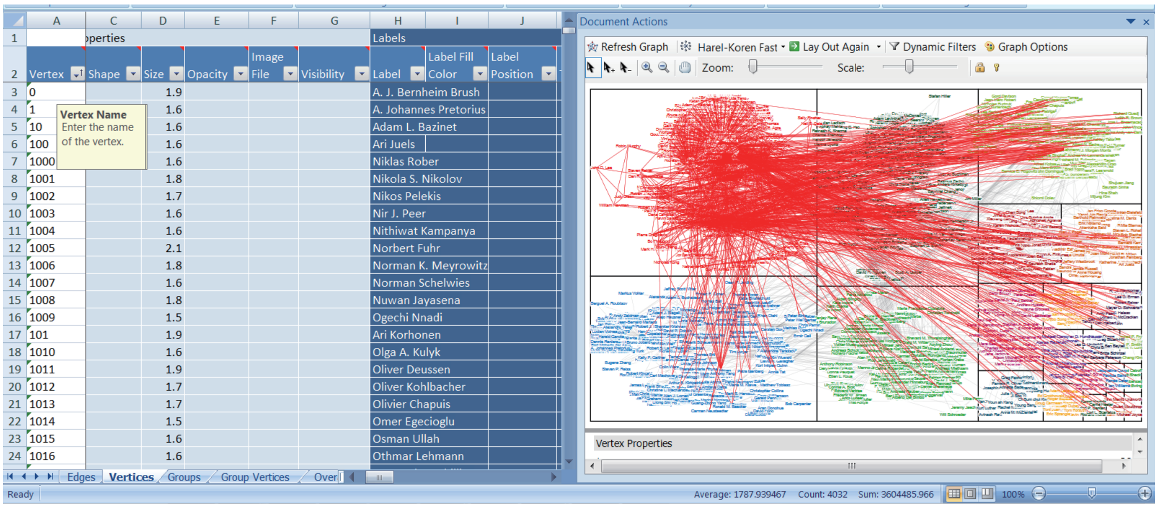

Figure 28: The InfoVis co-author network in NodeXL [58].

nodes, edges, and group details (see Fig. 28). Eight subjects were recruited. Seven of them were computer science students, and the other one was an electronic engineering graduate student. Their ages ranged from 21 to 31. They were not familiar with any of the systems. However, they had previous experiences with visualization tools and Excel. The subjects were evenly divided into two groups based on their expertise and were balanced by the order of system usage.

Table 11: User study tasks.

Task	Description
N1	Who are the leaders in the largest and the smallest communities?
N2	Identify the bridging nodes between the largest community and the other communities in its step 1, 2, and 3 neighborhoods.
N3	How does Catherine Plaisant collaborate with the research communities?
N4	Find the nodes that are two steps away from the neighborhood of Community 0.
N5	List three communities which work in “visual database ”.
L2	In which research area Community 0 has more collaboration, “graphical user ” or “virtual reality ”?
C2	Which community more closely collaborates with other comm -unities, Community 15 or Community 14?
C3	How are researchers in the area “rendering” distributed in the graph?
C4	List one central and one marginal community. Explain why.

Datasets: The Infovis co-author network (see Section 3.6) and two networks generated by randomizing its node labels were used in the user study. The three datasets have the same topology structure and different node labels. The same community detection results (generated using the AdjCluster [71]) were used in both systems. The Infovis co-author network was used in the training section and the other two datasets were evenly used in the two systems in the formal testing. In this way, the learning effect could be minimized in the within subject design.

Tasks: The proposed task taxonomy was used to develop the tasks (see Table 11). Rather than testing all tasks in the taxonomy, we focused on a subset of tasks requiring distance information, since representing distance information is the most significant feature of the proposed approaches. These tasks cover a majority of tasks in the taxonomy.

6.6.2 Procedure

The subjects conducted the study one by one in a laboratory setting. In each study, a subject used the two systems one by one. For each system, there was a training section followed by a formal testing section.

