Intelligence Analysis Using Titan

Patricia Crossno, Brian Wylie, Andrew Wilson, John Greenfield, Eric Stanton, Timothy Shead, Lisa Ice, and Kenneth Moreland*

Sandia National Laboratories

Jeffrey Baumes and Berk Geveci**

Kitware Inc.

Abstract

The open source Titan Informatics Toolkit Project, which extends the Visualization Toolkit (VTK) to include information visualization capabilities, is being developed by Sandia National Laboratories in collaboration with Kitware. The VAST Contest provided us with an opportunity to explore various ideas for constructing an analysis tool, while allowing us to exercise our architecture in the solution of a complex problem. As amateur analysts, we found the experience both enlightening and fun.

CR Categories and Subject Descriptors: H.5.2 [Information Interfaces & Presentations]: User Interfaces – Graphical User Interfaces (GUI); I.3.6 [Methodology and Techniques]: Interaction Techniques.

Additional Keywords: visual analytics, information visualization.

1 Overview

In our analysis, we used the open source Titan Informatics Toolkit [1] that's currently part of the Visualization Toolkit (VTK) [2]. The Titan components include a wide variety of modules including database drivers, informatics data structures, processing filters, and infovis 'views'. These components allowed us to quickly pull together an application, DatabaseView (seen in Figure 1), to do our analysis for the VAST contest.

DatabaseView is really just combining existing toolkit components. The database drivers allowed us to access any type of database, the query classes produced tables, the tables were transformed into graphs, and the graphs were sent to views. We were pleasantly surprised by how effectively we could build an application from scratch during just the short time frame of the contest. Users interact with DatabaseView by creating SQL queries for nodes, edges, and selections - which are displayed as interactive graphs with control over layout, color, and labeling.

We used a commercial tool, Clear Forest [3], to extract entities from the contest data and to build relationship tables that were then imported into a MySQL database using a custom parser and a schema of our own design. Additional links between entities were created by linking entities that appeared within the same document. Text within images was manually entered and entities were hand-tagged for inclusion in the database.

Our strategy was to use all entities, terms, source documents, and known relationships from the database to generate a graph that casts as wide a net as possible, and then to probe that graph with focused queries to generate subgraphs (seen inside the pink rectangle, Figure 1) containing relevant pieces of the puzzle. Although it requires a degree of SQL expertise, building graphs in this manner gives us the flexibility and control needed for rapid exploration of complex queries.

We iteratively build a large and complex query that adds nodes to a current hypothesis graph. Although we occasionally do a side query on an unrelated topic, we keep a copy of the hypothesis query for when we resume our solution building. When we have difficulty merging disparate graph sections, we can sometimes find a linking node through a breadth-first expansion around current nodes in our solution. We see this in the interface through the concept of a neighborhood parameter.

The search is done by taking a neighborhood of nodes and edges around each of the selected nodes from within the wide-net graph. A neighborhood of one includes all nodes directly connected to the selected nodes by an edge. A neighborhood of two includes all nodes that can be reached by traversing no more than two edges. Larger neighborhoods are rarely useful because the number of extraneous nodes and overlapping edges quickly grow to hide any useful information. The neighborhood is set through the parameter field (pink oval, Figure 1). The button to its left activates the graph generation. Alternatively, nodes can be selected through a non-SQL interface (green box) that lists Clear Forest tag categories in one window, and the values for the selected category within the other.

Once the subgraph is generated, entities within it can be examined in detail. Rubber-banding can be used to select source files (both text and image) from the subgraph. The selected files are then listed in the window shown in the upper right corner of the interface. As an example of this drill-down capability, an image from the selected source files (blue box, Figure 1) is displayed below (bottom of blue arrow, Figure 1).

Although a great deal can be inferred from just the entity/term relationships shown in the subgraph, much of our analysis involved reading articles or viewing images whose relevance was suggested by the subgraph.

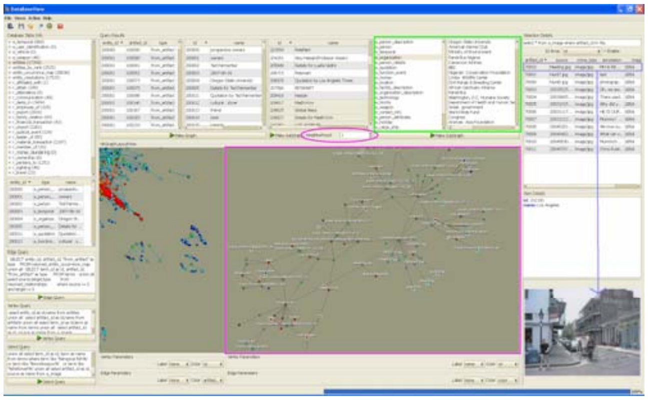

Figure 1. DatabaseView - full interface with all query windows.

