Helping Intelligence Analysts make Connections

M. Shahriar Hossain, Christopher Andrews, Naren Ramakrishnan, and Chris North
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061
Email: {msh, cpa, naren, north}@vt.edu

Abstract

Discovering latent connections between seemingly unconnected documents and constructing "stories" from scattered pieces of evidence are staple tasks in intelligence analysis. We have worked with government intelligence analysts to understand the strategies they use to make connections. Beyond techniques like clustering that aim to provide an initial broad summary of large document collections, an important goal of analysts in this domain is to assimilate and synthesize fine grained information from a smaller set of foraged documents. Further, analysts' domain expertise is crucial because it provides rich contextual background for making connections and thus the goal of KDD is to augment human discovery capabilities, not supplant it. We describe a visual analytics system we have built—Analyst's Workspace (AW)—that integrates browsing tools with a storytelling algorithm in a large screen display environment. AW helps analysts systematically construct stories of desired fidelity from document collections and helps marshall evidence as longer stories are constructed.

Introduction

What do the April'07 shootings at Virginia Tech, Bernard Madoff's Ponzi scheme uncovered in Dec'08, and the March'09 recall of Zencore plus have in common? They are all extreme happenings that lead us to question: 'Why didn't somebody connect the dots?' Our ongoing failures to do so have led to these and many other, arguably avoidable, catastrophes. Yet, piecing together a story between seemingly disconnected information remains an elusive skill and an understudied task.

Storytelling is an accepted metaphor in analytical reasoning and in visual analytics (Thomas and Cook (eds.) 2005). Many software tools exist to support story building activities (Eccles et al. 2008; Hsieh and Shipman 2002; Wright et al. 2006). Analysts are able to lay out evidence according to spatial cues and incrementally build connections between them. Such connections can then be chained together to create stories which either serve as end hypotheses or as templates of reasoning that can then be prototyped. However, there are severe limitations to human sensemaking capabilities, even on gigapixel-sized displays, when confronted with massive haystacks of data. Algorithmic support to help sift through the myriad of possibilities is crucial here. At the same time, storytelling is not entirely automatable since it is an exploratory activity and the analyst brings in valuable intuition and contextual cues to direct the story building process. Hence it is imperative that we view storytelling as a collaborative enterprise between algorithmic and human capabilities.

The focus of this paper is on exploring document collections and we present a visual analytics system called Analyst's Workspace (AW) that aids intelligence analysts in exploring connections and building stories between possibly disparate end points. Our key contributions are:

1. Design considerations that have emerged from a detailed user study with five analysts working on intelligence analysis tasks.
2. New algorithms that find stories through document collections and also help marshall evidence to support discovered stories.
3. Implementation of both interactive visualization and algorithmic storytelling support in AW; and a case study over a public domain dataset.

How Analysts make Connections

We recently had the opportunity to interview and perform a study with five intelligence analysts currently employed at a government organization. The detailed results are presented and discussed in [Andrews et al. 2010]. We begin by describing qualitative lessons from the interviews followed by a study of their strategies in solving analysis tasks.

Interviews with Analysts

For the purpose of this paper, it suffices to note that the goal of the interviews was to attempt to typify how analysts approached the large quantities of data they were required to sift through, and to learn what tools they used and how they used them. From these interviews, the most interesting fact that emerged was that the analysts largely used software tools only at the beginning and at the end of their analysis.

Basic search tools were used to filter down a dataset at the start of their analysis. At the end of the analysis, presen-

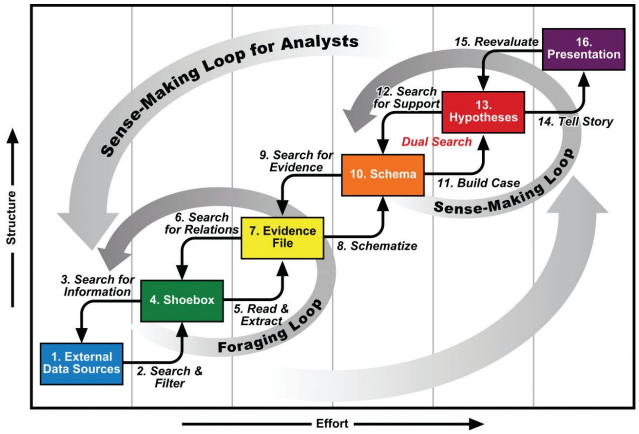


Figure 1: How intelligence analysts make connections (from (Pirolli and Card 2005).)

tation tools (such as PowerPoint) would be used to create reports. For the middle of the analytic process, where the actual sensemaking occurs, the analysts in our study reported that they tended to print out reports and other source materials. This allowed them to easily read them, annotate them with notes and highlights, sort them into physical folders, stack them in meaningful ways on the desk, and even lay all the documents out on a large table where they could be organized and rapidly skimmed.

A formal way to characterize the above observations is with reference to the schematic of Pirolli and Card (Pirolli and Card 2005). As Fig. 1 shows, the process by which intelligence analysts make connections is frequently tentative and evolutionary, with structures developing as understanding of the data increases. There are two 'subloops' in Fig. 1: information foraging and sense-making. Most analytic systems, such as *IN-SPIRE* (PNNL), *Jigsaw* (HCII), *ThemeRiver* (Havre et al. 2002), *NetLens* (Kang et al. 2007) focus on support for the information foraging loop, leaving the sensemaking to the analyst. Other tools, such as *Analyst's Notebook* (i2group), *Sentinel Visualizer* (FMS, Inc.), *Entity Workspace* (Bier et al. 2006), and *Palantir* (Khurana et al. 2009) focus more on the the sensemaking loop, and while many of them ostensibly support foraging, the analysts reported using these tools primarily for late stage sensemaking and presentation.

The key problem with this separation of the two halves of the sensemaking process is that the schematic is not meant to be a state diagram – it is a representation of some of the thought processes and structures that are identifiable during sensemaking and a description of how they relate. There is an overall trend from a collection of raw data to a final report, but inbetween, the analyst should be ranging widely across the entire process, building up an understanding through progressive foraging and structuring.

