Visual Text Analytics for Impromptu Analysts

Oriana J. Love, Daniel M. Best *Member, IEEE*, Joseph R. Bruce, Scott T. Dowson and Christopher S. Larmey

Abstract— The Scalable Reasoning System (SRS) is a lightweight visual analytics framework that makes analytical capabilities widely accessible to a class of users we have deemed “impromptu analysts.” By focusing on a deployment of SRS, the Lessons Learned Explorer (LLEx), we examine how to develop visualizations around analytical-oriented goals and data availability. We discuss how to help impromptu analysts to explore deeper patterns. Through designing consistent interactions, we arrive at an interdependent view capable of showcasing patterns. With the combination of SRS widget visualizations and interactions around the underlying textual data, we aim to transition the casual, infrequent user into a viable–albeit impromptu–analyst.

Index Terms—Visualization, Visual analytics, Text analytics, End user, Analyst

1 Introduction

Within the visual analytics community, we strive to construct windows into data to make way for discovery. Unfortunately, several rounds of data import, processing, filtering, sorting, manual reordering, outlier exclusion/inclusion, and other data massaging techniques often precede such moments of discovery. For analysts who have the time needed to maintain such a demanding relationship with their data, adjusting such parameters may be acceptable. For user populations that are infrequent analysts or unfamiliar with the data, such fine-tuning is unrealistic and may result in software rejection.

The Scalable Reasoning System (SRS) provides a web-accessible capability for “impromptu analysts” to visually represent their data without the need to learn complex software. We define the term impromptu analysts to include individuals who are not dedicated data analysts but instead use visual analytic tools casually or infrequently to aid in decisions. SRS can be tailored to include different widgets for unstructured or structured text depending on the users’ analytical goals and the type of data available. The term “widgets” describes the visualizations that combine to create an SRS application. Designed for customization and flexibility, each SRS application instance can include the set of visualization widgets most closely aligned with the analytical goals of its users. Additionally, SRS does not require data of a particular schema to be imported but rather custom data modules provide direct access to arbitrary data stores to create datasets for analysis. Text analysis algorithms then process the dataset and the results are transmitted to the visualizations available in the web client.

In this paper, we focus on the affordances of different widgets that can satisfy differing analytic goals for impromptu analysts through a case study around an SRS application, the Lessons Learned Explorer (LLEx). Then by leveraging the capabilities of the widgets in conjunction, we explore how one can find patterns among categorical, structured and unstructured portions of data. We detail the selection and drill-down paradigms used and the tradeoffs made to maintain an optimal balance between interaction simplicity and analytic power needed to support impromptu analysts.

2 Background

SRS was originally developed as a mechanism to distribute analytic capability to a wide audience [8]. The SRS framework is a collection of components that parse, analyze, and model information through a lightweight web client coupled with a service-oriented architecture.

Our original SRS prototype provided insight on the feasibility of providing analytics via a web front end. In the next version of the software, we improved the original model by addressing user experience inconsistencies and providing a componentized system where capabilities could be chosen a la carte per deployment.

2.1 Related Work

There are several notable applications in the family of text and semi-structured data analysis tools of which SRS is a part. Some, such as Analysts Notebook and Jigsaw, enable exploration through thick-client applications [6, 12]. While it is possible to provide a user with a richer experience through the use of standalone applications, requiring client-installation limits the applications flexibility and potential audience. We instead chose to provide our analytic capability using a standard web browser to broaden SRS’s potential audience.

Like SRS, other tools deliver their content over the web [7, 2, 3]. VisCept and Dashiki focus on the collaborative elements of sense making in their visualizations, while VizGets facilitates web searches. All have multiple linked views (maps, graphs, timelines) for exploring an information space. Dashiki allows users to customize their dashboard by specifying which predefined visualizations they want to use against either embedded or linked datasets. VizGets and VisCept have a predefined dashboard layout that does not support customization of the visualizations. While users cannot tailor the interface as in Dashiki, SRS does allow for customization of the dashboard as specified by application needs.