Training: First, an instructor demonstrated the system to the subject using tasks similar (but not identical) to the tasks that were used in the formal testing. Specifically, the instructor emphasized the following NodeXL interactions that are the most useful for conducting the tasks: (1) Community selection. Community can be directly selected from the "Group" worksheet by clicking the corresponding group or can be selected from the Node-Link Diagram by drawing a rectangle. A right click on a selected node will trigger a menu from which all neighbors of this node can be selected; (2) Dynamic filtering. Each column of the "Vertices" worksheet, such as degree or other node attributes, can be used for dynamic filtering. Also, a dynamic filtering dialog is offered in NodeXL which provides statistic information such as centrality and degree; (3) Searching and sorting. Users can search a node by name in Excel. They can also sort nodes by any node attributes in Excel. After the demonstration, the subject spent as much time as she wanted to interact with the system. She was encouraged to "think aloud" and ask questions. All the usability issues she mentioned were recorded by the instructor. To make sure that the subject had built a correct mental model of the system, she was asked to describe the system when she exited the training section. Corrections were given by the instructor when needed. The average training time for CAGVis and NodeXL were 36.6 and 35 minutes, respectively.

Formal testing: In this section, the subject conducted the tasks listed in Table 11 one by one on the test dataset. Task completion time was recorded for each task. After each task, the subject rated her level of cognitive load using a modified version of the NASA task load index questionnaire [22]. She was also asked to comment on the excitements or difficulties she met.

The subject was asked to compare the two systems and gave her comments after both systems were tested. The whole study took around 2 hours for each subject.

6.6.3 Results

I analyzed the results from three aspects: (1) Time and accuracy; (2) Cognitive load; and (3) Observations and comments.

Time and accuracy. User performances with the two systems were summarized in Table 12. The subjects had shorter average task completion time and more accurate

Table 12: Time and correctness results

	CAGVis		NodeXL	
	Time (min)	Correctness (%)	Time (min)	Correctness (%)
N1	1.11	100	5.78	100
N2	5.64	100	13.40	48
N3	1.19	100	6.00	50
N4	1.17	100	2.13	12
N5	1.17	100	3.96	66
L2	0.88	100	4.64	90
C2	0.82	100	2.53	66
C3	0.95	100	1.35	90
C4	0.66	100	0.77	97

answers for all tasks using CAGVis than using NodeXL. Except for the last task (Time: p = 0.2; Correctness: p = 0.1), ANOVA analysis revealed that all the time and correctness differences were significant between the two systems ($p < 0.05$). All subjects were able to solve all tasks with a 100% correctness rate using CAGVis. Four tasks (N1, N3, N4, N5) were completed within 1.2 minutes with CAGVis. The rest four tasks (L2, C2, C3, C4) took less than 1 minute with CAGVis. N2 took more time since the task itself was more complex than other tasks.

Table 13: Cognitive load results (scale: 1-5).

