Interactive Demo: Using CZSaw to Analyze Entities in Collections

Victor Chen, Dustin Dunsmuir, Nazanin Kadivar, Eric Lee, Jeffrey Guenther, Saba Alimadadi Jani,
John Dill, Chris Shaw, Robert Woodbury, Maureen Stone, Cheryl Qian

School of Interactive Arts and Technology, Simon Fraser University

ABSTRACT

CZSaw [1] is a visual analytics tool for sense-making across entities, entity collections, and relations with a focus on augmenting the analysis process. It uses a variety of flexible data visualizations to represent, explore, and compute networks of entities and relations from different perspectives. CZSaw is designed to provide a replayable record of the analysis process and to generate a reusable model of the analysis logic, structured as a dependency graph. To support these goals, semantically meaningful interactions are captured into a script. Replaying this script replays the analysis process, and editing it allows fine control and reuse of the process. Specialized viewers are also provided for the dependency graph and for the user's history, to provide more visual interaction. This demo shows how CZSaw can be used to analyze different types of datasets (structured and unstructured data), as well as some strategies (e.g. divide and conquer) used on analysis tasks.

KEYWORDS: visual analytics, investigative analysis, intelligence analysis, sense-making, analysis process

INDEX TERMS: I.3.8 [Computer Graphics]: Applications-Visual Analytics, I.6.9 [Visualization]: information visualization, H.5.2 [Information Systems]: Information Interfaces and Presentation.

1 SYSTEM DESCRIPTION

CZSaw uses visualization and manipulation of entities, entity collections, and relations to support the sense-making process. An entity collection can be a text document mentioning various entities (such as people, locations, dates, etc), or a row in a spreadsheet with columns as entities. CZSaw is inspired by Jigsaw [2], which seeks to help analysts discover hidden facts in large collections of text reports by making connections between disparate facts. CZSaw extends this approach by capturing and encoding the analysis process, enabling the user to replay, modify and reuse analysis procedures and relationships, which are modeled as a dependency graph. As well as providing a rich set of tools that give the analyst the power to visualize and manipulate entities and relations, CZSaw provides tools to visualize, replay and modify the complex analysis process itself.

1.1 Capturing the Analysis Process

The analytic process is a sense-making process. The analyst typically iterates through sequences of steps many times, varying parameters each time, to gradually make sense of the data. CZSaw provides an editable, re-playable, and re-useable mechanism to help analysts understand, explore, reference, and reuse their analysis.

E-mail: {yvchen, dtd, nka23, ela10, jguenthe, salimada, dill, shaw, rw}@sfu.ca, stone@stonesc.com, qianz@purdue.edu

User interactions are recorded and translated into a script language at a task level. We only record meaningful tasks such as "find all entities related to at least two entities in the selected group", and not every single mouse movement. Interaction and results (and parameters involved in the interaction) are encapsulated into variables and functions, and stored as a *transaction* consisting of one or more lines of script language code. CZSaw's *Script View* supports editing script transactions and navigation through the process by rewinding and replaying transactions of the analysis.

The script generated from a user's interactions creates a model of the analysis process in the form of a propagate-able dependency graph. The model comprises analysis results (variables) and connections among results (interactions to generate these variables). Any change in a variable triggers the underlying propagation mechanism to automatically update downstream variables in turn reflected in data views. It allows the user to quickly reuse parts of the analysis process by assigning new data to a variable. CZSaw's *Dependency Graph View* (Fig. 1) directly visualizes the dependency model, helping users understand the logic and relations of analysis results.

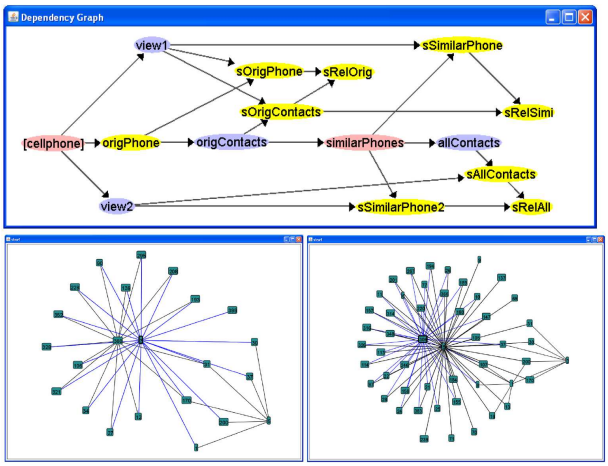

Figure 1. Above: Dependency graph of searching replacement cell phone (VAST 2008 mini-challenge). Bottom: Reuse of the dependency graph to generate the graph for multiple phones.

CZSaw also provides a *History View* to visualize the analysis process and to quickly explore and access previous states. The *History View* displays past states as screenshots in a temporal order, allowing the user to quickly recall the past analysis actions, and easily see an overview and access history states.

1.2 Data Views

CZSaw provides several views to visualize and interact with data from different perspectives:

Semantic Zoom View (SZV): examines documents at several levels of detail (overview, entities in the document, and detailed text). It incorporates continuous zoom and the visual information

seeking mantra: "overview first, zoom and filter, then details-on-demand". Each document is represented by a small rectangle; the user can zoom into each to see its set of entities. Further zooming allows reading the full text while retaining the context of surrounding documents (Fig. 2). A clustering algorithm places documents containing many of the same entities close together. The user can apply a divide-and-conquer strategy by creating groups of related documents to divide the set into smaller chunks.