2.2 Case Study: Lessons Learned Explorer

Pacific Northwest National Laboratory (PNNL) is a U.S. Department of Energy laboratory that performs a wide range of scientific research in areas such as cyber security, non-proliferation of weapons of mass destruction, hydrogen and biomass-based fuels, and environmental effects of energy generation. With nearly 5000 staff working on 2000 projects in these diverse disciplines, there is tremendous value in sharing lessons learned in critical areas, such as safety, management, and security. These lessons are captured as articles and shared across projects and organizations so all can benefit from experiences and improve performance across domains. PNNL hosts this information on an internal website that is accessible by all staff, including project managers, project teams, workflow process designers, and the Lessons Learned Operations team. The current website has limited search, tagging, and keyword capabilities. These limitations frustrate users who want to quickly find relevant articles. In particular, project managers use the articles to decide which risks are relevant to their project plans; thus it is imperative for them to understand the large corpus and identify applicable articles.

To improve service and value to website users, the Lessons Learned Operations team began looking for a faster, more accurate way to find articles based on different topics, keywords, and subsets of interests. While tagging had been a helpful way to find and manually group articles, it was cumbersome and difficult to keep consistent and current. The new system needed to be web-based, user-friendly, easily tailored, and capable of supporting thousands of articles and handling both structured and unstructured data. As the second version of SRS

- *Oriana Love, Daniel Best, Joseph Bruce, Scott Dowson and Christopher Larmey are with Pacific Northwest National Laboratory, E-mail: first.last@pnnl.gov.*

Table 1: Determining Widget Utility Based on Analytical Goals and Data Structure

Widget	LLEx	Analytical Goal	Data Structure
Word Clusters	✓	Understand the natural groupings of unstructured textual data based on content	Unstructured textual data
Streamgraph		Understand how key themes within the data shift over time	Temporal fields correlated with bodies of unstructured textual data
Faceted Browse	✓	Explore and drill into the relationships between categorical data	Structured categorical data
Treemap	✓	Visualize relative distributions of hierarchical data	Hierarchical data that can be counted (categorical) or has a numeric representation
Geospatial		Understand and explore the geographical distribution of a data set	Geographic coordinates
Timeline Histogram	✓	Visualize and interact with data based on date or time	Timestamps or temporal ranges
List	✓	Browse individual articles from the data collection	Uniquely identifiable and recognizable attributes of data

is optimized for lightweight visual analysis of structured and unstructured data, it is an appropriate framework from which to create the Lessons Learned Explorer (LLEx) application.

3 Visual Metaphors for Analytics

Each SRS application team must decide which widgets best fit the users' analytical goals. For example, do users need to make decisions based on the geo-location of a source? Will the age of the data help inform these decisions? If the widget helps meet a user's analytical goal, an additional feasibility check must be made to determine if the underlying data is capable of telling the story. Some widget visualizations lend themselves better to structured or unstructured data. Even if the data supported all widgets, the clutter resulting from displaying all widgets would be detrimental to the utility of an application and might distract impromptu analysts from the decisions at-hand. For LLEx, each widget was considered to ensure analytical goals were first addressed. After determining the utility of a widget in the decision-making process, the Lessons Learned articles were evaluated to make sure that they contained the proper data structures to produce a valid visualization. Table 1 shows describes available SRS widgets, if they were selected for LLEx, their analytical value, and their required data structures.

3.1 Unstructured Text Analysis

The vast majority of business-relevant information is captured in an unstructured form [4]. SRS has two text analytic algorithms available for unstructured text, supporting the word cluster and streamgraph widgets. *Text clustering* categorizes large collections of unstructured text documents into manageable groups of similar documents. *Theme generation* extracts major topical threads from documents through the use of the Rapid Automatic Keyword Extraction (RAKE) [10] algorithm.

3.1.1 Word Clusters

The word cluster widget analyzes unstructured text, partitioning the document collection using differentiating words detected within raw, descriptive text across the entire document space. The three most distinctive words in each cluster are presented as the identifier for that cluster (e.g., *fire, hazard, recall*). Each cluster can be further decomposed into smaller clusters in order to explore subsets or perform a more granular selection. This decomposition process results in a cluster hierarchy, much like the file system folder paradigm familiar to users but one that is produced automatically by SRS. The clusters are ordered based on the number of documents within those clusters.