Task		MD	PD	TD	OP	EF	FR
N1	CAGVis	**1 (0.0)**	**1.125 (0.13)**	**1 (0)**	**1 (0)**	**1.125 (0.13)**	1 (0)
	NodeXL	**3.125 (0.41)**	**3 (0.88)**	**2.5 (0.29)**	**1.75 (0.21)**	**3.125 (0.98)**	1.75 (1.07)
N2	CAGVis	**3.375 (0.55)**	3.25 (0.79)	3.25 (0.5)	**1.625 (0.55)**	**3.375 (0.55)**	**1.875 (0.70)**
	NodeXL	**4.75 (0.5)**	3.25 (1.07)	3.5 (0.29)	**4.375 (0.27)**	**4.5 (0.57)**	**3.875 (0.70)**
N3	CAGVis	**1.375 (0.27)**	**1 (0)**	**1.5 (0.29)**	**1 (0)**	**1.25 (0.21)**	**1 (0)**
	NodeXL	**4.375 (0.27)**	**3.75 (1.64)**	**3.5 (0.29)**	**3.875 (0.41)**	**4.25 (0.79)**	**4.375 (0.27)**
N4	CAGVis	**1 (0)**	**1.25 (0.21)**	**1 (0)**	**1 (0)**	**1 (0)**	**1 (0)**
	NodeXL	**5 (0)**	**4.5 (2)**	**5 (0)**	**5 (0)**	**5 (0)**	**5 (0)**
N5	CAGVis	**1 (0)**	**1.125 (0.13)**	**1 (0)**	1 (0)	**1.125 (0.13)**	**1 (0)**
	NodeXL	**2.75 (0.79)**	**3 (0.86)**	**2 (0.29)**	1.75 (1.93)	**2.875 (0.70)**	**1.75 (0.21)**
L2	CAGVis	**1 (0)**	**1.25 (0.21)**	**1 (0)**	**1 (0)**	**1.25 (0.21)**	**1 (0)**
	NodeXL	**2.5 (0.29)**	**3 (0.86)**	**3.25 (0.21)**	**1.875 (0.13)**	**2.5 (0.29)**	**2.75 (0.21)**
C2	CAGVis	**1.625 (0.27)**	**1.875 (0.41)**	**1.375 (0.27)**	**1.375 (0.27)**	**1.875 (0.41)**	**1 (0)**
	NodeXL	**5 (0)**	**4.5 (2)**	**5 (0)**	**5 (0)**	**5 (0)**	**5 (0)**
C3	CAGVis	1.625 (0.27)	**1 (0)**	**1.375 (0.27)**	1.125 (0.13)	**1.375 (0.27)**	1 (0)
	NodeXL	2.25 (0.21)	**3 (0.86)**	**2 (0)**	1.5 (0.57)	**2.625 (0.27)**	1.5 (0.57)
C4	CAGVis	1.125 (0.13)	1.25 (0.21)	1.125 (0.13)	1 (0)	1.25 (0.21)	1 (0)
	NodeXL	1.75 (1.93)	2 (0.57)	1.875 (1.84)	1.5 (2)	2.25 (1.64)	1.5 (2)

Cognitive load. The cognitive load ratings were reported in Table 13. In the table, each data is represented as "Mean (Standard Deviation)". MD represents mental demand; PD represents physical demand; TD represents time demand; OP: performance; EF represents effort; FR represents frustration. An ANOVA analysis was conducted for each cell in this table. Factors that were significantly different ($p < 0.05$) between CAGVis and NodeXL were shown in bold. CAGVis outperformed NodeXL in almost every cognitive load aspect in all tasks. Meaningful cognitive comments (156 comments for CAGVis and 174 comments for NodeXL) were identified by the instructor. These comments were encoded into “Encouragement”, “Neutral” and “Frustration” by two researchers (96% agreement rate). The conflicting coding was refined according to their discussion. Overall, the subjects had 9% comments in “Engagement”, 86% comments in “Neutral”, and 5% comments in ”Frustration” for CAGVis. The subjects had 55% comments in “Neutral”, 45% comments in “Frustration”, and no comment in “Engagement” for NodeXL. This indicated more work load resulting from the complexity of the dataset and less work load resulting from the lack of support to the tasks with CAGVis than with NodeXL. Thus, we conclude that the tasks were better supported in CAGVis than in NodeXL.

Observations and comments. (1) Stacked Parallel Coordinates. Training was necessary for Stacked Parallel Coordinates users. When the instructor demonstrated Stacked Parallel Coordinates in the training section, a coordinated Node-Link Diagram was displayed aside Stacked Parallel Coordinates. It helped the subjects understand Stacked Parallel Coordinates.

I observed that the subjects liked to select nodes at different distances to a community to how they were distributed in the neighborhoods of other communities. They gave comments such as:

"This is interesting and it makes connection vivid to me."

"This is simple, appealing to the eye, and presents a large amount of useful data very quickly."

"Every distance's distribution is easy to get and, what is important, is clear to understand."

Several usability issues were identified from the study. First, brushing needs improvement. The distance 0 stacks are frequently used in the selections. Thus the subjects desired options to automatically excluding or including the distance 0 stacks when a dimension was brushed. Second, it was desired to allow users turn on/off the curves and cylinders independently. Subjects asked for such options since they thought those options could keep them more focused. These issues have been addressed in the redesigned system shown in the case studies.

(2) Scatter Pie Matrices. We observed learning curves for Scatter Pie Matrices. It took the subjects a non-trivial amount of time to understand the information that can be obtained from it. Then, they were able to make use of it after some practices. Scatter Pie Matrices were used much less than Stacked Parallel Coordinates in the study. More user studies are needed here. One interesting finding was that the subjects liked using Scatter Pie Matrices as a navigation map for Stacked Parallel Coordinates. They commented that:

"Good to quickly do the selection from Scatter Pie Matrices, see the flow from Stacked Parallel Coordinates, and (then) see the node labels immediately from the semantic box."

