Visual Analysis of Large Heterogeneous Social Networks by Semantic and Structural Abstraction

Zeqian Shen, Kwan-liu Ma, and Tina Eliassi-Rad

***Abstract*— Social network analysis is an active area of study beyond sociology. It uncovers the invisible relationships between actors in a network and provides understanding of social processes and behaviors. It has become an important technique in a variety of application areas such as the Web, organizational studies, and homeland security. This paper presents a visual analytics tool, OntoVis, for understanding large, heterogeneous social networks, in which nodes and links could represent different concepts and relations, respectively. These concepts and relations are related through an *ontology* (a.k.a. a *schema*). OntoVis is named such because it uses information in the ontology associated with a social network to semantically prune a large, heterogeneous network. In addition to semantic abstraction, OntoVis also allows users to do structural abstraction and importance filtering to make large networks manageable and to facilitate analytic reasoning. All these unique capabilities of OntoVis are illustrated with several case studies.**

***Index Terms*— graph drawing, information visualization, ontology, semantic graphs, social networks, visual analytics**

I. INTRODUCTION

A social network represents relationships between actors (such as people, families, Internet social groups, corporate organizations, business partners, nations, etc). Social network analysis is an active area of study beyond sociology. Analyzing a social network can provide structural intuition according to the ties linking actors and lead to improved understanding of social processes. The analysis can help identify critical actors or links. For example, analyzing a terrorism network can help us understand terrorists, their organizations, and their associations to events. The goal of such analysis is to enable us to deduce new relations, reveal potential vulnerabilities, and identify an attack before it occurs. Social networks are inherently visual in nature. Visual analytics tools and techniques have been used in social network analysis [1]–[3]. These research results show the potential value of visualization directed analysis to complement the more traditional methods based on mathematical and combinatorial analysis.

Social network analysis for real-world applications presents several challenges. First, the network can be very large containing from thousands to even millions of nodes. For example, Kolda et al. [4] describe networks that can reach 10^9 nodes and an order of magnitude more links. The sheer number of nodes and links makes drawing the entire network computationally infeasible. It also results in a mismatch between the density of information and screen resolution available. Next, large social networks commonly have complex relations among their actors (such as the *small-world* property [5]). These structural properties make visual analysis difficult (*a.k.a.*, the death-star phenomenon). Finally, a social network may have heterogeneous nodes and links (*a.k.a.*, a semantic graph) [6]. In other words, the network may consist of different types of entities (multi-mode network) and relationships (multirelational network) [7]. For example, in a terrorism network the nodes may be terrorists, organizations, attacks, and other relevant events. This heterogeneity creates additional complexity of visual analysis. In this paper, we address some of these challenges.

We analyze large heterogeneous social networks by (*i*) applying a set of information visualization techniques and (*ii*) guiding the analysis with an auxiliary graph called *ontology*. An ontology graph (a.k.a blockmodel [8]) is a description of the relationships that can exist for an actor or a group of actors [6] and is generally much smaller (as in the number of actors and links) than the social network. The ontology may be generated prior to populating the network with data or be derived from the network data. In this paper, we assume the ontology graph is given with the semantic graph (i.e., it is not derived). We have introduced a set of novel abstraction and filtering techniques for ontology-based, interactive visual analysis of large heterogeneous social networks. These abstraction and filtering techniques are based on both semantic and structural information and allow users to find facts about actors and relations more easily by creating views of some isolated groups. We have built a prototype system, which we call OntoVis, and evaluated it using two case studies: a movie network [9] and a terrorism network [10]. The rest of this paper discusses related work, introduces our design and the interactive visual analysis process, presents our case studies, and suggests further research directions.

II. RELATED WORK

Social network visualization and analysis has attracted much interest from both sociology and information visualization areas. Linton Freeman [1] summarizes the rich history of social network visualization from the sociology perspective. A comprehensive list of social network analysis software packages can be found at [7], [11] and the INSNA website [12]. Research from the information visualization perspective put more efforts into the visual representation and exploration of social networks. Graph drawing methods have been developed for network visualization [13]–[15] and are applicable to the analysis of social networks.

Zeqian Shen and Kwan-liu Ma are with the Department of Computer Science, University of California at Davis, One Shields Ave., Davis, California 95616. E-mail: {shenz, ma}@cs.ucdavis.edu.

Tina Eliassi-Rad is with the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East Ave. L-560, Livermore, CA 94550. E-mail: eliassirad1@llnl.gov.

To overcome the visual complexity of very large networks, Abello and his colleagues introduce network visualization techniques that allow users to find facts about actors and relations among them more quickly through interactive navigation of the network at different levels of abstraction, from an overview to close-up views of some isolated groups [16], [17]. Focus+context techniques have also been used to allow users to simultaneously access overview information while working on details of interesting groups [18], [19].

The key to visual analysis of heterogeneous networks is proper utilization of the semantic information that resides on the nodes and links throughout the analysis. Despite the wealth of network visualization and analysis research, few were done for heterogeneous networks. Most previous works focus on visualizing the Semantic Web [20], [21]. NetDraw [22] is a program to visualize large social networks. It can handle heterogeneous networks with two node types. Pajek [2] is a widely used network visualization and analysis system, which employs a suite of network analysis algorithms. It can break a large heterogeneous network into small ones by clustering the nodes according to their types. However, Pajek does not use ontology information effectively, because it is not designed specifically for heterogeneous social network visual analysis.

Therefore, new designs and techniques for visual analysis of large heterogeneous social networks are needed. We propose to use both structural abstraction and filtering based on ontology information, which results in a dual-graph approach using both the semantic graph and its associated ontology graph to reduce the large social network, examine different instances of known relationships, infer new relationships, and reveal other hidden knowledge in the networks.

III. ONTOVIS

Throughout the paper, a semantic graph is defined as $G = (V, E, vt, et)$ and its associated ontology graph is defined as $OG = (T_V, T_E)$. V denotes the vertex set and E denotes the edge set. $T_V = \{t_1, t_2, ..., t_n\}$ is a set of vertex types and $T_E = \{(t_i, t_j) : t_i, t_j \in T_V\}$ is a set of edge types. vt denotes a mapping from V to T_V that associates a vertex to its type. If v is a vertex in the semantic graph, $vt(v)$ denotes the type for vertex v. Similarly, et denotes a mapping from E to T_E that associates an edge with its type. If e is an edge in the semantic graph, $et(e)$ denotes the type for edge e. It is important to note that a semantic graph cannot have vertices and edges with types that are not presented in its associated ontology graph. In other words, T_V and T_E are, respectively, supersets of the vertex and edge types that occur in the semantic graph G. Both semantic graphs and the associated ontology graphs discussed in this paper are undirected.

An ontology graph specifies the nature of actors and relations that can exist in a heterogeneous social network. OntoVis is a social network visual analytics tool that is designed to take advantage of this additional information. OntoVis allows users to more easily isolate actors and relationships for making inferences or identifying the key actors. OntoVis is unique because it can do both structural abstraction and semantic abstraction. Structural abstraction is achieved by using topological information such as node degree, connectivity, etc. Semantic abstraction is attained by utilizing the ontology graph such as node degree based on a specific type. OntoVis also allows the users to filter the graph substantially using statistical measures. The resulting graph is displayed using a clever graph layout method. In addition, the transition from one graph to the next is animated.