User Study

The tendency of analysts to resort to non-software methods for information organization suggested to us the potential

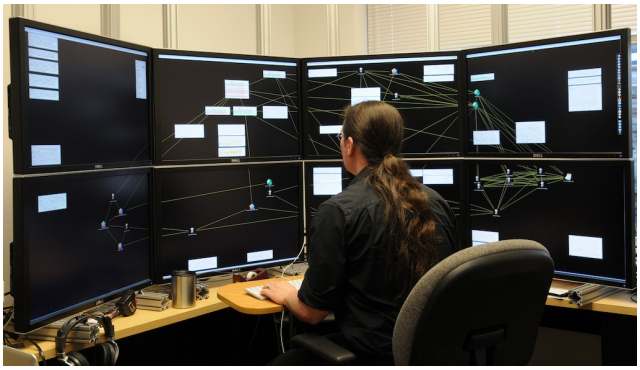


Figure 2: A user works with Analyst's Workspace on a 32 megapixel display.

for exploring the use of large screen displays and how they can be integrated into the sensemaking process. If the sensemaking can be drawn back into the computational realm, it provides the opportunity to better support the analysts.

We conducted a detailed user study with a large 32 megapixel (10,240×3,200) display, which consists of a 4×2 grid of 30″ LCD panels, each with a maximum resolution of 2560×1600. All of the panels in the display are driven by a single computer, allowing us to run conventional desktop applications on the display without modification. The display is configured for single-user use and is slightly curved around the user, who sits in the center, with the freedom to rotate around to access all parts of the display (Fig. 2).

For the study, we employed the VAST (Symposium on Visual Analytics Science and Technology) 2006 Challenge dataset. This dataset contains approximately 240 documents, which are primarily synthetic news stories from a fictitious city newspaper. Although this is a relatively small dataset, most of it is actually noise, with only about ten of the documents being relevant to uncovering the plot. Another feature of this dataset is that even if the analyst uncovers all ten documents, some analysis is still required to actually determine the nature of the synthetic threat.

Five analysts were presented with the dataset as a directory of files, with only the search facilities of Windows XP's File Explorer, WordPad for reading and annotating documents, and a simple image viewer for the couple of images included in the dataset. We asked them to uncover the buried plot using any approach that they desired, using the space afforded by the display in any way that they found useful.

A key conclusion from this study was that the large display was treated in a fundamentally different way from conventional displays. Conventional displays typically constrain the user to working with one or two applications or documents at a time. Interaction in this environment is primarily application oriented. The large display, on the other hand, permits the user to work with a large number of applications and documents simultaneously. In our study, we found that this simple change encouraged users to adopt a more document-centric approach, working with the documents in a fashion more akin to the way one would in-

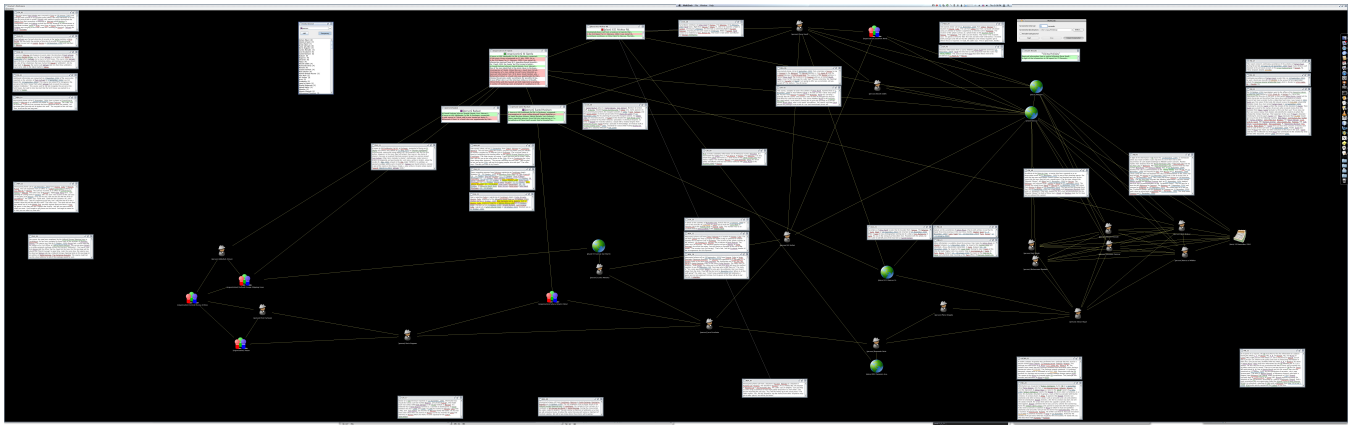


Figure 3: An active session in Analyst's Workspace. Full text documents and entities share the space, with a mixture of spatial metaphors, such as clusters, graphs, and timelines all in evidence. The yellow lines are the links of the derived social network.

teract with physical pieces of paper laid out on a physical desk. We found that our subjects freely moved documents around the space, creating a form of "semantic layer" over the document collection, in which position on the display helped to convey additional semantics, such as relationships between the documents. Using space to encode extra information about the relationship between objects has a rich history, rooted in human perceptual abilities (Kirsh 1995). A primary advantage of the use of space for this purpose is that it is very flexible and allows the user to express transitory or questionable relationships in a visually salient structure without committing to a strict and potentially confining structure (Shipman and Marshall 1999).

For example, most of the analysts used the space to cluster the documents that they found important. The interesting feature of these clusters is that they were frequently vague and grouped documents on an assortment of different levels. For instance, documents in the same workspace could be clustered because they related to a particular person or place, because they had a related theme such as weapons, or even because of how the analyst regarded the documents (e.g., many of the analysts created a pile of documents that they thought were probably junk but seemed related enough that they did not want to close them and lose them). Sometimes, clusters would form without the analyst having any clear thought about why the documents in the collection might fit together.

While the study demonstrated the appeal of working spatially for sensemaking, it is worth noting that most analysts did not solve the analysis task. At the end of most sessions, the analysts had all identified the major themes and created representative structures, but they did not connect the dots to put the entire story together. Here, we can point to the impoverished foraging support, which could not help them to identify the critical linchpins that would draw the whole story together.

The above observations motivated us to develop a visual analytics environment—Analyst's Workspace (AW), and open the door to our algorithmic assistance for foraging connections of exploration within AW. AW i) closely mimics information organization layouts employed by analysts, ii) relates multiple representations to accommodate different strategies of exploration, and iii) provide automated algorithmic assistance for foraging connections and hypothesis generation.