They also liked the region selection because *"it separates different information* *otherwise distracting".*

(3) NodeXL. Although NodeXL fits user perception of graphs better and is easier to learn, user comments suggested that the community related exploration tasks were not well supported in NodeXL. There were comments such as:

"There is no way to easily view the second neighborhood of a community. I can technically look at the nodes on the visualization view, but the view is disorganized, and there is no way to sort the neighbors in a particular community."

"I could manually click each node that a community is linked to and find which nodes each of those are linked to, but that would take ages."

"This task was not difficult, but I could not easily find the answers since the nodes were fading quickly. Actually I'm not sure if these results are correct."

6.7 Discussion

In this chapter, I have introduced two graph visualization techniques focused on community related tasks. By mapping topology information into multidimensional visualization techniques and combining with tailored interactions, it is possible to support various community analysis tasks, such tasks are largely ignored in existing graph visualization systems and task taxonomy. Our formal user study suggests that there is a great value of the system in helping users conduct community analysis.

CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 Conclusion of Completed Work

In this dissertation, I proposed an EGV system evaluation methodology. Rooted in cognitive load theory, the methodology provides practical means to control the variability among human subjects and measure the complexity in the underlying cognitive process. These methods can be feasibly adapted in EGV system evaluations, as our user study has demonstrated. Based on the insights from the evaluation, PIWI, an EGV system built on community structure, is proposed. It is a novel visualization system that allows users to interactively explore large multivariate graphs based on their community structure. Its "tag clouds + vertex plots" visualization presents the semantics, attributes, and communities of a large graph in an uncluttered way. Labels of a large number of nodes can be examined within their communities and neighborhood context simultaneously. From the researches in community related graph visualizations, I realized that a systematic approach must be employed: important tasks need to be discovered; major challenges need to be identified; effective visualization techniques addressing these challenges need to be developed. Thus, by surveying the existing work of graph analysis in data mining, sociology and visualization domains, I propose a taxonomy of community related graph visualization tasks, which has been verified through an expert survey. Furthermore, I tailored two multidimensional visualization techniques into graph visualizations, which aim at addressing the community related graph visualization tasks in the taxonomy. The main idea is to analogize community based graph visualization to multidimensional visualization. In particular, the communities are analogized to dimensions in a multidimensional dataset and the nodes are analogized to data items in a multidimensional dataset. Thus, the distance from a node to a community can be expressed by the value of a data item on a dimension and the correlation between two communities is mapped to correlation between two dimensions. Therefore, existing multidimensional visualization and interaction techniques can be customized to support community related graph visualization tasks.

7.2 Future Work

In the future, I plan to conduct a formal survey on existing systems in graph community analysis. Also, I will improve Stacked Parallel Coordinate and Scatter Pie Matrix along the following directions: (1) Interactions. More user studies are needed in the future to find out users' information needs and in which way they prefer to interact with the visualizations. (2) Clustering algorithms. As targets at community analysis, the two views rely on the initial clustering results. We plan to give users more choices on clustering algorithms in the future. In this way, they can see difference of clustering algorithms and explore the graph from difference aspects.

Also, I will continue use the topology mapping to see whether other multidimensional techniques can be made use of. A potential one is glyphs. It can present each node as a glyph, where each axe is the community in the graph and the length of the axe represents the node's topology distance to that community. The user could select a glyph by a mouse click. The Glyph View would be helpful to gain node level details, to examine sub groups in one community, and to find out boundary nodes among communities.

REFERENCES

[1] Carnegie Mellon University, http://www.cmu.edu/teaching/designteach /teach/priorknowledge.html.

[2] The new york times. http://www.nytimes.com.

[3] Tulip software. www.tulip-software.org.

[4] Web spam challenge, 2007. http://webspam.lip6.fr/wiki/pmwiki.php.

[5] Abello, J., and van Ham, F. Matrix zoom: A visual interface to semi-external graphs. In *Proceedings of the IEEE Symposium on Information Visualization* (Mar. 2004), IEEE Computer Society, pp. 183–190.