A. Semantic Abstraction

In OntoVis, users are able to construct a derived graph from the original graph by including only nodes whose types are selected in the ontology graph. It would be very helpful if the users could also remove nodes or add nodes back into the derived graph as desired during an analysis. In order to achieve this goal, abstraction is used to hide nodes. An abstraction is defined as an induced graph $G[TS_V] = (VS, ES, vt, et)$ of a selected set of node types $TS_V \subseteq T_V$, where $VS = \{v \in V : vt(v) \in TS_V\}$ and $ES = \{(v_i, v_j) \in E : vt(v_i), vt(v_j) \in TS_V\}$. Each node $v \in VS$ has an attribute set, which is defined as $va(v) = \{u \in V : (u, v) \in E, vt(u) \notin TS_V\}$. Since this induced graph is an abstraction based on semantic information, we call it semantic abstraction of the original graph. In OntoVis, as soon as a node type is chosen from the ontology graph, all nodes with this type are added to the semantic abstraction. All the nodes that are not included in the abstraction are not deleted; rather, they are converted into attributes of the nodes kept in the abstraction (which are their neighbors in the original graph).

Semantic abstraction not only provides a simple view of interest to help isolate key actors in social networks but also allows users to expand the abstraction back locally to see details. In other words, users can browse the attribute lists of nodes in the abstraction and convert a particular attribute of interest back to a node. In order to demonstrate this idea, a very simple example of a terrorism network is shown in Fig. 1. Please note that in OntoVis edges are drawn as Bézier curves, because we believe they look more pleasing than straight lines. There are three different node types in the network: country/area, terrorist organization, and classification. The task is to find the similarity and difference among all terrorist organizations. We first select only one node type: terrorist organization in the ontology graph. The induced semantic abstraction contains four organizations: Revolutionary Nuclei (RN), Revolutionary People's Struggle (RPS), Fighters for Freedom (FF) and Popular Revolutionary Action (PRA). The attributes of RPS and FF are displayed (see Fig. 1(a)). They have one common attribute: Greece, which is a country/area node in the original graph. We convert Greece to a node and add it to the abstraction. All the existing edges between node Greece and the nodes that are included in the abstraction are displayed (See Fig. 1(b)). We find the common characteristic of the four terrorist organizations is that they are all located in Greece. Next, we want to find the difference among four organizations. We go back to the first view, i.e., Fig. 1(a) by converting node Greece back to attributes. In Fig. 1(a), one of the attributes of RPS is Leftists, while that of FF is Anti-Globalization. They are both classification nodes in the original graph. We added them back into the abstraction.

Fig. 1(c) shows that these four terrorist organizations are classified into two categories: RPS and RN are classified as Anti-Globalization, while FF and PRA are Leftists. In this example, we demonstrate how to obtain a simple view of interest and expand it locally using semantic abstraction techniques. This is a powerful capability of OntoVis. For example, a terrorism network that contains a large number of attacks, if displayed directly, would lead to severe visual complexity. To suppress this complexity, users can choose to only display the terrorist organizations and make terrorist attack an attribute of the responsible organization. If users are interested in investigating a particular attack, it can be selected in the attribute list of the responsible organization and added to current abstraction of the semantic graph.

(a) A semantic abstraction that only contains terrorist organizations. Nodes with types of country/area and classification become attributes of their neighbors in the original graph.

(b) Greece is converted to a node. This view shows that all four organizations are located in Greece.

(c) From (a), Leftists and Anti-Globalization are converted and added. This views shows that two organizations are classified as Leftists and the other two are classified as Anti-Globalization.

Fig. 1. A simple example of terrorism network analysis using semantic abstraction techniques. The attributes of a node are displayed in a box besides it.

B. Structural Abstraction

The other way to manage the visual complexity of a large network is to make use of its structural information (such as connectivity and node degree) to condense the network. Such structural abstraction, however, must preserve the essential structure of the entire network. There are different approaches to making structural abstractions. For example, in [19], nodes are laid out by a force-directed method, and then based on the layout result, nodes close to each other are merged into super nodes. In [23], an algorithm is introduced for finding strongly connected cores of bipartite graphs. Most of social networks that we have worked on have two characteristics. First, the network contains many nodes of degree one connecting to specific nodes. Second, the network has duplicate paths. For example, the nodes circled in Fig. 2(a) connect two nodes that are also directly connected. In general, this kind of visual information is considered redundant for understanding the structure of the network. OntoVis can directly remove one-degree nodes and duplicate paths. Fig. 2(b) shows the result of applying such structural abstraction. The visual complexity is reduced such that the user can readily focus on the key actors in the network and the relationships among them.

C. Importance Filtering

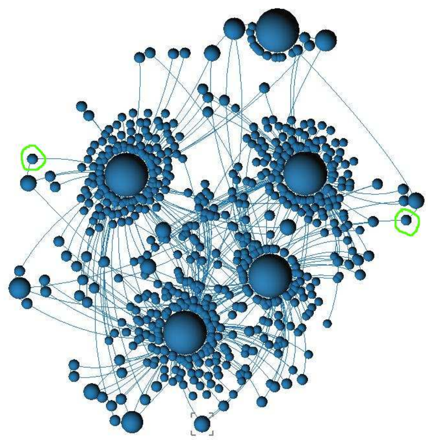
A main question in analysis of heterogeneous social networks is which types and relationships should be selected and investigated. The ontology of a heterogeneous social network can facilitate the selection of node types and relationships. Usually users make judgments according to their domain knowledge. Unfortunately, these judgments can be quite subjective and introduce biases into the analysis. Moreover, for unfamiliar data it is usually difficult for users to make choices. To achieve more objective reasoning, OntoVis makes use of the statistical measures proposed in [6] for evaluating connectivity and relevance between node types. For example, node degree is a useful measure in social network analysis. A node with very high degree is generally a crucial actor in the social network. This concept can be extended to heterogeneous networks as described in [6]. In particular, node degree is redefined to take into account the type information given by the ontology graph. Formally, let α, β be node types and $i = 1, ..., n$. Then, $k_{\alpha\beta}(i)$ is the number of neighbors with type β of a node i with type α. The total number of neighbors of node i is

$$k_\alpha(i) = \sum_\beta k_{\alpha\beta}(i) \tag{1}$$

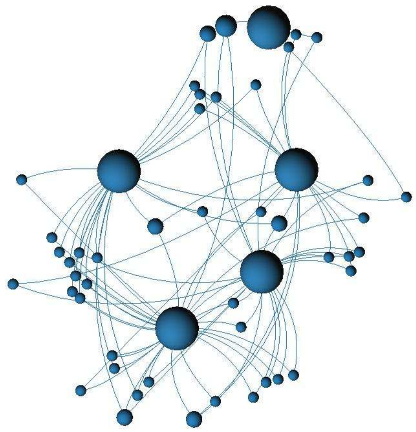
Let n_α be the number of nodes of type α. Function t_i gives the type of node i. The average number of neighbors over all nodes with type α is

$$\overline{k_\alpha} = \frac{1}{n_\alpha} \sum_{i, t_i = \alpha} k_\alpha(i) \tag{2}$$

To compare connectivity among different types, $\overline{k_\alpha}$ must be rescaled by the number of different neighbor types that type



(a) Original network has many one-degree nodes and duplicate paths, which increase the visual complexity of the graph.



(b) Structural abstraction after removing one-degree nodes and duplicate paths. The key nodes in each cluster and the connections between them become clearer.

Fig. 2. A simple example of structural abstraction

α can have, k_α^0. Then, the average connectivity per type is defined as follow:

$$m_\alpha = \frac{\overline{k_\alpha}}{k_\alpha^0} \tag{3}$$

This value measures the connectivity of nodes with a type α. The connectivity variance per type is

$$\overline{k_\alpha^2} = \frac{1}{n_\alpha} \sum_{i,t_i=\alpha} k_\alpha^2(i) \tag{4}$$

The dispersion of average connectivity per type is given by

$$\sigma_\alpha^k = \frac{[\overline{k_\alpha^2} - \overline{k_\alpha}^2]^{1/2}}{k_\alpha^0} \tag{5}$$

σ_α^k tells us if all nodes of type α have essentially the same connectivity.

Using *node degree per type and its dispersion*, users can get an idea about the connectivity of nodes belonging to each type. Typically nodes with large degrees per type and dispersions indicate key actors in a network.