Email: *{pjcross, bnwylie, atwilso, jagreen, etstant, tshead, lgice, kmorel} @sandia.gov,
**{jeff.baumes, berk.geveci}@kitware.com

2 Analysis

2.1 General Approach

Our general approach was an iterative one. Starting from some conceptual thread or idea, we performed the following steps repeatedly to extract an expanding set of entities and relationships to form a solution graph:

- Identify a thread to investigate.
- Do an SQL query on thread-related entities and terms.

  OR

  Select entities from the category view (green rectangle, Figure 1), where lists of entities are grouped by category.
- Select a subset of relevant entities and terms from a list returned by the query.
- Select a neighborhood distance value.
- Generate a subgraph containing nodes and edges within the neighborhood of the query result (pink rectangle, Figure 1).
- Select source documents in the subgraph for examination.
- View source documents to identify next query thread.

The subgraph integrates many different types of data into a single figure, including source files, people, places and organizations. This enables us to visualize a broad view of significant entities and relationships, while simultaneously keeping track of information sources and permitting full source access. Although the full interface has a large number of windows, unneeded ones can be hidden, while the others can be expanded to make full use of the available screen real estate.

One problem that came up repeatedly was entity resolution. There were several manifestations of the problem, including nicknames, aliases, misspellings or typos, and partial names or initials. We developed an entity resolution capability allowing us to map multiple entities to a single main entity description.

2.2 Results

The contest data contained a central plot to create a monkeypox outbreak carried by pet chinchillas intended to halt demand for, and poaching of, endangered wild chinchillas. Subplots included exotic animal smuggling, drug smuggling, and animal rights terrorism.

To give a flavor for how our tool visualizes part of the monkeypox plot, Figure 2 shows a section of the subgraph returned from a query on chinchillas with a neighborhood of two. The monkeypox outbreak is shown in the upper left portion of the graph (large pink circle). The relevant source articles are given by the three square green nodes. We also see references to poaching and Chile and Rosalind Baptista (pink circles). Baptista is connected to an image of a meeting in New Orleans (green circles). Key players, locations, concepts, information sources and relationships are shown in the graph. Text and image files can be selected and displayed in a linked view to analyze detailed source information. This is vital for finding additional entities or relationships that then become leads for the next iteration of queries.

3 Conclusions

We got involved in this contest to learn about analysis by doing it, and, in the end, we have learned a great deal. We found that the central strength of our tool is also its main weakness, namely the use of SQL queries as our user interface. Although it permits us to form complex and powerful queries, it is also a formidable obstacle to learning to operate the tool. In future work we intend to reserve this interface as an expert user mode while developing a more user-friendly alternative for novices.

We also discovered that there were a number of other features we would have liked to have had. We often felt the need for a 'scratch pad' into which we could pull interesting sections of the graph. We wanted to make annotations and have assistance in suggesting possible entity resolution candidates. Also, time is not seen in our graphs and we have some ideas about how to add time-based mappings, since the timeline of events is often critical to the analysis.

4 Acknowledgements

We would like to thank the contest organizers for putting together an entertaining and complex puzzle to solve. This has created an invaluable opportunity to walk in an analyst's shoes for awhile. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

References

[1] Infovis Support in VTK. In *Kitware Software Developer's Quarterly*, issue 4, page 11. April, 2007. http://www.kitware.com/products/newsletter/kitware_quarterly0507.pdf.

[2] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics, 4th Edition*. Kitware, 2006.

[3] Clear Forest. http://www.clearforest.com

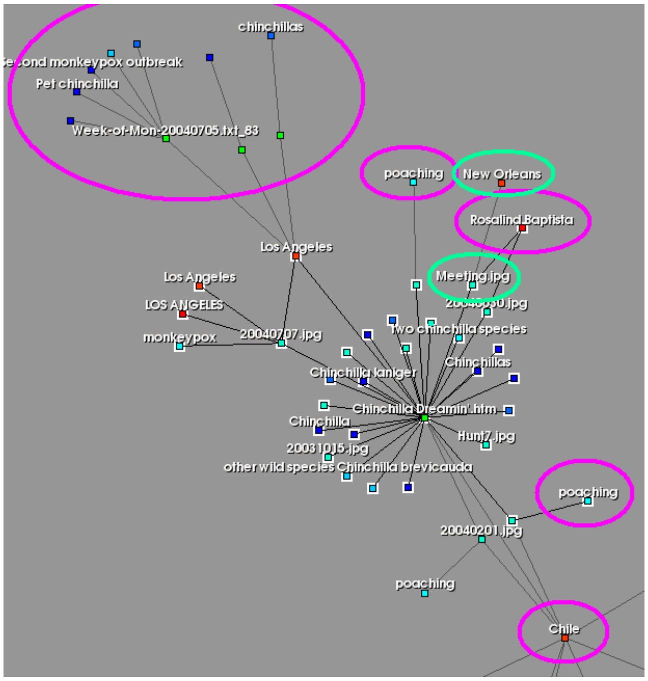

Figure 2. Closeup of chinchilla query subgraph generated with a neighborhood of two.