[6] ah Kang, Y., and Stasko, J. Lightweight task/application performance using single versus multiple monitors: A comparative study. In *Proc. Graphics Interface Conference* (June 2008), Canadian Information Processing Society, pp. 17–24.

[7] Anderson, E., Potter, K., Matzen, L., Shepherd, J., Preston, G., and Silva, C. A user study of visualization effectiveness using EEG and cognitive load. *Eurographics/IEEE Symposium on Visualization 2011 30*, 6 (2011), 36–104.

[8] Aris, A., and Shneiderman, B. Designing semantic substrates for visual network exploration. *Information Visualization 6*, 4 (2007), 281–300.

[9] Bausch, P., and Bumgardner, J. *Make a Flickr-Style Tag Cloud*, second ed. O'Reilly Press, 2006.

[10] Bertin, J. *Semiology of Graphics: Diagrams, Networks, Maps.* University of Wisconsin Press, Reading, Mass., 1983.

[11] Bezerianos1, A., Chevalier2, F., Dragicevic2, P., Elmqvist3, N., and Fekete2, J. Graphdice: A system for exploring multivariate social networks. *IEEE Trans. Vis. Comput. Graph. 29*, 3 (2010), 1100–1108.

[12] Brunken, R., Plass, J., and Leutner, D. Direct measurement of cognitive load in multimedia learning. *Educational Psychologist 38*, 1 (2003), 53–61.

[13] Chau, D. H., Kittur, A., Hong, J. I., and Faloutsos, C. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), pp. 167–176.

[14] Chuang, J., Ramage, D., Manning, C. D., and Heer, J. Interpretation and trust: Designing model-driven visualizations for text analysis. In *ACM Human Factors in Computing Systems (CHI)* (2012), pp. 443–452.

[15] Clauset, A., Newman, M. E. J., and Moore, C. Finding community structure in very large networks. *Physical Review E. 70:066111* (2004).

[16] Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., and Woodhull, G. Graphviz and dynagraph static and dynamic graph drawing tools. In *Graph Drawing Software* (2003), Springer-Verlag, pp. 127–148.

[17] Fortunato, S. Community detection in graphs. *PHYSICS REPORTS 486* (2010), 75–174.

[18] Fruchterman, T. M. J., and Reingold, E. M. Graph drawing by force-directed placement. *Softw. Pract. Exper. 21*, 11 (1991), 1129–1164.

[19] Fua, Y.-H., Ward, M. O., and Rundensteiner, E. A. Hierarchical parallel coordinates for exploration of large datasets. IEEE Computer Society Press, pp. 43–50.

[20] Girvan, M., and Newman, M. Community structure in social and biological networks. *Proc. National Academy of Sciences of the United States of America 99*, 12 (2002), 7821–7826.

[21] Green, T. M., Jeong, D. H., and Fisher, B. Using personality factors to predict interface learning performance. In *Proc. of Hawaii International Conference on System Sciences* (2010), pp. 1–10.

[22] Hart, S. G., and Stavenland, L. E. Development of the NASA-TLX (task load index): Results of empirical and theoretical research. In *Proc. of Human Mental Workload* (1988), pp. 139–183.

[23] He, W., Erdelez, S., Wang, F.-K., and Shyu, C.-R. The effects of conceptual description and search practice on users' mental models and information seeking in a case-based reasoning retrieval system. *Information Processing and Management 44*, 1 (Jan. 2008), 294–309.

[24] Hearst, M., Elliott, A., English, J., Sinha, R., Swearinged, K., and Yee, K.-P. Finding the flow in web site search. *Communications of the ACM 45*, 9 (2002), 42–49.

[25] Heer, J., and Boyd, D. Vizster: Visualizing online social networks. *Proc. IEEE Symposium on Information Visualization* (2005), 5–12.

[26] Henry, N., daniel Fekete, J., and Mcguffin, M. J. Nodetrix: a hybrid visualization of social networks. *IEEE Trans. Vis. Comput. Graph. 13*, 6 (2007), 1302–1309.

[27] Henry, N., and Fekete, J.-D. Matrixexplorer: a dual-representation system to explore social networks. *IEEE Trans. Vis. Comput. Graph. 12*, 1 (2006), 677–684.