Besides the average node degree per type and its dispersion, the *node type disparity* of each network connection is important. For example, suppose type A can link to both B and C given the ontology graph. However, nodes with type A preferentially connect to nodes with type B. This information is useful in heterogeneous social network analysis. In addition, identifying weak linkages are important [24]. For example, if the links between nodes with type A and nodes with type C are weak, we might want to look at the nodes with these two types together to investigate their relationships. This disparity value is formally defined in [25] and [26]. Specifically, for a given node i of type α

$$Y_2(i,\alpha) = \sum_\beta \left[\frac{k_{\alpha\beta}(i)}{k_\alpha(i)}\right]^2 \tag{6}$$

A small value of $Y_2(i,\alpha)$ indicates that the links from node i are evenly distributed to all types, which can connect to type α. If a few types dominate the most of connections from node i, $Y_2(i,\alpha)$ will be large. The average value of $Y_2(i,\alpha)$ over all the nodes with type α is

$$\overline{Y_2} = \frac{1}{n_\alpha} \sum_{i,t(i)=\alpha} Y_2(i,\alpha) \tag{7}$$

The corresponding dispersion is

$$\sigma_\alpha^Y = [\overline{Y_2^2(\alpha)} - \overline{Y_2(\alpha)}^2]^{1/2} \tag{8}$$

However, these results are affected by the fact that some types have more nodes than others. Let $V(\alpha)$ be the set of types, which can connect to α. Then, Y_2^T is defined

$$Y_2^r = \sum_{\beta \in V(\alpha)} \left[\frac{n_\beta}{n}\right]^2 \tag{9}$$

Rescale $Y_2(\alpha)$ by Y_2^r:

$$R(\alpha) = \frac{\overline{Y_2(\alpha)}}{Y_2^r} \tag{10}$$

The corresponding dispersion is

$$\sigma_\alpha^R = \frac{\sigma_\alpha^Y}{Y_2^r} \tag{11}$$

$R(\alpha)$ is called the *disparity of connected types* [6]. A node type with a large value for its disparity of connected types, $R(\alpha)$, preferentially links to a small number of types. For heterogeneous network analysis, node types with high disparity may be interesting because their preference of neighbors could lead to finding of weak linkages. The *dispersion* associated with disparity of connected types shows whether such a behavior is typical.

In OntoVis, users can select one of these measures to be visually encoded in the ontology graph. For example, OntoVis uses node sizes to represent disparity of connected types in an ontology graph. Fig. 3 depicts the ontology graph for a movie dataset. Type person has the largest disparity. In other words, nodes of type person preferentially connect to a small number of types. Based on such information, users can focus on the types with high connectivity and unbalanced connections (which have higher information content).

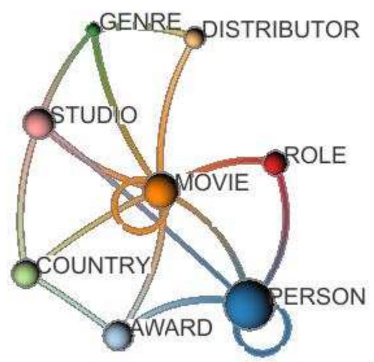


Fig. 3. Ontology graph of a movie dataset. Node sizes indicate disparity of connected types. Type person has the largest disparity.

D. Graph Layout and Animation

Whether we want to display a part or an abstraction of the social network, using an appropriate graph drawing method is essential to reveal the key structural information of the network. We have chosen to use force-directed layout graph drawing [27] because it is general enough to work with many types of networks, relatively easy to implement, and adaptable to satisfy different requirements.

Force-directed methods use a force model based on the graph layout aesthetics and an algorithm to search for the optimal state of the model. A variety of force models and optimization algorithms [28] have been introduced. Because a large percentage of social networks exhibit small-world properties [5], we need a layout that reflects the cluster structure. Thus, we use Linlog [29], which is a force model designed specifically for small-world networks.

One problem with most graph layout methods is the ignorance of visual overlapping of the nodes (see Fig. 4(a)). Generally, force-directed algorithms do not take into account the displayed size of each node in the network. In our design, node size is used to indicate node degree or other properties so we have to address the visual overlapping problem. Some refined force models are introduced in [30], [31], which eliminate overlappings caused by nodes with non-uniform sizes and shapes. Since our node shape is round, after the force-directed layout is computed, a refinement process based on a simple heuristic can be used to reduce most of the overlappings. The refinement process goes through all the overlapping nodes and pushes the smaller nodes away from the larger nodes. Fig. 4(b) shows the results after such refining process.

Furthermore, because force-directed layout methods are designed for connected networks, they cannot place unconnected nodes very well (to better utilize the display space). Since those unconnected nodes are usually of less interest in the social network analysis, they should not be placed in the center to distract the users. We can simply filter out all the unconnected nodes or place them in the peripheral area as shown in Fig. 4(c).

Whenever a node is added to or removed from the network, the force-directed layout is recomputed. The nodes positions after layout can be different from before. It is important to provide the viewers a smooth transition between the old and new layouts. Animation enables users to keep the mental map and better track the network structures [32], [33]. Therefore, the movement of nodes from their current positions to the new positions is smoothly animated. The key issue is selecting a meaningful initial position for new nodes. There are two different conditions:

1) Add a set of nodes with the same type to the network: Since the new nodes belong to the same type, we want them to be perceived as a group. Thus, all new nodes are initially placed together in a random fashion. Users see a transition of how they are pulled away by existing nodes.
2) Turn an attribute into a node of the network: Because the new node was an attribute of another node (which we named parent node) it is placed next to the parent node. Therefore, users can get the sense that the new node has "spawn" from the parent node. It helps users to recognize and track the new node and its relationship with its parent node.

IV. Case Studies

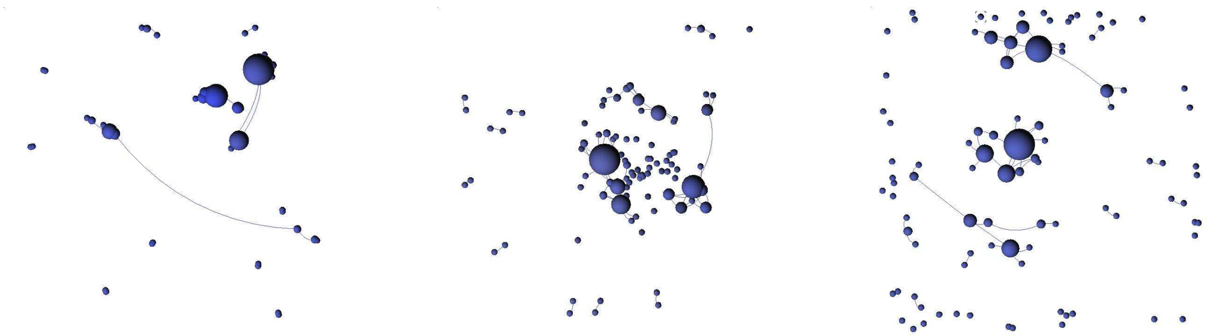
To demonstrate how visual analysis of large heterogeneous networks can be done with OntoVis, we present two case studies in this section.

A. Movie Domain

The movie network is compiled from the UCI KDD Archive [9]. The network contains eight node types, including person, movie, role, studio, distributor, genre, award and country. There are 35,312 nodes and 108,212 links in this network. It is generally impossible to visualize the entire network on a desktop screen. The data is noisy and contains missing relationships. Consequently, some of the conclusions drawn here may be inaccurate. However, we focus on demonstrating the visual analysis process using OntoVis.

The ontology graph of the movie network is shown in Fig. 5. Node sizes in the ontology graph are set to represent disparities of connected types. Type person is the one with largest disparity value. The number on each link is the frequency of links between two connected types. Type person is strongly connected to types movie and award. The relationship between type person and type role is weak. As mentioned before, in social network analysis weak linkages are often very important [24].