Analyst's Workspace

AW provides the user with a plethora of interaction tools for use with large screen displays (e.g., familiar click-and-drag, selection rectangles, multi-click selections) as well as information organization facilities (e.g., graph layout, temporal ordering). Because these operations are local, they only affect the local area or the currently selected documents and hence enable the analyst to freely mix spatial metaphors (see Fig. 3).

While the primary visual elements in AW are full text documents, we also provide support at the entity level. Documents are marked up based on extracted entities, and the analyst can use context menus to quickly identify new entities and create aliases between entities. Double clicking an entity of interest in a document opens an entity object, which is initially displayed as a list of documents in which that entity appears. Entities can also be collapsed down to a representational icon, and AW automatically draws links between entities when they co-occur in a document. These two features allow the analyst to rapidly construct and explore social networks, which are commonly used tools in intelligence analysis.

AW also provides basic facilities for text-based search. Search results are displayed as lists of matching documents in the space, like the entities. The documents are color coded to tell the analyst the state of a document: open, previously viewed, or never viewed.

Visual links play a strong role in AW. These allow a number of relationships to be expressed, freeing spatial proximity to be used to express more complex relationships more directly related to making sense of the dataset.

While Analyst's Workspace is designed to support a flex-

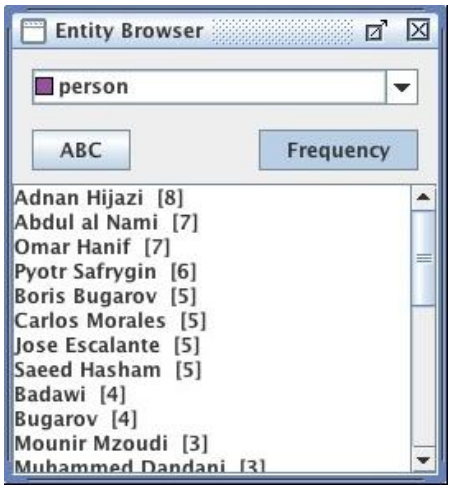


Figure 4: AW's entity browser, here showing the people identified in the dataset, sorted by the number of documents in which each appears.

ible approach to sensemaking, it does encourage a particular analytic approach that we observed being used by the analysts. This is a strategy that Kang et al. (2009) referred to as "Find a Clue, Follow the Trail". In this strategy, the analyst identifies some starting place and then branches out the investigation from that point, following keywords and entities.

In AW, a starting point can be provided by the entity browser (Fig. 4), which allows the analyst to order entities by the number of occurrences in the dataset. The analyst opens this entity and gets a list of documents in which this entity appears. The analyst then works through these documents, opening new entities or performing searches as new clues are found. Since all of the search results are independent objects in the space and there is a visual record of which documents have been visited, AW can support both a breadth-first and a depth-first search through the information. As the investigation progresses, the analyst uses the space to arrange the information as it is uncovered, building and rebuilding structures to reflect his or her current understanding of the underlying narrative.

While this approach has been shown to be fairly effective (Kang et al. 2009), it does not permit greater characterization of the dataset and does not support more complex questions that the analyst might ask. For example, this approach relies entirely on the analyst to pick the right keywords and entities to "chase," and can miss less direct lines of investigation. It is common for terrorists to use multiple aliases or code words that can easily thwart this approach. However, it is possible that common patterns of behavior or other document similarities might help the analyst to uncover some of these connections.

The analyst may also need the discovery of paths through the dataset to be more efficient. For example, the analyst may have uncovered that a revolutionary in South America shares the same last name as a farmer in the Pacific Northwest who has been implicated in some nefarious affairs and wishes to ask if there is any link between them or if their last name is a coincidence. An exhaustive background check of the two men is possible through AW if the dataset is relatively small, but it is an indirect and time consuming process.

Algorithmic Support for Storytelling

We attempted to formalize and support the ways by which an analyst conducts unstructured discovery, chases leads, and marshalls evidence to support or refute potentially promising chains. Our story generation framework is exploratory in nature so that, given starting and ending documents of interest, it explores candidate documents for path following, and heuristics to admissibly estimate the potential for paths to lead to a desired destination. The generated paths are then presented to the AW analyst who can choose to revise them or adapt them for his/her purposes.

A story between documents d_1 and d_n is a sequence of intermediate documents $d_2, d_3, ..., d_{n-1}$ such that every neighboring pair of documents satisfies some user defined criteria. Given a story connecting a start and an end document (see Fig. 7 (a)), analysts perform one of two tasks: they either aim to strengthen the individual connections, possibly leading to a longer chain (see Fig. 7 (b)), or alternatively they seek to organize evidence around the given connection (see Fig. 7 (c)). We use the notions of *distance threshold* and *clique size* to mimic these behaviors. We designed our storytelling algorithm to work with these two criteria that are under the AW analyst's control and experimentation. (These are not magic parameters whose values have to be tuned but are rather controls that mimic the natural process by which analysts tighten or strengthen their hypotheses.)

The distance threshold refers to the maximum acceptable distance between two neighboring documents in a story. Lower distance thresholds impose stricter requirements and lead to longer paths. The clique size threshold refers to the minimum size of the clique that every pair of neighboring documents must participate in. Thus, greater clique sizes impose greater neighborhood constraints and lead to longer paths. See Fig 7 (d) for a new path with both stricter clique size and stricter distance thresholds. These two parameters hence essentially map the story finding problem to one of uncovering clique paths in the underlying induced similarity network between documents.

We use the term "clique chain" to refer to a story along with its surroundings connections of evidence. In contrast, a story only constitutes the junction points between consecutive cliques. Another way to characterize them is that a clique chain constitues many stories.

Fig. 5 describes the steps involved in generating stories for interaction by the AW analyst. For document modeling, we use a bag-of-words (vector) representation where the terms are weighted by tf-idf with cosine normalization. Our search framework has three key computational stages:

1. construction of a concept lattice,
2. generating promising candidates for path following, and
3. evaluating candidates for potential to lead to destination.

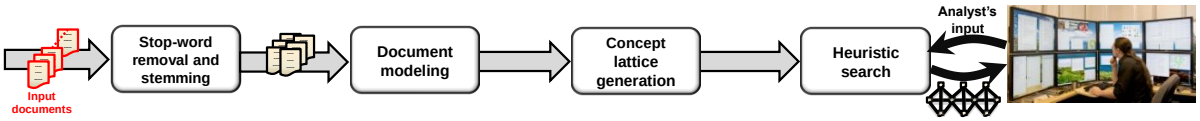


Figure 5: Pipeline of the storytelling framework in AW.