Before LLEx, data curators had to expend great effort to tag the articles for easy referencing. The tagging process was cumbersome and time consuming, resulting in many untagged articles. The word cluster widget provides a way to circumvent manual tagging and explore the articles based on hierarchies of word clusters detected straight from the text of the articles. Because nearly all the articles contained full text and abstracts, these could be analyzed to generate word clusters. For LLEx, the word cluster visualization was chosen as an exploratory widget where users could discover collections of related content.

3.1.2 Streamgraph

Streamgraphs are not new in the field of visual analytics [1, 5] but have recently increased in popularity. They are ideal for displaying trends in subsets of a dataset over time. SRS's implementation, known as *StoryFlow* [9], accentuates the increase or decrease in prominence of a stream over time. It uses output similar to the word clusters, with sensitivity to the temporal values associated with the data. Thus the streams represent textual themes threaded through the data and depict the change in prominence of these themes over time.

Because understanding data themes over time would not aid the selection of relevant articles, the StoryFlow widget was not chosen for LLEx.

3.2 Structured Text Analysis

The visualizations that rely on structured text leverage more well-known techniques for categorizing and interacting with structured data.

3.2.1 Faceted Browse

The faceted browse paradigm allows users to explore different dimensions of structured, categorical data. Because impromptu analysts using LLEx needed to understand relevant articles based on known properties of the data, the faceted browse widget was deployed as part of the application. The articles had categorical properties divergent enough to nicely partition the data, yet that data was not so widely dispersed as to over-partition and overwhelm users with options. By working closely with the Lessons Learned Operations team, useful facet categories were identified. Because the most utility was found in the faceted browse view, the visualization occupies the central, primary widget location by default.

Considering simplicity for the impromptu analysts, other faceted browse widget design considerations were also made. First, the number of facets was limited to avoid overwhelming the analyst with unnecessary decision points.

Faceted browse widgets can be configured to support OR-ing or AND-ing of the faceted values. For simplicity, SRS performs an AND operation to narrow down choices in exploration rather than an OR operation, which exposes every possible facet value.

3.2.2 Treemap

The treemap partitions data according to chosen fields and emphasizes the relative distribution in a hierarchical layout. Treemaps are well-documented tools in the field of visual analytics [11]. The "parents" in the hierarchy are represented by rectangles where the area granted each rectangle is relative to the number of articles within that parent category. In LLEx, just two levels of hierarchy are visible to avoid over-partitioning and overcomplicating the view.

As categorical information was of high interest to LLEx users, the treemap visually displayed a subset of the categories used within the faceted browse widget. Because the treemap and faceted browser leverage the exact same portions of categorical data, it was important to verify that the additional widget really satisfied an analytical goal. Furthermore, because faceted browse allows users to select and drill down into the values, the faceted browse view is best for narrowing down a set of articles given a particular value of interest. Finally, because the treemap does not filter other values, it was a better exploration tool than the faceted browser; users could visually approximate the importance of certain articles among all categories, not just unfiltered categories.

3.2.3 Geospatial

The SRS geospatial widget overlays documents on a map based on the documents origin or other location-based information available in the document. The map may be of the world, a country, or even a building. The more documents that originate from a source, the larger the dot representing those documents becomes on the map, making it easy to detect common or uncommon locations within the information space.

As articles are gathered across varying data sources of the different national laboratories, location information is available. While the structure of the articles lend themselves well to the geospatial widget, the analytical goals of LLEx impromptu analysts do not justify the use of such a widget. When project managers, project teams, workflow process designers, or the Lessons Learned Operations team are looking for articles, the origin of an article has little bearing on its applicability. It was determined that introducing the geospatial widget would be a distraction from the main analytical goals of LLEx users.