[28] Herman, I., Society, I. C., Melanon, G., and Marshall, M. S. Graph visualization and navigation in information visualization: a survey. *IEEE Trans. Vis. Comput. Graph. 6*, 2 (2000), 24–43.

[29] Hu, Y., Gansner, E. R., and Kobourov, S. G. Visualizing graphs and clusters as maps. *IEEE Computer Graphics and Applications 30*, 6 (2010), 54–66.

[30] Inselberg, A., and Dimsdale, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on Visualization 90* (1990), pp. 361–378.

[31] Itoh, T., Muelder, C., Ma, K.-L., and Sese, J. A hybrid space-filling and force-directed layout method for visualizing multiple-category graphs. In *Proceedings of the 2009 IEEE Pacific Visualization Symposium* (2009), pp. 121–128.

[32] Johnson-Laird, P. N. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* Harvard University Press, 1983.

[33] Käki, M., and Aula, A. Controlling the complexity in comparing search user interfaces via user studies. *Information Processing and Management 44*, 1 (2008), 82–91.

[34] Kammerer, Y., Nairn, R., Pirolli, P., and Chi, H. Signpost from the masses: learning effects in an exploratory social tag search browser. In *Proc. of the 27th international Conference on Human Factors in Computing Systems* (2009), pp. 625–634.

[35] Kang, H., Getoor, L., and Singh, L. C-group: A visual analytic tool for pairwise analysis of dynamic group membership. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (2007), pp. 211–212.

[36] Kellar, M., Watters, C., and Shepherd, M. A goal-based classification of web information tasks. In *Proc. ACM SIGCOMM* (2006), pp. 101–112.

[37] Kules, B., and Shneiderman, B. Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing and Management 44*, 2 (2008), 463–484.

[38] Landesberger, T. V., Kuijper, A., Schreck, T., Kohlhammer, J., Wijk, J. V., Fekete, J.-D., and Fellner, D. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum, Eurographics Association 30*, 6 (2011), 1719–1749.

[39] LeBlanc, J., Ward, M. O., and Wittels, N. Exploring n-dimensional databases. In *Proceedings of the 1st conference on Visualization '90* (1990), pp. 230–237.

[40] Lee, B., Czerwinski, M., Robertson, G., and B.Bederson, B. Understanding research trends in conferences using paperlens. In *Proc. Extended Abstracts of the ACM Conference on Human Factors in Computing Systems* (2005), pp. 1969–1972.

[41] Lee, B., Plaisant, C., and Parr, C. S. Task taxonomy for graph visualization. In *In Proceedings of BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV06* (2006), pp. 82–86.

[42] Liu, Y., Barlowe, S., Feng, Y., Yang, J., and Jiang, M. Evaluating exploratory visualization systems: A user study on how clustering-based visualization systems support information seeking from large document collections. *Information Visualization* (to appear).

[43] Mcguffin, M. J., and Jurisica, I. Interaction techniques for selecting and manipulating subgraphs in network visualizations. *IEEE Trans. Vis. Comput. Graph. 15*, 6 (2009), 937–1944.

[44] network analysis project, S. S. http://snap.standford.edu/.

[45] Newman, M. Finding community structure in networks using the eigenvectors of matrices. *Physical review E 7*, 1 (2006), 36–104.

[46] Nhon, D. T., Wilkinson, L., and Anand, A. Stacking graphic elements to avoid over-plotting. *IEEE Trans. Vis. Comput. Graph. 16*, 6 (2010), 1044–1052.

[47] Paas, F., Tuovinen, J. E., Tabbers, H., and Gerven, P. W. M. V. Cognitive load measurement as a means to advance cognitive load theory. *Educational Psychologist 38*, 1 (2003), 63–71.

[48] Palla, G., Derenyi, I., Farkas, I., and Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature 435* (2005).

[49] Perer, A., and Shneiderman, B. Balancing systematic and flexible exploration of social networks. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 639–700.

[50] Perer, A., and Shneiderman, B. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. In *Proceedings of the 13th international conference on Intelligent user interfaces* (2008), pp. 109–118.