We first investigate the relationships between persons and roles. Our belief is that a good actor should be able to play different roles. There are 115 different roles, including hero, savior, scientist, wimp, and so on. We start by selecting the node type role from the ontology graph. As a result, all other types of nodes become the attributes of role nodes. Next we choose five particular roles: hero, scientist, love interest, sidekick and wimp from the list of role nodes. We can then go through each role to select from its attribute list 'person,' which picks all the people who acted in that role. At this time, the derived abstraction is constructed accordingly and shows exactly this role-actor relationship (see Fig. 6).



(a) Layout using the original Linlog method. The node-overlapping problem is obvious.

(b) Overlappings are reduced by refinement processing. However, in the center, there are many unconnected nodes, which are of less interest in social network analysis. These nodes can distract the users.

(c) Unconnected nodes are placed in the peripheral area, such that the structures of clusters are clearer.

Fig. 4. Layout and Refinement for Social Networks

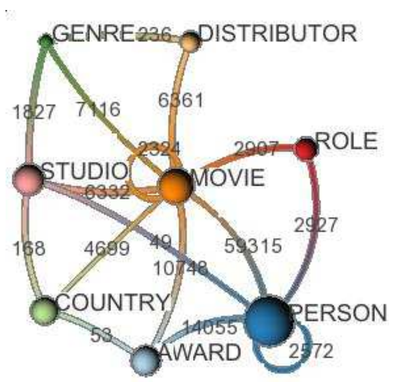


Fig. 5. Ontology Graph of the Movie Network. Node sizes represent the disparity of connected types for each node type. The numbers on edges are the frequencies of links between two types. Type person has the maximum disparity of connected types. Person nodes are preferentially connected to movie nodes and award nodes. The relationship between type person and type role is weak.

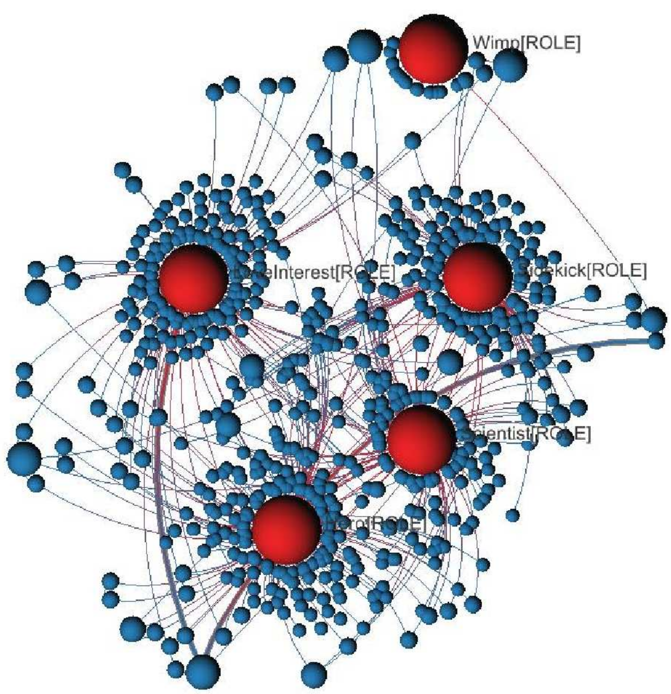


Fig. 6. Visualization of all the people have played any of the following roles: hero, scientist, love interest, sidekick and wimp. The red nodes are roles and the blue nodes are actors.

The sheer number of nodes and links make it hard to find actors that played multiple roles. To reduce the visual complexity, we select to view the derived graph with structural abstraction, which removes all the one-degree nodes and duplicate paths (see Fig. 7). We observe that Woody Allen, Sandra Bullock, Archibald Leach (*a.k.a.*, Cary Grant), Thomas Sean Connery and Bill Paxton are the actors that played three different roles. They are likely to be very good actors. Also we find that Maria "Mia" Farrow, who played two different roles, is related to Woody Allen. The relationship between two good actors can be interesting. Thus, we decided to do further investigation on their movies and relationships.

First, we want to clear out the unnecessary information. People nodes except Woody Allen and Maria "Mia" Farrow, are removed by deselecting them in the attribute lists of the five roles. The roles are also removed by deselecting the type role in the ontology graph. Next, new nodes and types needed for current analysis are added. All of Woody Allen's movies are selected from his attribute list. People who worked in those movies are selected from the movies' attribute lists. The orange nodes represent movies, and the blue nodes represent people. The largest blue node in the center is Woody Allen. Since the node sizes represent their degrees, people who worked more often with Woody Allen should be represented by larger nodes. In Fig. 8, we can find Maria "Mia" Farrow, Louise Lasser and Diane Keaton are the actors who worked most often with Woody Allen. Maria "Mia" Farrow worked on 13 of Woody Allen's movies, and Diane Keaton made 8 movies with him. Moreover, we find that Maria "Mia" Farrow and Louise Lasser are both connected to Woody Allen directly. The information on IMDB [34] shows that Maria "Mia" Farrow had children with Woody Allen and Louise Lasser was his ex-wife. It is also reported that he dated

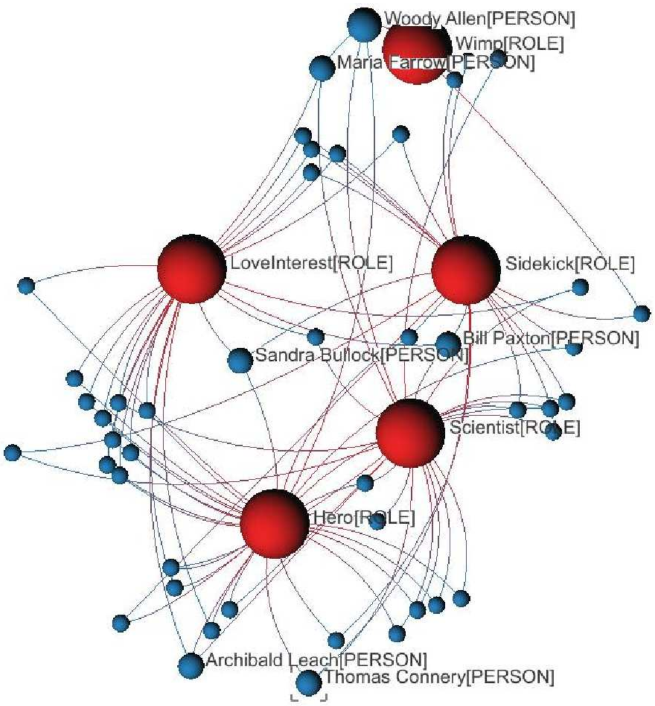


Fig. 7. Abstraction of the visualization of five roles and related actors. Woody Allen, Sandra Bullock, Archibald Leach (*a.k.a.*, Cary Grant), Thomas Sean Connery and Bill Paxton are the actors that played three different roles. Woody Allen is related to actress Maria "Mia" Farrow.

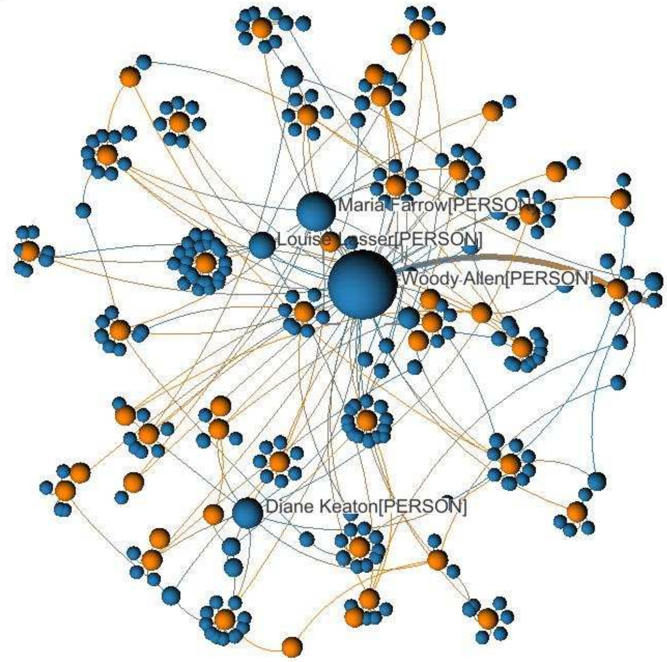


Fig. 8. Visualization of Movies (in orange) and People (in blue) Related to Woody Allen. The actors who worked most often with Woody Allen are Maria "Mia" Farrow, Louise Lasser and Diane Keaton.