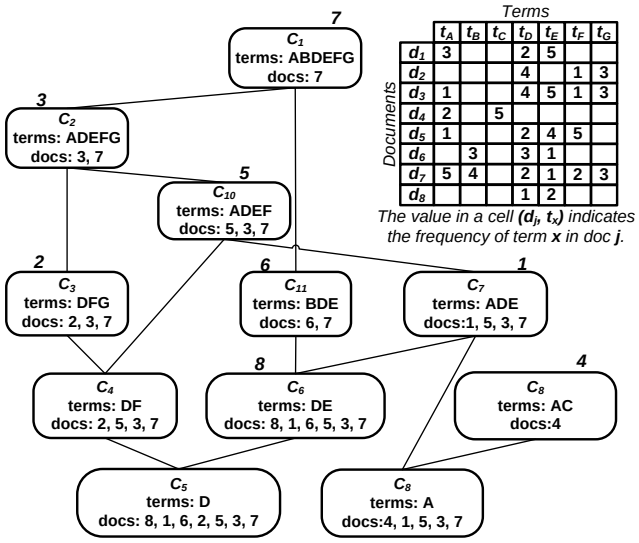


Figure 6: A dataset and its concept lattice.

Of these, the first stage can be viewed as a startup cost that can be amortized over multiple path finding tasks. The second and third stages are organized as part of an A* search algorithm that begins with the starting document, uses the concept lattice to identify candidates satisfying the distance and clique size requirements, and evaluates them heuristically for their promise in leading to the end document.

Concept Lattice Construction

The concept lattice is a data structure that models conceptual clusters of document and term overlaps and is used here as a quick lookup of potential neighbors that will satisfy the distance threshold and clique constraints. Given a (weighted) term-document matrix, we use the CHARM-L (Zaki and Ramakrishnan 2005) closed set mining algorithm on a boolean version of this matrix to generate a concept lattice. Each concept is a pair: (document set, term set) as shown in Fig. 6. Further, we order the document list for each concept by the number of terms. Note that we can find an approximate set of nearest neighbors for a document d from the document list of the concept containing d and the longest term set.

Successor Generation

Successor generation is the task of, given a document, using the distance threshold and clique size requirements to identify a set of possible successors for path following. Note that this does not use the end document in its computation.

The basic idea of our successor generation approach is, in addition to finding a good set of successor nodes for a given document d, to be able to have sufficient number of them so that, combinatorially, they contribute a desired number of cliques. With a clique size constraint of k, it is not sufficient to merely pick the top k neighbors of the given document, since the successor generation function expects multiple clique candidates. (Note that, even if we picked the top k neighbors, we will still need to subject them to a check to verify that every pair satisfies the distance threshold.) Given that this function expects b clique candidates (where b is the branching factor), a minimum m documents must be identified where m is given by the solution to the inequalities:

$$\binom{m-1}{k} < b \text{ and } \binom{m}{k} \geq b$$

For a given document, we pick the top m candidate documents from the concept lattice and form combinations of size k. Our successor generator thus forms combinations of size k from these m documents to obtain a total of b k-cliques. Since m is calculated using the two inequalities, the total number of such combinations is equal to or slightly greater than b (but never less than b). Each clique is given an average distance score calculated from the distances of the documents of the clique and the current document d. This aids in returning a priority queue of exactly b candidate k-cliques.

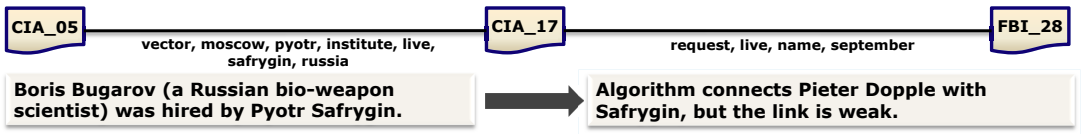
We evaluated our successor generation mechanism by comparing it to the brute force nearest neighbor search and the cover tree based (Beygelzimer et al. 2006) nearest neighbor search mechanisms. We found that our concept lattice based successor generation mechanism works faster than these other approaches (not described due to space limitations). Therefore we adopt the concept lattice in our successor generation procedure.

Evaluating Candidates

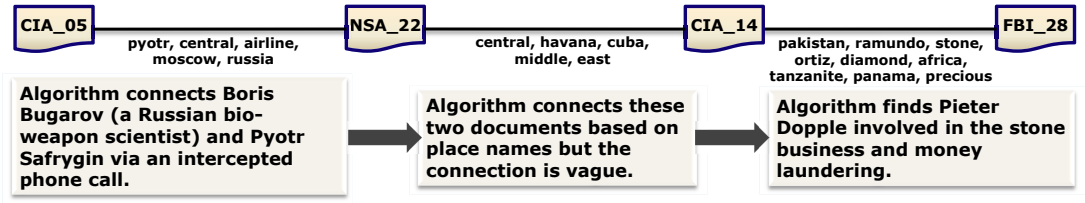
We now have a basket of candidates that are close to the current document and we must determine which of these has the potential to lead to the destination document. The primary criteria of optimality for the A* search procedure of our framework is the cumulative Soergel distance of the path. The Soergel distance between two documents $d_1$and d_2 is given by:

$$D(d_1, d_2) = \frac{\sum_t |w_{t,d_1} - w_{t,d_2}|}{\sum_t \max(w_{t,d_1}, w_{t,d_2})}$$

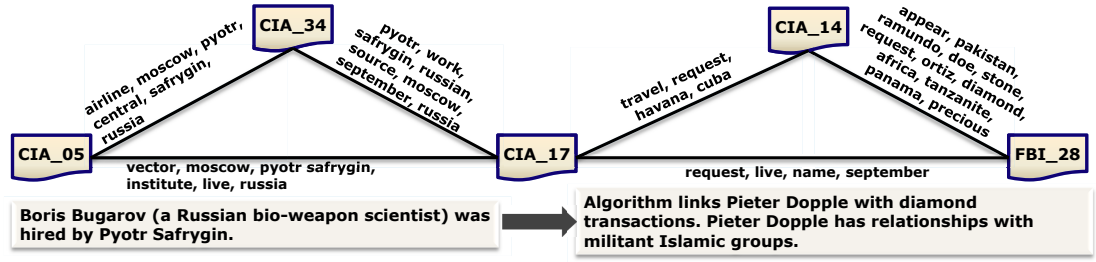
where w_{t,d_i} indicates the weight for term t of document d_i. We use the straight line Soergel distance for the heuristic