[51] Pirolli, P., and Card, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *In Proc International Conference on Intelligence Analysis* (2005), pp. 1237–1246.

[52] Pirolli, P., Schank, P., Hearst, M., and Diehl, C. Scatter/gather browsing communicates the topic structure of a very large text collection. In *Proc. of the SIGCHI conference on Human factors in computing systems* (1996), pp. 213–220.

[53] Pretorius, A. J., and van Wijk, J. J. Visual analysis of multivariate state transition graphs. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 685–692.

[54] Riche, N. H., and Dwyer, T. Untangling euler diagrams. *IEEE Trans. Vis. Comput. Graph. 16*, 6 (2010), 1090–1099.

[55] Shen, Z., Ma, K.-L., and Eliassi-Rad, T. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Trans. Vis. Comput. Graph. 12*, 6 (2006), 1427–1439.

[56] Shneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley, 2010.

[57] Shneiderman, B., and Aris, A. Network visualization by semantic substrates. *IEEE Trans. Vis. Comput. Graph. 12*, 5 (2006), 733 – 740.

[58] Smith, M. A., Shneiderman, B., Milic-Frayling, N., Mendes Rodrigues, E., Barash, V., Dunne, C., Capone, T., Perer, A., and Gleave, E. Analyzing (social media) networks with nodexl. *Proceedings of the fourth international conference on Communities and technologies 16*, 6 (2009), 255–264.

[59] Stasko, J., Grg, C., and Liu, Z. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (2008), 118–132.

[60] Sweller, J. Cognitive load during problem solving: Effects on learning. *Cognitive Science 12*, 2 (1988), 257–285.

[61] Utts, J. M. *Seeing Through Statistics.* Thomson Brooks, 2005.

[62] van Dongen, S. Graph clustering by flow simulation. *PhD thesis, University of Utrecht* (2000).

[63] van Ham, F., and Perer, A. Search, show context, expand on demand: Supporting large graph exploration with degree-of-interest. *IEEE Trans. Vis. Comput. Graph. 15*, 6 (2009), 953–960.

[64] Veerasamy, A., and Belkin, N. J. Evaluation of a tool for visualization of information retrieval results. In *SIGIR 1996* (1996), ACM Press, pp. 85–92.

[65] Viau, C., McGuffin, M. J., Chiricota, Y., and Jurisica, I. The flowvizmenu and parallel scatterplot matrix: Hybrid multidimensional visualizations for network exploration. *IEEE Trans. Vis. Comput. Graph. 16*, 6 (2010), 1100–1108.

[66] Viegas, F. B., Wattenberg, M., and Feinberg, J. Participatory visualization with wordle. *IEEE Trans. Vis. Comput. Graph. 15*, 6 (2009), 1137–1144.

[67] V.Krebs. The polbooks dataset, 2007. http://www.orgnet.com/.

[68] Wattenberg, M. Visual exploration of multivariate graphs. In *In Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), pp. 811–819.

[69] White, R. W., Marchionini, G., and Muresan, G. Editorial: Evaluating exploratory search systems. *Information Processing and Management 44*, 2 (2008), 433–436.

[70] Wise, J. A., Thomas, J. J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V. Visualizing the non-visual: spatial analysis and interaction with information from text documents. In *Proc. IEEE Information Visualization* (1995), pp. 51–58.

[71] Wu, L., Ying, X., Wu, X., and Zhou, Z.-H. Line orthogonality in adjacency eigenspace with application to community partition. pp. 2349–2354.

[72] Yang, J., Liu, Y., Zhang, X., Yuan, X., Zhao, Y., Barlowe, S., and Liu, S. Piwi: Visually exploring graphs based on their community structure. *IEEE Trans. Vis. Comput. Graph.* (to appear).

[73] Yang, J., Luo, D., and Liu, Y. Newdle: Interactive visual exploration of large online news collections. *IEEE Computer Graphics and Applications 30*, 5 (2010), 32–41.

[74] Ying, X., Wu, X., and Barbará, D. Spectrum based fraud detection in social networks. In *ICDE* (2011), pp. 912–923.

[75] Zamir, O., and Etzioni, O. Grouper: A dynamic clustering interface to web search results. *Computer Networks 31*, 11 (1999), 1361–1374.