Diane Keanton. Therefore, we conclude that Woody Allen often worked with his girl-friends on his movies.

We also want to study the genres of Woody Allen's movies. Since the people nodes are no longer useful for our task, they are removed. Genre nodes are added by selecting type genre in the ontology graph (see Fig. 9). Note that the genres of some movies are missing. Based on the available information, the main genre for Woody Allen's movies is comedy. Romantic and drama are two other major genres for his movies. Interestingly, he also worked (as the voice of one the characters) in the cartoon movie Antz.

In the movies' case study, we show that OntoVis can analyze a large graph and lead users to some interesting and useful findings. Specifically, users can use the ontology graph to construct an abstraction that best relates to their analysis objective.

B. Terrorism Domain

The terrorism network used in this case study is compiled from the MIPT Terrorism Knowledge Base [10]. There are nine different types of nodes in the network. Fig. 10 shows its ontology graph. The most crucial types are terrorist organizations and terrorist attacks. Classification indicates the type of a terrorist organization. Terrorists can be the leaders or members of an organization or defendants involving in a legal case against it. A terrorist organization is connected to its base countries and areas. A terrorist attack is described by its location, tactics, targets, weapons and the organizations, which are responsible for it. The number besides each type is the frequency of that node type in the network. This terrorism

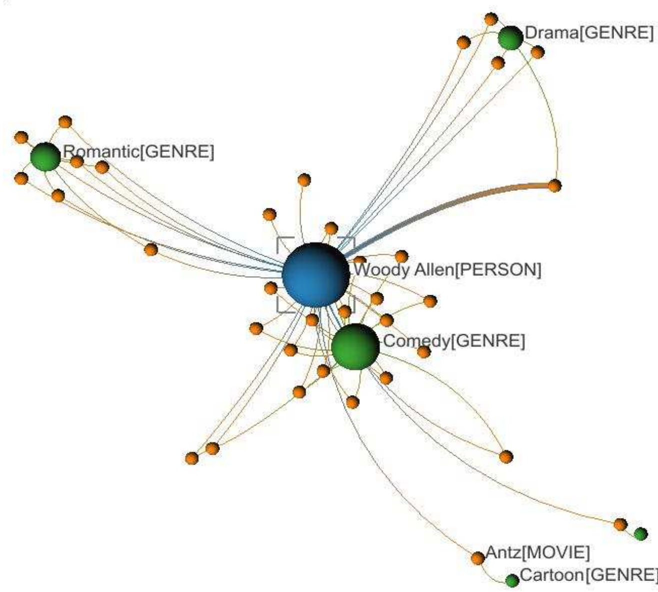


Fig. 9. The three major genres (in green) of Woody Allen's movies are comedy, romantic and drama.

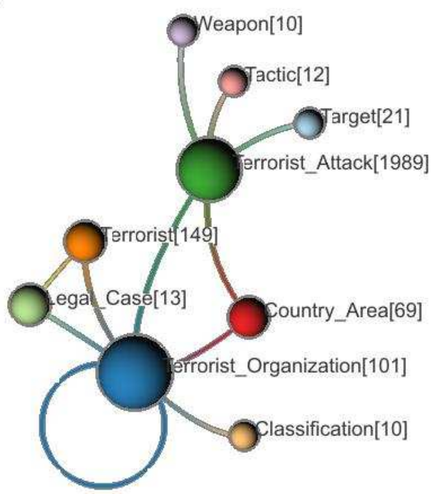


Fig. 10. Ontology Graph of the Terrorism Network. There are nine node types including terrorist organization, classification, terrorist, legal case, country/area, attack, attack target, weapon and tactic. The number in the square brackets is the count frequency for the corresponding node type.

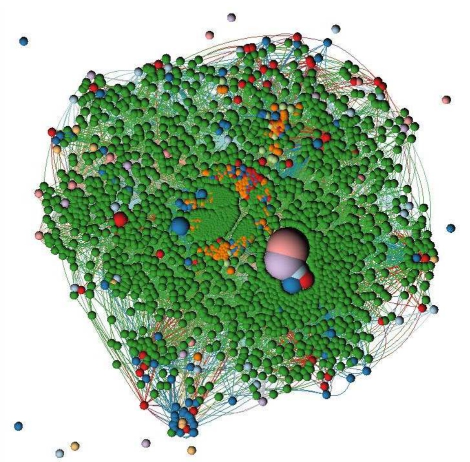


Fig. 11. Visualization of the Entire Terrorism Network. There are 2,374 nodes and 8,767 links.

network contains a total of 2,374 nodes and 8,767 links. Fig. 11 shows a visualization of the entire network. Although different colors are assigned to indicate various node types and node sizes are used to indicate node degrees, the sheer number of nodes and links creates an overwhelming visual complexity for users.

Tasks for terrorism network analysis are to find key terrorist organizations and try to characterize their behaviors. Analysis of current behaviors of a terrorist organization can help identify and understand future threats, assess vulnerabilities and determine potential impacts. In addition, we are interested in relationships between organizations, key terrorists, and attacks.

Based on our tasks and knowledge about the terrorism network, organizations are first studied. By selecting the type terrorist organization in the ontology graph, the network of organizations is visualized (see Fig. 12). In order to identify key actors (which have more connections than others), node size is set according to its degree. Al-Qaeda is one of the most dominant organizations in the network and is related to most organizations. Since OntoVis uses the force-directed layout, connected nodes are placed close to each other. In the visualization, we observe that all the organizations are placed into clusters based on their connectivity. Largest nodes in each cluster are its center actors (such as al-Qaeda in the group near the center and PFLF and Hamas in the group in the upper-left corner). Although al-Qaeda connects to most of other key organizations, it does not directly connect to Hamas. They are instead connected through an organization named Hezbollah. The description of Hezbollah on MIPT is as follows: "Hezbollah is an umbrella organization of various radical Islamic Shi'ite groups and organizations which receives substantial financial and philosophical support from Iran" [10].

OntoVis can show us the commonalities in organizations belonging to the same cluster. In other words, what are the reasons they are closely connected together? By selecting nodes in the derived abstraction, we investigate the attributes of organizations in the upper-left corner. Both Hamas and PFLF are located in the Gaza Strip, West Bank and Israel. Thus, location might be the reason for these organization clusters. We add in the location information by selecting the type country/area in the ontology graph (see Fig. 13). The red nodes in the network are the base locations of organizations. After force-direct layout, the location nodes are close to the related organizations. In the visualization, each cluster has several location nodes inside or nearby. Most organizations in a cluster share common location nodes (except for the al-Qaeda cluster). Organizations in the cluster placed in the bottom-left corner have bases in Kashmir, India and Pakistan. Organizations in the upper-right cluster are located in Colombia. Al-Qaeda has connections in most countries and areas including United States, Austria, Yemen and Iraq. Compare to other regional organizations, al-Qaeda is an international terrorist organization. This clearly shows that the organization clusters are based on locations.