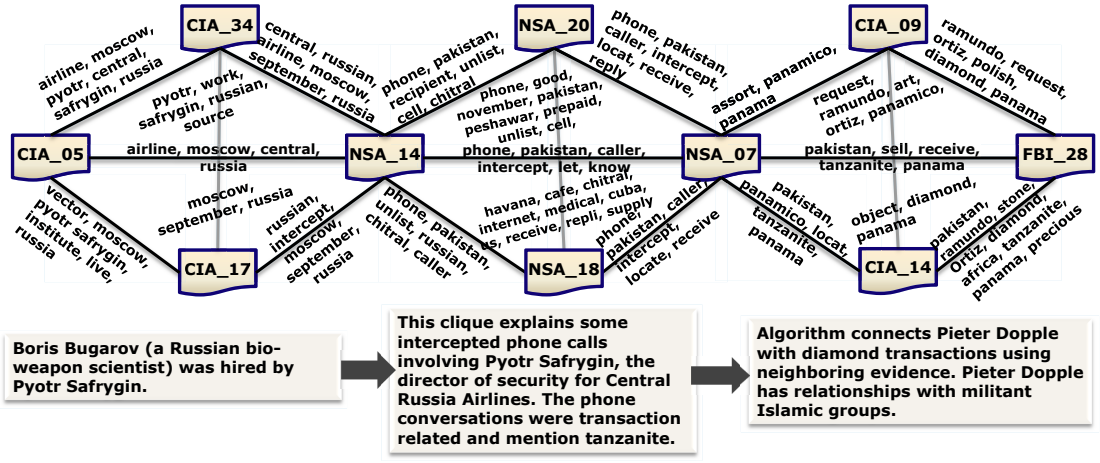
(a) Clique size=2, distance threshold=0.99: Example of a story with weak connections.



(b) Clique size=2, distance threshold=0.96 : Example of a story with stricter links.



(c) Clique size=3, distance threshold=0.99 : Example of a better story with small amount of surrounding evidence.



(d) Clique size=4, distance threshold=0.95 : Example of a better story with more surrounding evidence.

Figure 7: A sample story illustrating the impact of change of clique size and distance threshold. The goal is to connect bio-weapon scientist Boris Bugarov with money launderer Pieter Dopple. As the distance and clique size thresholds are experimented with, we observe surrounding evidence connecting Pieter Dopple with militant Islamic groups.

and, because it obeys the triangle inequality, it can be shown that this will never estimate the cost of a path from any document d to the goal. Therefore our heuristic is admissible and our A* search will yield the optimal path.

It is important to note that our algorithm never explicitly computes or materializes the underlying network of similarities at any time. As a result, it is very easy for the AW analyst to vary the clique size and distance thresholds to analyze different stories for the same start and end pairs.

Experimental Results

We conduct both quantitative and qualitative evaluation of AW's visual and algorithmic support for storytelling. The questions we seek to assess are:

1. What is the interplay between distance threshold and clique size constraints in story construction? How does

our heuristic fare in reference to an uninformed search and as a function of the constraints?

2. What is the quality of stories discovered by our algorithm?
3. How do the algorithmically discovered stories compare to those found by analysts?
4. How can analysts mix-and-match algorithmic capabilities with their intuitive expertise in story construction?

For our experiments, we used an analysis exercise (Hughes 2005) developed at the Joint Military Intelligence College. The exercise dataset is sometimes referred to as the Atlantic Storm dataset.

Evaluating Story Construction

To study the relationship between distance threshold and clique size constraints, we generated thousands of stories with different distance and clique size requirements from the Atlantic Storm dataset, and computed the maximum clique size for which at least one story was found. As expected, we see an anti-monotonic relationship and that it is more difficult to marshall evidence as distance thresholds get stricter (Fig. 8).

To study the performance of AW's heuristic over a non-heuristic based search, we picked 1000 random start-end document pairs from our document collection and generated stories with different distance threshold and clique size requirement. The non-heuristic search is simply a breadth-first search version of our A* search framework (in other words, the heuristic returns zero for all inputs). Fig. 9 compares average runtimes of AW's heuristic based search against the non-heuristic search. From top to bottom, three consecutive plots of Fig. 9 depict the average runtimes respectively as functions of story length, distance threshold, and clique size. Astute readers might expect a monotonic increase of average runtime with longer stories in Fig. 9 (top). Stories tend to become longer with stringent distance threshold and clique size. Further stringency, however, results in broken stories (the length of the story theoretically becomes infinite). As a result, we found a smaller number of longer stories than

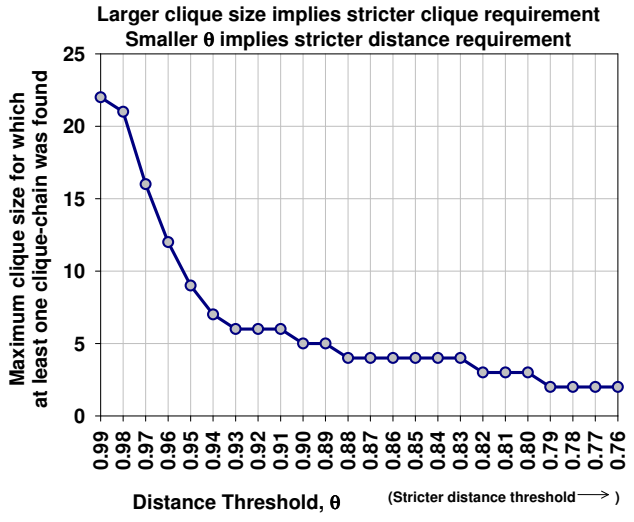


Figure 8: Atlantic storm dataset: interplay between distance threshold and clique size constraints.