3.2.4 Timeline Histogram

The timeline histogram provides an understanding of how many articles were generated over a period of time. The data is grouped into bins and summed so that it can be distributed in a histogram over the length of the timeline. The bins are chosen from the full time range of the dataset, and then adjusted to fit logical human boundaries (e.g., days, weeks, months). The number of bins is chosen to keep the rendered bin size approximately five pixels in width. That size is generally useful to keep the peaks from being too pronounced or too dampened.

The analytical value of the timeline widget was initially unknown for LLEx as date published did not appear to be a relevant aspect of the articles. Through working with the Lessons Learned Operations team, we came to understand a new use case that boosted the importance of the timeline histogram. The articles had evolved to contain a "Most Recent" flag that allowed users to see the most recently posted articles, which should be read first. While the publish date itself did not have a bearing on the applicability of such an article, understanding if the article was in the far right along the timeline accomplished a similar goal. As such, the publish date of the article was mapped to the timeline histogram and included within LLEx.

3.2.5 List

The list widget contains the individual articles and is the only widget that does not represent aggregations of data. Articles may be selected or deselected from the list widget. Likewise, users can open any article in the list to view it in full detail.

As the eventual goal of all LLEx inquiries was to arrive at a small set of relevant articles, the list was a practical solution for this last step. Because publish date is important for LLEX, this attribute is also encoded in the list widget as a secondary sort order, after primarly sorting by selection status. By sorting secondarily by publication date, users visit the articles in temporal order. While the list has a practical application in that it closes the explore-discover-find loop, we learned that the list served an additional purpose. In the visual analytics field, the list or table may be among the least intriguing widgets as it represents articles as individuals rather than aggregations. In discussions with the Lessons Learned Operations team, we discovered that the list served as a familiar user interface paradigm and allowed users to build trust with the system. Because the impromptu analysts could see the actual articles, they were able to establish a common language and then build further trust and understanding of the other widgets based on this list.

4 Interaction

While each widget holds its own unique analytic capabilities, the deeper analytical value of SRS exists among the interaction between widgets. In the second release of SRS, several design iterations were made to improve the user interaction paradigms. Because version one implemented different user experience paradigms in different widgets, several questions were asked to address interaction consistency: Should a user's mouse click result in selection or drill-down into the data for deeper analysis? Does the selection made within a widget replace or augment the current selection? Does selection affect drill-down? How should selection be visible within the widgets?

Based on the answer to each of the above questions, certain analytic affordances were encouraged and nurtured. Likewise, some analytic capabilities were knowingly sacrificed to maintain simplicity and transparency. The interaction choices below detail the user interaction paradigm decisions made in an effort to provide easy-to-use analytic capability to impromptu analysts.

4.1 Widget Layout

SRS is designed for one widget to take center stage in the primary view while all other widgets serve in supporting roles on the periphery. Any widget can be promoted to the primary view to allow users to interact more intensely or view the details of the selected items. At the time of the widget swap, the formerly primary widget will trade places with the widget that is being promoted.