Inter-group connections make clusters close to each other. For example, al-Qaeda connects to the cluster at Gaza Strip, West Bank and another cluster at Kashmir, India and Pakistan. However, it does not connect to the cluster at Colombia. Do those connected clusters share something in common? By investigating the attributes of related organizations, we find that al-Qaeda and most organizations related to it are classified as religious and nationalist/separatist organizations. Therefore, nodes of type classification are added. In Fig. 14, the classification nodes that closely connect to al-Qaeda and its related clusters, are religious and nationalist/separatist. Organizations in Colombia are classified as communist/socialist. Thus, organization clusters with the same classifications tend to connect to each other. Organizations that are located in the same countries or areas, tend to be of the same type. Moreover, religious and nationalist/separatist classification nodes connect to most organizations. In other words, they are the most common organization types.

We are interested in the relationships between terrorist organizations, terrorists and legal cases against them. Since location and classification information are not needed in this task, these types are deselected in the ontology graph to reduce

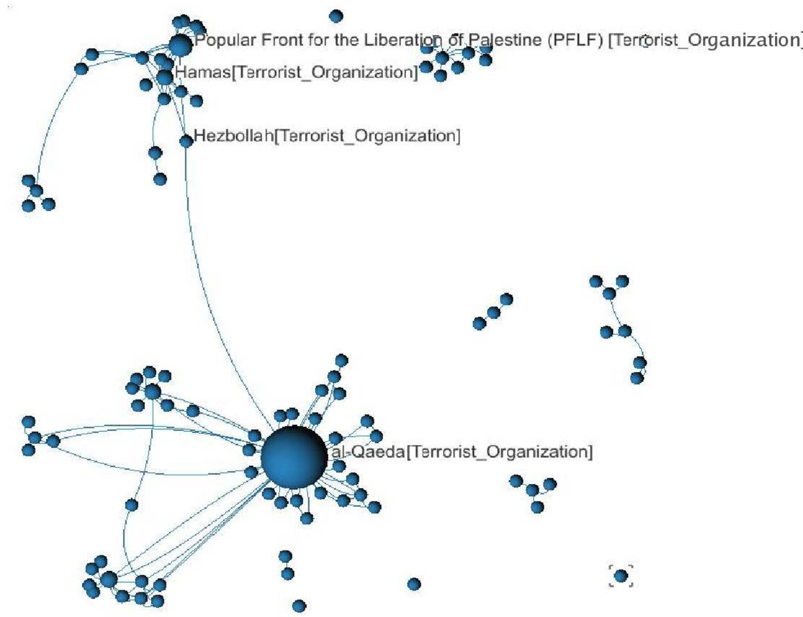


Fig. 12. Visualization of the Terrorist Organization Network. The organizations are placed into several clusters based on their connectivity by force-directed layout. The cluster near the center includes al-Qaeda, and the cluster in the upper-left corner contains Hamas and PFLF.

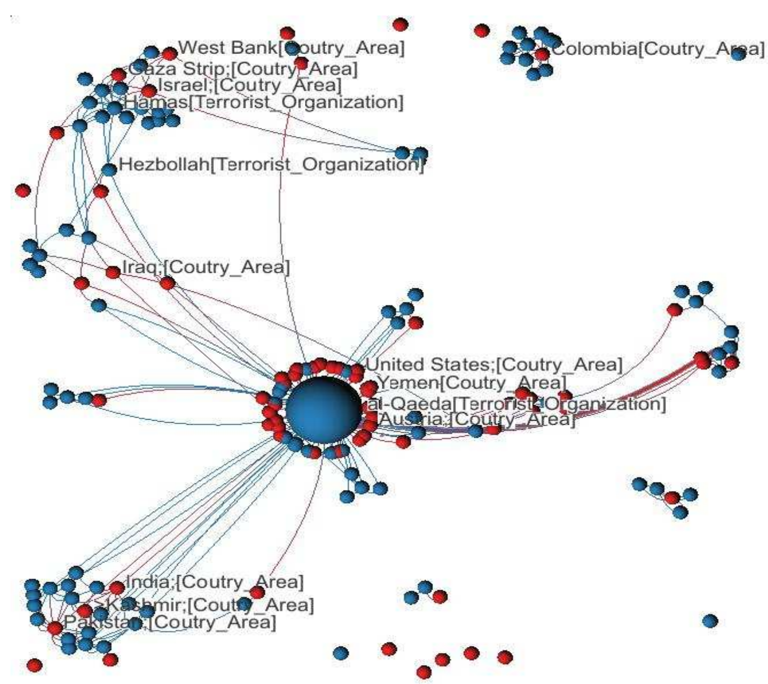


Fig. 13. Visualization of Terrorist Organizations (in blue) and Locations (in red). Clustering of terrorist organizations is consistent with their locations. The organizations in the upper-left cluster are located in Gaza Strip, West Bank and Israel. Al-Qaeda is an international organization with connections in many countries and areas.

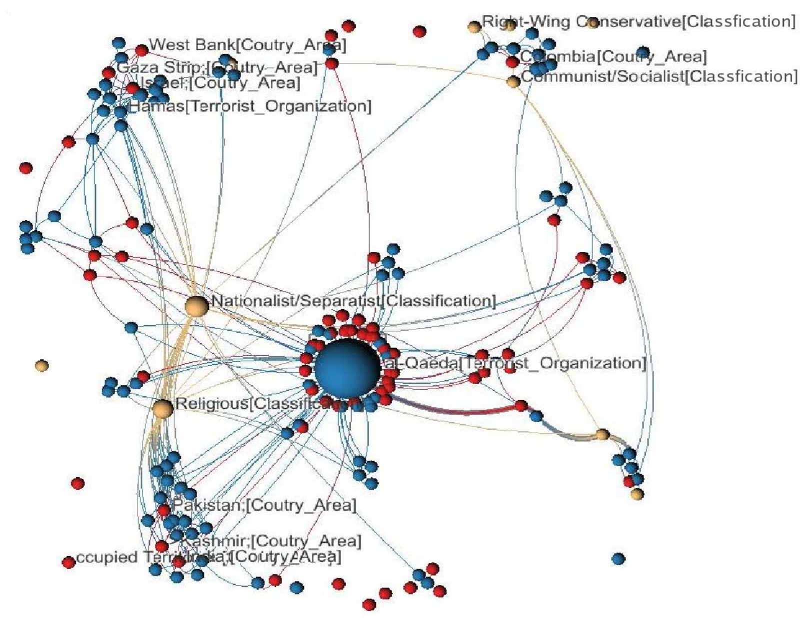


Fig. 14. Visualization of Terrorist Organizations (in blue), Locations (in red) and Classifications (in tan). Religious and nationalist/separatist are the most common terrorist organization types. Most organizations in Gaza Strip, West Bank, India and Pakistan are religious and nationalist/separatist. Organizations in Colombia are communist/socialist and right-wing conservatives.

the visual complexity. Terrorists and legal cases are inserted into the derived abstraction. In Fig. 15, the orange nodes are terrorists and the light green nodes are legal cases. Al-Qaeda is the organization with the most terrorists and legal cases. The case that involves most defendants is USA vs. Wadih Ei Hage et al. 98-CR-1023. There are only two legal cases against Hamas.

One of the most important tasks is to characterize the terrorist attacks. However, there are over 1900 attacks in the network, which is too much for visualizing and analysis. From the analysis above, Hamas and al-Qaeda are the two most crucial organizations in the terrorism network. We choose to focus on their attacks. Attacking targets, tactics and weapons used are utilized to describe attacks. These three types are selected in the ontology graph. All attacks are selected from the attribute list of al-Qaeda. The green nodes are attacks, the light blue ones are targets, the purple ones are weapons and the pink ones are tactics. Most attacks by al-Qaeda are bombing attacks using explosive weapons. The attacking targets vary from business, airport and government to diplomatic objects. We are able to identify 911 attacks, which are classified as unconventional attacks.