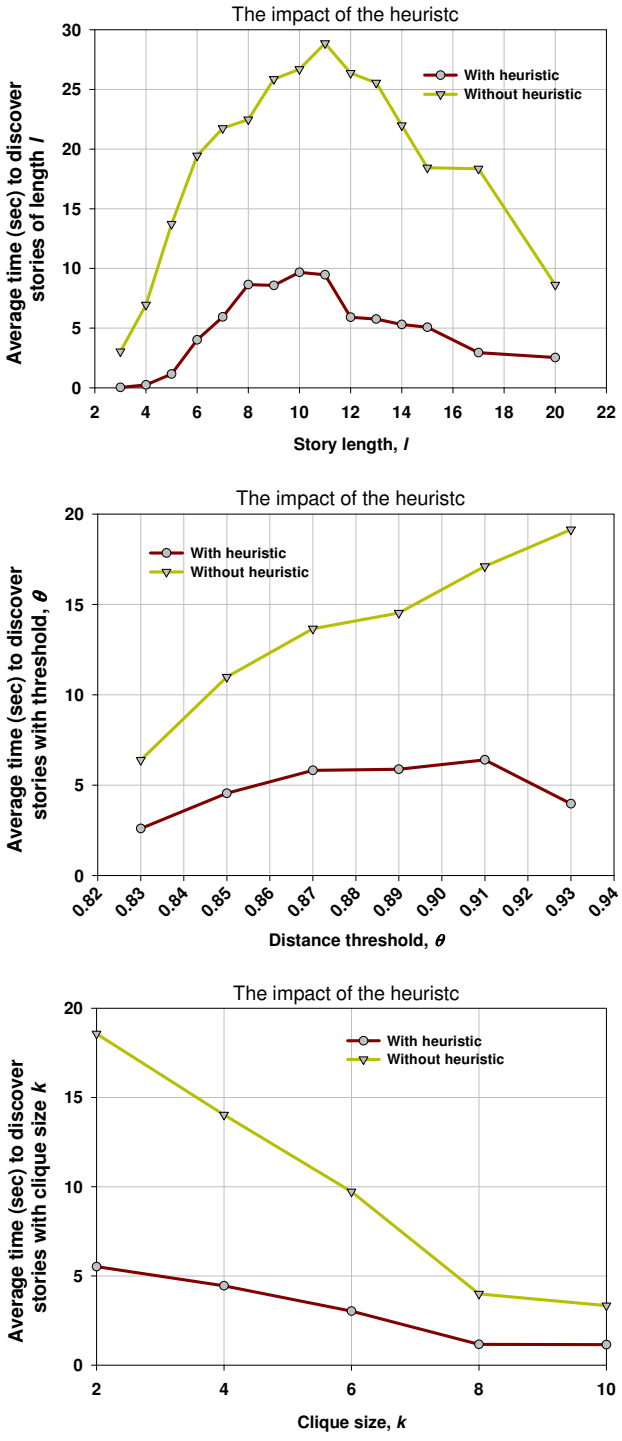


Figure 9: We used 1000 random start-end pairs to compare the performance of AW's heuristic search against uninformed search.

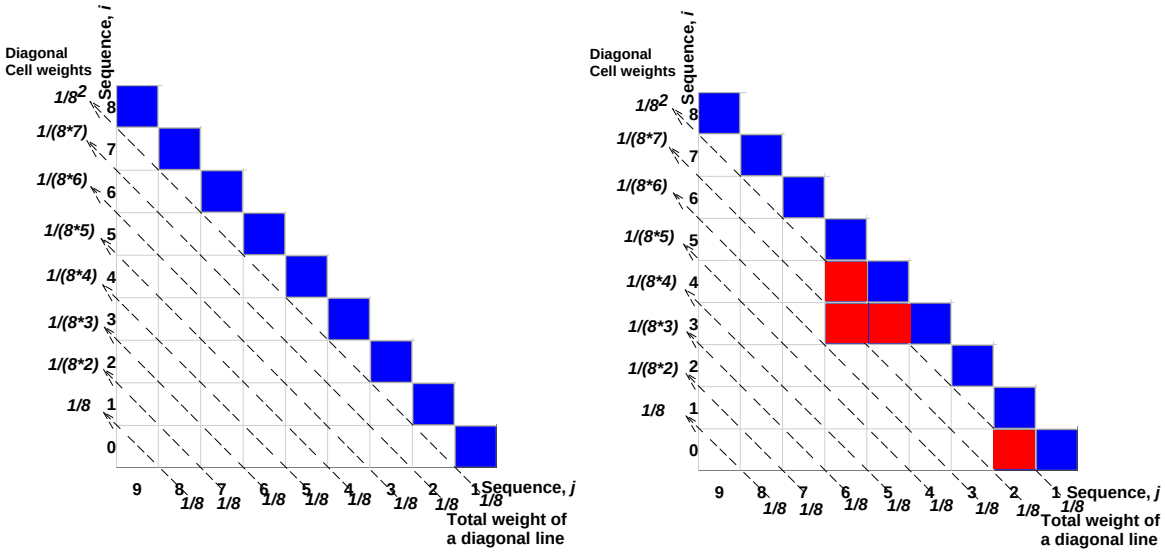


Figure 10: *(left)* A dispersion plot of an ideal story. The dispersion coefficient $\vartheta = 1.0$. *(right)* A dispersion plot of a non-ideal story of same length. The dispersion coefficient $\vartheta = 1 - \frac{3}{8\times8} - \frac{1}{8\times7} = 0.94$.

the shorter ones. In all the plots of Fig. 9, we calculated the average time over only the discovered stories. Since most of the long stories were found quickly by our algorithms, the curves of Fig. 9 (top) increase first and then descrease instead of being monotonically increasing. All the plots of Fig. 9 depict that the heuristic yields significant gains over the uninformed search.

Evaluating Story Quality

It is difficult to objectively evaluate the quality of stories. Here, we adopt Swanson's complimentary but disjoint (CBD) hypothesis (Swanson 1991) and assess the pairwise Soergel distance between documents in a story, between consecutive as well as non-consecutive documents. An ideal story is one that meets the Soergel distance threshold θ only between consecutive pairs whereas a non-ideal story "oversatisfies" the distance threshold and meets it even between non-consecutive pairs. As shown in Fig. 10 (left), an ideal story has only diagonal entries in its dispersion plot (contrast with Fig. 10 (right)). If n documents of a story are d_0, d_1, ..., d_{n-1}, then our formula for dispersion coefficient is given by:

$$\vartheta = 1 - \frac{1}{n-2}\sum_{i=0}^{n-3}\sum_{j=i+2}^{n-1} \text{disp}\left(d_i, d_j\right)$$

where

$$\text{disp}\left(d_i, d_j\right) = \begin{cases} \frac{1}{n+i-j}, & \text{if } D\left(d_i, d_j\right) > \theta \\ 0, & \text{otherwise} \end{cases}$$

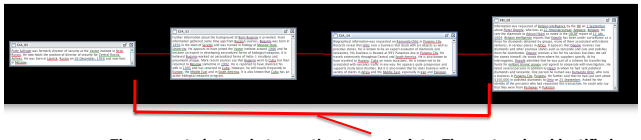
We also compute p-values for each generated story. Recall that at each step of the A* search we build a queue of candidate documents by investigating the corresponding concepts of the concept lattice. To calculate the p-value of a clique of size k, we randomly select $k-1$ documents from the entire candidate pool and check if all the edges of the formed k-clique satisfy the distance threshold θ, iterating the test 50,000 times. This allows us to find p-values down to 2×10^{-5}. We repeat this process for every junction-document of a discovered clique chain. The overall p-value of a clique chain is calculated by multiplying all the p-values of every clique of the chain.