4.2 Selection versus Drill Down

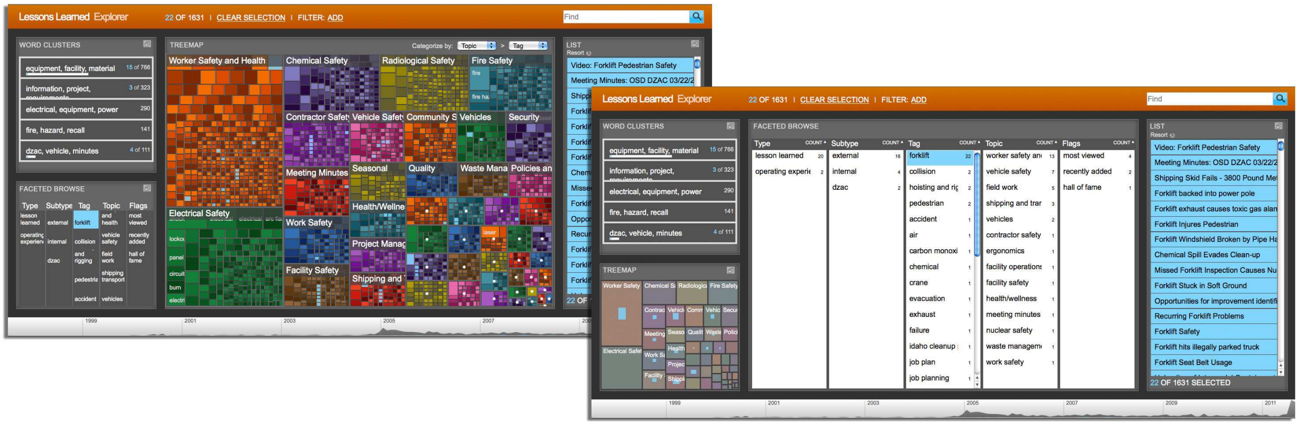
Originally in SRS, selection and drill-down interactions were inconsistent. Sometimes a user was able to left-click on an item to select it and CTRL-click to drill down, while for other widgets, the user CTRL-clicked to select an item and left-clicked to drill down. In version two, we chose left-click as the consistent method of selection. For those widgets that support drill-down–faceted browse and word clusters–selection prompts drill-down so that the impromptu analysts can more quickly explore the finer details of the selection. When a word cluster is selected, the sub-word clusters are automatically displayed so users can immediately refine the results if needed.

4.3 Selection Across the Views

Selection is the most visually prioritized state that binds all views across SRS. Selecting articles within one view will always show the articles selected across the other widgets, as shown in Figure 1. The bright blue color used for selection across all LLEx widgets allows impromptu analysts to visualize deeper patterns across widgets as a way to triangulate decision points or discover interesting articles.

To illustrate, a project manager uses LLEx to identify relevant risks in using forklifts. The project manager knows of three relevant lessons learned based on previously planned projects, so hypothesizes that there may be one or two more published. To begin, the project manager selects the forklift value of the topic field in the faceted browse view. Looking within the faceted widget, the project manager narrows

Fig. 1: Lessons Learned Explorer: Treemap and Faceted Browse in Maximized, Center View



WORD CLUSTERS

- equipment, facility, material 15 of 766
- information, project, requirements 3 of 323
- electrical, equipment, power 290
- fire, hazard, recall 141
- dzac, vehicle, minutes 4 of 111

FACETED BROWSE

Type	COUNT	Subtype	COUNT	Tag	COUNT	Topic	COUNT	Flags	COUNT
lesson learned	20	external	16	forklift	22	worker safety an	13	most viewed	4
operating experie	2	internal	4	collision	2	vehicle safety	7	recently added	2
		dzac	2	hoisting and rig	2	field work	5	hall of fame	1
				pedestrian	2	shipping and trar	3		
				accident	1	vehicles	2		
				air	1	contractor safety	1		
				carbon monoxi	1	ergonomics	1		
				chemical	1	facility operations	1		
				crane	1	facility safety	1		
				evacuation	1	health/wellness	1		
				exhaust	1	meeting minutes	1		
				failure	1	nuclear safety	1		
				idaho cleanup	1	waste managem	1		
				job plan	1	work safety	1		
				job planning	1				

LIST

- Video: Forklift Pedestrian Safety
- Meeting Minutes: OSD DZAC 03/22/2
- Shipping Skid Fails - 3800 Pound Me
- Forklift backed into power pole
- Forklift exhaust causes toxic gas alar
- Forklift Injures Pedestrian
- Forklift Windshield Broken by Pipe Ha
- Chemical Spill Evades Clean-up
- Missed Forklift Inspection Causes Nu
- Forklift Stuck in Soft Ground
- Opportunities for improvement identifi
- Recurring Forklift Problems
- Forklift Safety
- Forklift hits illegally parked truck
- Forklift Seat Belt Usage