Next, attacks of Hamas are visualized in Fig. 17. There are 500 attacks related to Hamas. We select the 158 attacks from 2005 for visualization. Almost all of them are bombing attacks using explosive weapons. Different from al-Qaeda, most of Hamas' attack targets are citizens and private properties.

In our case study of the terrorism network, we show that OntoVis helps identify the most important organizations and analyze their relationships based on their locations and classifications. Also, OntoVis can do further investigations of a particular organization's behavior. The ontology graph and the abstractions provide strong support in visual analysis.

V. Future Work

OntoVis system makes use of the ontology information and several statistical measures to reveal the hidden knowledge in heterogeneous social networks. Other social network analysis techniques developed in sociology research [7], [35] can be incorporated to extend the capabilities of OntoVis. For example, evaluating the importance of relationships and filtering the trivial ones can reduce the visual complexity caused by the large number of links in the visualization.

In OntoVis, the ontology graph is only used for selecting node types. There are other potential uses of ontology information in network analysis [36]. In addition, more interaction between the derived abstraction and the ontology graph can be useful. For example, if the ontology graph associated with the original network would contain errors, analyzing the derived abstraction could uncover the errors and help users correct and refine the ontology graph.

An important task in social network analysis is to find "interesting" connections between two nodes such as the *connection subgraphs* algorithm [37]. Connection subgraphs capture the strongest relationships between two nodes and can be readily incorporated into OntoVis.

VI. Conclusion

In this paper we present OntoVis, a visual analysis tool for exploring and understanding large heterogeneous social networks. Ontology graphs are used as a guide for interactive exploration, coupled with simple but powerful semantic and structural abstractions and filtering using statistical measures of the nodes and links in the network. The resulting networks are intuitively displayed and respond to user interaction. Visual analytics tools such as OntoVis are expected to enable

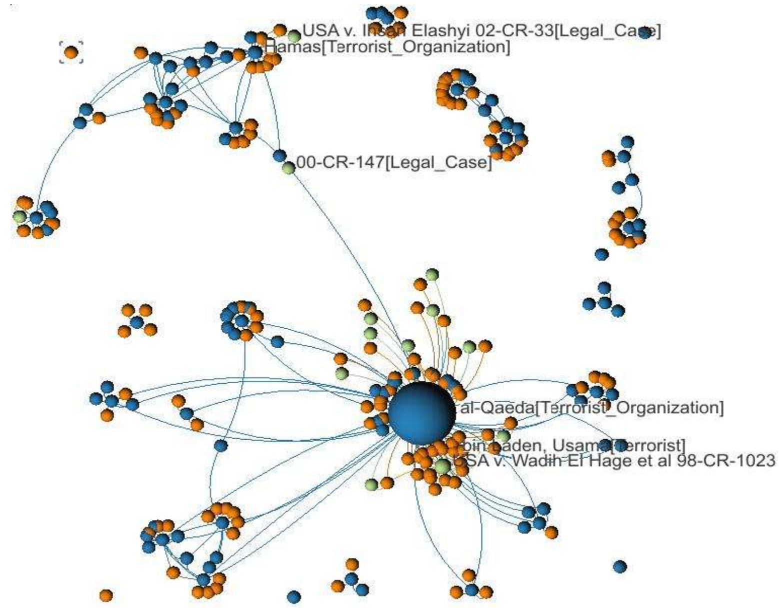


Fig. 15. Visualization of Terrorists (in orange), Legal Cases (in lightgreen) and Terrorist Organizations (in blue). Al-Qaeda is the organization related to most legal cases. The case involving most terrorists is USA vs. Wadih Ei Hage et al. 98-CR-1023.

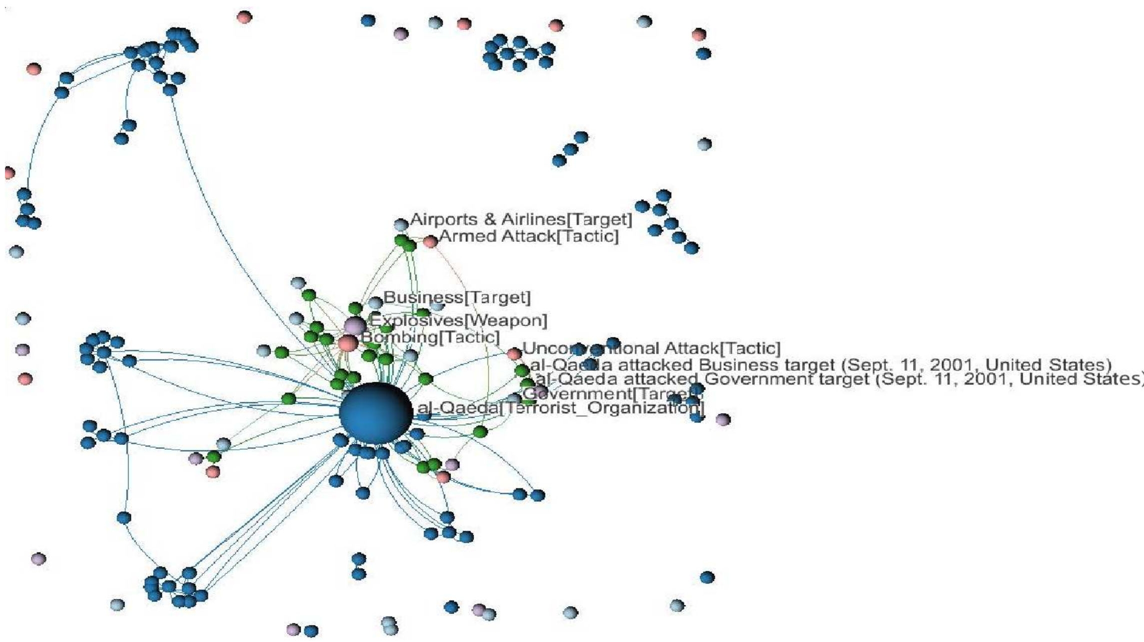


Fig. 16. Visualization of Terrorist Attacks (in green) Related to al-Qaeda. Al-Qaeda's major attacking tactic is bombing. Their attacking targets vary from business, airport, government to diplomatic objects.

breakthroughs in data exploration where information overload is a barrier to insight.

Acknowledgment

This work has been sponsored in part by the U.S. National Science Foundation under contracts CCF 9983641, CCF 0222991, OCI 0325934, and IIS 0552334, and the U.S. Department of Energy SciDAC program and under Lawrence Livermore National Laboratory Agreement No. B550194. The authors would also like to thank the MIPT Terrorism Knowledge Base and the UCI KDD Archive for making available the terrorism network and the movie network data sets, respectively.

References

[1] L.C. Freeman, "Visualizing Social Networks," *J. Social Structure*, vol. 1, no. 1, 2000.

[2] V. Batagelj and A. Mrvar, "Pajek: Analysis and Visualization of Large Networks," *Graph Drawing Software*, pp. 77-103, Springer, 2003.

[3] J.J. Thomas and K.A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*, IEEE CS Press, 2005.