Table 1: Sample story fragments from an analyst. How did our algorithm fare in discovering them?

Story	Found by algorithm	Found in the clique path	Found by merging stories
FBI_30→FBI_35→FBI_41→CIA_43	√		
CIA_41→CIA_34→CIA_39→NSA_09 →NSA_16			√
CIA_01→CIA_05→CIA_34→CIA_41→CIA_17→ CIA_39→NSA_22	√		
NSA_11→NSA_18→NSA_16	√		
CIA_06→CIA_22→CIA_21	√		
CIA_24→FBI_24	√		
NSA_06→CIA_32→CIA_42	√		
NSA_16→CIA_38→CIA_42		√	

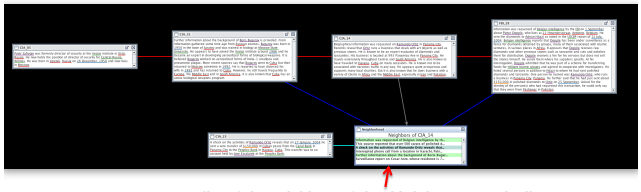
Story Validation

We have depicted stories with different distance and clique size requirements in Fig. 7. The story connects a Russian bio-weapon scientist (Boris Bugarov) with a money launderer (Pieter Dopple) who has ties to militant Islamic groups. In Table 1 we compared some discovered stories with fragments put together by analysts. The inputs from the analysts are not complete stories but rather scattered,



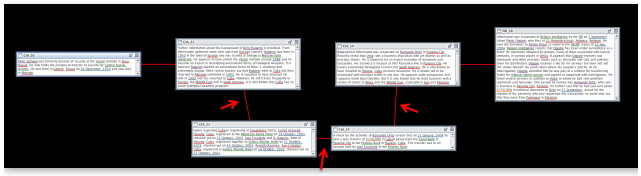
The generated story between the two endpoints. The system has identified two linking documents, and connected them together into a linked story.

The analyst requests a story connecting a pair of interesting documents.



A list of the neighbors of the third document. The lines provide visual links to open documents.

Unsatisfied with the strength of the connection, the analyst requests information about documents in the surrounding neighborhood (i.e., within the local clique).



New connections have been manually added to extend the story

Having explored the local neighborhood, the analyst has identified two additional documents that form a more meaningful connection and extends the original story.

Figure 11: Illustration of AW usage.

piecewise connections. The table illustrates that all the stories were discovered by our algorithm with two exceptions: the stories were not in the directly discovered path but were present in the clique chain (i.e., the story did not exhibit the same junction points), or the fragment can be discovered by merging multiple stories together. This depicts the potential of our heuristic in helping AW analysts discover stories algorithmically.

Illustration of AW Usage

Fig. 11 shows an example of the usage of AW and our algorithms. In this scenario, the analyst requests a story connecting a pair of interesting documents. The algorithm returns a story but the analyst is not satisfied with parts of the story. The analyst then requests information about documents in the surrounding neighborhood (i.e., within the local clique) of an intermediate document. Having explored the local neighborhood, the analyst identified two additional documents that form a more meaningful connection and extends the original story. The two story fragments of Table 1 that were not directly found by the algoritm could be modified by the analyst to obtain more meaningful stories.

Related Literature

We organize related work in this space under various categories.

Relationships via associations: Jayadevaprakash et al. (2005) advocate a transitive method to generate an association graph to find relationships between non-cooccurring text objects. The authors advocate the use of transitive methods because transitive methods do not require expensive training by human experts. Similarly, our approach does not require expensive training, but we situate our methods in a visual analytics setting with intelligence experts providing active feedback in the discovery process. Vaka and Mukhopadhyay (2009) describe a method to extract transitive associations among diseases and herbs related to Ayurveda. The method is based on a text-mining technique designed for discovering transitive associations among biological objects. It uses a vocabulary discovery method from a subset of PubMed corpora to associate herbs and diseases. Thaicharoen (2009) aims to discover relational knowledge in the form of frequent relational patterns and relational association rules from disjoint sets of literature. Although the aim of the research of Vaka and Mukhopadhyay and Thaicharoen is somewhat similar to our objective, we focus on finding connecting chains in an induced similarity network of documents rather than finding a chain of associations via external knowledge.

Topic based hypotheses generation: Jin et al. (2007) present a tool based on link analysis, and text mining methodologies to detect links between two topics across two individual documents. Srinivasan (2004) presents text mining algorithms that are built within the framework established by Swanson (1991). The algorithms generate ranked term lists where the key terms represent novel relationships between topics. Although we do not conduct explicit topic modeling in our work, the requirement to impose clique constraints in story construction essentially helps transduce slowly between topics.

Classification and clustering for hypotheses generation: Glance et al. (2005) describe a system that gathers specific types of online content and delivers analytics based on classification, natural language processing, and other mining technologies in a marketing intelligence application. Faro et al. (2009) propose a clustering method aimed at discovering hidden relationships for hypothesis generation and suitable for semi-interactive querying. Our method does not depend on classification/clustering for information organization but harnesses CBD structures in finding chains between documents of different clusters.

Connecting the dots: The "connecting the dots" problem has appeared in the literature in different guises and for different applications: cellular networks (Brassard et al. 1980), social networks (Faloutsos et al. 2004), image collections (Heath et al. 2010), and document collections (Das-Neves et al. 2005; Kumar et al. 2006; Shahaf and Guestrin 2010). Our work explicitly harnesses CBD structures whereas many of these works focused on contexts with weaker dispersion requirements. For instance, the model proposed by Shahaf and Guestrin (2010) explicitly requires a connecting thread of commonality through all documents in a story.