22 OF 1631 SELECTED

down terms to further describe the forklift articles. In the word cluster widget, the project manager sees that the forklift values fall into three different top-level clusters: *equipment, facility, material*, then *information, project, management*, followed by *vehicle, vechicles, dzac*. The categorization of forklift in the word cluster widget appears to be right on track, so the project manager decides to look at all of these articles. The project manager chooses to read the article that fell into *information, project, management* category first, because it may cover much of the project-management-related risks. Next, with forklift still selected in the faceted browse view, the project manager wants to understand the article categories. Looking at the treemap allows the project manager to quickly see that forklift has a large presence in the *Worker Safety and Health, Facility Safety, Vehicle Safety, Shipping and Transportation* and the *Field Work* categories. Then taking a quick look at the timeline, the project manager understands the publish date distribution. In the end, the project manager discovers there are several recent lessons learned around forklifts relevant to the project.

Through selection, the hypothesis of the impromptu analyst can be confirmed or negated across the different widgets, while each widget may lead to discovery of other relevant articles.

4.4 Selection Replacement

The first selection made within any SRS widget always dictates the current selection across the entire application. For most widgets, the first selection made within that widget will always replace the current selection, making it obvious what is selected. The list widget is the only exception to the selection replacement paradigm; selection is toggled in the list widget to allow users to more easily remove or add to the collection of documents selected.

The replacement selection paradigm was chosen to constrain and simplify selection and provide quick, easy-to-use analytical capabilities. While such a selection model is certainly simple, some analytical capabilities were relinquished. For example, if a a project manager wanted to refine selection by selecting in another widget–e.g., the timeline–after selecting the “forklift” facet value, such an interaction would not be possible, as selection in the timeline would replace, rather than refine, the selection. While a refinement selection model would allow for deeper analysis, the obvious nature of selection would be lost completely. To make sure that more users were able to use SRS and LLEx as an impromptu analytic tool, the simplified replacement selection model was implemented.

5 Conclusion

Looking at an SRS application, LLEx, we explored the applicability of each analytical widget offered by SRS. By focusing on the users analytical goals in combination with the underlying available data structures, one can determine the utility and prioritization of available SRS widgets. Then, to gain deeper analytical value, we considered user interaction. As the selection model is pervasive and visually prioritized within the tool, users can quickly and easily detect patterns to inform decisions while discovering the undiscovered. Using simplified, yet powerful interaction paradigms across SRS enables impromptu analysts to quickly find analytical value to solidify or question decisions. In the future, we look toward introducing additional drill-down and customization capabilities to augment the analytical SRS capabilities.

References

[1] L. Byron and M. Wattenberg. Stacked graphs – geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14:1245–1252, 2008.

[2] H. Chung, S. Yang, N. Massjouni, C. Andrews, R. Kanna, and C. North. Vizcept: Supporting synchronous collaboration for constructing visualizations in intelligence analysis. In *IEEE VAST Conference*, Salt Lake City, Utah, October 2010.

[3] M. Dork, S. Carpendale, C. Collins, and C. Williamson. Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1205–1212, November/December 2008.

[4] S. Grimes. Unstructured data and the 80 percent rule. *Experts Corner*, (3), 2008.

[5] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS ’00, pages 115–, Washington, DC, USA, 2000. IEEE Computer Society.

[6] i2. Analysts notebook assisted analysis and visualization, 2011.

[7] M. McKeon. Harnessing the information ecosystem with wiki-based visualization dashboards. *IEEE Transactions on Visualization and Computer Graphics*, 15:1081–1088, November 2009.

[8] W. Pike, J. Bruce, B. Baddeley, D. Best, L. Franklin, R. May, D. Rice, R. Riensche, and K. Younkin. The Scalable Reasoning System: Lightweight visualization for distributed analytics. *Information Visualization*, 8(1):71–84, 2009.

[9] S. Rose, S. Butner, W. Cowley, M. Gregory, and J. Walker. Describing story evolution from dynamic information streams. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 99 –106, Oct 2009.

[10] S. Rose, D. Engel, N. Cramer, and W. Cowley. *Automatic Keyword Extraction from Individual Documents*, pages 1–20. John Wiley & Sons, Ltd, 2010.

[11] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11:92–99, January 1992.

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