[4] T. Kolda, D. Brown, J. Corones, T. Critchlow, T. Eliassi-Rad, L. Getoor, B. Hendrickson, V. Kumar, D. Lambert, C. Matarazzo, K. McCurley, M. Merrill, N. Samatova, D. Speck, R. Srikant, J. Thomas, M. Wertheimer

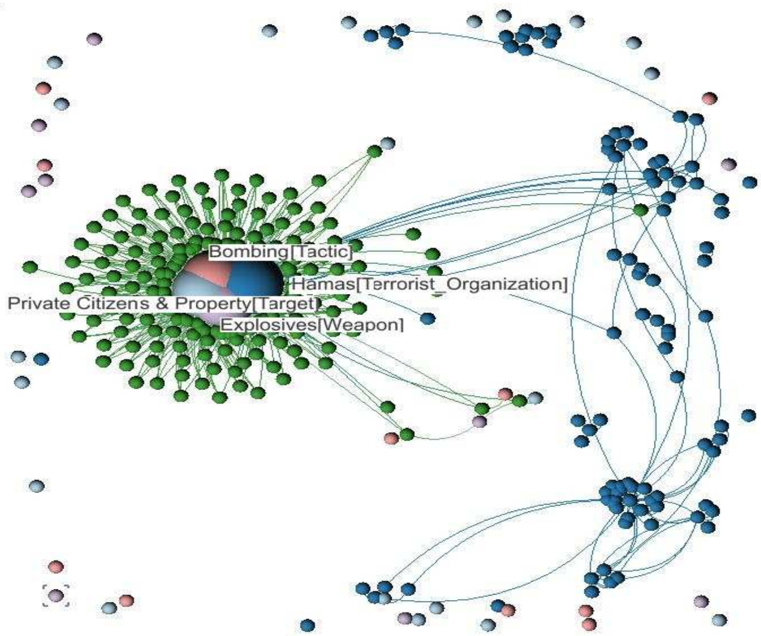


Fig. 17. Visualization of Terrorist Attacks (in green) Related to Hamas in 2005. Bombing is also the major tactic used by Hamas. However, their attacks focus on private citizens and properties.

and P.C. Wong, “Data Sciences Technology for Homeland Security Information Management and Knowledge Discovery,” Technical Report UCRL-TR-208926, Lawrence Livermore National Laboratory, 2004.

[5] D.J. Watts, *Small Worlds: The Dynamics of Networks Between Order and Randomness*, Princeton University Press, 1999.

[6] M. Barthelemy, E. Chow and T. Eliassi-Rad, “Knowledge Representation Issues in Semantic Graphs for Relationship Detection,” *AI Technologies for Homeland Security: Papers from the 2005 AAAI Spring Symposium*, AAAI Press, pp. 91-98, 2005.

[7] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.

[8] P. Doreian, V. Batagelj and A. Ferligoj, *Generalized Blockmodeling*, Cambridge University Press, 2005.

[9] S. Hettich and S.D. Bay, The UCI KDD Archive, http://kdd.ics.uci.edu, University of California, Irvine, Department of Information and Computer Science, 1999.

[10] MIPT Terrorism Knowledge Base, http://www.tkb.org/

[11] M. Huisman and M.A.J. Van Duijn, “Software for Social Network Analysis,” *Models and Methods in Social Network Analysis*, P.J. Carrington, J. Scott and S. Wasserman, eds., pp. 270-316, Cambridge University Press, 2005.

[12] International Network for Social Network Analysis (INSNA), http://www.sfu.ca/ insna/

[13] G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis, *Graph Drawing : Algorithms for the Visualization of Graphs*, Prentice Hall, 1999.

[14] I. Herman, G. Melancon, and M.S. Marshall, “Graph Visualization and Navigation in Information Visualization: A Survey,” *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 1, pp. 24-43, 2000.

[15] M. Jünger and P. Mutzel, *Graph Drawing Software*, Springer, 2004.

[16] J. Abello, J. Korn and I. Finocchi, “Graph Sketches,” *Proc. IEEE Symp. Information Visualization*, pp. 67, 2001.

[17] J. Abello and J. Korn, “MGV: A System for Visualizing Massive Multidigraphs,” *IEEE Trans. Visualization and Computer Graphics*, vol. 8, no. 1, pp. 21-38, 2002.

[18] E. Gansner, Y. Koren, and S. North, “Topological Fisheye Views for Visualizing Large Graphs,” *Proc. IEEE Symp. Information Visualization*, pp. 175-182, 2004.

[19] F. Van Ham and J.J. Van Wijk, “Interactive Visualization of Small World Graphs,” *Proc. IEEE Symp. Information Visualization*, pp. 199-206, 2004.

[20] V. Geroimenko and C. Chen, *Visualizing the Semantic Web*, Springer, 2002.

[21] F. Van Harmelen and C. Fluit, “Ontology-Based Information Visualization,” *Proc. the Fifth International Conference on Information Visualization*, pp. 546, 2001.

[22] NetDraw, http://www.analytictech.com/netdraw.htm

[23] S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, “Trawling Emerging Cyber-communities Automatically,” *Proc. 8th International World Wide Web Conference*, 1999.

[24] M. Granovetter, “The Strength of Weak Ties,” *American Journal of Sociology*, vol. 78, no. 6, pp. 1360-1380, 1973.

[25] B. Derrida and H. Flyvbjerg, “Statistical Properties of Randomly Broken Objects and of Multivalley Structures in Disordered Systems,” *J. Physics A*, vol. 20, no. 15, pp. 2573-2588, 1987

[26] M. Barthelemy, B. Gondran and E. Guichard, “Spatial Structure of the Internet Traffic,” *Physica A*, vol. 319, pp. 633-643, 2003.

[27] P. Eades, “A Heuristic for Graph Drawing,” *Congressus Numerantium*, vol. 42, pp. 149-160, 1984.

[28] U. Brandes. “Drawing on Physical Analogies,” *Drawing Graphs, LNCS 2025,* M. Kaufmann and D. Wagner, eds., pp. 71-86. Springer-Verlag, 2001.

[29] A. Noack. “An Energy Model for Visual Graph Clustering,” *Proc. Graph Drawing 2004*, pp. 425-436, Springer, 2004.

[30] D. Harel and Y. Koren, “Drawing Graphs with Non-Uniform Vertices,” *Proc. Working Conference on Advanced Visual Interfaces*, pp. 157-166, ACM Press, 2002

[31] E.R. Gansner and S.C. North, “Improved Force-Directed Layouts,” *Proc. 6th International Symposium on Graph Drawing*, pp. 364-373, Springer-Verlag, 1998.

[32] G.G. Robertson, J.D. Mackinlay and S.K. Card, “Cone Trees: Animated 3D Visualizations of Hierarchical Information,” *Proc. Human Factors in Computing Systems*, pp. 189-194, ACM Press, 1991.

[33] K. Misue, P. Eades, W. Lai and K. Sugiyama, “Layout Adjustment and the Mental Map,” *J. Visual Languages and Computing*, vol. 6, no. 2, pp. 183-210, 1995.

[34] Internet Movie Database, http://www.imdb.com

[35] D. Jensen and J. Neville, “Data Mining in Social Networks,” *Symp. Dynamic Social Network Modeling and Analysis*, National Academy Press, 2002.

[36] H. Alani, S. Dasmahapatra, K. O’Hara and N. Shadbolt, “Identifying Communities of Practice through Ontology Network Analysis,” *IEEE Intelligent Systems* vol. 18, no. 2, pp.18-25, 2003.

[37] C. Faloutsos, K. McCurley and A. Tomkins, “Fast Discovery of Connection Subgraphs,” *Proc. the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 118-127, ACM Press, 2004.

Zeqian Shen received the BS in Computer Science from the ZheJiang University, China, in 2001 and the MS from University of Tennessee, Knoxville, in 2004. He is currently working toward the Ph.D. degree at University of California, Davis. His research interests include information visualization and data mining, with focus on interactive graph visualization and analysis.

Kwan-liu Ma is a professor of computer science at the University of California at Davis. His research interests include visualization, user interfaces, and high-performance computing. He has a PhD in computer science from the University of Utah. He is a senior member of the IEEE and a member of the ACM.

Tina Eliassi-Rad earned a Ph.D. in Computer Sciences (with a minor in Mathematical Statistics) at the University of Wisconsin-Madison. She is currently a computer scientist at the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory. Her research interests include artificial intelligence, machine learning, knowledge discovery and data mining. Her work has been applied to the World-Wide Web, large-scale scientific simulation data, and complex networks. She is a member of AAAI, ACM, IEEE, and SIAM. For more details, visit http://www.llnl.gov/CASC/people/eliassi-rad/.