VAST 2014 Mini-Challenge 1: Team Aptima

Caroline Ziemkiewicz* Aptima, Inc.
Adam Fouse* Aptima, Inc.
Gabriel Ganberg* Aptima, Inc.
Ryan Mullins* Aptima, Inc.
Stacy Pfautz* Aptima, Inc.

ABSTRACT

We present a prototype visual analytics tool for facilitating analysis in a collaborative team setting by providing a flexible platform for visualizing and exploring multiple linked datasets as a unified graph model. Our approach uses a Layered Graph Model to connect multiple datasets and afford querying and analysis across linked data. We used this tool to explore and form hypotheses about the disappearance on Kronos in Mini-Challenge 1, revealing some initial strengths and weaknesses of the approach.

Keywords: Visual analytics, graph-based data models

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Miscellaneous

1 INTRODUCTION

Researchers at Aptima have developed a prototype visual analytics tool designed to help analysts find meaningful connections among multiple models and datasets. This tool was inspired by problems in highly collaborative analytic environments characterized by small teams working with multisource heterogeneous data. When a request for information is received, each analyst works on a piece of the problem using their own workflow and tools. For example, one analyst might perform a social network analysis, one might develop statistical models from demographic and survey data, and another might analyze topics contained in a corpus of text documents. After the individual analyses are done, the team must bring their work together to form a complete picture of the problem space.

This process is challenging and time consuming, and can lead to missed connections between analyses done at different times, by different people, in different formats. Our work addresses this challenge by making it possible for a single user to import the outputs of their colleagues' work into a unified data model and explore the results.

The 2014 VAST Challenge provided a good model for the types of questions that our tool is intended to help answer. We imported and analyzed Mini-Challenge 1 as a test case, and have continued to import the rest of the challenge data for evaluation use. The challenge use case demonstrates the advantages of our approach and areas for further research.

2 VISUAL ANALYSIS SYSTEM

2.1 Data Import and Integration

The first step to analyzing data in our tool is to import and integrate the data and models. This import process is designed to be highly flexible to multiple data types and formats. Our approach uses a Layered Graph Model which creates a flexible schema on top of a basic graph structure. Graph databases represent data as a set of vertices and edges connecting vertices. Edges and vertices can both have a set of properties associated with them, and edges have a label that allows for multiple types of relationships between vertices to be represented. This structure is particularly well suited for representing large dynamic datasets where relationships are important but the schema is not fixed. The Layered Graph metamodel defines the types of entities and relationships allowed in the graph, but can be changed at runtime to adjust to a dynamic dataset.

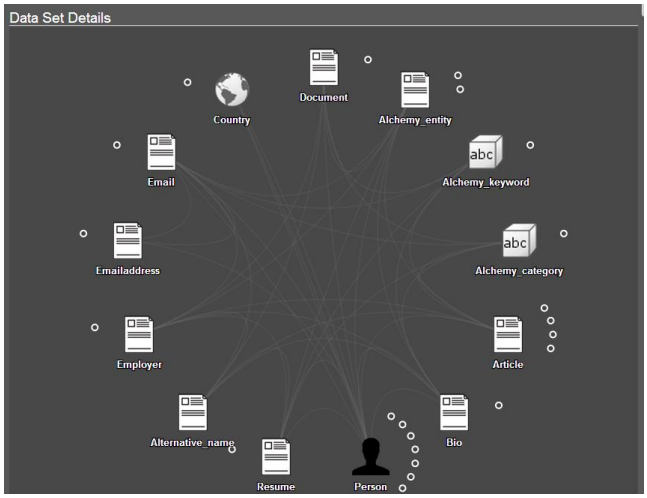

Figure 1. The Overview tab summarizes the entities and relationships in the layered graph model of the MC1 data.

The Layered Graph model is implemented as a Java application called the Context Engine. Scripts that construct and run the import functions are available through a command line interface as well as a web-based REST API [2]. Once the relevant data is modeled within the graph, graph query and inference algorithms can be used to reason about the knowledge contained within the graph. Typically these queries are accomplished through graph traversing or graph pattern matching.

To prepare the mini-challenge data, we performed some additional analyses using off-the-shelf software and techniques. We performed entity and keyword extraction over the news stories using open-source Natural Language Processing tools. We then performed a simple text search with the extracted entities and keywords to additionally link them to the historical documents, resumés, and email headers.

Using import scripting, we imported each of these individual datasets and analyses as a data layer. In the Layered Graph model, a layer is a graph partition that contains either an imported raw dataset or an integrated dataset based on merging, joining, and/or analyzing a set of source datasets. We then ran a script to generate a combined layer with all of the data together. This script matches

* [cziemkiewicz, afouse, ganberg, rmullins, spfautz]@aptima.com

IEEE Symposium on Visual Analytics Science and Technology 2014
November 9-14, Paris, France
978-1-4799-6227-3/14/$31.00 ©2014 IEEE

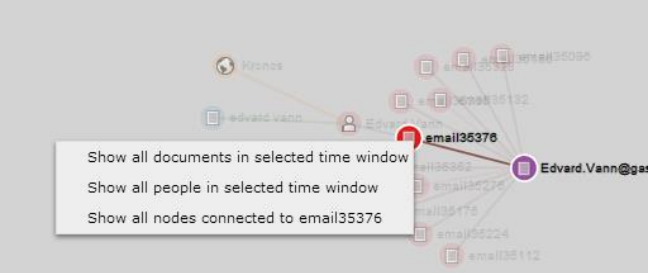

Figure 2. The context menu querying function suggests queries for related entities based on a selected graph node.

entities on designated identifying fields and combines their relationship links and attributes to form a single entity in the combined layer. The result is a unified graph model that represents all of the data and connections between them that can be automatically extracted.