Discussion

We have described a visual analytics system (AW) that provides both exploratory and algorithmic support for analysts in making connections. Privacy considerations pro-

hibit us from describing the new applications that AW is being used for but the experimental results demonstrate its range of capabilities. Future work is geared toward more mixed-initiative facilities for story generation and probabilistic methods to accommodate richer forms of analyst's feedback. We are also working toward techniques to do automatic story summarization and concept map generation.

Acknowledgments

This work is supported in part by the Institute for Critical Technology and Applied Science, Virginia Tech, and the US National Science Foundation through grant CCF-0937133.

References

Andrews, C.; Endert, A.; and North, C. 2010. Space to Think: Large High-resolution Displays for Sensemaking. In *CHI '10*, 55–64.

Beygelzimer, A.; Kakade, S.; and Langford, J. 2006. Cover Trees for Nearest Neighbor. In *ICML '06*, 97–104.

Bier, E.; Ishak, E.; and Chi, E. 2006. Entity Workspace: An Evidence File That Aids Memory, Inference, and Reading. In *ISI '06*, 466–472.

Brassard, J.-P., and Gecsei, J. 1980. Path Building in Cellular Partitioning Networks. In *ISCA '80*, 44–50.

Das-Neves, F.; Fox, E. A.; and Yu, X. 2005. Connecting Topics in Document Collections with Stepping Stones and Pathways. In *CIKM '05*, 91–98.

Eccles, R.; Kapler, T.; Harper, R.; and Wright, W. 2008. Stories in GeoTime. *Info. Vis.* 7(1):3–17.

Faloutsos, C.; McCurley, K. S.; and Tomkins, A. 2004. Fast Discovery of Connection Subgraphs. In *KDD '04*, 118–127.

Faro, A.; Giordano, D.; Maiorana, F.; and Spampinato, C. 2009. Discovering Genes-diseases Associations from Specialized Literature using the Grid. *Trans. Info. Tech. Biomed.* 13:554–560.

FMS, Inc. FMS Advanced Systems Group, Sentinel Visualizer. Last accessed: May 26, 2011, http://www.fmsasg.com/.

Glance, N.; Hurst, M.; Nigam, K.; Siegler, M.; Stockton, R.; and Tomokiyo, T. 2005. Deriving Marketing Intelligence from Online Discussion. In *KDD '05*, 419–428.

Havre, S.; Hetzler, E.; Whitney, P.; and Nowell, L. 2002. ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE TVCG* 8(1):9–20.

HCII. Human Computer Interaction Institute, Carnegie Mellon University, Jigsaw. Last accessed: May 26, 2011, http://www.hcii.cmu.edu/mhci/projects/jigsaw.

Heath, K.; Gelfand, N.; Ovsjanikov, M.; Aanjaneya, M.; and Guibas, L. 2010. Image Webs: Computing and Exploiting Connectivity in Image Collections. In *CVPR*, 3432 –3439.

Hsieh, H., and Shipman, F. M. 2002. Manipulating Structured Information in a Visual Workspace. In *UIST'02*, 217–226.

Hughes, F. J. 2005. Discovery, Proof, Choice: The Art and Science of the Process of Intelligence Analysis, Case Study 6, "All Fall Down", Unpublished report.

i2group. The Analyst's Notebook. Last accessed: May 26, 2011, http://www.i2group.com/us.

Jayadevaprakash, N.; Mukhopadhyay, S.; and Palakal, M. 2005. Generating Association Graphs of Non-cooccurring Text Objects using Transitive Methods. In *SAC '05*, 141–145.

Jin, W.; Srihari, R. K.; and Ho, H. H. 2007. A Text Mining Model for Hypothesis Generation. In *ICTAI '07*, 156–162.

Kang, H.; Plaisant, C.; Lee, B.; and Bederson, B. B. 2007. NetLens: Iterative Exploration of Content-actor Network Data. *Info. Vis.* 6(1):18–31.

Kang, Y.; Görg, C.; and Stasko, J. 2009. The Evaluation of Visual Analytics Systems for Investigative Analysis: Deriving Design Principles from a Case Study. In *VAST*, 139–146.

Khurana, H.; Basney, J.; Bakht, M.; Freemon, M.; Welch, V.; and Butler, R. 2009. Palantir: a Framework for Collaborative Incident Response and Investigation. In *IDtrust '09*, 38–51.

Kirsh, D. 1995. The Intelligent Use of Space. *Artif. Intell.* 73(1-2):31–68.

Kumar, D.; Ramakrishnan, N.; Helm, R. F.; and Potts, M. 2006. Algorithms for Storytelling. In *KDD '06*, 604–610.

Pirolli, P., and Card, S. 2005. The Sensemaking Process and Leverage Points for Analyst Technology as Identified through Cognitive Task Analysis. In *ICIA '05*.

PNNL. Pacific Northwest National Laboratory, INSPIRE visual document analysis. Last accessed: May 26, 2011, http://in-spire.pnl.gov.

Shahaf, D., and Guestrin, C. 2010. Connecting the Dots between News Articles. In *KDD '10*, 623–632.

Shipman, F. M., and Marshall, C. C. 1999. Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems. *CSCW* 8:333–352.

Srinivasan, P. 2004. Text Mining: Generating Hypotheses from MEDLINE. *J. Am. Soc. Inf. Sci. Technol.* 55:396–413.

Swanson, D. R. 1991. Complementary Structures in Disjoint Science Literatures. In *SIGIR '91*, 280–289.

Thaicharoen, S. 2009. *Text Association Mining with Cross-sentence Inference, Structure-based Document Model and Multi-relational Text Mining*. Ph.D. Dissertation, Univ. of Colorado at Denver.

Thomas, J. J., and Cook (eds.), K. A. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press.

Vaka, H. G. G., and Mukhopadhyay, S. 2009. Hypotheses Generation Pertaining to Ayurveda Using Automated Vocabulary Generation and Transitive Text Mining. In *NBIS '09*, 200–205.

Wright, W.; Schroh, D.; Proulx, P.; Skaburskis, A.; and Cort, B. 2006. The Sandbox for Analysis: Concepts and Methods. In *CHI '06*, 801–810.

Zaki, M. J., and Ramakrishnan, N. 2005. Reasoning About Sets Using Redescription Mining. In *KDD '05*, 364–373.