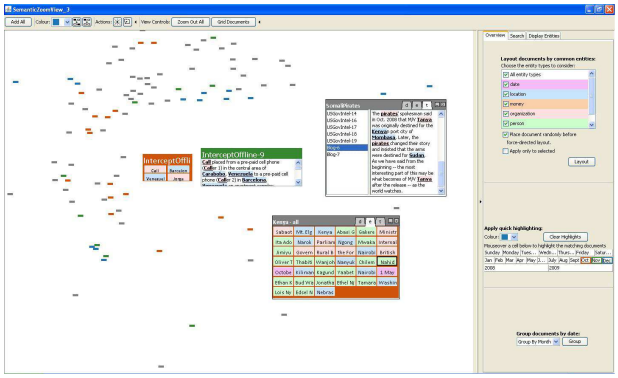

Figure 2. Semantic zoom view visualizes documents with different level of details from grouped overview to textual contents.

Hybrid View: An enhanced graph visualization showing entities and relations in a node-link graph where the nodes can be represented with visualization techniques (e.g. list, grouped nodes, bars, temporal and spatial layouts) to visualize different types of data (Fig. 3). It allows quick access to temporal and spatial layouts without losing the relationships to non-temporal and non-spatial data. By changing a node's visual state, the level of data detail can be adjusted to drill down into a dataset without losing the context of the analysis.

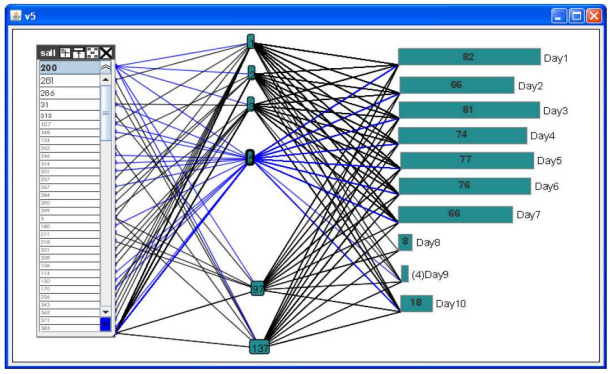

Figure 3. Hybrid View. Communications in the 10 day period, VAST'08 challenge

Document View: For datasets containing document collections, the analyst must read at least some documents. This view displays the content of selected documents highlighting mentioned entities.

These views are driven by CZSaw's dependency propagation mechanism. Views automatically update themselves to reflect the content change caused by the propagation from the dependency graph.

1.3 Data Query and Entity Management Functions

CZSaw provides computational power for data query, computation, and management. These functions can be accessed through the user interface, or via the script view (for advanced control). Script commands include:

Data view commands: Control visualization states, such as show/hide, layout, and aggregation levels of the visualization (e.g. showing a set of entities as scattered nodes, in a list, or grouped as a single node).

Data query commands: Query, filter (entities or relations) from CZSaw's database. For example, search for entities by value comparison, search for related entities to a set of entities, or get relations between two entity sets. In CZSaw, users need not deal with a very crowded graph containing all the entities. They can work selectively on a set of entities.

Entity refinement commands: Extract, merge, edit, and link entities. CZSaw relies on entities and relations to generate visualizations. Text documents involved in real world problems are usually messy and contain inconsistencies and errors and automated entity extraction is often not satisfactory. Thus, we provide entity management commands to allow users to refine entities on the fly within the analysis. Within the data views the user can extract new entities, merge entities (e.g. an entity with different aliases or misspelled names), edit/remove entities, and link entities (e.g. associate phone numbers with individuals).

2 Applications of CZSaw to a Variety of Data

We have applied CZSaw to different types of data to demonstrate its features and how it supports analytical strategies. Datasets used include VAST challenge data (2008 and 2010) and FAA wildlife strike data. One key problem in the VAST 2008 cell phone mini-challenge is finding several phones' replacement phones. The underlying logic is that if two phones call a similar set of phones, these two phones may belong to the same person. We will demonstrate how CZSaw tests this hypothesis through interaction for one phone and then apply the same set of actions for the remaining phones. The interactions are captured into a script and construct a dependency graph. The dependency graph is then reused to find replacements for other phones (Fig. 1). In conjunction with this, CZSaw's hybrid view visualizes the temporal pattern of phone usage, which strengthens the hypothesis since the phones are used at different times.

The 2010 arms dealer mini-challenge is mainly textual data. Its multiple threads resulted in an inaccurate entity extraction. We use this problem to demonstrate CZSaw's entity refinement functions, the *Semantic Zoom View* (SZV), and the divide-and-conquer strategy of handling multiple threads by grouping documents. Both the SZV and the *Hybrid View* can be used to manipulate and view clusters of documents. The analyst can then work on a smaller number of documents from each cluster. While examining details of documents and entities, the user may find many imperfect machine-extracted entities, and use entity refinement commands (editing, extracting and merging) to refine them. These operations will change the entities and relations, which cause the dependency graph to propagate from the root. This leads to layout changes in views, creating new clusters or merging existing clusters.

The FAA wildlife strike data is a large dataset containing both structured data and textual content. We use CZSaw to explore, analyze and report relations among wildlife species, seasons, time of day, aircraft types, and text reports.

References

[1] N. Kadivar, V.Chen, D.Dunsmuir, E.Lee, C.Qian, J.Dill, C.Shaw and R.Woodbury, "Capturing and Supporting the Analysis Process", Proceedings of IEEE Visual Analytics Science & Technology 2009, IEEE, Atlantic City, NJ, Oct 11-16, 2009, pp. 131-138.

[2] J. Stasko, C. Gorg, and Z., Liu, "Jigsaw: Supporting Investigative Analysis through Interactive Visualization", Information *Visualization*, Vol. 7, No. 2, Summer 2008, pp. 118–132