2.2 Visualization and Exploration

The front end of our analysis tool is a web application developed in HTML5/JavaScript using D3 [1]. The visual analysis portion has two tabs: the Overview and the Exploration view. The Overview tab summarizes the combined model so a viewer can see what entity types exist in the data, how they relate to one another, and what attributes they have (Figure 1). This screen serves as a starting point for further analysis of the data. The user can click on one of the entities to load entities of that type into the Exploration view, which allows interactive visual analysis of the data. The user can also go directly to the Exploration view and start searching or building queries to load in data.

The Exploration view consists of three linked views: a timeline, a map, and a node-link graph of the data model itself. These are standard components that can be relevant to a wide range of data, so they are the default views provided to maintain flexibility.

Even in a relatively small data collection such as that in MC1, the combined graph model is too large and heavily interconnected to be legibly visualized as a node-link diagram. Therefore, interaction in the Exploration view is focused on allowing the user to load in subsets of the data for visualization focused on segments of interest. The analyst can load specific entities through free text searches and an advanced Query Builder interface, or follow branching paths of inquiry by selecting items in the visualizations loading connected entities through a context menu. The support for multiple querying modes is designed to help the user explore the underlying graph model in a variety of ways.

3 Analyzing Mini-Challenge 1

In our analysis of MC1, we most frequently used the free text search and the context menu queries. This confirmed our intuition that the Query Builder is a more advanced functionality that would be rarely used for open exploration. By contrast, the context menu queries allowed us to explore the data more naturally through association (Figure 2).

As an example of the process, we discovered a group of GasTech employees with possible links to Protectors of Kronos through a combination of search and associative querying. We started by bringing up the GasTech personnel network, including employees, their resumes, and previous employers. We then visually scanned the network and noticed a cluster of people with previous experience in the Armed Forces of Kronos. One of these people was, Isia Vann, who also appears in the historical documents on the PoK. We cleared the previous queries and searched for Vann specifically, then used context menu querying to load everyone who emailed him in the data. Exploring this

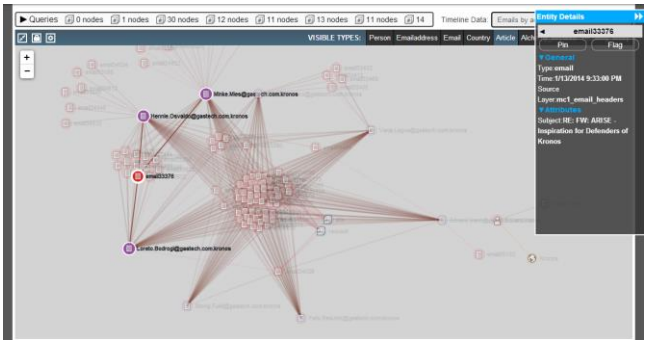

Figure 3. Exploring entities in the graph view reveals a suspicious email thread among members of the GasTech security team.

network revealed that he frequently emailed several other employees with surnames shared by current or former PoK members. Scanning the attributes of these entities in the detail view showed that they were all part of the security team, and further time filtering brought up potentially suspicious email threads among them (Figure 3). This exploration served as evidence for one of our two hypotheses about the disappearances.

4 Lessons Learned

This process has revealed some advantages and disadvantages of our approach in its current form. One of our primary goals in entering the challenge was to test our ability to import a new data collection and integrate it into a combined graph model, and this test proved successful. The architecture we developed not only supported efficient integration of the MC1 data, but also made it possible to easily add additional data as we continued the analysis. We also found that our interaction design supports a wide range of useful querying and exploration behaviors. The context menu querying in particular makes good use of the graph model to let analysts follow a hypothesis by association. We will test this hypothesis, among others, in an upcoming evaluation study.

The process also revealed some areas that require further work. While our tool is designed for exploration, the MC1 questions focused on dissemination: that is, producing a report for a decision-maker. We do not currently have a way to express findings in a visually concise way. While this hasn't been a priority for our current use case, it is a limitation on use in a collaborative setting, and should be addressed in longer term research. Our tool also does not support built-in entity resolution beyond basic string matching. This caused some issues in the MC1 data since there were several cases of people or groups having alternate names or spellings, and some of these had to be resolved manually. Adding this functionality would be an obvious improvement to support flexible import.

5 Conclusion

Our goal is to facilitate collaborative analysis by providing a flexible platform for visualizing and exploring multiple linked datasets as a unified graph model. Mini-Challenge 1 provided a realistic test scenario that demonstrated our system's flexibility and querying power, while identifying some areas for further research and development.

References

[1] Bostock, M., Ogievetsky, V., & Heer, J. (2011). D³ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on, 17*(12), 2301-2309.

[2] Fielding, R. (2000). Representational state transfer. *Architectural Styles and the Design of Network-based Software Architecture*, 76-85.